ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Московский физико-технический институт
(национальный исследовательский университет)»
МФТИ, Физтех

На правах рукописи

Безносиков Александр Николаевич

О некоторых продвинутых сюжетах теории сложности численных методов оптимизации: распределенные и стохастические постановки, обобщение на более широкие классы задач

Специальность: 1.2.2. Математическое моделирование,
численные методы и комплексы программ

ДИССЕРТАЦИЯ
на соискание ученой степени
доктора физико-математических наук

Научный консультант:
Гасников А. В. – д.ф.-м.н., чл.-корр. РАН

Долгопрудный — 2025

MOSCOW INSTITUTE OF PHYSICS AND TECHNOLOGY
MIPT, Phystech

As a manuscript

Beznosikov Aleksandr

On Advanced Topics in Complexity Theory of Numerical Optimization Methods: Distributivity, Stochasticity and Generalization to Broader Classes of Problems

Specialty: 1.2.2. Mathematical modeling, numerical methods and program complexes

DISSERTATION
for the Degree of Doctor of Physical and Mathematical Science

Consultant:
Gasnikov A. – Sc.D. phys.-math., RAS Corr. Memb.

Dolgoprudny — 2025

To my mother

Acknowledgments

I express my deepest gratitude to my advisor, Alexander Gasnikov. His advice guided me from undergraduate studies through my master's and PhD, and now to the defense of the habilitation. I have learned many new aspects of science and realized my potential. I am forever grateful for the experience, the opportunities, and the time I have been given.

I thank Eduard Gorbunov for his responsiveness. Without his help, I would not have been able to take my first and most important steps in science. I extend my gratitude to Martin Takac for his hospitality. Martin remains for me an example of a true applied scientist. My respect also goes to Peter Richtarik, who showed me what beautifully framed science looks like at the beginning of my journey.

I would like to recognize my young team as well. Their strong desire to engage in science non-stop energizes me each day. Thanks, (in alphabetical order) Nursultan Abdullaev, Mikhail Aleksandrov, Vladimir Aletov, Artem Andreev, Nail Bashirov, Dmitry Beresnev, Alexander Bogdanov, Dmitry Bylinkin, Semyon Chebykin, Grigoriy Evseev, Dmitrii Feoktistov, Evgeny Gurov, Igor Ignashin, Ninel Kamyshnikova, Fanis Khafizov, Leonid Levin, Daniil Medyakov, Gleb Molodtsov, Alexey Orlov, Valery Parfenov, Maksim Pechin, Egor Petrov, Alexander Pichugin, Sofya Polozova, Boris Prokhorov, Alexey Rebrikov, Nikolai Rekut, Mikhail Rudakov, Yuri Sapronov, Andrei Semenov, Kseniya Shastakova, Alexander Shestakov, Vladimir Solodkin, Sergey Stanko, Svetlana Tkachenko, Stephan Trifonov, Andrey Veprikov, Arkadiy Vladimirov, Roman Voronov, Ivan Yanko, Artur Zagitov, Philip Zmushko. On a special note, I thank my first PhD student Savelii Chezhegov and wish him all the best on his academic journey.

I would like to pay great respect to my alma mater and second home, Phystech University. It is hard to express my appreciation for all the support I received from MIPT and, in particular, from Andrei Raigorodskii and Olga Fedko. Gratitude is also directed to the Institute of System Programming of the Russian Academy of Sciences, especially to the director Arutyun Avetisyan, along with my colleagues Yury Markin and Aram Avetisyan. I also thank Innopolis University for the opportunity to develop science outside Moscow.

I thank all my co-authors for their work, fruitful discussions, and great impact on my research (in the order of appearance of joint works): Samuel Horvath, Mher Safaryan, Abdurakhmon Sadiev, Aleksandra Bazarova, Pavel Dvurechensky, Valentin Samokhin, Alexander Rogozin, Darina Dvinskikh, Dmitry Kovalev, Vasillii Novitskii,

Egor Gladin, Mohammad Alkousa, Ekaterina Borodich, Vadim Sushko, Nikolay Savelyev, Anastasia Koloskova, Sebastian Stich, Ivan Stepanov, Artyom Voronov, Rachael Tappenden, Gesualdo Scutari, Michael Diskin, Max Ryabinin, Karina Zainulina, Alexander Maslovskiy, Dmitry Pasechnyuk, Anton Novitskii, Farshed Abdukhakimov, Bin Gu, Michael Persiianov, Hugo Berard, Nicolas Loizou, Abdulla Jasem Almansoori, Farshed Abdukhakimov, Dmitry Kamzolov, Aibek Alanov, Boris Polyak, Alexander Lobanov, David Dobre, Gauthier Gidel, Sergey Samsonov, Marina Sheshukova, Alexey Naumov, Eric Moulines, Georgiy Konin, Dmitry Metelev, Anton Proskurnikov, Georgiy Kormakov, Aleksey Ustimenko, Seydamet Ablaev, Sergei Puchinin, Fedor Stonyakin, Vitali Pirau, Vladislav Matyukhin, Yaroslav Kholodov, Andrey Savchenko, Alexander Sholokhov, Roman Emelyanov, Andrey Tikhomirov, Timur Karimullin, Vladimir Ivanov, Sergey Skorik, Nikolas Khachaturov, Danil Shalagin, Yaroslav Klyukin, Ruslan Nazikov, Vladislav Minashkin, Andrey Sadchikov, Michael Ermoshin, Roman Gavrilenko, Kirill Degtyarev, Egor Maslennikov, Marat Mansurov, Ruslan Svirschevski, Denis Kuznedelev, Nikita Kornilov, Klea Ziu, Elizaveta Starykh.

Finaly, I would like to say a huge thank you to Marina for her care, patience, attentiveness and support!

Contents

1 Introduction **10**

1.1 Relevance and Importance 10

1.2 Goal and Objectives 14

1.3 Research Methodology and Methods 15

1.4 Scientific Novelty 15

1.5 Theoretical and Practical Significance 16

1.6 Main Proposals for Defense 16

1.7 Compliance with Passport of Speciality 18

1.8 Reliability 18

1.9 Presentations and Validation 19

1.10 Publications 21

1.11 Personal Contribution 22

1.12 Summary of Content and Structure 22

2 Distributed Saddle Point Problems and Variational Inequalities **25**

2.1 Upper and Lower Bounds. General Stochastic Case 29

2.1.1 Related Works and Contributions 30

2.1.2 Assumptions 32

2.1.3 Lower Bounds 33

2.1.4 Optimal Algorithms 37

2.2 Upper and Lower Bounds. Finite-Sum Stochastic Case 42

2.2.1 Contributions 42

2.2.2 Assumptions 42

2.2.3 Lower Bounds 43

2.2.4 Optimal Algorithms 45

2.2.5 Experiments 49

2.3 Local Approach 51

2.3.1 Related Works and Contributions 51

2.3.2 Algorithm 53

2.3.3 Assumptions 54

2.3.4 Convergence 54

2.3.5 Experiments 56

2.4 Using Similarity 62

2.4.1 Related Works and Contributions 62

2.4.2 Assumptions 63

2.4.3 Lower Bounds 64
2.4.4 Optimal Algorithm 65
2.4.5 Experiments 66
2.5 Compressed Communications 68
2.5.1 Related Works and Contributions 68
2.5.2 Algorithms 71
2.5.3 Extension to Stochastic Case 77
2.5.4 Extension to Federated Learning Case 80
2.5.5 Experiments 83
2.6 Combination 87
2.6.1 Related Works and Contributions 87
2.6.2 Assumptions 89
2.6.3 Algorithms 90
2.6.4 Experiments 101

3 Stochastic Saddle Point Problems and Variational Inequalities 104
3.1 Cocoercive Variational Inequalities 104
3.1.1 Assumptions 105
3.1.2 Contributions 106
3.1.3 Main Theorems 106
3.1.4 Special Cases 110
3.1.5 Beyond Unified Assumptions 120
3.1.6 Experiments 122
3.2 Optimal Batching 125
3.2.1 Related Works and Contributions 125
3.2.2 Assumptions 126
3.2.3 Algorithms 129
3.2.4 Experiments 132

4 Stochastic Frank-Wolfe 139
4.1 Stochastic Frank-Wolfe for Finite-Sum Problems 140
4.1.1 Related Works and Contributions 141
4.1.2 Assumptions 143
4.1.3 Algorithms 143
4.1.4 Experiments 149
4.2 Zero-Order Frank-Wolfe with Memory 156
4.2.1 Related Works and Contributions 157
4.2.2 Algorithms 158
4.2.3 Experiments 165
4.3 Unified Frank-Wolfe 169
4.3.1 Contributions 169

4.3.2 Main Theorems 170
4.3.3 Special Cases 172
4.3.4 Experiments 184

5 Beyond Unbiased Stochasticity 187
5.1 Variance Reduction without Full Gradients Computing via Shuffling 187
5.1.1 Related Works and Contributions 188
5.1.2 Algorithms 191
5.1.3 Experiments 194
5.2 Biased Compression for Distributed Learning 200
5.2.1 Related Works and Contributions 200
5.2.2 Assumptions 202
5.2.3 Compression Operators 202
5.2.4 Basic Algorithms 211
5.2.5 Distributed Algorithms 213
5.2.6 Experiments 217
5.3 Markovian Stochasticity 224
5.3.1 Related Works and Contributions 224
5.3.2 Assumptions 227
5.3.3 Algorithms and Lower Bounds 229

6 Sliding for Distributed and Federated Optimization 238
6.1 Sliding for Black Box Non-Smooth Case 241
6.1.1 Related Works and Contributions 242
6.1.2 Assumptions 243
6.1.3 Algorithm 244
6.1.4 Experiments 249
6.2 Sliding for First-Order Smooth Case 251
6.2.1 Contributions 251
6.2.2 Lower Bounds 252
6.2.3 Optimal Algorithms 253
6.2.4 Experiments 257

7 Conclusion 266

References 268

A Missing Details 308
A.1 Notation and Basic Facts 308
A.2 Selected Proofs 310
A.2.1 Proof of Theorems 2.2.4, 2.2.6 310
A.2.2 Proof of Theorem 2.2.8 322

A.2.3 Proof of Theorem 2.2.10 334
A.2.4 Proof of Theorem 2.4.5 354
A.2.5 Proof of Theorem 2.5.4 358
A.2.6 Proof of Theorems 3.1.2, 3.1.5, 3.1.7 368
A.2.7 Proof of Theorems 4.3.2, 4.3.3 380
A.2.8 Proof of Theorem 5.3.7 387
List of Figures 393
List of Tables 398

Chapter 1

Introduction

In this chapter, we provide a general introduction along with an overview of the results developed in this thesis. All subsequent chapters also contain their own detailed introductions.

1.1 Relevance and Importance

Numerical methods are inherently associated with computing devices. Although the first optimization methods appeared long before computers. Back in the middle of the nineteenth century, the French mathematician Augustin Cauchy invented one of the simplest optimization methods, Gradient Descent [26]. Surprisingly, this method remains an optimization cornerstone today. Despite the simplicity and intuitiveness of Gradient Descent, historically it is not the first optimization method. Even before Cauchy, Isaac Newton [27] in the second half of the seventeenth century proposed the first variant of Newton's method, which is now well known in optimization [1].

However, it is important to note that up to the middle of the twentieth century, there were no guarantees on the convergence rate of either Newton's method or Gradient Descent. Thus, the full-fledged mathematical development of numerical optimization methods began only in the middle of the last century. This was primarily due to the advent of linear programming and the explosion of popularity around this problem [28, 29]. Of course, the appearance of the first computers also had an important influence. The first attempts to prove the convergence of Gradient Descent appeared in [30]. Haskell Curry proved the asymptotic convergence of Gradient Descent for general optimization problems, but nothing about the rate of convergence was given in that paper.

The breakthrough in the development of the theory of numerical methods of optimization was made by the Soviet and Russian mathematician Boris Theodorovich Polyak in 1963: in the paper [31] Boris Theodorovich proved the first non-asymptotic estimates for the convergence rate of Gradient Descent. A year later, Boris Theodorovich developed the Heavy Ball method, also called Gradient Descent with momentum [32]. Surprisingly, this idea underlies the training of almost all neural networks.

The next significant step occurred in 1979, when Soviet mathematicians Arkadi Semyonovich

[1] It is worth noting that Newton developed his original method for finding the roots of a function of a single argument, not the optimal point.

Nemirovski and David Berkovich Yudin published their foundational work "Complexity of Problems and Efficiency of Optimization Methods" [33]. In this book, Nemirovski and Yudin proved lower bounds on the rate of convergence. The importance of this work can hardly be overestimated: thanks to Nemirovski and Yudin, community understood which methods and for which problems can be improved and which cannot.

And of course, it is worth mentioning another creator of the complexity theory of optimization methods: the Soviet mathematician Yurii Evgenyevich Nesterov. As a student of Polyak, he refined the Heavy Ball method [34] and, later, in collaboration with Nimerovski, developed the fundamental theory for solving problems with constraints [35].

Of course, numerical methods can only exist in code format on a computer. However, it is the development of theory that has elevated optimization science to a completely new level of understanding methods, their working principles, peculiarities, and the competent comparison of which method is better or worse. Time progresses, and life compels the creation of new optimization methods applicable to contemporary optimization problems. In particular, optimization formulations underlying learning problems are at the forefront. Within the framework of this thesis, we discuss three aspects of modern optimization methods. These three aspects are outlined in the title.

Distributivity. Today's realities offer solutions to computationally complex optimization problems. This is largely due to the development of machine learning and the engineers' ambition to create sophisticated artificial intelligence solutions. To achieve this, they strive for greater generalization of models and use larger architectures, trained on vast amounts of data. In this context, it is challenging to proceed without parallelization and distributed computing [36]. A well-known and understandable bottleneck for this approach is communication complexity. Due to the extra costs incurred by devices communicating with each other, perfect parallelization of the optimization process cannot be achieved. Several ideas exist to enhance communication efficiency.

The first approach involves the use of local steps. The essence of this technique is to employ a large number of updates between synchronization moments. The simplest method with local steps consists of independently running gradient descent on all devices, intermixed with occasional averaging of current points [37]. With the rapid development of distributed learning, there are more and more results around local methods. Advanced techniques emerges to deal with, for example, the heterogeneity of data on computing devices [38, 39].

Closely related to the technique of local steps is the concept of similarity of target functions on devices. It turns out that even with a small degree of homogeneity, local methods work much better [40]. Similarity can be formalized in various ways, but the most theoretically sound approach involves the proximity of Hessians [41]. This subject has been of theoretical interest for a long time. Obtaining an optimal method under similarity conditions was a major challenge, because it took decades from the first methods to obtain unimproved results.

In addition to local steps, there is a somewhat orthogonal approach that involves a lot of cheap

communication. The essence is the use of compressed forwarded information. There is also a wide range of research in this area. For the most part, the community either seeks to devise new efficient compression operators [42, 6] or aims to enhance the methods in which these compressions are integrated. Thus, there are approaches ranging from the simplest modifications of gradient descent [43] to advanced error-compression techniques [44, 45].

Distributed optimization continues to evolve. Above, we mentioned only some popular stories from this area. However, this is not an exhaustive list. In the last decade, federated learning, a branch of distributed optimization focused on private personal data, has gained significant popularity. This formulation raises numerous new questions, ranging from data protection to the personalization of results.

It is worth noting that one can still do without distributed computing for certain tasks. However, it is much more difficult to assert the same regarding the next aspect.

Stochasticity. Stochastic optimization is a relatively young and burgeoning field. However, it is worth noting that the first works surrounding Gradient Descent methods, which utilized stochastic realizations instead of a deterministic gradient, emerged in the middle of the last century [46]. This paper not only proposed the method but also proved its convergence.

From the perspective of complexity theory, the most classical and well-known result regarding the convergence of Stochastic Gradient Descent is the analysis under unbiased and variance-bounded assumptions of the stochastic gradient [47, 48]. Currently, the study of methods utilizing stochastic gradients encompasses a broad scope. The community is interested in a wide range of questions, spanning from pure theory [49, 50] to practical applications in real-world problems. Considering that the thesis is primarily based on theoretical aspects, we mention some stories on theoretical achievements related to the stochastic gradient.

In particular, research around the finite sum problem has become and remains very popular. This formulation assumes that the target function has the form of a large number of summands [51]. The randomness and inaccuracy of the gradient computation arises because we do not want to compute the full gradient, but rather limit ourselves to using only a small number of summands at each iteration. This formulation departs from the general stochastic formulation and offers advantages. It has been shown that due to the specificity of the problem, it is possible to construct methods that have more optimistic convergence estimates compared to classical SGD for the same problem [52, 53, 54]. This family of methods is named variance reduced methods. In the last decade, a large number of advanced results have emerged around this topic [55, 56, 57]. The variance reduction technique is excellent for solving small problems in classical machine learning, but unfortunately falls short of the simplest SGD in training neural networks [58]. At the same time, the idea of variance reduction has organically migrated to other areas of optimization, where it remains at the forefront to this day and does not lose its relevance [59, 60, 45].

In addition, one can mention a rather important story related to the study of convergence of stochastic gradient methods under the assumption of heavy tail noise [61, 62]. This line of research

is motivated by the challenges encountered in training modern large models [63]. Moreover, modern machine learning involves obtaining more optimistic convergence guarantees. In particular, there is a preference for high probability guarantees rather than expectation bounds, as seen in the classical case [64]. Continuing the discussion on variations of assumptions regarding randomness in SGD, one can note the results in the so-called overparameterized case, where the variance of the stochastic gradient is proportional to the norm of the honest gradient [65]. It turns out that such an assumption is also more accurate in terms of the nature of the learning process of large networks.

If we discuss a stochastic gradient, it is sometimes important to understand the causes of randomness [66]. In the case of the finite sum, it relates to the choice of a summand for calculating a part of the gradient. However, there are other approaches that involve the calculation of stochastic gradients, focusing solely on the derivative along a certain direction [67, 68]. This topic is also widely studied. A special role in this area is played by the gradient-free case when the directional derivative is approximated through a finite difference of the target function values at close points [69, 70]. This formulation makes it possible to consider two sources of stochasticity simultaneously, arising from the random direction and from the randomness in the calculation of the target function.

Generalization to Broader Classes of Problems. It remains to highlight the last main topic of this thesis related to generalization to broad classes of problems. This point is about the fact that, for obvious reasons, most past and current research in the optimization community is devoted to minimization problems. However, there are other classes of problems that are equally interesting from both theoretical and practical perspectives. Such problems include, for example, saddle point (minmax) problems. But of particular interest here are variational inequalities, which encompass both minimization and saddle point problems, while themselves representing an important class of problems with distinct applications [71].

It is clear that such classes of problems are far from new, and research surrounding them has persisted for quite a long time [72, 73]. From the perspective of numerical methods, we can mention works that transferred the idea of Gradient Descent from minimization problems to variational inequalities [74]. Unfortunately, the physics of classical Gradient Descent performs poorly even on the simplest saddle point problems [75]. There were other attempts to develop advanced gradient-like methods for saddle point problems and variational inequalities based on various techniques [76, 77]. We emphasize an elegant Extragradient scheme proposed by Galina Mikhailovna Korpelevich [78]. The essence of this method is a trial extra step inserted into the usual gradient scheme. In some sense, it is a simulation of implicit or proximal Gradient Descent. Subsequently, the idea of the Extragradient method received many different modifications [79, 80].

The area around numerical methods for solving saddle problems and variational inequalities got a new breath with the relevance of these problems in machine learning. Numerous applications emerged that extend beyond minimization problems. The methods developed 50 years ago by Korpelevich was proved to be more robust, not only for simple examples but also for complex and non-convex problems related to neural network training [81]. But it is clear that even now

there are still many open questions in this area, in particular in terms of the distribution and stochasticity mentioned above.

Thus, we arrive at the goal of this thesis.

1.2 Goal and Objectives

The goal is to study open questions in distributed and stochastic optimization, emphasizing their application to broad formulation classes such as saddle point problems and variational inequalities.

Based on the stated goal, the following objectives are established:

- Develop methods for solving distributed stochastic saddle point problems and variational inequalities in various formulations from the perspectives of both the target function/operator and the communication network. Investigate lower bounds for the formulations considered.
- Develop methods for solving distributed saddle point problems and variational inequalities using the local steps technique. Investigate their convergence also under conditions of similarity of local functions/operator.
- Develop methods for solving distributed saddle point problems and variational inequalities using compression of forwarded information. Consider the use cases of both unbiased randomized and biased waited compression. Investigate the convergence of the obtained methods.
- Combine compression and local step techniques into a single method for efficient solution of distributed saddle problems and variational inequalities. Investigate its convergence.
- Develop an optimal method for solving stochastic saddle point problems and variational inequalities of finite sum type. Consider different formulations in terms of the geometry of the problem.
- Propose a unified view on stochastic methods for saddle point problems and variational inequalities. Based on the presented theory, derive a family of new methods that transfer known techniques for minimization problems to saddle point problems and variational inequalities.
- Propose a unified view of the stochastic Frank-Wolfe method for the minimization problem with constraints. Based on the presented theory, obtain a family of new methods combining known techniques for unconstrained minimization problems with Frank-Wolfe iteration.
- Develop a variance-reduced method that addresses the disadvantages of existing analogues, specifically avoiding the calculation of the full gradient and minimizing the expenditure of additional memory. Simultaneously, the proposed method should maintain the convergence characteristics of existing advanced variance reduction techniques.

- Develop methods that solve problems involving Markovian randomness in the stochastic gradient. Investigate the lower bounds for this problem formulation, as well as the optimality of the proposed algorithms.
- Investigate the convergence of distributed methods using biased greedy compression. Show the necessity of using the error compensation technique in these methods.
- Investigate various sliding techniques for specialized formulations of distributed optimization problems.

1.3 Research Methodology and Methods

The research methodology is based on on the complex use and development of the following methods:

- Methods design. The thesis uses ideas from existing optimization methods for various problems to derive new algorithms.
- Techniques for proving upper estimates. The approaches to obtaining proofs are based on existing techniques for proving convergence of various existing algorithms under diverse settings and assumptions.
- Techniques for obtaining lower bounds. Approaches to obtain lower bounds and prove the optimality of new algorithms are based on existing examples of "bad" problems, as well as techniques of analyzing the behavior of any algorithms that work with these problematic functions.

1.4 Scientific Novelty

In terms of distributed variational inequalities, this work makes a fundamental contribution to science. Prior to this paper, this area was poorly researched, and some issues remained unstudied. The thesis aims to provide a comprehensive view of upper and lower bounds for different problem formulations of distributed variational inequalities. Moreover, the thesis is the first to investigate methods for distributed variational inequalities using compression, as well as methods under conditions of similarity of local functions/operators. In terms of stochastic algorithms for variational inequalities, the thesis develops optimal methods for solving finite sum inequalities under different geometries of the problem. In addition, a unified view of a large family of stochastic methods for solving variational inequalities in various formulations is given.

The work also closes a large number of important subjects in the field of numerical methods for minimization problems. The thesis provides a general perspective on stochastic methods based on Frank-Wolfe iteration while developing a comprehensive family of new methods. This work provides record results that significantly enhance the existing literature in the areas of Markovian optimization, variance-reduced methods, and distributed optimization with compression. The

thesis also addresses intriguing topics related to sliding techniques for obtaining record convergence guarantees for solving specialized distributed optimization problems.

1.5 Theoretical and Practical Significance

The thesis is largely a theoretical result. It provides a fundamental view of complexity theory concerning classes such as saddle point problems and variational inequalities. The thesis also addresses many specialized theoretical topics related to variance reduction methods, Markovian optimization, and distributed computing. Almost all algorithms presented in the thesis provide record convergence guarantees compared to existing competitors in the literature. Moreover, a large number of lower bounds are obtained that guarantee the optimality of many of the presented algorithms.

Many methods presented in this thesis are investigated from a practical point of view for solving problems in classical learning, game theory, and statistics. Almost all methods demonstrate their superiority over existing analogues. Moreover, some of the methods are tested on large problems of neural networks training, where our algorithms show positive results. On this basis, we can make an optimistic conclusion about the possible wide applicability of the proposed approaches in applied tasks from machine learning and beyond.

1.6 Main Proposals for Defense

The main results of the work presented for defense are summarized as follows:

- An extensive study of distributed smooth saddle point problems and variational inequalities with the Lipschitz operator. The thesis considers various formulations of these problems. From the perspective of the assumption on the stochasticity of the target function/operator, the general case with bounded variance and the finite sum case are considered. From the point of view of the communication organization, centralized communication networks and decentralized ones with fixed and time-varying topologies are studied. For each problem formulation, an algorithm and its convergence theory are provided. In addition, the thesis presents lower bounds to demonstrate the optimality and unimprovability of the proposed methods.

- Local methods for distributed variational inequalities. This thesis presents distributed algorithms that assume a large number of local steps between communications. Variations for both centralized communication and time-varying decentralized networks are considered. From a theoretical point of view, an important aspect of the thesis is the analysis of how local updates can be used efficiently when there is similarity among local operators on the devices.

- Compressed methods for distributed variational inequalities. The thesis provides algorithms that utilize unbiased and more practical biased compression operators. It shows that these

algorithms are theoretically and practically superior to the uncompressed counterparts presented in this thesis. The thesis also presents and analyzes methods that, in addition to compression, incorporate the similarity of local operators.

- A unified view on the stochastic methods for cocoercive variational inequalities. The thesis considers the Stochastic Gradient Descent-Ascent (SGDA) method and proposes a universal theorem that allows for the analysis of a whole family of methods surrounding this method. Building on the obtained theory, the convergence of existing versions of SGDA and seven new methods combining SGDA with variance reduction, compression, and coordinate selection techniques is investigated.

- Algorithms with optimal batching for solving stochastic variational inequalities of finite sum form with Lipschitz and monotone operators. The presented algorithms are adapted to both standard Euclidean geometry and arbitrary Bregman settings.

- A unified view on the stochastic Frank-Wolfe method. The thesis proposes a universal theorem that allows for the analysis of a whole family of methods based on the Frank-Wolfe concept. In addition, ten new methods are presented, which combine the Frank-Wolfe iteration with various stochastic techniques, including variance reduction, coordinate methods, and distributed computing.

- The first variance-reduced method that requires neither significant additional memory overhead nor regular computation of the full gradients of the target function. The essence of the technique is the use of a shuffling technique. It guarantees that the gradient is called over all the batches, which, in turn, allows to recover the delayed full gradient dynamically.

- Stochastic methods with Markovian stochasticity in the gradient. The thesis proposes accelerated methods that achieve advanced convergence estimates for smooth minimization problems. In addition, lower bounds guaranteeing the optimality of the proposed methods are presented. Moreover, the results are carried over to variational inequalities.

- A theoretical analysis of distributed optimization methods using biased compression operators. Several different formalizations of the biased operator are investigated. The divergence of the classical distributed gradient descent with the biased compression operator is demonstrated. The convergence of the method, with an additionally included error compression technique, is proven.

- Several sliding algorithms for solving distributed optimization problems with various features in a communication-efficient manner. The thesis proposes methods for dealing with personalized formulations and with non-smooth problems, with access only to zero-order information.

A number of the proposed results were used in the development of the courses "Numerical Optimization", "Optimization in Machine Learning", and "Machine Learning" at Moscow Institute of Physics and Technology, Moscow State University, Skoltech University, Innopolis University,

Institute for System Programming of the RAS, Institute for Information Transmission Problems of the RAS, Computing Centre of the RAS, Central University and MADE: Big Data Academy Mail.Ru Group.

1.7 Compliance with Passport of Speciality

Proposals presented for defense in the thesis correspond to the passport of specialty 1.2.2. «Mathematical modelling, numerical methods, and program complexes», in particular, to points

1. Development of new mathematical methods for modeling objects and phenomena.
2. Development, justification and testing of efficient computational methods using modern computer technology.
3. Implementation of effective numerical methods and algorithms in the form of complex problem-oriented programs for carrying out computational experiments.
4. Development of new mathematical methods and algorithms for interpreting a full-scale experiment on the basis of its mathematical model.
5. Development of new mathematical methods and algorithms for validation of mathematical models of objects based on the data of field experiments or on the basis of analysis of mathematical models.
8. Comprehensive research of scientific and technical problems using modern technology of mathematical modeling and computational experiment.

1.8 Reliability

Reliability of the results obtained in the thesis is confirmed by

- Background. The new algorithms presented in this thesis are based on existing approaches and concepts of numerical methods of optimization, which have demonstrated their effectiveness from both theoretical and practical perspectives.
- Fundamental theory. All problems considered in the thesis are clearly formalized and described using classical assumptions widely spread in the field. Convergence guarantees are proved for each proposed algorithm. Moreover, for most algorithms, it is shown that they are optimal for a certain class of problems, which strengthens the fundamentality of the results.
- Numerical experiments. Most of the methods and algorithms presented in this thesis are tested in numerical simulations on classical problems from game theory, statistics, and machine learning. Some promising methods are validated for solving complex and heavyweight problems related to neural network training.

- Community approbation. The conclusions formulated in the thesis received qualified approbation at scientific conferences and seminars. Reliability is also confirmed by publication of research results in peer-reviewed scientific journals.

The main results were obtained by the author within the scope of scientific projects supported by the following grants:

- Russian Foundation for Basic Research project 19-31-51001 «Distributed and parallel algorithms for solving data analysis problems» (2020–2021)
- Russian Science Foundation project 21-71-30005 «Development of numerical optimization methods in applications to control, inverse and learning problems» (2021–2024)
- Russian Science Foundation project 23-11-00229 «Development of effective distributed algorithms for solving optimization problems» (2023–2025)
- Goszadaniye 075-03-2024-214 «Development of federated learning algorithms: safe and efficient training of large models with applications to medical problems» (2024–2026)

1.9 Presentations and Validation

The results of this thesis were presented at the following conferences and seminars:

- 21st International Federation of Automatic Control World Congress, "On Distributed Methods for Variational Inequalities and Beyond", online, 13 July, 2020.
- 12th International Conference Optimization and Applications (OPTIMA 2021), "Near-Optimal Decentralized Algorithms for Saddle Point Problems over Time-Varying Networks", Petrovac, Montenegro, 28 September, 2021.
- 35th Annual Conference on Neural Information Processing Systems (NeurIPS 2021), "Distributed Saddle-Point Problems Under Similarity", online, 9 December, 2021.
- 13th International Conference Optimization and Applications (OPTIMA 2022), "Compression and Data Similarity: Combination of Two Techniques for Communication-Efficient Solving of Distributed Variational Inequalities", Petrovac, Montenegro, 27 September, 2022.
- 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022), "Decentralized Local Stochastic Extra-Gradient for Variational Inequalities", New Orleans, USA, 1 December, 2022.
- 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022), "Distributed Methods with Compressed Communication for Solving Variational Inequalities, with Theoretical Guarantees", New Orleans, USA, 1 December, 2022.
- 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022), "Optimal Algorithms for Decentralized Stochastic Variational Inequalities", New Orleans, USA, 1 December, 2022.

- 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022), "Optimal Gradient Sliding and its Application to Distributed Optimization Under Similarity", New Orleans, USA, 1 December, 2022.
- 26th International Conference on Artificial Intelligence and Statistics (AISTATS 2023), "Stochastic Gradient Descent-Ascent: Unified Theory and New Efficient Methods", Valencia, Spain, 26 April, 2023.
- 37th Annual Conference on Neural Information Processing Systems (NeurIPS 2023), "Similarity, Compression and Local Steps: Three Pillars of Efficient Communications for Distributed Variational Inequalities", New Orleans, USA, 13 December, 2023.
- 37th Annual Conference on Neural Information Processing Systems (NeurIPS 2023), "First Order Methods with Markovian Noise: from Acceleration to Variational Inequalities", New Orleans, USA, 13 December, 2023.
- Scientific seminar of Laboratory of Algorithms and Technologies for Networks Analysis (Higher School of Economics), "On Distributed Methods for Variational Inequalities and Beyond", Nizhny Novgorod, Russia, 17 January, 2024.
- Scientific seminar of International Laboratory of Stochastic Algorithms and High-Dimensional Inference (Higher School of Economics), "On Distributed Methods for Variational Inequalities and Beyond", Moscow, Russia, 25 January, 2024.
- Scientific seminar of Laboratory of Matrix Methods in Mathematics (Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences), "Distributed Optimization Methods with Compression", Moscow, Russia, 27 January, 2024.
- Main scientific seminar "Innopolis. Science" (Innopolis University), "Different Aspects of Communication Organization in Distributed and Federated Learning", Innopolis, Russia, 29 January, 2024.
- Scientific seminar of Department of Informatics and Computational Mathematics (Moscow Institute of Physics and Technology), "Distributed Optimization Methods with Compression", Moscow, Russia, 1 February, 2024.
- Scientific seminar of Dobrushin Mathematics Laboratory (Kharkevich Institute for Information Transmission Problems of the Russian Academy of Sciences), "On Distributed Methods for Variational Inequalities", Moscow, Russia, 6 February, 2024.
- St. Petersburg Seminar on Control Theory (Institute for Problems in Mechanical Engineering of the Russian Academy of Sciences), "On Decentralized Distributed Methods for Solving Variational Inequalities", St. Petersburg, Russia, 12 February, 2024.
- Scientific seminar "Maths of AI" of Center for Applied Artificial Intelligence (Skoltech), "Stochastic Optimization Methods for Variational Inequalities", Moscow, Russia, 16 February, 2024.

- Scientific seminar of Optimization Laboratory (Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences), "Error Compensation Technique in Distributed Optimization", Moscow, Russia, 21 February, 2024.
- Scientific seminar of Computing Systems Laboratory (Moscow State University), "Use of Compressions in Distributed Optimization: Classical and Modern Approaches", Moscow, Russia, 22 February, 2024.
- Scientific seminar of Institute for Artificial Intelligence (Moscow State University), "State-of-the-art Distributed and Federated Optimization Methods: Compression and Local Steps", Moscow, Russia, 11 March, 2024.
- Scientific seminar of Artificial Intelligence Research Institute, "Local Methods of Distributed Optimization", Moscow, 10 April, 2024.
- Scientific seminar of Yandex Laboratory (Higher School of Economics), "Local Methods of Distributed Optimization", Moscow, Russia, 17 April, 2024.
- 27th International Conference on Artificial Intelligence and Statistics (AISTATS 2024), "Stochastic Frank-Wolfe: Unified Analysis and Zoo of Special Cases", Valencia, Spain, 3 May, 2024.
- 41st International Conference on Machine Learning (ICML 2024), "Sarah Frank-Wolfe: Methods for Constrained Optimization with Best Rates and Practical Features", Vienna, Austria, 23 July, 2024.
- Conference on Probability Techniques in Analysis and Approximation Theory, "On Distributed Optimization Problems under Similarity: Optimal Algorithms and Fast Extensions", St. Petersburg, Russia, 25 November, 2024.

The research upon which this thesis is based was received several awards in the field of artificial intelligence:

- AI Leaders in the nomination "Researcher",
- Yandex ML Prize in the nomination "Yandex Researcher".

1.10 Publications

The materials of the thesis were published in 25 peer-reviewed papers belonging to the following categories:

- 15 papers in K1 journals from MIPT's own list: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15],
- 10 papers in the proceedings of A*/A conferences indexed in Web of Science and/or Scopus: [16, 17, 18, 19, 20, 21, 22, 23, 24, 25].

1.11 Personal Contribution

The content of the thesis and the main points presented for defense reflect the author's personal contributions to the published works. All results included in the thesis were obtained personally by the author. In the papers [3, 4, 20, 7, 9], the author conducted a comprehensive theoretical analysis and performed all the numerical experiments. The theoretical advances of [1, 16, 17, 18, 6, 22, 8] were fully attributable to the author of the thesis, although some of the experiments were done jointly with other authors. In the works [2, 21], the author proposed algorithms and fully obtained all upper estimates of their convergence, while the lower bounds were determined jointly with colleagues. In the papers [19, 5], the author was responsible for the results concerning the unified analyses. All special cases of the universal method, both in theory and in practice, were considered by the author with the help of colleagues. In [23], the author obtained the lower bounds. In terms of the upper estimates, the author made the final algorithms and their convergence, although the building blocks of the algorithm were obtained by the co-authors. The experiments from [23] also partially are conducted by the author. In the paper [24], the author thoroughly addressed the aspect of variational inequalities. The lower bounds from paper [10] are the author's contribution. In the survey paper [11], the author was responsible for a part of the related work [1]. In the work [12], the author is fully responsible for the lower bounds, as well as the ideas of algorithms, analysis and convergence, and the methodology of experiments. In the papers [13, 14, 25, 15], the author was responsible for the ideas of the method as well as the concept of theoretical analysis. In addition, the author participated in the development of the methodology and code of the numerical experiments for [13, 14, 25, 15].

1.12 Summary of Content and Structure

The thesis consists of an introduction, 5 chapters, a conclusion and an appendix with basic notation and proofs. The full volume of the thesis is 400 pages, including 59 figures and 29 tables. The list of literature contains 484 items. The thesis includes only selected proofs. All other proofs are available in the original papers upon which the thesis relies.

The introduction establishes the relevance of the thesis topic, formulates the research object, and outlines the goals and objectives. It describes the main results proposed for defense and provides details regarding the approbation and verification of these results by the global scientific community.

Chapter 2 is devoted to solving saddle problems and variational inequalities on distributed computing architectures. Different formulations are considered, both in terms of assumptions regarding the target functions/operators and the specifics of the communication network. This chapter is based on the papers [2, 16, 3, 17, 18, 20, 5, 8, 23, 24, 10].

Sections 2.1, 2.2 are devoted to classical problems of variational inequalities. These sections develop lower bounds for various classes of distributed variational inequality problems and optimal

algorithms that achieve these bounds.

Section 2.3 deals with more advanced algorithms that utilize the local technique. Both centralized and decentralized (fixed and changing) communication networks are considered.

Section 2.4 investigates how the additional similarity assumption of local operators/functions can be useful in terms of communication costs. New algorithms that outperform the competitors from Sections 2.1, 2.2 and 2.3 are presented. Additionally, lower bounds are provided to confirm the optimality of the proposed methods.

Section 2.5 also proposes algorithms that may outperform the methods proposed in Sections 2.1 and 2.2, as this section assumes the use of compression for transmitted information. Both unbiased randomized compression strategies and biased greedy methods are studied.

Section 2.6 combines the results of Sections 2.4 and 2.5 obtaining the most efficient methods for distributed saddle point problems and variational inequalities that simultaneously utilize both the additional similarity assumption and compression.

Chapter 3 continues the study variational inequalities, but is not necessarily devoted to distributed formulations, and deals with special stochastic formulations. The main results of the chapter were published in the works [19, 4, 13, 14].

Section 3.1 studies a special class of variational inequalities problems, namely cocoercive inequalities. The section offers a unified view of stochastic methods for solving these problems and, with the help of a general theoretical framework, analyzes a large family of methods, ranging from variance reduction to coordinate methods.

Section 3.2 returns to classical variational inequalities and studies the case when the target operator is of the sum type. In this setting, we propose a stochastic method that is not sensitive to the batch size from the perspective of computational complexity. The methods are presented for both classical Euclidean geometry and non-Euclidean formulations.

Chapter 4 departs from the solution of variational inequalities and proposes solving classical minimization problems with constraints. The well-known Frank-Wolfe method is considered. Stochastic variations of this method are of particular interest. This chapter is based on the papers [22, 25, 15].

In Section 4.1, a combination of the Frank-Wolfe method and the well-known SARAH variance reduction technique is proposed. The theoretical analysis and numerical experiments demonstrate the superiority of this method over existing analogs.

In Section 4.2, a zero-order modification of the classical Frank-Wolfe method is proposed. In particular, a combination of this method and the technique of memorizing past calculations of derivatives is presented.

In Section 4.3, we propose to look at the stochastic Frank-Wolfe method in a unified way. By introducing universal assumptions, a single convergence theorem is constructed. It allows us to

analyze various methods, including those studied in Sections 4.1 and 4.2.

Chapter 5 departs from the classical assumption of stochastic optimization theory regarding the independence and unbiasedness of the stochastic gradient. We consider various scenarios where either or both of these properties are absent. The main results of the chapter were published in the works [21, 6, 7].

Section 5.1 discusses variance-reduced methods in which a shuffling technique is utilized instead of independent sampling. This technique addresses a critical issue of existing variance reduction methods, namely the computation of the full gradient.

Section 5.2 is devoted to the use of biased compressed compressive operators in distributed optimization. This section examines various formalizations of biased compression operators and introduces additional error compression techniques to enhance the convergence of distributed optimization methods with compression.

Section 5.3 deals with stochastic methods involving Markovian stochastic gradients. This section covers both accelerated optimal algorithms for solving such problems and lower bound estimates that prove the optimality of the proposed approaches. An important detail is that both minimization problems and variational inequalities are considered.

The final chapter (Chapter 6) contains two stories regarding the sliding technique for splitting oracle complexity in distributed optimization. The main results of the chapter were published in the papers [1, 9, 11, 12].

Section 6.1 studies sliding for non-smooth distributed optimization problems. Moreover, the formulation of gradient-free optimization for this kind of problems is considered.

Section 6.2 is devoted to personalized formulations of federated learning problems. In this case, the sliding technique allows us to organize communication rounds between devices much less frequently, given the non-negligible degree of personalization of the result.

Chapter 2

Distributed Saddle Point Problems and Variational Inequalities

Variational inequalities have a rather long history of research. Their first applications were the equilibrium problems in economics and game theory [82, 83, 84]. Variational inequalities were seen as a universal paradigm, involving other attractive problems, including minimization, saddle point problems, fixed point problems, and others. Whereas research for minimization was actively developed separately, research for saddle point problems was often coupled with the development of science around variational inequalities. This trend continues even now. This is partly related to the fact that the methods for minimization problems are not always suitable for saddle point problems and variational inequalities. Moreover, from the theoretical point of view, these methods give relatively poor or even zero guarantees of convergence. Therefore, a separate branch of algorithms has been developed for variational inequalities and saddle point problems. Such methods include, e.g. Extragradient [78, 80, 85], Tseng's method [86] and many others [75].

Over time, variational inequalities found new applications, e.g. in signal processing [71]. In the last decade, they also attracted the interest of the machine learning community. Nowadays, variational inequalities and saddle point problems are used in various aspects of machine learning. For example, there are numerous applications for the already classical learning tasks in supervised learning (with non-separable loss [87]; with non-separable regularizer [88]), unsupervised learning (discriminative clustering [89]; matrix factorization [90]), image denoising [91, 92], and robust optimization [93]. It turns out that variational inequalities can be also linked with modern areas of learning science, including reinforcement learning [94, 95], adversarial training [96], and GANs [97, 98, 81, 99, 100, 101, 102] to name a few.

Machine learning's rapid development offers more possibilities for consumers but poses challenges for engineers. To meet modern society's demands, the performance and generalization capacity of the model must be improved. This is currently achieved by increasing training data and model size [103]. Hardware acceleration and, most importantly, distributed algorithm design are essential for effectively handling these workloads [36].

Thus, it becomes clear that to solve the distributed problem, it is necessary to construct a distributed algorithm that combines local computation on each of the devices and communication between them. Such an algorithm can be organized as follows: all devices communicate only with

the main device (server). This approach is called centralized. The main problem is the importance of the server – it can crash and interrupt the whole process. Therefore, along with the centralized approach, the decentralized [104] one is also popular. In this case, all devices are connected by a network, and communication occurs along the edges of this network. In the thesis, we focus on both settings. We formally introduce them in the corresponding sections below.

The distributed nature of the problem arises not only from a desire to accelerate the learning process through parallelization but also from severe external constraints, as seen in federated learning [105, 106]. Private data transfer from local devices is prohibited due to privacy constraints, making distributed approaches the only practical option. Most distributed algorithms have communications as a bottleneck. Communication wastes time, slowing down training/optimization and preventing effective parallelization. Different approaches can reduce communication rounds and cost per round. The first popular approach is the compression of transmitted information [43]; it helps to increase the speed of the communication round without much loss in the quality of learning. The second approach is the use of local steps [107]; by increasing the number of local computations, it is sometimes possible to significantly reduce the number of communication rounds. Finally, the third basic approach relies on data similarity [108]. Local steps are often employed to help with this approach. Data similarity allows us to assume local loss functions on devices are similar, meaning local computations are close and replaceable. Both approaches and combinations have been extensively researched for minimizations, but a synergy of all three pillars of distributed algorithms has not been presented for either minimization problems or variational inequalities.

In this chapter, we proceed step by step through different formulations of distributed variational inequalities. We first consider a general setting and aim to obtain lower bounds and algorithms that achieve these bounds. Then, we examine additional features, such as local steps, similarity, and compressions. Finally, we consolidate all results into one efficient method.

Now let us formalize the problem. We study variational inequalities of the form

$$\text{Find} \quad z^* \in \mathbb{R}^d \quad \text{such that} \quad \langle F(z^*), z - z^* \rangle + g(z) - g(z^*) \geq 0, \quad \text{for all} \quad z \in \mathbb{R}^d, \tag{2.1}$$

where $F : \mathbb{R}^d \to \mathbb{R}^d$ is an operator, and $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a proper lower semicontinuous convex function. We also assume that g is proximal friendly, i.e. the computation of the operator $\text{prox}_{\rho g}(z) = \arg\min_{y \in \mathbb{R}^d} \{\rho g(y) + \frac{1}{2}\|y - z\|^2\}$ (with $\rho > 0$) is done for free or costs very low.

To showcase the expressive power of the formalism (2.1), we now provide a few examples of variational inequalities arising in machine learning.

Example 2.0.1 (Minimization)**.** Consider the convex regularized minimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) + g(x), \tag{2.2}$$

where f is typically a smooth data-fidelity term, and g a possibly nonsmooth regularizer. If we

define $F(z) := \nabla f(x)$, then it can be proved that $x^* \in \operatorname{dom} g$ is a solution for (2.1) if and only if $x^* \in \operatorname{dom} g$ is a solution for (2.2). So, the regularized optimization problem (2.2) can be cast as a VI (2.1). See [84, Section 1.4.1].

As mentioned, while minimization problems are widely studied in separate literature, the next class of problems is more strongly tied to variational inequalities.

Example 2.0.2 (Saddle point)**.** Consider the convex-concave saddle point problem

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} f(x, y) + g_1(x) - g_2(y), \tag{2.3}$$

where g_1 and g_2 can also be interpreted as regularizers.
If we let $F(z) := F(x, y) = [\nabla_x f(x, y), -\nabla_y f(x, y)]$ and $g(z) = g(x, y) = g_1(x) + g_2(y)$, then it can be proved that $z^* \in \operatorname{dom} g$ is a solution for (2.1) if and only if $z^* \in \operatorname{dom} g$ is a solution for (2.3). So, convex-concave saddle point problems (2.3) can be cast as a variational inequality (2.1). See [84, Section 1.4.1]. It gives that convex-concave saddle point problems (2.3) can be investigated via a reformulation in the form of the variational inequality (2.1).

Saddle point problems are strongly related to variational inequalities. In particular, lower bounds for the former are also valid for the latter. Moreover, upper bounds for variational inequalities are valid for saddle point problems. However, what is perhaps more important is that these lower and upper bounds match. This is in contrast to minimization, where the lower bounds are weaker.

One more example is the fixed-point problem.

Example 2.0.3 (Fixed point problem)**.** Consider the fixed point problem:

$$\text{Find} \quad z^* \in \mathbb{R}^d \quad \text{such that} \quad T(z^*) = z^*, \tag{2.4}$$

where $T : \mathbb{R}^d \to \mathbb{R}^d$ is an operator. With $F(z) = z - T(z)$, it can be proved that $z^* \in \mathbb{R}^d$ is a solution for (2.1) if and only if $F(z^*) = 0$, i.e. $z^* \in \mathbb{R}^d$ is a solution for (2.4).

In the view of machine learning, the most widely used example is the empirical risk minimization problem [51]:

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} l(\text{model}(x, a_i), b_i) + \frac{\lambda}{2} \|x\|^2, \tag{2.5}$$

where $\text{model}(x, a_i)$ is the output of a machine learning model (e.g. linear regression model or neural network) with parameters x on the input a_i, b_i is the label for the object a_i, l is the loss function, and n is the number of samples. One can extend the empirical risk minimization problem (2.5). Many machine learning models, including neural networks [109], are vulnerable to adversarial examples, where nearly indistinguishable inputs force the model to produce very different outputs. To improve the model's resistance to adversarial attacks, one can formulate the following saddle

point problem [96]:

$$\min_{x\in\mathbb{R}^{d_x}} \max_{\|r_i\|\leq\mathbb{R}^{d_a}} \frac{1}{n}\sum_{i=1}^{n} l(\text{model}(x, a_i + r_i), b_i) + \frac{\lambda}{2}\|x\|^2 - \frac{\beta}{2}\|r_i\|^2, \tag{2.6}$$

where r_i is the adversarial noise applied to the input a_i.

Since the purpose of this chapter is distributed variational inequalities, we assume that F from (2.1) has a finite sum structure:

$$F(z) := \frac{1}{M}\sum_{m=1}^{M} F_m(z). \tag{2.7}$$

This means that the target operator is distributed among M devices, and the mth device only has access to F_m.

Next, we list two key assumptions - both are standard in the literature on variational inequalities.

Assumption 2.0.4 (Lipschitzness)**.** The operator F is L-Lipschitz continuous, i.e. for all $z_1, z_2 \in \mathbb{R}^d$ we have $\|F(z_1) - F(z_2)\| \leq L\|z_1 - z_2\|$.

Each operator F_m is L_m-Lipschitz continuous, i.e. for all $z_1, z_2 \in \mathbb{R}^d$ it holds $\|F_m(z_1) - F_m(z_2)\| \leq L_m\|z_1 - z_2\|$. Let us define new constant $\tilde{L}$ as follows $\tilde{L}^2 = \frac{1}{M}\sum\limits_{m=1}^{M} L_m^2$ and $L_{\max}$ as $L_{\max} = \max L_m$.

For saddle point problems, these properties are equivalent to smoothness.

Assumption 2.0.5 (Monotonicity)**.** We need three cases of monotonicity.

(SM) Strong monotonicity. The operator F is μ-strongly monotone, i.e. for all $z_1, z_2 \in \mathbb{R}^d$ we have $\langle F(z_1) - F(z_2), z_1 - z_2\rangle \geq \mu\|z_1 - z_2\|^2$.

(M) Monotonicity. The operator F is monotone, i.e. for all $z_1, z_2 \in \mathbb{R}^d$ we have $\langle F(z_1) - F(z_2), z_1 - z_2\rangle \geq 0$.

(NM) Non-monotonicity. The operator F is non-monotone (minty), if and only if there exists $z^* \in \mathbb{R}^d$ such that for all $z \in \mathbb{R}^d$ we have $\langle F(z), z - z^*\rangle \geq 0$.

The last assumption is called the minty or variational stability condition. It is not a general non-monotonicity, but it is already associated in the community with non-monotonicity [110, 111, 99, 112, 113, 114, 115], particularly within a setup that is somewhat appropriate for GANs [116, 117, 118, 119].

We need to make other assumptions; however, these will be detailed in the appropriate sections.

We also need to introduce a convergence criterion for the monotone case of (2.1). In this setting, we focus on the following quantity called a restricted gap-function [120] defined for any $z \in \mathbb{R}^d$

and any bounded $C \subset \mathbb{R}^d$:

$$\mathrm{Gap}_C(z) := \max_{u \in \mathcal{C}} \left[\langle F(u), z - u \rangle + g(z) - g(u) \right]. \tag{2.8}$$

Here we do not take the maximum over the entire set $\mathbb{R}^d$, but over $\mathcal{C}$ – a compact subset of $\mathbb{R}^d$. Thus, we can also consider unbounded sets $\mathbb{R}^d$. This is permissible, since such a version of the criterion is valid if the solution z^* lies in $\mathcal{C}$; for details see the work of [120].

In this chapter, we consider two cases of stochastic settings of (2.1)+(2.7): the general and the finite sum cases. The simpler deterministic case immediately follows from these two by removing stochasticity from the formulation. In each setting, we internally study three setups in terms of the communication graph: centralized, decentralized fixed, and decentralized time-varying.

Let us note some of the terms we use in the chapter. By local steps/updates, we mean that some device m, using only local information about operator F_m, changes the local value of variable z. Local computations/calculations includes all computations of F_m on a local device: local updates/steps may take place, or the value of the operator is accumulated and is sent to the other device without changes in local variables z. The local device m complexity we mean the number of local computations of F_m during the entire operation of algorithms, and by the total local complexity – the total number of computations of F_m for all m. A communication round refers to establishing a connection between two devices and transmitting information in one or both directions. When multiple devices communicate in parallel—for instance, if all nodes send information to a central device—it is counted as a single communication round. Since we employ sparsifiers as compressors, we quantify the transmitted information by the number of coordinates sent. Communication complexity is the total volume of information exchanged between all devices during the execution of the algorithm. Communication time, on the other hand, is the duration spent on all such communication rounds.

2.1 Upper and Lower Bounds. General Stochastic Case

We consider the following operators F_m from (2.1)+(2.7):

$$F_m(z) := \mathbb{E}_{\xi_m \sim \pi_m}[F_m(z, \xi_m)] \tag{2.9}$$

and suppose that we have access to realizations of the operator $F_m(z, \xi_m)$, but not to $F_m(z)$ itself. This is due to the fact that, from the perspective of machine learning, the distribution π_m is an unknown target distribution. Consequently, we cannot compute the mathematical expectation directly, but can only work with some realizations on the objects of the training sample.

To solve (2.1)+(2.7)+(2.9), it becomes necessary to construct a distributed algorithm that combines computations on each of the devices and communication between them. Such an algorithm can be organized as follows: all devices communicate only with the main device (server). This approach is called centralized. The main problem is the importance of the server – it can crash

and interrupt the whole process. Therefore, alongside the centralized approach, the decentralized [104] one is also popular. In this case, all devices are connected by a network, communication occurs along the edges of this network.

2.1.1 Related Works and Contributions

Methods for solving centralized minimization problems have a substantial literature background. This is due to the fact that any non-distributed method can be considered as a centralized one. Therefore, we do not attempt to summarize all that has been accomplished in centralized minimization. Decentralized optimization on both fixed and time-varying graphs is a narrow and specialized area. Although, in the case of minimization, it can be regarded as well studied.

• **Minimization on static networks.** Decentralized methods are based on combining iterations of classical first-order methods with communication steps via the gossip protocol. One of the first distributed gradient dynamics studied in the literature [121], where, in fact, it alternates between the gossip protocol step and the usual Gradient Descent. Later, the original idea of EXTRA was introduced, proposing the use of two different gossip matrices: the original and a modified by unit matrix [122]. These ideas are grounded in the Gradient Descent method; thus, they yield convergence results that are far from optimal. The development of accelerated methods relies on two main approaches: primal and dual. Dual methods reformulate the original minimization problem using a Laplacian matrix induced by the communication network. The basic idea behind the dual approach is to apply first-order methods to this dual reformulation. This was realized in [123], where Chebyshev acceleration was also used to accelerate the consensus [124]. When it comes to primal methods, the simplest idea, namely a direct combination of Nesterov acceleration and gossip iteration was proposed in [125]. The algorithm from this work resembles another popular scheme in distributed decentralized optimization called Gradient Tracking, the essence of which is to maintain consensus not only on the optimization variables but also on the current gradients [125, 126, 127, 128, 129, 130, 131]. Additionally, we can mention works that accelerate Extra using the Catalyst envelope [132]. Based on gradient tracking, Nesterov acceleration, and Chebyshev acceleration, the MUDAG scheme was developed, which is one of the cutting-edge schemes in the literature [127, 133]. Finally, we would like to mention the works that proposed a direct method APAPC, based on the idea of saddle point reformulation of the decentralized minimization problem [134].

• **Minimization on time-varying networks.** Some schemes from the constant topology case have been directly transferred to time-varying graphs. This includes, for example, the simplest idea [135] and the technique of gradient tracking [121]. Next, we highlight PANDA, which is a dual method capable of working over time-varying graphs [136]. Analysis of Gradient Tracking and PANDA assumes that the underlying network is B-connected, that is, the union of B consequent networks is connected, while the network is allowed to be disconnected at some steps. Considering the time-varying graphs which stay connected at each iteration, decentralized Nesterov method [137] has an accelerated rate under the condition that graph changes happen rarely enough, ADOM

[138] and ADOM+ [139] are first-order optimization methods which achieve lower complexity bounds [139]. APM-C [140], Acc-GT [124] are accelerated methods over time-varying graphs, as well.

• **Applications of decentralized optimization.** Decentralized algorithms find applications in sensor networks [141], network resource allocation [142], cooperative control [143], distributed spectrum sensing [144], distributed tracking [145], power system control [146, 147], formation control [148, 149, 150], distributed load balancing [151, 152] and, of course, in machine learning [153, 154, 155].

Next, we highlight the necessary works on variational inequalities and saddle point problems.

• **Distributed variational iniqulities and saddle point problems.** It is worth noting that there are not many works on distributed optimization for these classes. In the decentralized case, we can mention works proposing deterministic methods [112, 156] as well as stochastic methods [117]. As for centralized approaches, in addition to the originally non-distributed methods that can be directly transferred to the centralized case, we note the following works [157, 158].

• **Lower bounds for variational inequalities.** Since in this section we move towards optimality of proposed algorithms, it is important to understand how lower bounds are organized for the class of problems known as variational inequalities. Typically, these lower bounds are obtained by considering special cases of saddle point and minimization problems. In the non-distributed case, the lower bounds for smooth strongly convex-strongly concave saddle point problems are given in [159], while those for convex-concave problems are presented in [160]. In smooth stochastic convex optimization, we highlight works on lower bounds [33, 161]. It is also important to note the works devoted to the lower bounds for centralized and decentralized distributed convex optimization [123, 162].

We are now ready to summarize the contributions of our work.

• **Lower bounds.** We present lower bounds for distributed general stochastic strongly monotone and monotone variational inequalities in both distributed settings: centralized and decentralized (static and time-varying). In particular, for a given budget on the number of communications and on the number of local computations for each node, we provide the lower bound on the resulting accuracy. From such kind of estimates, given the accuracy, we can solve the opposite problem and obtain estimates on the number of communications and local oracle calls.

• **Optimal algorithms.** Next, we obtain the optimal algorithms. For the centralized problem, we construct our method based on the Extragradient method [85] (a classical and optimal method for non-distributed problems) with the correct batch size. In the decentralized case, we also use the Extragradient method as a basis, but communication is done using the (accelerated) gossip consensus procedure [163].

For the summary and comparison of the lower and upper convergence rate bounds, we refer to Table 2.1.

• **Non-monotone analysis.** We analyze our new algorithms not only in the monotone case but also in the non-monotone case under the Minty assumption [164, 115]. Minty is the weakest additional assumption for a non-monotone problem found in the literature. Under this weak assumption only a few results on distributed methods are available in the literature [112, 117]. In particular, our analysis covers the estimates of the decentralized but deterministic method from [112] and also generalizes and overlaps the estimates for the stochastic method for homogeneous data ($F_m = F$) from [117].

Table 2.1: Lower and upper bounds for distributed smooth general stochastic strongly monotone (sm) or monotone (`c`) `variational inequalities in the centralized and decentralized (constant and time-varying) cases.`

	lower	**upper**
	`centralized`	
`sm`	$\Omega\left(R_0^2 \exp\left(-\frac{32\mu \min\{K;T\}}{L\Delta}\right) + \frac{\sigma^2}{\mu^2 MT}\right)$	$\tilde{\mathcal{O}}\left(R_0^2 \exp\left(-\frac{\mu \min\{K;T\}}{4L\Delta}\right) + \frac{\sigma^2}{\mu^2 MT}\right)$
`m`	$\Omega\left(\frac{L\Omega^2\Delta}{K} + \frac{\sigma\Omega}{\sqrt{MT}}\right)$	$\mathcal{O}\left(\frac{L\Omega_z^2\Delta}{\min\{K;T\}} + \frac{\sigma\Omega}{\sqrt{MT}}\right)$
	`decentralized (constant)`	
`sm`	$\Omega\left(R_0^2 \exp\left(-\frac{128\mu \min\{K;T\}}{L\sqrt{\chi}}\right) + \frac{\sigma^2}{\mu^2 MT}\right)$	$\tilde{\mathcal{O}}\left(R_0^2 \exp\left(-\frac{\mu \min\{K;T\}}{8L\sqrt{\chi}}\right) + \frac{\sigma^2}{\mu^2 MT}\right)$
`m`	$\Omega\left(\frac{L\Omega^2\sqrt{\chi}}{\min\{K;T\}} + \frac{\sigma\Omega}{\sqrt{MT}}\right)$	$\tilde{\mathcal{O}}\left(\frac{L\Omega^2\sqrt{\chi}}{\min\{K;T\}} + \frac{\sigma\Omega}{\sqrt{MT}}\right)$
	`decentralized (time-varying)`	
`sm`	$\Omega\left(R_0^2 \exp\left(-\frac{256\mu \min\{K;T\}}{L\chi}\right) + \frac{\sigma^2}{\mu^2 MT}\right)$	$\tilde{\mathcal{O}}\left(R_0^2 \exp\left(-\frac{\mu \min\{K;T\}}{8L\chi}\right) + \frac{\sigma^2}{\mu^2 MT}\right)$
`m`	$\Omega\left(\frac{L\Omega^2\chi}{\min\{K;T\}} + \frac{\sigma\Omega}{\sqrt{MT}}\right)$	$\tilde{\mathcal{O}}\left(\frac{L\Omega^2\chi}{\min\{K;T\}} + \frac{\sigma\Omega}{\sqrt{MT}}\right)$

Notation: L – Lipschitz constant of F, μ – strong monotonicity constant, $R_0^2 = \|z^0 - z^*\|^2$, Ω – diameter of the optimization set, Δ, χ – diameter and condition number of the communication graph (in time-varying case maximum of all graphs), K – number of communication rounds, T – number of local calls of the gradient oracle on each node. In the monotone case, the bounds are in terms of the gap function (see (2.8)), in the strongly monotone case – in terms of the (squared) distance to the solution.

2.1.2 Assumptions

Let us introduce the assumptions for our setting (2.9).

Assumption 2.1.1 (Boundness of variance)**.** $F_m(z, \xi)$ is unbiased and has bounded variance, i.e. for all $z \in \mathcal{Z}$ it holds that

$$\mathbb{E}_{\xi \sim \mathcal{D}_m}[F_m(z, \xi)] = F_m(z), \quad \mathbb{E}_{\xi \sim \mathcal{D}_m}[\|F_m(z, \xi) - F_m(z)\|^2] \leq \sigma^2. \tag{2.10}$$

Assumption 2.1.2 (Compactness)**.** $\mathcal{Z}$ is compact, i.e. for all $z, z' \in \mathcal{Z}$

$$\|z - z'\| \leq \Omega_z.$$

Typically, decentralized communication is realized via a *gossip protocol* [165, 166, 121], which is merely matrix-vector multiplication with a gossip matrix W, described below, which is different in the fixed and time-varying cases. Let $\mathcal{L} = \{u_1, \ldots, u_M) \in \mathbb{R}^M : u_1 = \ldots = u_M\}$ be the *consensus space*.

Assumption 2.1.3 (Fixed network [123])**.** For a fixed network, communication can be modeled via an undirected connected graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = [M]$ are vertices (workers) and $\mathcal{E} = \{(i, j) \,|\, i, j \in \mathcal{V}\}$ are edges. Note that $(i, j) \in \mathcal{E}$ if and only if there exists a communication link between agents i and j. The gossip matrix W satisfies the following three assumptions: 1) W is symmetric positive semi-definite; 2) $\ker W \supset \mathcal{L}$; 3) W is supported on the vertices and edges of the network only: $w_{i,j} \neq 0$ if and only if $i = j$ or $(i, j) \in \mathcal{E}$.

To characterize the matrix W, which captures the properties of the network, we denote $\lambda_{\max}(W) = 1$ as the maximum eigenvalue of W, $\lambda^+_{\min}(W)$ as the minimum positive eigenvalue of W, and the characteristic number $\chi = \lambda_{\max}(W)/\lambda^+_{\min}(W) = 1/\lambda^+_{\min}(W)$.

Assumption 2.1.4 (Time-varying network [167])**.** For a time-varying network, at any moment t, communication network can be modeled as a directed B-connected graph, $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{E}(t) = \{(i, j) \,|\, i, j \in \mathcal{V}\}$ are directed edges. B-connectedness means that for any time t, the graph $\mathcal{G}_B(t)$ with the set of edges $\bigcup_{\tau=t}^{t+B-1} \mathcal{E}(t)$ is connected. To describe the gossip protocol for time-varying case, we define the multi-consensus gossip matrix

$$W_T(t) = I - \prod_{\tau=t}^{t+T-1} (I - W(\tau)).$$

One can also observe that multiplication with the matrix W_T requires to perform multiplication with T gossip matrices $W(t), \ldots, W(t + T - 1)$, i.e., it requires T decentralized communications. We further assume that the gossip matrices $W(t)$ (for $\mathcal{G}(t)$) and $W_B(t)$ satisfy: 1) $W(t)$ is supported on the nodes and edges of the network: $w_{i,j}(t) \neq 0$ if and only if $i = j$ or $(i, j) \in \mathcal{E}(t)$; 2) $\ker W(t) \supset \mathcal{L}$; 3) $\text{range} W(t) \subset \{u \in \mathbb{R}^M : \sum_{m=1}^M u_m = 0\}$; 4) there exists a characteristic number $\chi \geq 1$ such that $\|W_B(t)u - u\|^2 \leq (1 - \chi^{-1})\|u\|^2$ for all $z \in \text{range} W_B(t)$.

2.1.3 Lower Bounds

Our lower bounds apply to a specific class of algorithms that are, loosely speaking, permitted to communicate with neighbors and compute any local first-order information. We now provide a formal definition.

Definition 2.1.5 (Oracle)**.** Each agent m has its own local memory $\mathcal{M}_m$ with initialization $\mathcal{M}_m = \{0\}$. $\mathcal{M}_m$ is updated as follows. At each iteration, the algorithm either performs local computations or communicates.

• **Local computation:** At each local iteration, device m can sample uniformly and independently ξ_m for $F_m(z'', \xi_m)$ and adds to its $\mathcal{M}_m$ a finite number of points z, satisfying

$$z \in \text{span}\left\{z' \ , \ F_m(z'', \xi_m), \text{prox}_{\rho g}\left(\text{span}\left\{z''' \ , \ F_m(z'', \xi_m)\right\}\right)\right\}$$

for $z', z'', z''' \in \mathcal{M}_m$ and $\rho > 0$.

• **Communication:** Upon communication rounds among the neighboring nodes, and at communication time t, $\mathcal{M}_m$ is updated according to

$$\mathcal{M}_m := \text{span}\left\{\bigcup_{(i,m)\in\mathcal{E}(t)} \mathcal{M}_i\right\}.$$

• **Output:** The final global output is calculated as $\hat{z} \in \text{span}\left\{\bigcup_{m=1}^{M} \mathcal{M}_m\right\}$. We assume that each node makes no more than T local iterations (for simplicity, that exactly T) during the operation of the algorithm. The number of communication rounds is also limited to a certain number of $K < T$.

The structure of the above definition is typical for distributed lower bounds [123] and for stochastic lower bounds [168].

Following the classical results on obtaining lower bounds, it is sufficient to provide an example of a «bad» problem. As already mentioned, for variational inequalities, bad examples can be found among saddle point problems and minimization as special cases. In particular, we seek to identify a «bad» function and the «bad» partitioning of this function between nodes [123]. We begin with the «bad» function. The first important point in constructing of our «bad» function is that it consists of two independent parts: deterministic and stochastic. Consider $f_m(x, y) = f_m^{deter}(x^{deter}, y) + f^{stoch}(x^{stoch})$, where the vectors x^{deter} and x^{stoch} together give the vector $x = (x^{deter}, x^{stoch})$. At the same time, we have access to $F_m(x, y, \xi) = F_m^{deter}(x^{deter}, y) + \nabla f^{stoch}(x^{stoch}, \xi)$. This means that we have a deterministic oracle for f_m^{deter} and a stochastic one – for f^{stoch}. Such f_m helps to rewrite the original problem (2.1)+(2.7) as follows:

$$\min_{x^{deter}\in\mathcal{X}^{deter}} \max_{y\in\mathcal{Y}} \frac{1}{M}\sum_{m=1}^{M} f_m^{deter}(x^{deter}, y) + \min_{x^{stoch}\in\mathcal{X}^{stoch}} f^{stoch}(x^{stoch}). \tag{2.11}$$

Therefore, we separately prove the estimates for each of the problems and then combine them.

Deterministic lower bounds

In this part, we provide lower bounds for the centralized (Theorem 2.1.6) and decentralized (Theorem 2.1.7, Theorem 2.1.8) cases.

Theorem 2.1.6. For any $L > \mu > 0$ and any connected graph, there exists a distributed saddle point problem on $\mathcal{X} \times \mathcal{Y} = \mathbb{R}^d \times \mathbb{R}^d$ with $x^*, y^* \neq 0$ over a fixed network, such that the following statements hold:

- the diameter of the network is equal to Δ,
- $f = \frac{1}{M} \sum\limits_{m=1}^{M} f_m$ is L-smooth, μ-strongly convex-strongly concave ,
- size $d \geq \max\left\{2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right), 2K\right\}$, where $\alpha = \frac{4\mu^2}{L^2}$ and $q = \frac{1}{2}\left(2 + \alpha - \sqrt{\alpha^2 + 4\alpha}\right) \in (0; 1)$,
- for any output $\hat{x}, \hat{y}$ of any procedure satisfying Definition 2.1.5, the following estimate hold:

$$\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2 = \Omega\left(\exp\left(-\frac{4\mu}{L-\mu} \cdot \frac{K}{\Delta}\right)\|y^0 - y^*\|^2\right).$$

Theorem 2.1.7. For any $L > \mu > 0$ and any connected graph, there exists a distributed saddle point problem on $\mathcal{X} \times \mathcal{Y} = \mathbb{R}^d \times \mathbb{R}^d$ with $x^*, y^* \neq 0$ over a fixed network characterized by a gossip matrix, such that the following statements hold:

- the gossip matrix have the condition number χ,
- $f = \frac{1}{M} \sum\limits_{m=1}^{M} f_m$ is L-smooth, μ-strongly convex-strongly concave,
- size $d \geq \max\left\{2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right), 2K\right\}$, where $\alpha = \frac{4\mu^2}{L^2}$ and $q = \frac{1}{2}\left(2 + \alpha - \sqrt{\alpha^2 + 4\alpha}\right)$,
- for any output $\hat{x}, \hat{y}$ of any procedure satisfying Definition 2.1.5, the following estimate hold:

$$\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2 = \Omega\left(\exp\left(-\frac{128\mu}{L-\mu} \cdot \frac{K}{\sqrt{\chi}}\right)\|y^0 - y^*\|^2\right).$$

Theorem 2.1.8. For any $L > \mu > 0$ and any sequence of time-varying connected any moment graphs, there exists a distributed saddle point problem on $\mathcal{X} \times \mathcal{Y} = \mathbb{R}^d \times \mathbb{R}^d$ with $x^*, y^* \neq 0$ over a time-varying network characterized by a sequence of gossip matrices, such that the following statements hold:

- all gossip matrices have the condition number not greater than χ,
- $f = \frac{1}{M} \sum\limits_{m=1}^{M} f_m$ is L-smooth, μ-strongly convex-strongly concave,
- size $d \geq \max\left\{2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right), 2K\right\}$, where $\alpha = \frac{4\mu^2}{L^2}$ and $q = \frac{1}{2}\left(2 + \alpha - \sqrt{\alpha^2 + 4\alpha}\right)$,
- for any output $\hat{x}, \hat{y}$ of any procedure satisfying Definition 2.1.5, the following estimate hold:

$$\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2 = \Omega\left(\exp\left(-\frac{256\mu}{L-\mu} \cdot \frac{K}{\chi}\right)\|y^0 - y^*\|^2\right).$$

Convex-concave case. Note that in the convex-concave case, the problem is usually considered on a bounded set (see Assumption 2.1.2). Moreover, the convergence criterion for algorithms is

formulated in terms of the following gap function ((2.8) for saddle point problems):

$$\text{Gap}(x, y) := \max_{y' \in \mathcal{Y}} f(x, y') - \min_{x' \in \mathcal{X}} f(x', y).$$

Therefore, the lower bounds are also needed in terms of (2.8). Following the inequality 6 of [159], we can rewrite the estimates from Theorems 2.1.6 and 2.1.7 as follows:

$$\text{Gap}(x, y) \geq \frac{\mu}{2}\|x - x^*\|^2 + \frac{\mu}{2}\|y - y^*\|^2. \tag{2.12}$$

In Table 2.1, the lower bounds for the convex-concave case are already declared. Let us provide an intuition for how these estimates can be obtained from Theorems 2.1.6 and 2.1.7.

One way to get the lower estimates is to prove by contradiction and assume that the estimate given in Table 2.1 is not valid: there exists an algorithm that converges better than the given bound, namely for some numerical constant C the output of this method after k communications satisfies

$$\text{Gap}(\hat{x}^k, \hat{y}^k) < \frac{CL\Delta(\|x^0 - x^*\|^2 + \|y^0 - y^*\|^2)}{k}. \tag{2.13}$$

Using (2.12), we have

$$\|\hat{x}^k - x^*\|^2 + \|\hat{y}^k - y^*\|^2 < \frac{2CL\Delta(\|x^0 - x^*\|^2 + \|y^0 - y^*\|^2)}{\mu k}.$$

It follows that for some $k < \frac{2CL\Delta}{\mu}$ one can guarantee that $(\|\hat{x}^k - x^*\|^2 + \|\hat{y}^k - y^*\|^2) < (\|x^0 - x^*\|^2 + \|y^0 - y^*\|^2)/2$. Using the restarts idea, specifically running the algorithm for k communications several times from the output of the previous run, after

$$\begin{aligned} K &= k \cdot \log_2 \frac{\|x^0 - x^*\|^2 + \|y^0 - y^*\|^2}{\varepsilon} \\ &< \frac{2CL\Delta}{\mu} \cdot \log_2 \frac{\|x^0 - x^*\|^2 + \|y^0 - y^*\|^2}{\varepsilon} \quad \text{communications} \end{aligned}$$

one can find some ε-solution of the strongly convex-strongly concave problem. But this contradicts the results of Theorem 2.1.6, which states that

$$K = \Omega\left(\frac{L\Delta}{\mu}\log_2 \frac{\|y^0 - y^*\|^2}{\varepsilon}\right).$$

Similar considerations can be made for Theorem 2.1.7. But this reasoning has a drawback due to the fact that in the lower bounds from Table 1 (and further in the upper bounds from Section 2.1.4), Ω_z appears, while in (2.13) we used $(\|x^0 - x^*\|^2 + \|y^0 - y^*\|^2)$. Therefore, this reasoning is not completely correct.

Another way to obtain the result is to consider the same line as in the proofs of Theorems 2.1.6 and 2.1.7, but replace μ with $\frac{\varepsilon}{\Omega_z^2}$. For any $\mu > 0$, a strongly convex-strongly concave function is also convex-concave. Therefore, this substitution of μ can serve as an example of a convex-concave

function.

Stochastic lower bounds

Due to our choice of f_m from (2.11), one can note that in order to obtain lower stochastic bounds, we consider the minimization problem rather than the saddle point problems, and moreover, this problem is the same for each node. Therefore, it is suggested that we are in an ideal situation when each node communicates to each and collectively minimizes the same function. Then the total number of stochastic oracle calls is MT. It may seem that such a formulation simplifies the problem and may not yield the most advanced lower bounds. But in the next section, we will give upper bounds that coincide with the lower bounds, and this will verify that considering such an idealistic setting is sufficient.

Let us formulate two theorems for the convex and strongly convex cases of f^{stoch}.

Theorem 2.1.9. For any $L > \mu > 0$, there exists a stochastic minimization problem with L-smooth and μ-strongly convex function (i.e., satisfying Assumptions 2.0.4 and 2.0.5(SM)), such that for any output $\hat{x}$ of any procedure satisfying Definition 2.1.5 with M workers, one can obtain the following estimate:

$$\mathbb{E}\left[\|\hat{x} - x^*\|^2\right] = \Omega\left(\frac{\sigma^2}{MT\mu^2}\right).$$

Theorem 2.1.10. For any $L > 0$ and any $M, T \in \mathbb{N}$, there exists a stochastic minimization problem with L-smooth and convex function (i.e., satisfying Assumptions 2.0.4 and 2.0.5(SM)) on a bounded $\mathcal{X}^{stoch}$ with a diameter Ω_z (i.e., satisfying Assumption 2.1.2), such that for any output $\hat{x}$ of any procedure satisfying Definition 2.1.5 with M workers, one can obtain the following estimate:

$$\mathbb{E}\left[f^{\text{stoch}}(\hat{x}) - f^{\text{stoch}}(x^*)\right] = \Omega\left(\frac{\sigma\Omega_z}{\sqrt{MT}}\right).$$

Connection of lower bounds

The connecting of deterministic and stochastic bounds follows from (2.11). The results for the centralized and decentralized cases are presented in Table 2.1. To verify the tightness of our lower bounds, the next section designs algorithms that achieve such bounds.

2.1.4 Optimal Algorithms

This section focuses on theoretically near-optimal algorithms. It is straightforward to verify that our algorithms satisfy Definition 2.1.5.

Centralized case

We design our algorithm based on MiniBatch SGD and Extragradient. For this algorithm, we introduce r as the maximum distance from nodes to the server. It is easy to note that r is upper

bounded by the diameter Δ.

Algorithm 1 Centralized Extra Step Method (`CESM`)

Parameters: stepsize γ; communication rounds K, number of local steps T
Initialization: choose $z^0 \in \mathcal{Z}$, $t = \left\lfloor \frac{K}{r} \right\rfloor$ and batch size $b = \left\lfloor \frac{T}{2t} \right\rfloor$

1: **for** $k = 0, 1, 2, \ldots, t-1$ **do**
2: Generate batch ξ_m^k on each machine independently
3: Each machine m computes $g_m^t = \frac{1}{b}\sum_{i=1}^{b} F_m(z^k, \xi_m^{t,i})$ and sends g_m^k to server
4: Server computes $z^{k+1/2} = \text{prox}_{\gamma g}(z^k - \frac{\gamma}{M}\sum_{m=1}^{M} g_m^k)$ and then sends $z^{k+1/2}$ to machines
5: Generate batch $\xi_m^{k+1/2}$ on each machine independently
6: Each machine m computes $g_m^{k+1/2} = \frac{1}{b}\sum_{i=1}^{b} F_m(z^{k+1/2}, \xi_m^{k+1/2,i})$ and sends $g_m^{k+1/2}$ to server
7: Server computes $z^{k+1} = \text{prox}_{\gamma g}(z^k - \frac{\gamma}{M}\sum_{m=1}^{M} g_m^{k+1/2})$ and then sends z^{k+1} to machines
8: **end for**

Theorem 2.1.11. Let $\{z^k\}_{k\geq 0}$ denote the iterates of Algorithm 1 for solving the problem (2.1)+(2.7)+(2.9). Let Assumptions 2.0.4, 2.1.1 be satisfied. Then if $\gamma \leq \frac{1}{4L}$, we have the following estimates in

- the μ-strongly convex-strongly concave case (Assumption 2.0.5(SM)):

$$\mathbb{E}[\|z^t - z^*\|^2] = \tilde{\mathcal{O}}\left(\|z^0 - z^*\|^2 \exp\left(-\frac{\mu K}{4L\Delta}\right) + \frac{\sigma^2}{\mu^2 MT}\right),$$

- the convex-concave case (Assumptions 2.0.5(M) and 2.1.2):

$$\mathbb{E}[\text{Gap}(z_{avg}^t)] = \mathcal{O}\left(\frac{L\Omega_z^2\Delta}{K} + \frac{\sigma\Omega_z}{\sqrt{MT}}\right),$$

- the non-convex-non-concave case (Assumptions 2.0.5(NM) and 2.1.2):

$$\mathbb{E}\left[\frac{1}{t}\sum_{k=0}^{t-1}\|F(z^k)\|^2\right] = \mathcal{O}\left(\frac{L^2\Omega_z^2\Delta}{K} + \frac{\sigma^2 K}{MT\Delta}\right),$$

where $z_{avg}^t = \frac{1}{t}\sum_{k=0}^{t-1} z^{k+1/2}$.

Decentralized static case

The idea of Algorithm 3 combines three components: Extragradient, accelerated consensus – `FastMix` (see Algorithm 2) and the optimal size of batches. With the appropriate selection of P, averaging of all vectors can be achieved with good accuracy. In particular, it can be assumed that $z_1^k \approx \ldots \approx z_M^k$.

Algorithm 2 `FastMix`

Parameters: communication rounds P, $\eta = \frac{1-\sqrt{1-\lambda_2^2(\mathbf{W})}}{1+\sqrt{1-\lambda_2^2(\mathbf{W})}}$

Initialization: $\mathbf{z}^{-1} = \mathbf{z}^0 = \mathbf{z} = (z_1, \ldots, z_M)^T$

for $k = 0, 1, 2, \ldots, P-1$ **do**

  $\mathbf{z}^{k+1} = (1+\eta)(I - W \otimes I)\mathbf{z}^k - \eta \mathbf{z}^{k-1}$

end for

Output: $z_1, \ldots, z_M$ of $\mathbf{z}^P$

The essence of the `FastMix` is straightforward. Initially, there are vectors z_1 and z_M, which are stored on their respective devices. Our goal is to obtain a vector that approximates the $\bar{z} = \frac{1}{M}\sum_{m=1}^M z_m$ vector on all devices. At each iteration, each device exchanges local vectors with its neighbors and then modifies its local vector by averaging its local vector with the vectors of its neighbors, weighted by the matrix $\mathbf{W}$.

Algorithm 3 Decentralized Extra Step Method (`DESM`)

Parameters: stepsize γ; communication rounds K, number of local calls T, number of `FastMix` steps P

Initialization: choose $z^0 \in \mathcal{Z}$, $z_m^0 = z^0$, $t = \left\lfloor \frac{K}{P} \right\rfloor$ and batch size $b = \left\lfloor \frac{T}{2t} \right\rfloor$

for $k = 0, 1, 2, \ldots, t-1$ **do**

  Generate batch ξ_m^k on each machine independently

  Each machine m computes $\hat{z}_m^{k+1/2} = z_m^k - \gamma \cdot \frac{1}{b}\sum_{i=1}^b F_m(z_m^k, \xi_m^{k,i})$

  Communication: $\tilde{z}_1^{k+1/2}, \ldots, \tilde{z}_M^{k+1/2} =$`FastMix`$(\hat{z}_1^{k+1/2}, \ldots, \hat{z}_M^{k+1/2}, P)$

  Each machine m computes $z_m^{t+1/2} = \mathrm{prox}_{\gamma g}(\tilde{z}_m^{k+1/2})$

  Generate batch $\xi_m^{k+1/2}$ on each machine independently

  Each machine m compute $\hat{z}_m^{k+1} = z_m^k - \gamma \cdot \frac{1}{b}\sum_{i=1}^b F_m(z_m^{k+1/2}, \xi_m^{k+1/2,i})$

  Communication: $\tilde{z}_1^{k+1}, \ldots, \tilde{z}_M^{k+1} =$`FastMix`$(\hat{z}_1^{k+1}, \ldots, \hat{z}_M^{k+1}, P)$

  Each machine m compute $z_m^{k+1} = \mathrm{prox}_{\gamma g}(\tilde{z}_m^{k+1})$

end for

The analysis of Algorithm 3 is derived from the analysis of the classical Extragradient method. We study the convergence properties of the sequence $\{\bar{z}^k\}_{k=0}^\infty$, where $\bar{z}^k = \frac{1}{M}\sum_{m=1}^M z_m^k$. Note that $\bar{z}^k$ is not held at any agent; instead, this quantity is only used in the analysis. Algorithm 3 employs gossip averaging after each local step. Therefore, the method uses an approximate value of $F(\bar{z}^k)$ when performing updates, and the approximation error is driven by the number of gossip iterations H. The analysis of Algorithm 3 comes down to studying the Extragradient method, which uses inexact values of F at each iteration. Given a target accuracy ε, we choose the number of gossip iterations P proportional to ε. Since `FastMix` (Algorithm 2) is a linearly convergent method, P is proportional to $\log(1/\varepsilon)$. As a result, we have a $\log^2(1/\varepsilon)$ term in the number of communication rounds of Algorithm 3.

Theorem 2.1.12. Let $\{z_m^k\}_{k\geq 0}$ denote the iterates of Algorithm 3 for solving the problem (2.1)+(2.7)+(2.9). Let Assumptions 2.0.4, 2.1.1 be satisfied. Then if $\gamma \leq \frac{1}{4L}$ and $P = \mathcal{O}\left(\sqrt{\chi}\log\frac{1}{\varepsilon}\right)$, we have the following estimates in

- the μ-strongly convex-strongly concave case (Assumption 2.0.5(SM)):

$$\mathbb{E}[\|\bar{z}^t - z^*\|^2] = \tilde{\mathcal{O}}\left(\|z^0 - z^*\|^2 \exp\left(-\frac{\mu K}{8L\sqrt{\chi}}\right) + \frac{\sigma^2}{\mu^2 MT}\right),$$

- the convex-concave case (Assumptions 2.0.5(M) and 2.1.2):

$$\mathbb{E}[\mathrm{Gap}(\bar{z}_{avg}^t)] = \tilde{\mathcal{O}}\left(\frac{L\Omega_z^2\sqrt{\chi}}{K} + \frac{\sigma\Omega_z}{\sqrt{MT}}\right),$$

- the non-convex-non-concave case (Assumptions 2.0.5(NM) and 2.1.2):

$$\mathbb{E}\left[\frac{1}{t}\sum_{k=0}^{t-1}\|F(\bar{z}^k)\|^2\right] = \tilde{\mathcal{O}}\left(\frac{L^2\Omega_z^2\sqrt{\chi}}{K} + \frac{\sigma^2 K}{MT\sqrt{\chi}}\right),$$

where $\bar{z}^k = \frac{1}{M}\sum\limits_{m=1}^{M} z_m^k$ and $\bar{z}_{avg}^{t+1} = \frac{1}{Mt}\sum\limits_{k=0}^{t-1}\sum\limits_{m=1}^{M} z_m^{k+1/2}$.

Decentralized time-varying case

In time-varying case, we do exactly the same as in static. We only change `FastMix` to the simpler `Gossip`, since `FastMix` does not work for time-varying graphs.

Algorithm 4 `Gossip`

Parameters: communication rounds P
Initialization: $\mathbf{z}^{-1} = \mathbf{z}^0 = \mathbf{z} = (z_1, \ldots, z_M)^T$
1: **for** $k = 0, 1, 2, \ldots, P-1$ **do**
2: $\quad \mathbf{z}^{k+1} = (I - W(k) \otimes I)\mathbf{z}^k$
3: **end for**
4: **Output:** $z_1, ..., z_M$ of $\mathbf{z}^P$

Theorem 2.1.13. Let $\{z_m^k\}_{k\geq 0}$ denote the iterates of Algorithm 3 for solving the problem (2.1)+(2.7)+(2.9). Let Assumptions 2.0.4, 2.1.1 be satisfied. Then if $\gamma \leq \frac{1}{4L}$ and $P = \mathcal{O}\left(\chi\log\frac{1}{\varepsilon}\right)$, we have the following estimates in

- the μ-strongly convex-strongly concave case (Assumption 2.0.5(SM)):

$$\mathbb{E}[\|\bar{z}^t - z^*\|^2] = \tilde{\mathcal{O}}\left(\|z^0 - z^*\|^2 \exp\left(-\frac{\mu K}{8L\chi}\right) + \frac{\sigma^2}{\mu^2 MT}\right),$$

Algorithm 5 Time-Varying Decentralized Extra Step Method (`TVDESM`)

Parameters: stepsize γ; communication rounds K, number of local calls T, number of `Gossip` steps P.

Initialization: choose $z^0 \in \mathcal{Z}$, $z_m^0 = z^0$, $t = \left\lfloor \frac{K}{P} \right\rfloor$ and batch size $b = \left\lfloor \frac{T}{2t} \right\rfloor$.

for $k = 0, 1, 2, \ldots, t-1$ **do**
  Generate batch ξ_m^k on each machine independently
  Each machine m computes $\hat{z}_m^{k+1/2} = z_m^k - \gamma \cdot \frac{1}{b} \sum_{i=1}^{b} F_m(z_m^k, \xi_m^{k,i})$
  Communication: $\tilde{z}_1^{k+1/2}, \ldots, \tilde{z}_M^{k+1/2} =$`Gossip`$(\hat{z}_1^{k+1/2}, \ldots, \hat{z}_M^{k+1/2}, P)$
  Each machine m computes $z_m^{t+1/2} = \mathrm{prox}_{\gamma g}(\tilde{z}_m^{k+1/2})$
  Generate batch $\xi_m^{k+1/2}$ on each machine independently
  Each machine m compute $\hat{z}_m^{k+1} = z_m^k - \gamma \cdot \frac{1}{b} \sum_{i=1}^{b} F_m(z_m^{k+1/2}, \xi_m^{k+1/2,i})$
  Communication: $\tilde{z}_1^{k+1}, \ldots, \tilde{z}_M^{k+1} =$`Gossip`$(\hat{z}_1^{k+1}, \ldots, \hat{z}_M^{k+1}, P)$
  Each machine m compute $z_m^{k+1} = \mathrm{prox}_{\gamma g}(\tilde{z}_m^{k+1})$
end for

- the convex-concave case (Assumptions 2.0.5(M) and 2.1.2):

$$\mathbb{E}[\mathrm{Gap}(\bar{z}_{avg}^t)] = \tilde{\mathcal{O}}\left(\frac{L\Omega_z^2\chi}{K} + \frac{\sigma\Omega_z}{\sqrt{MT}}\right),$$

- the non-convex-non-concave case (Assumptions 2.0.5(NM) and 2.1.2):

$$\mathbb{E}\left[\frac{1}{t}\sum_{k=0}^{t-1}\|F(\bar{z}^k)\|^2\right] = \tilde{\mathcal{O}}\left(\frac{L^2\Omega_z^2\chi}{K} + \frac{\sigma^2 K}{MT\sqrt{\chi}}\right),$$

where $\bar{z}^k = \frac{1}{M}\sum_{m=1}^{M} z_m^k$ and $\bar{z}_{avg}^{t+1} = \frac{1}{Mt}\sum_{k=0}^{t-1}\sum_{m=1}^{M} z_m^{k+1/2}$.

Discussions. Let us provide some comments on our algorithms:

- It is easy to see that our algorithms are optimal in the strongly monotone and monotone – see Table 2.1 for details. However, there are several practical drawbacks of these algorithms. The first is related to the fact that in Algorithm 3 and 5 we have to take multi consensus steps at each iteration. This approach does not always pay off in practice. Second, if $T \gg K$, we collect a very large batch at each iteration, in practice such batches do not make sense. Therefore, the idea arises to use these local computations of gradients more efficiently, e.g. by doing local steps. This brings us to Section 2.3.

- In the non-monotone case, we do not guarantee the convergence if $T \approx K$. However, the method converges sublinearly when $\sigma = 0$. In this case, we cover the deterministic results of [112]. In the stochastic case ($\sigma > 0$), convergence is also not guaranteed in [117, 169]. Therefore, we cover and even overlap their analysis, as they consider only the homogeneous case ($F_m = F$).

2.2 Upper and Lower Bounds. Finite-Sum Stochastic Case

We consider the following operators F_m from (2.1)+(2.7):

$$F_m(z) := \frac{1}{n}\sum_{i=1}^{n} F_{m,i}(z). \tag{2.14}$$

This formulation is less general than the previous one (2.9) but refers to the popular empirical risk setting. In particular, the data describing F_m is stored on worker m. For example, $F_{m,i}$ can correspond to the value of the operator on the ith data point of the m-th dataset.

In this section, we do not consider the centralized case separately, as the decentralized case with a complete graph can be easily reduced to it. We also focus exclusively on the strongly monotone case.

2.2.1 Contributions

All the necessary surveys have already been conducted in the previous section.

We now briefly summarize our main contributions.

• **Lower bounds.** We present the first lower bounds for the communication and local computation complexities of decentralized variational inequalities in the stochastic (finite sum) case, in both the fixed and time-varying network topology regimes. Existing literature contains lower bounds for non-distributed finite-sum variational inequalities [170], which we recover as a special case. The previous section contains lower bounds for deterministic (if $\sigma = 0$) decentralized variational inequalities in the fixed and time-varying. New bounds covers these results too.

• **Optimal decentralized algorithms.** We construct new algorithms for stochastic (finite sum) decentralized variational inequalities for fixed networks and for time-varying networks. These algorithms match our lower bounds and are therefore optimal in terms of communication and local iteration complexities. These are the first algorithms for stochastic (finite sum) decentralized variational inequalities over fixed and time-varying networks.

• **Experiments.** Numerical experiments on bilinear problems and robust regression problems confirm the practical efficiency of our methods, both in the non-distributed stochastic setup and in the decentralized deterministic one.

2.2.2 Assumptions

Let us introduce the assumptions for our setting (2.14).

Assumption 2.2.1 (Lipschitzness)**.** Each collection of operators $\{F_{m,i}\}$ is $\overline{L}$-average Lipschitz continuous, i.e., for all $u, v \in \mathbb{R}^d$ it holds $\frac{1}{n}\sum_{i=1}^{n} \|F_{m,i}(u) - F_{m,i}(v)\|^2 \leq \overline{L}^2\|u - v\|^2$.

Assumption 2.2.2 (Strong monotonicity)**.** Each operator F_m is μ-strongly monotone, i.e., for all $u, v \in \mathbb{R}^d$ we have $\langle F_m(u) - F_m(v); u - v\rangle \geq \mu\|u - v\|^2$.

Note that Assumption 2.0.5 employs the strong monotonicity of only F.

All the necessary details regarding the decentralized communication graph were introduced in the previous section (Subsection 2.1.2).

2.2.3 Lower Bounds

Our lower bounds apply to a specific class of algorithms that are, loosely speaking, permitted to communicate with neighbors and compute any local first-order information. We now provide a formal definition. The definition closely resembles what we presented in the previous section (Definition 2.1.5)

Definition 2.2.3 (Oracle)**.** Each agent m has its own local memory $\mathcal{M}_m$ with initialization $\mathcal{M}_m = \{0\}$. $\mathcal{M}_m$ is updated as follows. At each iteration, the algorithm either performs local computations or communicates.

• **Local computation:** At each local iteration, device m can sample uniformly and independently batch S_m of any size b from $\{F_{m,i}\}$ and adds to its $\mathcal{M}_m$ a finite number of points z, satisfying

$$z \in \operatorname{span}\left\{z' \ , \ \sum_{i_m \in S_m} F_{m,i_m}(z''), \operatorname{prox}_{\rho g}\left(\operatorname{span}\left\{z''' \ , \ \sum_{i_m \in S_m} F_{m,i_m}(z'')\right\}\right)\right\}$$

for $z', z'', z''' \in \mathcal{M}_m$ and $\rho > 0$. Such a call needs b local computations to collect the batch. Batch of size n represents F_m;

• **Communication:** Upon communication rounds among the neighboring nodes, and at communication time t, $\mathcal{M}_m$ is updated according to

$$\mathcal{M}_m := \operatorname{span}\left\{\bigcup_{(i,m)\in\mathcal{E}(t)} \mathcal{M}_i\right\}.$$

• **Output:** The final global output is calculated as $\hat{z} \in \operatorname{span}\left\{\bigcup_{m=1}^{M} \mathcal{M}_m\right\}$.

Definition 2.2.3 includes all approaches for working with stochastic problems, such as SGD or variance reduction techniques (SVRG, SARAH). Note that while our algorithm can invoke the deterministic oracle (full F_m) in local computations, in the work on lower bounds in the non-distributed case [170], there is no such possibility. This narrows the class of algorithms for which the results of [170] are valid. In particular, they cannot perform SVRG-type updates.

Theorem 2.2.4 (Lower bound - fixed network)**.** Let $\overline{L} > \mu > 0$, $n \in \mathbb{N}$ (with $\overline{L}/\mu \geq \sqrt{n}$), $\chi \geq 1$ and $K, N \in \mathbb{N}$. There exists a distributed variational inequality over fixed network (Assumption

2.1.3). For which the following statements are true:

- a gossip matrix W has $\chi(W) = \chi$,
- F is μ – strongly monotone,
- F_m are $\overline{L}$-average Lipschitz and F is $L = \frac{\overline{L}}{\sqrt{n}}$ - Lipschitz,
- size $d \geq \max\left\{2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right), 2K, 2N\right\}$, where $\alpha = \frac{2n\mu^2}{L^2}$ and $q = \frac{1}{2}\left(2 + \alpha - \sqrt{\alpha^2 + 4\alpha}\right) \in (0; 1)$,
- the solution of the problem is non-zero: $z^* \neq 0$.

Then for any output $\hat{z}$ of any procedure (Definition 2.2.3) with K communication rounds and N local computations, one can obtain the following estimate:

$$\|\hat{z} - z^*\|^2 = \Omega\left(\exp\left(-\frac{80}{1 + \sqrt{\frac{2L^2}{\mu^2} + 1}} \cdot \frac{K}{\sqrt{\chi}}\right)\|z_0 - z^*\|^2\right);$$

$$\|\hat{z} - z^*\|^2 = \Omega\left(\exp\left(-\frac{16}{n + \sqrt{\frac{2n\overline{L}^2}{\mu^2} + n^2}} \cdot N\right)\|z_0 - z^*\|^2\right).$$

Corollary 2.2.5. In the setting of Theorem 2.2.4, the number of communication rounds and local computations required to obtain an ε-solution (in expectation) is lower bounded by

$$\Omega\left(\sqrt{\chi}\left(1 + \frac{L}{\mu}\right) \cdot \log\left(\frac{R_0^2}{\varepsilon}\right)\right) \quad \text{and} \quad \Omega\left(\left(n + \sqrt{n} \cdot \frac{\overline{L}}{\mu}\right) \cdot \log\left(\frac{R_0^2}{\varepsilon}\right)\right), \quad \text{respectively.}$$

Theorem 2.2.6 (Lower bound - time varying network)**.** Let $\overline{L} > \mu > 0$, $n \in \mathbb{N}$ (with $\overline{L}/\mu \geq \sqrt{n}$), $\hat{\chi} \geq 3$ and $K, N \in \mathbb{N}$. There exists a distributed variational inequality over time-varying network (Assumption 2.1.4). For which the following statements are true:

- Assumption 2.1.4 holds with $\chi = \hat{\chi}$,
- F is μ – strongly monotone,
- F_m are $\overline{L}$-average Lipschitz and F is $L = \frac{\overline{L}}{\sqrt{n}}$ - Lipschitz,
- size $d \geq \max\left\{2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right), 2K, 2N\right\}$, where $\alpha = \frac{2n\mu^2}{L^2}$ and $q = \frac{1}{2}\left(2 + \alpha - \sqrt{\alpha^2 + 4\alpha}\right) \in (0; 1)$,
- the solution of the problem is non-zero: $z^* \neq 0$.

Then for any output $\hat{z}$ of any procedure (Definition 2.2.3) with K communication rounds and

N local computations, one can obtain the following estimate:

$$\|\hat{z} - z^*\|^2 = \Omega\left(\exp\left(-\frac{64}{1+\sqrt{\frac{2L^2}{\mu^2}+1}} \cdot \frac{K}{B\hat{\chi}}\right)\|z_0 - z^*\|^2\right);$$

$$\|\hat{z} - z^*\|^2 = \Omega\left(\exp\left(-\frac{16}{n+\sqrt{\frac{2n\overline{L}^2}{\mu^2}+n^2}} \cdot N\right)\|z_0 - z^*\|^2\right).$$

Corollary 2.2.7. In the setting of Theorem 2.2.6, the number of communication rounds and local computations required to obtain an ε-solution (in expectation) is lower bounded by

$$\Omega\left(B\chi\left(1+\frac{L}{\mu}\right)\cdot\log\left(\frac{R_0^2}{\varepsilon}\right)\right) \quad \text{and} \quad \Omega\left(\left(n+\sqrt{n}\cdot\frac{\overline{L}}{\mu}\right)\cdot\log\left(\frac{R_0^2}{\varepsilon}\right)\right), \quad \text{respectively.}$$

The proof uses the idea (example of bad functions) of non-distributed deterministic lower bounds from [159]. This concept is further extended to address distributed stochastic variational inequalities. See Section A.2.1

Note that in the time-varying case, the lower bounds for communication differ by the constant B from the estimates encountered in the previous section. This is due to the consideration of a more general setup with a B-connected graph (see Assumption 2.1.4), while the existing literature on lower bounds focuses on the simpler $B = 1$ case.

2.2.4 Optimal Algorithms

Since in the decentralized gossip protocol each local worker m stores its own z_m vector, we consider the problem

$$\text{Find} \quad \mathbf{z}^* \in (\mathbb{R}^d)^M \quad \text{such that} \quad \langle \mathbf{F}(\mathbf{z}^*), \mathbf{z} - \mathbf{z}^* \rangle + \mathbf{g}(\mathbf{z}) - \mathbf{g}(\mathbf{z}^*) \geq 0, \forall \mathbf{z} \in (\mathbb{R}^d)^M, \tag{2.15}$$

where we use new notation: $\mathbf{z} = (z_1, \ldots, z_M)^\top$ and $\mathbf{z}^* = (z_1^*, \ldots, z_M^*)^\top$. Additionally, here we introduce the lifted operator $\mathbf{F} : (\mathbb{R}^d)^M \to (\mathbb{R}^d)^M$ given as $\mathbf{F}(\mathbf{z}) = (F_1(z_1), \ldots, F_M(z_M))^\top$, and the lifted operator $\mathbf{g} : (\mathbb{R}^d)^M \to \mathbb{R} \cup \{+\infty\}$ defined by $\mathbf{g}(\mathbf{z}) = \frac{1}{M}\sum_{m=1}^M g(z_m)$.

One can note that (2.15) is a set of M unrelated variational inequalities with their own variables. But the original problem (2.1) + (2.7) + (2.14) is a sum of variational inequalities with the same variables: $\sum_{m=1}^M \left[\langle F(z^*), z - z^* \rangle + \frac{1}{M} g(z) - \frac{1}{M} g(z^*)\right]$. To eliminate this issue and move on to problem (2.1) + (2.7) + (2.14), it is easy to get the following modification of (2.15)

$$\text{Find} \quad \mathbf{z}^* \in \mathcal{L} \quad \text{such that} \quad \langle \mathbf{F}(\mathbf{z}^*), \mathbf{z} - \mathbf{z}^* \rangle + \mathbf{g}(\mathbf{z}) - \mathbf{g}(\mathbf{z}^*) \geq 0, \quad \forall \mathbf{z} \in \mathcal{L}, \tag{2.16}$$

where $\mathcal{L}$ is the consensus space. Problem (2.16) is equivalent to (2.1) + (2.7) + (2.14). Due to Assumptions 2.0.4, 2.2.1 and 2.2.2, $\mathbf{F}$ is L-Lipschitz continuous, $\overline{L}$-average Lipschitz continuous,

and μ-strongly monotone.

Static case

We present Algorithm 6 for fixed networks. In particular, we provide the deterministic variant as well as the non-distributed version of Algorithm 6. The next result gives the iteration complexity of Algorithm 6. In this algorithm, we use the follwong notation: $\mathbf{F}_j(\mathbf{z}) = (F_{1,j_{1,l}}(z_1), \ldots, F_{M,j_{M,l}}(z_M))^T$, $l \in \{1, \ldots, b\}$

Algorithm 6

Parameters: stepsizes $\eta, \theta > 0$, momentums α, β, γ, batchsize $b \in \{1, \ldots, n\}$, probability $p \in (0, 1)$

Initialization: choose $\mathbf{z}^0 = \mathbf{w}^0 \in (\operatorname{dom} g)^M$, $\mathbf{y}^0 \in \mathcal{L}^\perp$. Put $\mathbf{z}^{-1} = \mathbf{z}^0, \mathbf{w}^{-1} = \mathbf{w}^0$, $\mathbf{y}^{-1} = \mathbf{y}^0$

for $k = 0, 1, 2 \ldots$ **do**

Sample $j^k_{m,1}, \ldots, j^k_{m,b}$ independently from $[n]$

$S^k = \{j^k_{m,1}, \ldots, j^k_{m,b}\}$

Sample $j^{k+1/2}_{m,1}, \ldots, j^{k+1/2}_{m,b}$ independently from $[n]$

$S^{k+1/2} = \{j^{k+1/2}_{m,1}, \ldots, j^{k+1/2}_{m,b}\}$

$\delta^k = \frac{1}{b} \sum_{j \in S^k} \left(\mathbf{F}_j(\mathbf{z}^k) - \mathbf{F}_j(\mathbf{w}^{k-1}) + \alpha[\mathbf{F}_j(\mathbf{z}^k) - \mathbf{F}_j(\mathbf{z}^{k-1})] \right) + \mathbf{F}(\mathbf{w}^{k-1})$

$\Delta^k = \delta^k - (\mathbf{y}^k + \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}))$

$\mathbf{z}^{k+1} = \operatorname{prox}_{\eta \mathbf{g}}(\mathbf{z}^k + \gamma(\mathbf{w}^k - \mathbf{z}^k) - \eta \Delta^k)$

$\Delta^{k+1/2} = \frac{1}{b} \sum_{j \in S^{k+1/2}} \left(\mathbf{F}_j(\mathbf{z}^{k+1}) - \mathbf{F}_j(\mathbf{w}^k) \right) + \mathbf{F}(\mathbf{w}^k)$

$\mathbf{y}^{k+1} = \mathbf{y}^k - \theta(W \otimes I)(\mathbf{z}^{k+1} - \beta(\Delta^{k+1/2} - \mathbf{y}^k))$

$\mathbf{w}^{k+1} = \begin{cases} \mathbf{z}^k, & \text{with probability } p \\ \mathbf{w}^k, & \text{with probability } 1 - p \end{cases}$

end for

Theorem 2.2.8 (Upper bound - fixed network)**.** Consider the problem (2.16) (or (2.1) + (2.7) + (2.14)) under Assumptions 2.0.4, 2.2.1 and 2.2.2 over a fixed graph $\mathcal{G}$ (Assumption 2.1.3) with a gossip matrix W. Let $\{\mathbf{z}^k\}$ be the sequence generated by Algorithm 6 with tuning of $\eta, \theta, \alpha, \beta, \gamma$ as described in Appendix A.2.2. Then, given $\varepsilon > 0$, the number of iterations for $\mathbb{E}[\|\mathbf{z}^k - \mathbf{z}^*\|^2] \leq \varepsilon$ is

$$\mathcal{O}\left(\left[\frac{1}{p} + \chi + \frac{1}{\sqrt{pb}} \frac{\overline{L}}{\mu} + \sqrt{\chi} \frac{L}{\mu} \right] \log \frac{1}{\varepsilon} \right).$$

Let us discuss the results of the theorem. First of all, we are interested in how to obtain the complexity of communications and local computations from iterative complexity. At each iteration, we require (on average) $\mathcal{O}(b + pn)$ local computations, as we need to store batch b twice, and with probability p, we update the point $\mathbf{w}^{k+1}$ by $\mathbf{z}^k$. This requires calculating the full $\mathbf{F}$ in the next iteration. Then, as the optimal p, one can choose $p \sim b/n$. With this choice of p, we have the following local and communication complexities.

$$\mathcal{O}\left(\left[n + b\chi + \sqrt{n} \frac{\overline{L}}{\mu} + b\sqrt{\chi} \frac{L}{\mu} \right] \log \frac{1}{\varepsilon} \right) \quad \text{and} \quad \mathcal{O}\left(\left[\frac{n}{b} + \chi + \frac{\sqrt{n}}{b} \frac{\overline{L}}{\mu} + \sqrt{\chi} \frac{L}{\mu} \right] \log \frac{1}{\varepsilon} \right),$$

respectively (since at each iteration Algorithm 6 performs $\mathcal{O}(1)$ communications). Depending on $\max\{\sqrt{n}; \sqrt{\chi}\}$, we have the optimality of either local communications or decentralized communications. It is sufficient to take $b \geq {}^{\overline{L}\sqrt{n}}/_{L}$ to ensure the optimal communication complexity (see Corollary 2.2.5), but this leads to non-optimality in local iterations.

To make the algorithm optimal both in terms of communications and local computations, we need to slightly modify it. This can be achieved using `FastMix` (Algorithm 2). We then modify Algorithm 6 by replacing W with the output of Algorithm 2 to obtain

Theorem 2.2.9 (Upper bound - fixed network)**.** Consider the problem (2.16) (or (2.1) + (2.14)) under Assumptions 2.2.1 and 2.2.2 over a fixed connected graph $\mathcal{G}$ (Assumption 2.1.3) with a gossip matrix W. Let $\{\mathbf{z}^k\}$ be the sequence generated by Algorithm 6 with Chebyshev polynomial $P(W)$ as a gossip matrix and with tuning of $\eta, \theta, \alpha, \beta, \gamma$ as described in Appendix A.2.2. Then, given $\varepsilon > 0$, the number of iterations for $\mathbb{E}[\|\mathbf{z}^k - \mathbf{z}^*\|^2] \leq \varepsilon$ is

$$\mathcal{O}\left(\left[\frac{1}{p} + \frac{1}{\sqrt{pb}}\frac{\overline{L}}{\mu} + \frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right).$$

In this case, the communication complexity of one iteration is χ, and the local complexity (on average) remains $\mathcal{O}(b + pn)$. Then with $p = {}^{b}/_{n}$, we obtain the following local and communication complexities.

$$\mathcal{O}\left(\left[n + \sqrt{n}\frac{\overline{L}}{\mu} + b\frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right) \quad \text{and} \quad \mathcal{O}\left(\left[\sqrt{\chi}\frac{n}{b} + \sqrt{\chi}\frac{\sqrt{n}}{b}\frac{\overline{L}}{\mu} + \sqrt{\chi}\frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right),$$

respectively. To achieve optimal results, we only need to take $b = {}^{\overline{L}\sqrt{n}}/_{L}$.

In contrast to the algorithm from the previous section (Algorithm 3), our Algorithm 6 requires `FastMix` for both optimal rates, but can work without these additional procedures. Algorithm 3 requires $\mathcal{O}(\sqrt{\chi}\log\varepsilon^{-1})$ iterations for `FastMix`, which makes the algorithms less practical.

Time-varying networks

We present Algorithm 7 for time-varying networks. It needs to compute W_T using `Gossip` (Algorithm 4), it requires T communications. The following result provides the iteration complexity of Algorithm 7.

Theorem 2.2.10 (Upper bound - time varying network)**.** Consider the problem (2.16) (or (2.1) + (2.14)) under Assumptions 2.2.1 and 2.2.2 over a sequence of time-varying graphs $\mathcal{G}(k)$ (Assumption 2.1.4) with gossip matrices $W(k)$. Let $\{\mathbf{z}^k\}$ be the sequence generated by Algorithm 7 with $T \geq B$ and tuning of parameters described in Appendix A.2.3. Let the choice of T guarantees contraction property (Assumption 2.1.4 point 4) with $\chi(T)$. Then , given $\varepsilon > 0$, the number of

Algorithm 7

Parameters: stepsizes $\eta_z, \eta_y, \eta_x, \theta > 0$, momentums $\alpha, \gamma, \omega, \tau$, parameters ν, β, batchsize $b \in \{1, \dots, n\}$, probability $p \in (0, 1)$

Initialization: choose $\mathbf{z}^0 = \mathbf{w}^0 \in (\operatorname{dom} g)^M$, $\mathbf{y}^0 \in (\mathbb{R}^d)^M$, $\mathbf{x}^0 \in \mathcal{L}^\perp$. Put $\mathbf{z}^{-1} = \mathbf{z}^0, \mathbf{w}^{-1} = \mathbf{w}^0$, $\mathbf{y}_f = \mathbf{y}^{-1} = \mathbf{y}^0$, $\mathbf{x}_f = \mathbf{x}^{-1} = \mathbf{x}^0$, $m_0 = \mathbf{0}^{dM}$

for $k = 0, 1, 2, \dots$ **do**

Sample $j^k_{m,1}, \dots, j^k_{m,b}$ independently from $[n]$

$S^k = \{j^k_{m,1}, \dots, j^k_{m,b}\}$

Sample $j^{k+1/2}_{m,1}, \dots, j^{k+1/2}_{m,b}$ independently from $[n]$

$S^{k+1/2} = \{j^{k+1/2}_{m,1}, \dots, j^{k+1/2}_{m,b}\}$

$\delta^k = \frac{1}{b} \sum_{j \in S^k} \left(\mathbf{F}_j(\mathbf{z}^k) - \mathbf{F}_j(\mathbf{w}^{k-1} + \alpha[\mathbf{F}_j(\mathbf{z}^k) - \mathbf{F}_j(\mathbf{z}^{k-1})] \right) + \mathbf{F}(\mathbf{w}^{k-1})$

$\Delta^k_z = \delta^k - \nu \mathbf{z}^k - \mathbf{y}^k - \alpha(\mathbf{y}^k - \mathbf{y}^{k-1})$

$\mathbf{z}^{k+1} = \operatorname{prox}_{\eta_z \mathbf{g}}(\mathbf{z}^k + \omega(\mathbf{w}^k - \mathbf{z}^k) - \eta_z \Delta^k_z)$

$\mathbf{y}^k_c = \tau \mathbf{y}^k + (1 - \tau)\mathbf{y}^k_f$

$\mathbf{x}^k_c = \tau \mathbf{x}^k + (1 - \tau)\mathbf{x}^k_f$

$\Delta^k_y = \nu^{-1}(\mathbf{y}^k_c + \mathbf{x}^k_c) + \mathbf{z}^{k+1} + \gamma(\mathbf{y}^k + \mathbf{x}^k + \nu \mathbf{z}^k)$

$\delta^{k+1/2} = \frac{1}{b} \sum_{j \in S^{k+1/2}} \left(\mathbf{F}_j(\mathbf{z}^{k+1}) - \mathbf{F}_j(\mathbf{w}^k) \right) + \mathbf{F}(\mathbf{w}^k)$

$\Delta^k_x = \nu^{-1}(\mathbf{y}^k_c + \mathbf{x}^k_c) + \beta(\mathbf{x}^k + \delta^{k+1/2})$

$\mathbf{y}^{k+1} = \mathbf{y}^k - \eta_y \Delta^k_y$

$\mathbf{x}^{k+1} = \mathbf{x}^k - (W_T(Tk) \otimes I_d)(\eta_x \Delta^k_x + m^k)$

$m^{k+1} = \eta_x \Delta^k_x + m^k - (W_T(Tk) \otimes I_d)(\eta_x \Delta^k_x + m^k)$

$\mathbf{y}^{k+1}_f = \mathbf{y}^k_c + \tau(\mathbf{y}^{k+1} - \mathbf{y}^k)$

$\mathbf{x}^{k+1}_f = \mathbf{x}^k_c - \theta(W_T(Tk) \otimes I)(\mathbf{y}^k_c + \mathbf{x}^k_c)$

$\mathbf{w}^{k+1} = \begin{cases} \mathbf{z}^k, & \text{with probability } p \\ \mathbf{w}^k, & \text{with probability } 1 - p \end{cases}$

end for

iterations for $\mathbb{E}[\|\mathbf{z}^k - \mathbf{z}^*\|^2] \leq \varepsilon$ is

$$\tilde{\mathcal{O}}\left(\chi^2(T) + \frac{1}{p} + \chi(T)\frac{L}{\mu} + \frac{1}{\sqrt{bp}}\frac{\overline{L}}{\mu} \right).$$

Note that an important detail of the method is that $T \geq B$. This limitation is due to the fact that the network is B-connected. In particular, for $B > 1$, it can occur that in some communications we utilize empty graphs. Therefore, the requirement $T \geq B$ is natural to guarantee the contraction property (Assumption 2.1.4, point 4). This means that if $B > 1$, we must use `Gossip`.

But with $B = T = 1$, we can avoid multi-consensus; let us consider this case first. In the same way as in the fixed graph case, we choose $p = b/n$. Then, we obtain the estimates on communications and local calls.

$$\mathcal{O}\left(\left[\chi^2 + \frac{n}{b} + \chi\frac{L}{\mu} + \frac{\sqrt{n}}{b}\frac{\overline{L}}{\mu} \right] \log\frac{1}{\varepsilon} \right).$$

If we put $b = 1$, we have the optimal result.

Now we consider the general case with any B. We use a multi-gossip step and take $T > 1$. In particular, let us choose $T = B \cdot \lceil \chi \ln 2 \rceil$. Then, using `Gossip` and point 4 of Assumption 2.1.4, we can guarantee that $\|W_T(t)z - z\|^2 \leq \frac{1}{2}\|z\|^2$. Therefore, $\chi(T) = 2$, but we need T communications per iteration. With $p = b/n$ and $b = \overline{L}\sqrt{n}/L$, the iteration complexity from Theorem 2.2.10 can be rewritten as follows

$$\mathcal{O}\left(\left[1 + \frac{\sqrt{n}L}{\overline{L}} + \frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right).$$

Using that per iteration we make $\mathcal{O}(B \cdot \lceil \chi \ln 2 \rceil)$ communications and $\mathcal{O}\left(\overline{L}\sqrt{n}/L\right)$ local computations, we obtain

$$\mathcal{O}\left(\left[B\chi + B\chi\frac{\sqrt{n}L}{\overline{L}} + B\chi\frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right) \text{ commn-s and } \mathcal{O}\left(\left[n + \frac{\overline{L}\sqrt{n}}{L} + \sqrt{n}\frac{\overline{L}}{\mu}\right]\log\frac{1}{\varepsilon}\right) \text{ local calls.}$$

2.2.5 Experiments

In this subsection, we compare the state-of-the-art methods for solving strongly monotone decentralized variational inequalities over fixed and time-varying networks with our Algorithms 6 and 7.

We consider robust linear regression (the particular case of (2.6)):

$$\min_{x\in\mathbb{R}^d}\max_{\|r_i\|\leq R}\frac{1}{2n}\sum_{i=1}^{n}(x^T(a_i + r_i) - b_i)^2 + \frac{\lambda}{2}\|x\|^2 - \frac{\beta}{2}\|r_i\|^2. \tag{2.17}$$

For comparison, we consider methods for decentralized problems over fixed and time-varying networks. In particular, we select `EGD-GT` from [156], Algorithms 3, 5 (`EGD-Con`) and Algorithm 9 (`Sliding`) from the next section.

For a fair comparison, we consider the deterministic setup, i.e., each worker can compute full gradients. We take datasets from LiBSVM [171] and divide them unevenly across $M = 25$ workers. For communication networks, we choose the star, the ring, and the grid topologies. For time-varying networks, the topologies remain the same, but the locations of the vertices in them change randomly. All methods are tuned for the best convergence. The comparison criterion is the number of communication rounds.

Fixed networks. The plots from Figure 2.1 demonstrate that our Algorithm 6 outperforms other methods. Among other aspects, it surpasses `Sliding`, which exhibits a rapid theoretical communication complexity. However, this performance occurs when the dataset is relatively homogeneous and uniformly distributed across the devices. In our setting, this is not the case.

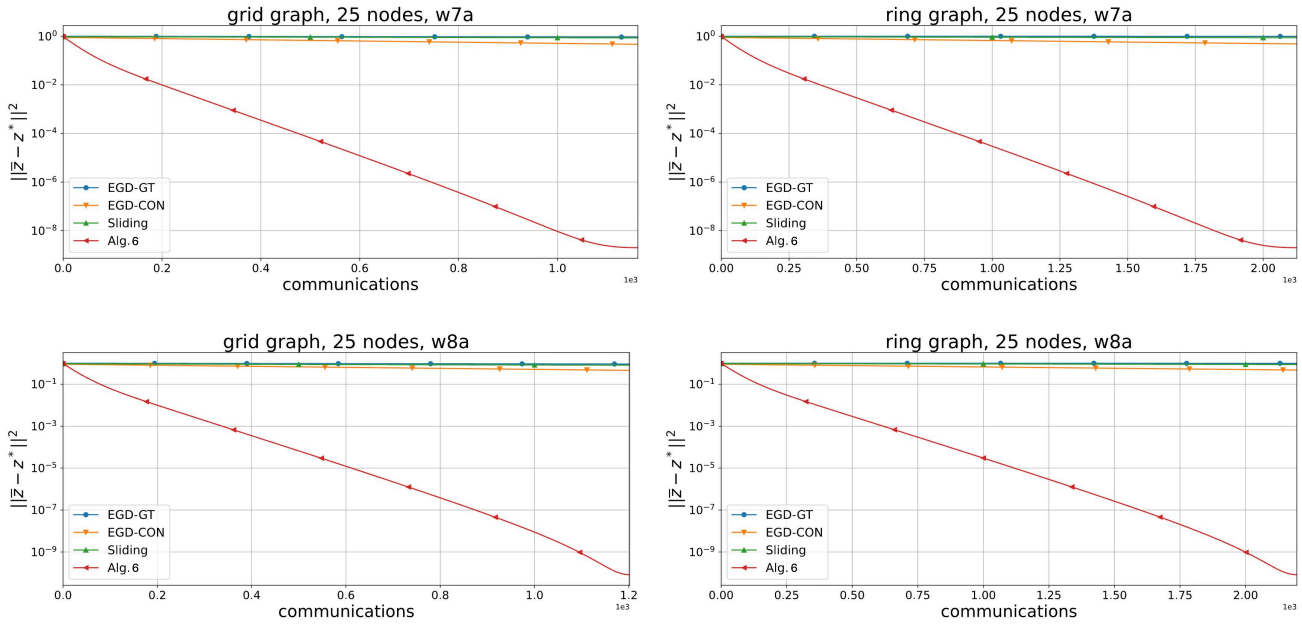


Figure 2.1: Comparison communication complexities of Algorithm 6, EGD-GT, EGD-Con and Sliding on (2.17) over fixed networks.

Time-varying networks. The plots from Figure 2.2 demonstrate that our Algorithm 7 outperforms other methods in the grid network scenario. In the case of the ring topology, `EGD-Con` yields the best results. Indeed, this algorithm implements centralized communications through a decentralized protocol; this approach is not always the fastest, but it is reliable.

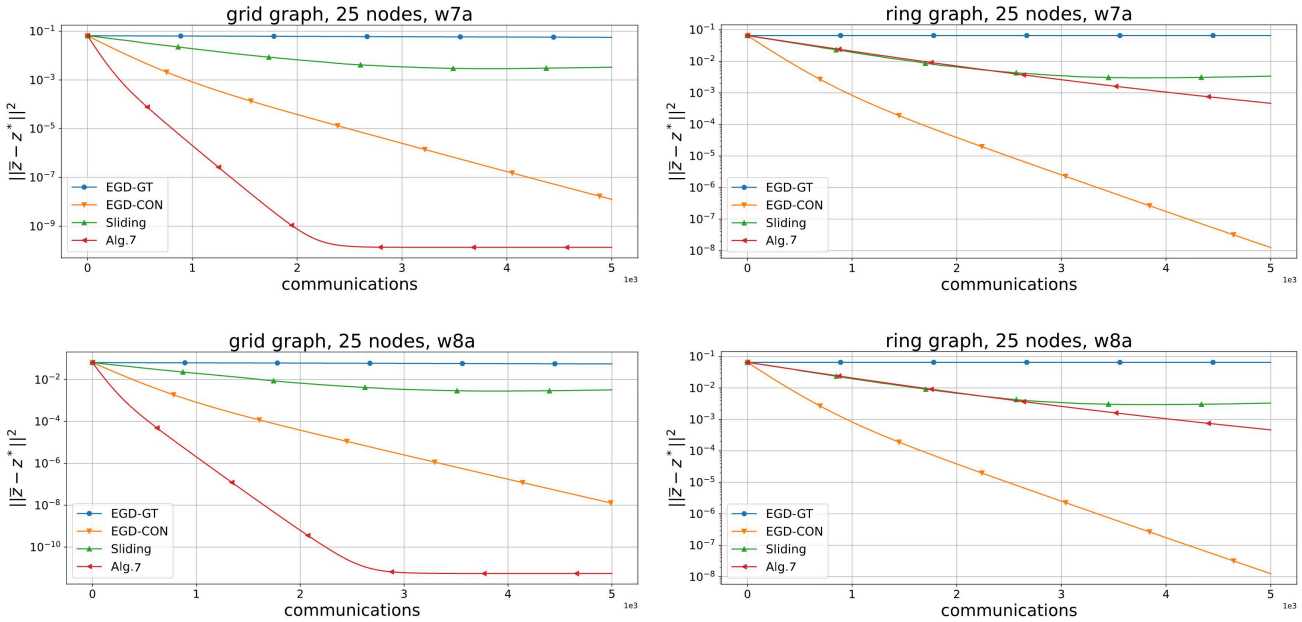


Figure 2.2: Comparison communication complexities of Algorithm 7, EGD-GT, EGD-Con and Sliding on (2.17) over time-varying networks.

2.3 Local Approach

The previous two sections provided a general overview of distributed stochastic variational inequalities. Now we try to move towards the struggle for efficient communication in solving problem (2.1)+(2.7). We begin with local methods.

In this section, as in Section 2.1, we consider a general stochastic setting:

$$F_m(z) := \mathbb{E}_{\xi_m \sim \mathcal{D}_m} \left[F_m(z, \xi_m) \right].$$

Note at once that in this section we put $g \equiv 0$.

The goal is not simply to develop the ability to add local steps to a centralized method (e.g., Algorithm 1). But to look at the local technique in a unified way, in order to be able to implement it uniformly in decentralized optimization as well.

The use of local steps is probably the most popular approach to addressing the communication bottleneck. Methods with local updates are also widely utilized in general distributed and federated learning. The idea of distributed minimization using methods similar to Local SGD is not an achievement of recent years and comes back to [107, 172, 173]. The development of theoretical analysis and new modifications for the original methods can be traced in the following works [174, 175, 176, 40, 177, 178, 135]. With the appearance of [39], local methods for distributed minimization problems receive a new boost.

2.3.1 Related Works and Contributions

The first local methods for variational inequalities and saddle point problems were adaptations of local algorithms for minimization problems. In particular, the `Local SGD`/`FedAvg` method was adapted in [157, 158]. [158] also analyzed the SCAFFOLD method [40] for saddle point problems.

We summarize our main contributions:

• **New local method.** Based on extragradient updates, we develop a novel algorithm for distributed stochastic variational inequalities (and, as a special case, for distributed stochastic saddle point problems) with heterogeneous data. Our scheme supports a very general communication protocol that covers centralized settings as in Federated Learning, fully decentralized settings, local steps in both the centralized/decentralized settings, and time-varying network topologies. In particular, we are not aware of earlier works proposing or analyzing extragradient methods with local steps for the fully decentralized setting or decentralized algorithms for stochastic variational inequalities over time-varying networks.

• **General setting.** Under the very general communication protocol and in the three settings of variational inequalities, i.e., with an operator that is strongly monotone, monotone, or non-monotone under the Minty condition, we prove the convergence of our algorithm and provide an explicit dependence of the rates on the problem parameters: characteristics of the network (e.g.,

mixing time), data heterogeneity, variance of the data, number of devices, and other standard parameters. These theoretical results translate to the corresponding three settings of saddle point problems (strongly convex-strongly concave, convex-concave, non-convex-non-concave under the Minty condition). All our theoretical results are valid in the important heterogeneous data regime and allow judging in a quantifiable way how different properties, e.g., data heterogeneity, the scale of the noise in the data, and network characteristics, influence the convergence rate of the algorithm. Even for decentralized settings, our results are novel for time-varying graphs and three different settings of monotonicity. See also Table 2.2 that gives more details on our contribution compared to the existing literature. The main challenge of our analysis is to deal with the very general assumption about the communication protocol and cope with the errors caused by the stochastic nature and heterogeneity of the data and limited information exchange between the nodes of the communication network. As a byproduct of independent interest, we analyze the stochastic extragradient method with biased oracle on unbounded domains, which was not done so far in the literature.

Table 2.2: Comparison of approaches for distributed strongly monotone (SM), monotone (M), and non-monotone (NM) variational inequalities or, respectively, strongly convex-strongly concave, convex-concave, non-convex-non-concave saddle point problems.

Reference	base method	arbitrary network	time-varying	local updates	no multiple gossip steps	SM	M	NM
Liu et al. 2019 [117]	Stoch. ES	✔	✘	✘	✘	✘	✘	✔†
This thesis (Alg. 3)	Stoch. ES	✔	✘	✘	✘	✔	✔	✘
Barazandeh et el. 2021 [119]	Stoch. ES	✔	✘	✘	✘	✘	✘	✔
Liu et al. 2019 [112]	Deter. prox	✔	✘	✘	✔	✘	✘	✔
Mukherjee and Chakraborty 2020 [156]	Deter. ES	✔	✘	✘	✔	✔	✘	✘
Tsaknakis et al. 2020 [179]	Stoch. DA	✔	✘	✘	✔	✘	✘	✔‡
Rogozin et al. 2021[180]	Deter. ES	✔	✘	✘	✔	✘	✔	✘
Xian et al. 2021 [181]	Stoch. DA	✔	✘	✘	✔	✘	✘	✔‡
Beznosikov et al. 2021 [16] Alg 3	Stoch. ES	✘	✘	✔	-§	✔	✘	✘
Deng and Mahdavi 2021[157]	Stoch. DA	✘	✘	✔	-	✔	✘	✔‡
Hou et al. 2021 [158]	Stoch. DA	✘	✘	✔	-	✔	✘	✘
Ours	Stoch. ES	✔	✔	✔	✔	✔	✔	✔

† – homogeneous case, ‡ – non-convex-concave saddle point problems (other works use minty condition – (2.0.5)(NM)), § – this column does not apply to centralized algorithms.
Definitions of columns: base method — the non-distributed algorithm that is taken as the basis for the distributed method, typically it is either the extragradient method (EGM) or the descent-ascent (DA); arbitrary network — supporting fully decentralized vs. only centralized topology; time-varying — decentralized method supporting time-varying network topology; local updates — method supporting local steps between communications; no multiple gossip steps — at one global iteration the method does not use many iterations of gossip averaging to reach a good consensus accuracy; SM, M, NM — monotonicity assumption, see Assumption 2.0.5.

• **Experiments.** We verify our theoretical results through numerical experiments and demonstrate the practical effectiveness of the proposed scheme. In particular, we train the DCGAN [182] architecture on the CIFAR-10 [183] dataset.

2.3.2 Algorithm

In this section, we present and discuss the proposed algorithm (Algorithm 8) that is based on two main ideas: (i) the Extragradient step, as in the classical methods for variational inequalities [78, 80], and (ii) the gossip averaging (see previous sections and [166, 121]) widely used in decentralized optimization methods and in the literature on diffusion strategies in distributed learning [184, 185, 186, 187, 188]. Unlike these papers that propose algorithms for optimization problems by exploiting gradient descent, our algorithm is based on the Extragradient method and is designed to solve variational inequalities and saddle point problems. Moreover, unlike the mentioned works, our method also allows for local steps in-between the communication rounds and for time-varying networks and has non-asymptotic theoretical convergence rate guarantees.

Note that our setting possesses great flexibility, as the topology of the communication graph is allowed to change between iterations; thus, the matrix $W(k)$, which encodes the current structure of the network, changes accordingly. This is encoded in line 2, where the matrix $W(k)$ is generated by a specific rule $\mathcal{W}(k)$ that can vary in nature. Examples include a deterministic choice of a sequence of matrices $W(k)$, sampling from a time-varying probability distribution on matrices. Even local steps without communication can be encoded using a diagonal matrix $W(k)$.

Algorithm 8 Extra Step Time-Varying Gossip Method (`ESTVGM`)

Parameters: stepsize $\gamma > 0$, $\{\mathcal{W}(k)\}_{k\geq 0}$ – rules or distributions for gossip matrix
Initialization: choose $z^0 \in \mathcal{Z}, \forall m : z_m^0 = z^0$

1: **for** $k = 0, 1, 2, \dots K-1$ **do**
2: Sample gossip matrix $W(k)$ from $\mathcal{W}(k)$
3: **for** each node m **do**
4: Generate independently $\xi_m^k \sim \mathcal{D}_k$, $\xi_m^{k+1/3} \sim \mathcal{D}_k$
5: $z_m^{k+1/3} = z_m^k - \gamma F_m(z_m^k, \xi_m^k)$
6: $z_m^{k+2/3} = z_m^k - \gamma F_m(z_m^{k+1/3}, \xi_m^{k+1/3})$
7: **end for**
8: $z_1^{k+1}, \dots, z_M^{k+1} =$`Gossip`$(z_1^{k+2/3}, \dots, z_M^{k+2/3}, 1)$
9: **end for**

To ensure that it is possible to approach consensus among the nodes, we need the following assumption on the mixing properties of the matrix sequence $W(k)$. In fact, this assumption is a slight modification of Assumption 2.1.4, specifically regarding its last property.

Assumption 2.3.1 (Expected Consensus Rate)**.** We assume that there exists a constant $\chi \geq 1$ and an integer $B \geq 1$ such that, after K iterations, for any $u \in \mathbb{R}^M$ and all $l \in \{0, \dots, K/B\}$, it holds

$$\mathbb{E}_W[\|W_{l,\tau}u - u\|^2] \leq (1 - \chi^{-1})\|u\|^2,$$

where $W_{l\tau}(l\tau) = I - (I - W(l\tau)) \cdot \ldots \cdot (I - W((l+1)\tau - 1))$ and the expectation $\mathbb{E}_W$ is taken over distributions of $W(t)$ and indices $t \in \{l\tau, \dots, (l+1)\tau - 1\}$.

This assumption ensures that, in expectation after τ gossip steps with such time-varying matrices,

we improve the consensus between the nodes, i.e., how close each z_m is to $\bar{z}$, by the factor of $\frac{1}{1-p}$. As mentioned, this assumption is very general and covers many special cases of decentralized and centralized algorithms. For example, if we fix $W(k) = W$ for some fixed connected graph, we obtain a decentralized algorithm on this graph. If, at the same time, we set the matrix $W = \frac{1}{M}\mathbf{1}\mathbf{1}^T$, then it is easy to see that we get an analogue of the centralized setting with the averaging over all nodes performed in each communication step. If we take $W(k) = W$ for some fixed connected graph at every B-th step and in other steps use $W(k) = I$, we have a decentralized (and, in particular, centralized) algorithm with local steps [175, 189, 135] and communications after each B iterations. Generic Assumption 2.3.1 covers also many other settings of time-varying decentralized topologies, e.g., random topologies, cliques, B-connected graphs [148, 190]. Below we show that, under an appropriate choice of the stepsize, our Extragradient method provably converges under such a general assumption that covers centralized and decentralized settings, local steps in both centralized and decentralized settings, and changing topologies of the communication graph. Even for decentralized settings, this is novel for time-varying graphs and three different settings of monotonicity which we consider.

2.3.3 Assumptions

The next assumption is standard for the stochastic setting.

Assumption 2.3.2 (Bounded noise)**.** $F_m(z, \xi)$ is unbiased and has bounded variance, i.e., for all z,

$$\mathbb{E}[F_m(z, \xi)] = F_m(z), \ \mathbb{E}[\|F_m(z, \xi) - F_m(z)\|^2] \leq \sigma^2.$$

Our second assumption reflects the variability of the local operators compared to their mean and is usually called D-heterogeneity. This assumption is widely used in the analysis of local-steps (and not only) algorithms for minimization problems [176, 178, 189, 108, 162, 191, 41, 192, 193]. Moreover, [157, 158] employs this assumption for the analysis of centralized local-steps methods for saddle point problems. The authors of [117] assume $D = 0$ for the decentralized training of GANs. Even in this case, the analysis of algorithms can be challenging.

Assumption 2.3.3 (D-heterogeneity)**.** The values of the local operator exhibit bounded variability, i.e., for all z,

$$\|F_m(z) - F(z)\|^2 \leq D^2.$$

2.3.4 Convergence

In this section, we present the convergence rate results for the proposed method under different settings of Assumption 2.0.5. To present the main result, we introduce notation $\bar{z}^k := \frac{1}{M}\sum_{m=1}^M z_m^k$, $\bar{z}^{k+1/3} := \frac{1}{M}\sum_{m=1}^M z_m^{k+1/3}$ for the average among the devices' iterates and $\widehat{z}^k = \frac{1}{k+1}\sum_{i=0}^k \bar{z}^{i+1/3}$ for the average among the devices and iterates sequence, also known as the ergodic average. Finally, we

denote $\Delta = B\chi\big(D^2 B\chi + \sigma^2\big)$, which plays the role of the consensus error, i.e., the error caused by the impossibility of reaching exact consensus between the nodes. Note that the data heterogeneity appears in the convergence rates only through the quantity Δ.

Theorem 2.3.4. Let Assumptions 2.3.1, 2.0.4, 2.3.2, 2.3.3 hold and the sequences $\bar{z}^k$, $\widehat{z}^k$ be generated by Algorithm 8 that is run for $K > 0$ iterations. Then,

• **Strongly-monotone case:** under Assumption 2.0.5(SM), with $\gamma = \tilde{\mathcal{O}}\left(\min\left\{\frac{1}{B\chi L}, \frac{1}{\mu K}\right\}\right)$ it holds that

$$\mathbb{E}\left[\|\bar{z}^{K+1} - z^*\|^2\right] = \tilde{\mathcal{O}}\left(\|z^0 - z^*\|^2 \cdot \exp\left(-\frac{\mu K}{240 L B\chi}\right) + \frac{\sigma^2}{\mu^2 MK} + \frac{L^2\Delta}{\mu^4 K^2}\right); \tag{2.18}$$

• **Monotone case:** under Assumption 2.0.5(M), for any convex compact $\mathcal{C}$ s.t. $z^0, z^* \in \mathcal{C}$ and $\max_{z,z'\in\mathcal{C}} \|z - z'\| \leq \Omega_{\mathcal{C}}$, with $\gamma = \mathcal{O}\left(\min\left\{\frac{1}{L}, \left(\frac{\Omega_{\mathcal{C}}^2 M}{K\sigma^2}\right)^{\frac{1}{2}}, \left(\frac{\Omega_{\mathcal{C}}^2}{K^2L^2\Delta}\right)^{\frac{1}{4}}\right\}\right)$ it holds that

$$\mathrm{Gap}\left(\widehat{z}^K\right) = \mathcal{O}\left(\frac{L\Omega_{\mathcal{C}}^2}{K} + \frac{\sigma\Omega_{\mathcal{C}}}{\sqrt{MK}} + \frac{\sqrt{L\Omega_{\mathcal{C}}^3\sqrt{\Delta}}}{\sqrt{K}} + \sqrt{\frac{(\Delta + L^2\Omega_{\mathcal{C}}^2)\Omega_{\mathcal{C}}\sqrt{\Delta}}{KL}}\right).$$

Under the additional assumption that, for all k, $\|\bar{z}^k\| \leq \Omega$,
with $\gamma = \mathcal{O}\left(\min\left\{\frac{1}{L}, \left(\frac{\Omega_{\mathcal{C}}^2 M}{K\sigma^2}\right)^{\frac{1}{2}}, \left(\frac{\Omega_{\mathcal{C}}^2}{K^2L^2\Delta}\right)^{\frac{1}{4}}, \left(\frac{\Omega_{\mathcal{C}}^2}{K((\Omega+\Omega_{\mathcal{C}})L\sqrt{\Delta}+\Delta)}\right)^{\frac{1}{2}}\right\}\right)$, we have that

$$\mathrm{Gap}\left(\widehat{z}^K\right) = \mathcal{O}\left(\frac{L\Omega_{\mathcal{C}}^2}{K} + \frac{\sigma\Omega_{\mathcal{C}}}{\sqrt{MK}} + \frac{\sqrt{L\Omega_{\mathcal{C}}^3\sqrt{\Delta}}}{K^{3/4}} + \sqrt{\frac{((\Omega + \Omega_{\mathcal{C}})L\sqrt{\Delta} + \Delta)\Omega_{\mathcal{C}}^2}{K}}\right);$$

• **Non-monotone case:** under Assumption 2.0.5(NM) and if $\|z^0\| \leq \Omega, \|z^*\| \leq \Omega$, with $\gamma = \mathcal{O}\left(\min\left\{\frac{1}{L}, \left(\frac{\Omega^2}{K^2L^2\Delta}\right)^{\frac{1}{4}}\right\}\right)$:

$$\mathbb{E}\left[\frac{1}{K+1}\sum_{k=0}^{K}\|F(\bar{z}^k)\|^2\right] = \mathcal{O}\left(\frac{L^2\Omega^2}{K} + \frac{\sigma^2}{M} + L\Omega\sqrt{\Delta} + \frac{\sqrt{L\Omega\Delta^{3/4}}}{\sqrt{K}}\right).$$

Under the additional assumption that, for all k, $\|\bar{z}^k\| \leq \Omega$, with $\gamma = \mathcal{O}\left(\min\left\{\frac{1}{L}, \left(\frac{\Omega^2}{KL\Delta}\right)^{\frac{1}{3}}\right\}\right)$,
we have that

$$\mathbb{E}\left[\frac{1}{K+1}\sum_{k=0}^{K}\|F(\bar{z}^k)\|^2\right] = \mathcal{O}\left(\frac{L^2\Omega^2}{K} + \frac{\sigma^2}{M} + \frac{(L\Omega\Delta)^{2/3}}{K^{1/3}} + L\Omega\sqrt{\Delta}\right).$$

We underline that the standard analysis [85] does not apply for the following reasons. Our algorithm includes an additional communication step (line 8) between the computational nodes, which results in the impossibility for all the nodes to possess the same information regarding the global operator $F(z)$ and the current iterate z. This, in order, leads to a biased oracle that,

unlike existing works, has to be analyzed in the setting of an unbounded feasible set, which is quite challenging. Surprisingly, in the end, we still manage to analyze our algorithm under the very general Assumption 2.3.1 and we are not aware of any results with similar generality of the settings: different network topologies (including time-varying), distributed architectures, different monotonicity assumptions.

The provided convergence rates exhibit explicit dependence on the problem parameters: the network characterized by the mixing time B and the mixing factor χ, the data heterogeneity D (these three quantities appear in the convergence rates only through the quantity Δ), the variance σ^2 of the noise in the data, the Lipschitz constant L, the strong monotonicity parameter μ, and the number of nodes/devices M. Thus, our rates facilitate the evaluation of how different properties, such as data heterogeneity, noise level, and network characteristics, influence the convergence rates. This, in particular, opens an opportunity for a meta-optimization process if we can design the network and adjust M, B, χ to achieve faster convergence.

We now discuss the convergence results obtained in the theorem and compare them with existing algorithms (see Table 2.2) and their guarantees. Firstly, all the estimates have a similar multi-term structure. The first term corresponds to the deterministic setting and is similar to existing methods for smooth variational inequalities in the non-distributed setting. Only in the strongly convex case, there is an additional factor τ/p that increases the condition number L/μ of the problem. The second (stochastic) term is also standard for the non-distributed setting and corresponds to the stochastic nature of the problem. Note that, for a very general distributed setting, we have managed to obtain the corresponding terms similar to the non-distributed setting. Moreover, we can see the benefit of exploiting distributed computations: the leading stochastic term depends on σ^2/M that decreases as the number M of the nodes increases. The other terms correspond to the consensus error Δ and are due to the imperfect communications between the nodes, i.e., that all the nodes cannot have exactly the same information about the current iterate. Importantly, in all cases, this error does not make the overall convergence worse since the dependence on K is no worse in these terms than the dependence on K in the stochastic term. In the experimental section, we illustrate that the network error is not an artifact of the analysis but is indeed present in practice.

Theorem 2.3.4 is formulated for a fixed budget of iterations K and the corresponding stepsizes γ that depend on K, which is pretty standard in the literature [85, 194], where many algorithms fix the stepsize depending on the budget of the iterations. One can use a simple restarting procedure that allows to extend the results of Theorem 2.3.4 to any-time convergence without a-priori fixing K. The idea is to set $K_t = 2^t$ for $t = 0, 1, \ldots$ and restart the algorithm after each K_t iterations. Next, we provide refined comments for each particular setting of monotonicity.

2.3.5 Experiments

In this subsection, we present two sets of experiments to validate the performance of Algorithm 8. We verify the obtained convergence guarantees on two examples: a strongly monotone bilinear

problem and a monotone bilinear problem. In the second group of experiments, we explore the non-monotone case with an application to GAN training.

Verifying Theoretical Convergence Rate

First, we focus on verifying whether the actual behavior of Algorithm 8 is predicted by the theoretical convergence rate in Theorem 2.3.4.

We consider a distributed bilinear saddle point problem (2.3) with the objective functions

$$f_m(x, y) = \frac{a}{2}\|x\|^2 + bx^\top y - \frac{a}{2}\|y\|^2 + c_m^\top x,$$

where $x, y, c_m \in \mathbb{R}^d$, $a, b \in \mathbb{R}$ and $m \in \{1, \ldots, M\}$. This set of functions satisfy Assumptions 2.0.4, 2.0.5, 2.3.3 with constants $\mu = a, L^2 = a^2 + b^2$, $D = \max_m \|c_m - \bar{c}\|$. In this section, we use a ring topology on $M = 20$ nodes with uniform averaging weights, and we set the dimension $n = 5$, $b = 1$, $D \approx 3$, and keep $\tau = 1$. The value of the parameter p in this setting is approximately 0.288 [195, Table 1]. To satisfy Assumption 2.3.2, we generate stochastic gradients by adding to the exact gradients unbiased Gaussian noise with variance σ^2.

Convergence Behaviour. In Figure 2.3, we show the convergence of Algorithm 8 with a fixed stepsize on the strongly monotone ($a = 1$) and monotone ($a = 0$) instances. In the strongly monotone case, we see a linear convergence up to some level defined by the heterogeneity parameter and the noise. The convergence for the non-strongly monotone problem is slower, but we also see a linear convergence up to some level (for bilinear problems this behavior is expected from the theoretical point of view [78]). Note that the convergence to some limiting accuracy is expected since when a constant stepsize is used in stochastic optimization/stochastic variational inequalities with strong convexity/monotonicity, algorithms are usually guaranteed to converge only to a vicinity of the solution, see, e.g., Theorem 2 in [196]. This is also in accordance with Theorem 2.3.4 that, for a fixed stepsize, guarantees the convergence to some non-zero limit accuracy and says that, to achieve the zero error, one needs to choose a decreasing stepsize. We additionally validate later that with a decreasing stepsize, the algorithm can converge to the zero error.

Dependence on the Heterogeneity parameter D. In the second set of experiments, we aim to verify the dependence on the data heterogeneity parameter D. Therefore, we consider the setting when $\sigma^2 = 0$. From our theory, equation (2.18), we predict that the most significant term in the convergence rate when $\sigma^2 = 0$ scales as $\mathcal{O}\left(\frac{D^2}{p^2K^2}\right)$ (since the primary goal of this experiment is to study the dependence on p, D, K, we omit all the other fixed parameters for simplicity). We take $b = 1$, $a = 1$ and conduct experiments with the number of iterations needed to achieve the error $\frac{1}{M}\sum_{m=1}^M \|z_m^k - z^*\|^2 < \varepsilon$ for different ε. In all these experiments, the stepsize is tuned individually.

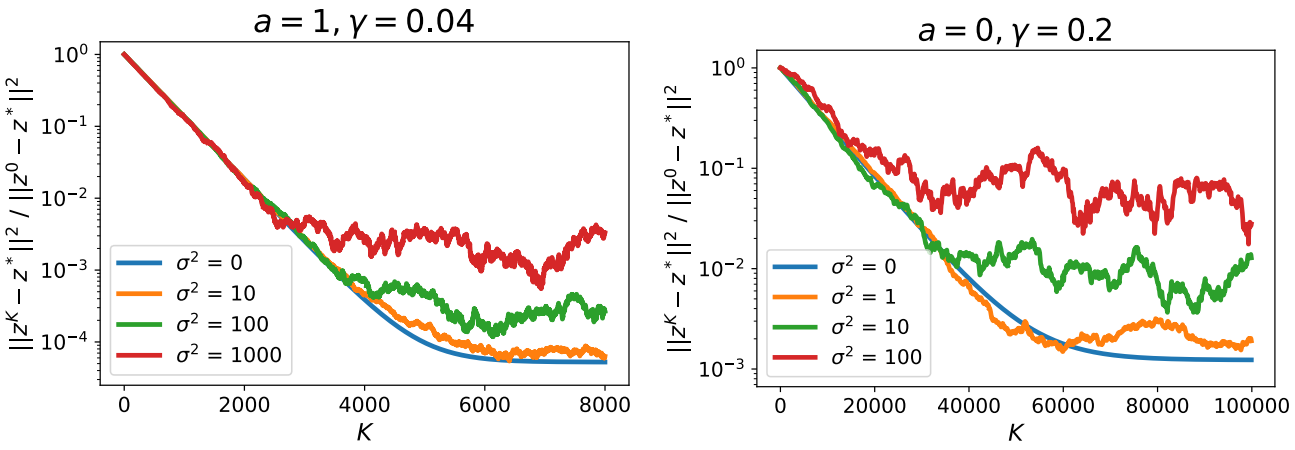


Figure 2.3: Convergence of Algorithm 8 with constant stepsize in the presence of stochastic noise in strongly monotone (left) and monotone (right) cases. We observe linear convergence up to an error floor depending on the noise variance and problem parameters (Theorem 2.3.4).

First, we verify the power of K in the bounds. For this experiment, we keep D, p constant and vary the accuracy ε. As we can see from the leftmost subplot in Figure 2.4, the number of iterations scales as $K \propto \frac{1}{\sqrt{\varepsilon}}$, confirming the predicted $\mathcal{O}\left(\frac{1}{K^2}\right)$ dependency of the error on K. Next, we measure the number of iterations sufficient to reach the error $\varepsilon = 0.01$ while varying D. The middle plot shows that the number of iterations scales proportionally to D, (showing $D \propto K$). Lastly, we depict the number of iterations to reach $\varepsilon = 0.01$ while changing the graph parameter p and again observe $\frac{1}{p} \propto K$. Summarizing, these experiments verify the $\mathcal{O}\left(\frac{D^2}{p^2K^2}\right)$ term in the convergence rate.

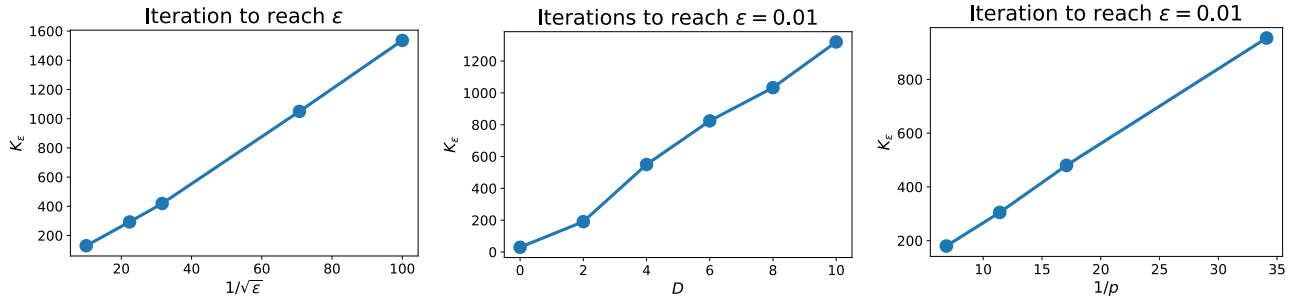


Figure 2.4: Verifying the $\mathcal{O}\left(\frac{D^2}{p^2K^2}\right)$ convergence rate for the strongly monotone noiseless ($\sigma^2 = 0$) case.

Decreasing Stepsize. As mentioned before, our theoretical results in Theorem 2.3.4 hold for the fixed stepsizes that optimize the error of the obtained approximate solution given the budget of K iterations. Thus, given a target accuracy $\epsilon > 0$ and using the bounds in Theorem 2.3.4, we can choose the number of iterations $K = K(\epsilon)$ to guarantee the accuracy ϵ. In turn, based on the value $K(\epsilon)$, we choose the fixed stepsize $\gamma = \gamma(K(\epsilon))$. This procedure of defining the stepsize based on the target accuracy and the corresponding budget of iterations is standard in the literature, see, e.g., [85, 194].

In the third group of experiments, we numerically illustrate that Algorithm 8 can achieve

arbitrarily small error when implemented with relatively simple decreasing steps. We note that, despite not analyzed theoretically, the used in the experiments decreasing stepsize leads to a good performance of the algorithm, which additionally illustrates the flexibility of our approach for practical purposes.

We consider two cases: with and without stochastic noise, i.e., we fix either $\sigma = 0$ or $\sigma = 100$. During the training, we decrease the stepsize as $\gamma_k = \frac{\alpha}{k+\beta}$, where k is the current iteration number. We set $\alpha = 40, \beta = 800$ in the noiseless case and $\alpha = 15, \beta = 150$ when $\sigma^2 = 100$. In Figure 2.5, we observe that the error decreases to zero at a sublinear rate. This contrasts with the limiting behavior we see in Figure 2.3, where the algorithm cannot optimize below a certain threshold. Sublinear convergence may be explained by the second and third terms in the estimate (2.18).

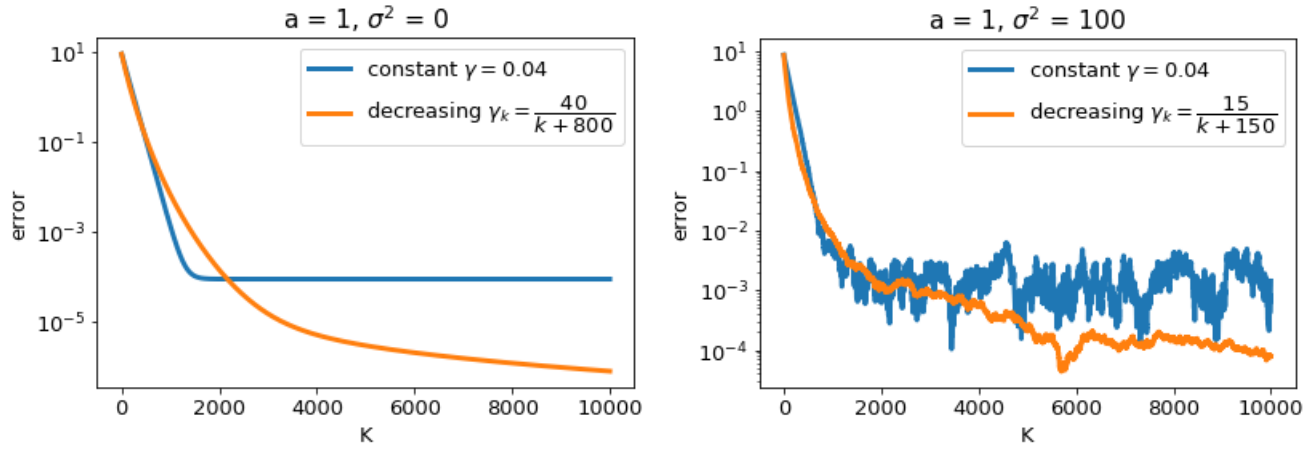


Figure 2.5: Convergence of Algorithm 8 with the decreasing stepsizes in the noiseless (left) and stochastic (right) cases.

Training GANs

Our algorithm allows for the combination of different communication graph topologies and local steps in the distributed learning setting. Thus, the goal of this section is to illustrate this empirically through experiments on GANs training.

Data and model. We consider the CIFAR-10 [183] dataset containing 60000 images, equally distributed over 10 classes. We increase the size of the dataset by four times using transformations and adding noise. We simulate a distributed setup of 16 nodes on two GPUs and use Ray [197]. To emulate the heterogeneous setting, we partition the dataset into 16 subsets. For each subset, we select a major class that forms 20% of the data, while the rest of the data split is filled uniformly by the other classes. As a basic architecture, we choose DCGAN [182], conditioned by class labels, similarly to [198]. Next, we comment on the relation between the main assumptions of the theoretical analysis and the example of training GANs. First of all, the goal of the DCGAN training experiment is to study how the network topology influences the convergence of the algorithm. Even if the assumptions do not hold, we observe that the algorithm performs quite well and is flexible regarding the choice of the topology. Secondly, as we write in the main text, the made assumptions

are quite standard and widely used in the literature. In particular, Assumptions 2.0.4 and 2.10 are classical and are often used in the literature, see, e.g. [85], including the literature on the neural networks training. Assumption 2.3.3 is also widely used [176, 178, 189, 108, 162, 41, 157, 158, 117], and holds with a small constant D when the data is uniformly split among the devices. Such splitting can be easily made when one uses a computational cluster with a large amount of data, e.g., images. In Section 2.3.5, we deliberately consider a more difficult setup and make the distribution of images over the nodes not uniform, but heterogeneous. As we see, the results of the experiments are quite promising in this case. Assumption 2.0.5 (NM) is also used in the literature on the algorithms for training GANs and their analysis [117, 99, 116]. Moreover, this assumption is shown to hold in some nonconvex minimization problems, for example, when SGD is employed for training neural networks [199, 200, 201].

We chose Adam [202] as the optimizer. We make one local Adam step and then one gossip averaging step with time-varying matrices W^k—similar to Algorithm 8. In Table 2.3, we give the hyperparameters for all the experiments.

Table 2.3: Hyperparameters for DCGAN training.

Hyperparameters	
Batch size	=64
Weight clipping for the discriminator	=0.01
Learning rate for generator and discriminator	=0.002
Initialization:	normal
Other parameters:	default in PyTorch

Setting. We compare the following three topologies (and the corresponding matrices W^k):
• **Full.** Full graph at the end of each epoch; otherwise, local steps. This indicates that 120 communication rounds (by communication round, we refer to the exchange of information between a pair of devices) in an epoch.
• **Local.** A full graph is generated at the conclusion of each 5th epoch; otherwise, local steps are utilized. This indicates that 24 communication rounds occur within an epoch, on average: 4 epochs without communications and 1 epoch with 120 rounds.
• **Clusters.** At the end of each epoch, clique clusters of size 4 are randomly formed (in total four cliques). This results in 24 communication rounds per epoch.

Note that the communication budget of the first approach is five times larger.

We use the same learning rate equal to 0.002 for the generator and the discriminator. The remaining parameters and features of the architecture can be found in the supplementary material.

Results. The results of the experiment are presented in Figure 2.6. In terms of the number of local epochs, all the methods converged quite close to each other and produced similar images.

In terms of communications, Local and Cluster topologies lead to much better results, and the Cluster topology is slightly better than the Local.

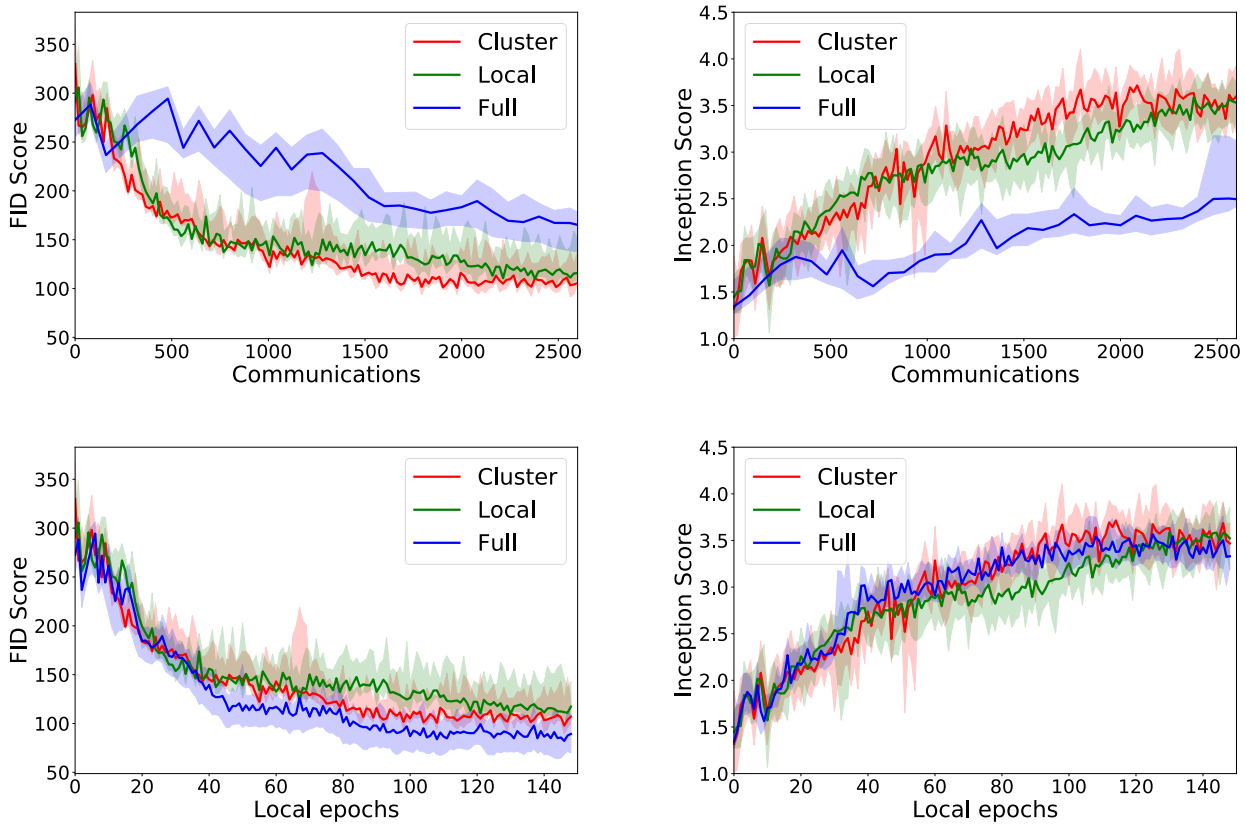


Figure 2.6: Comparison of the three network topologies in DCGAN distributed decentralized learning on CIFAR-10. FID Score and Inception Score vs the number of communications (two top), and the same Scores vs the number of local epochs (two bottom). The experiment was repeated 5 times on different random data splitting, the maximum and minimum deviations are depicted in the plots by the shade.

2.4 Using Similarity

We have already encountered the description of similarity in the previous section when we introduced Assumption 2.3.3. In the case of the minimization problem, this assumption corresponds to the uniform boundedness of the gradient difference: $\|\nabla f_m(x) - \nabla f(x)\| \leq D$. This kind of condition is indeed used in the literature, but it is not satisfied even for quadratic problems. Consequently, another description in homogeneity, namely the similarity of Hessians, has appeared:

$$\|\nabla^2 f_m - \nabla^2 f\| \leq \delta. \tag{2.19}$$

Let $f(x) = \frac{1}{N}\sum_{i=1}^{N} l(\text{model}(x, a_i), b_i)$, and $f_1(x) = \frac{1}{n}\sum_{i=1}^{n} l(\text{model}(x, a_i), b_i)$. If the objects $\{a_i\}_{i=1}^n$ for f_1 are chosen randomly and uniformly from N of f, it turns out that using facts from measure concentration (see Theorem 1.3 from [203], Theorem 2 from [41]), one can show

$$\delta = \tilde{\mathcal{O}}\left(\frac{L}{\sqrt{n}}\right) \quad \text{or even} \quad \delta = \tilde{\mathcal{O}}\left(\frac{L}{n}\right),$$

where L is the smoothness constant of $l(\text{model}(x, a), b)$. It can be seen that with increasing n, δ can decrease dramatically. This is what all works around similarity take advantage of.

2.4.1 Related Works and Contributions

For minimization problems, the study of distributed centralized problems under the Hessian similarity condition presents an interesting theoretical framework. The basic additional assumptions for this formulation include μ-strong convexity and L-smoothness. Almost all current results in the field are obtained in this setting. Already the first results have shown that under hessian similarity it is possible to obtain very optimistic guarantees in terms of communication costs.

The seminal paper [162] established lower communication complexity bounds under δ-similarity: ε-optimality cannot be achieve in less than $\Omega(\sqrt{\delta/\mu}\log 1/\varepsilon)$ communication rounds. Since then, a lot of effort has been devoted to design distributed schemes aiming at achieving optimal communication complexity. The authors in [108] proposed DANE, a Mirror Descent based algorithm whereby workers perform local data preconditioning via a suitably chosen Bregman divergence, and the master averages the solutions of the workers. For *quadratic* losses, DANE achieves communication complexity $\widetilde{\mathcal{O}}((\delta/\mu)^2 \log 1/\varepsilon)$; this was later improved to $\mathcal{O}((\delta/\mu)\log 1/\varepsilon)$ for nonquadratic losses in [204], where the SONATA algorithm was proposed (also implementable over mesh-networks). Improvements were achieved by employing acceleration; efforts include: DiSCO [205], an inexact damped Newton method coupled with a preconditioned conjugate gradient (to compute the Newton direction), which achieves communication complexity $\widetilde{\mathcal{O}}(\sqrt{\delta/\mu})\log 1/\varepsilon)$ for self-concordant losses; AIDE [206], which uses the Catalyst framework [207], matching the rate of DiSCO for quadratic losses; DANE-HB [208], a variant of DANE equipped with Heavy Ball momentum and matching for quadratic functions the communication complexity of DiSCO and AIDE; and SPAG [41], a preconditioned direct accelerated method, achieving for non-quadratic losses *asymptotically*

the convergence rate $\mathcal{O}((1-1/\sqrt{\beta/\mu})^k)$ (k is the iteration index)–the worst-case rate is still $\mathcal{O}(\sqrt{L/\mu})\log 1/\varepsilon)$.

Finally, higher-order methods employing preconditioning have been studied in [192, 191, 209]: [192] proposed DiRegINA, a decentralization of the cubic regularization of the Newton method, where workers build Newton directions by sampling local Hessians; [191] introduced ACN, an inexact accelerated cubic-regularized Newton's method, with improved complexity with respect to [192]; and [209] extended the Catalyst framework [207] to the distributed setting (including mesh networks), proposing Acc SONATA.

In summary, the above tour of the relevant literature shows that none of the existing works consider similarity for saddle point problems and variational inequalities. The goal of this section is to close this gap .

Our major results are summarized below.

• **Lower bounds.** We establish lower complexity bounds for the δ-related variational inequalities (Assumption 2.4.1).

• **Optimal algorithm.** We propose an algorithm for variational inequalities that matches the lower bounds up to logarithmic factors and exactly. It is provably faster than existing solution methods for the general problems of the form (2.1)+(2.7) under Assumptions 2.0.4+2.0.5, which do not exploit function similarity. Preliminary numerical results on distributed robust logistic regression support our theoretical findings.

2.4.2 Assumptions

We start immediately with a key assumption.

Assumption 2.4.1 (δ-relatedness)**.** Each operator F_m is δ-related. It means that the operator $F_m - F$ is δ-Lipschitz continuous, i.e. for all $u, v \in \mathbb{R}^d$ we have

$$\|F_m(u) - F(u) - F_m(v) + F(v)\| \leq \delta\|u - v\|.$$

At first glance, this assumption seems unrelated to the similarity of Hessians. However, in (2.19) the boundedness of the Hessian of $(f - f_m)$ is written out, which is equivariant to the δ-smoothness of the function $(f - f_m)$. Moreover, smoothness is equivalent to the Lipschitz continuity of the operator $\nabla(f - f_m)$, as stated in Assumption 2.4.1.

For saddle point problems, the analogue of (2.19) and Assumption 2.4.1 is written in the form:

$$\begin{aligned}
\|\nabla^2_{xx} f(x,y) - \nabla^2_{xx} f_m(x,y)\| &\leq \delta,\\
\|\nabla^2_{xy} f(x,y) - \nabla^2_{xy} f_m(x,y)\| &\leq \delta,\\
\|\nabla^2_{yy} f(x,y) - \nabla^2_{yy} f_m(x,y)\| &\leq \delta.
\end{aligned}$$

2.4.3 Lower Bounds

In this subsection, we establish lower complexity bounds for centralized algorithms under Assumption 2.4.1. As for past lower bounds, we introduce a definition of the oracle for which they are valid. Definition 2.4.2 is very close to what we introduced earlier in Definitions 2.1.5 and 2.2.3.

Definition 2.4.2 (Oracle)**.** Each agent m has its own local memory $\mathcal{M}_m$ with initialization $\mathcal{M}_m = \{0\}$. $\mathcal{M}_m$ is updated as follows. At each iteration, the algorithm either performs local computations or communicates.

• **Local computation:** At each local iteration, device m can sample uniformly and independently batch S_m of any size b from $\{F_{m,i}\}$ and adds to its $\mathcal{M}_m$ a finite number of points z, satisfying

$$z \in \operatorname{span}\left\{z' \ , \ F_m(z''), \operatorname{prox}_{\rho g}\left(\operatorname{span}\left\{z''' \ , \ F_m(z'')\right\}\right)\right\}$$

for $z', z'', z''' \in \mathcal{M}_m$ and $\rho > 0$.

• **Communication:** Upon communication rounds among the neighboring nodes, and at communication time t, $\mathcal{M}_m$ is updated according to

$$\mathcal{M}_m := \operatorname{span}\left\{\bigcup_{i=1}^{M} \mathcal{M}_i\right\}.$$

• **Output:** The final global output is calculated as $\hat{z} \in \operatorname{span}\left\{\bigcup_{m=1}^{M} \mathcal{M}_m\right\}$.

We are in the position to state our main results on lower communication complexity–Theorem 2.4.3.

Theorem 2.4.3. Let $L, \mu, \delta > 0$ (with $L > \mu$ and $L > \delta$), and $K \in \mathcal{N}$. There exists a centralized variational inequality for which the following statements are true:

- F are L-Lipschitz continuous, μ – strongly monotone,
- F_m are L-Lipschitz continuous, μ – monotone, δ-related,
- size $d \geq \max\left\{2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right), 2K\right\}$, where $\alpha = \left(\frac{64\mu}{\delta}\right)^2$ and $q = \frac{1}{2}\left(2 + \alpha - \sqrt{\alpha^2 + 4\alpha}\right) \in (0; 1)$,
- the solution of the problem is non-zero: $z^* \neq 0$.

Then for any output $\hat{z}$ of any procedure (Definition 2.4.2) with K communication rounds, one can obtain the following estimate:

$$\|\hat{z} - z^*\|^2 = \Omega\left(\exp\left(-\frac{K}{\Delta} \cdot \frac{1}{\frac{1}{8}\sqrt{1 + \left(\frac{\delta}{32\mu}\right)^2} - \frac{1}{8}}\right)\|z_0 - z^*\|^2\right).$$

Corollary 2.4.4. In the setting of Theorem 2.4.3, the number of communication rounds required

to obtain a ε-solution is lower bounded by

$$\Omega\left(\Delta\left(1+\frac{\delta}{\mu}\right)\cdot\log\left(\frac{\|z^*\|^2}{\varepsilon}\right)\right). \tag{2.20}$$

These lower complexity bounds show an expected dependence on the optimization parameters and network quantities. Specifically, the number of communications scale proportionally to δ/μ–this generalizes prvious lower bounds (Sections 2.1 and 2.2) that do not account for such similarity, resulting instead in the more pessimistic dependence on L/μ–as we konw, typically $\delta \leq L$.

2.4.4 Optimal Algorithm

Our optimal algorithm is formally described in Algorithm 9. Some insights on the genesis of this method are discussed next.

• Consider for a moment the convex minimization problem $\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{M}\sum_{m=1}^M f_m(x)$, under (2.19). Following [108] we can solve it by invoking the Mirror Descent algorithm, which reads

$$x^{k+1} = \arg\min_{x\in\mathbb{R}^d}\left[\langle\eta\nabla f(x^k), x\rangle + D_\phi(x, x^k)\right], \tag{2.21}$$

where $D_\phi(x,y) = \phi(x) - \phi(y) - \langle\nabla\phi(y), x-y\rangle$ is the Bregman divergence, with function $\phi(x) = f_1(x) + \frac{\delta}{2}\|x\|^2$. It is shown that we can take stepsize $\eta = 1$ [210, 41]. Therefore, (2.21) can be rewritten as

$$x^{k+1} = \arg\min_x\left[\frac{1}{\delta}f_1(x) + \frac{1}{2}\left\|x - x^k + \frac{1}{\delta}(\nabla f(x^k) - \nabla f_1(x^k))\right\|^2\right].$$

Using the operator $H(u) := \frac{1}{\delta}\nabla f_1(u) + \left(u - x^k + \frac{1}{\delta}(\nabla f(x^k) - \nabla f_1(x^k))\right)$ we can write this as a variational inequality:

$$\text{Find}\ \ \hat{u}^k \in \mathbb{R}^d\ \ \text{such that}\ \ \langle H(\hat{u}^k), x - \hat{u}^k\rangle \geq 0,\ \ \forall x \in \mathbb{R}^d; \tag{2.22}$$

Noting that in Algorithm 9 $\gamma \sim \frac{1}{\delta}$ (see Appendix A.2.4), one infers the connection between (2.22) and the updates in lines 3 (i) and 3 (ii). The extra step as in line 3 (iii) is due to the fact that Algorithm 9 solves variational inequalities (and not a classical minimization as postulated above). That is why we combine the idea of (2.22) with the Forward-Backward-Forward algorithm [86] or the Extragradient [78] method, which leads to the step in line 3 (iii).

• Another interpretation of the proposed algorithm comes from looking at the problem (2.1)+(2.7) as a composite problem, with objective operator $R(z) + Q(z)$, with $R(z) = F_1(z)$ and $Q(z) = \frac{1}{M}\sum_{m=1}^M(F_m(z) - F_1(z))$. The first operator F_1 is L-Lipschitz and monotone while Q is δ-Lipschitz and, in general, non-monotone. Such type of problems can be solved invoking sliding techniques [211].

It is not difficult to check that Algorithm 9 is an instance of the oracle introduced in Defini-

Algorithm 9 Server Mirror Method for Data Similarity `SMMDS`

Parameters: stepsize γ, accuracy e;
Initialization: Choose $z^0 \in \mathbb{R}^d$, $z_m^0 = z^0$, for all $m \in [M]$;

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Each worker m computes $F_m(z^k)$ and sends it to the master;
3: The master node:
(i) computes $v^k = z^k - \gamma \cdot \big(F(z^k) - F_1(z^k)\big)$;
(ii) finds u^k, s.t. $\|u^k - \hat{u}^k\|^2 \leq e$, where $\hat{u}^k$ is the solution of:

$$\text{Find} \;\; \hat{u}^k \in \mathbb{R}^d \;\; \text{such that} \;\; \langle G(\hat{u}^k), z - \hat{u}^k \rangle + g(z) - g(\hat{u}^k) \geq 0, \;\; \forall z \in \mathbb{R}^d; \tag{2.23}$$

with $G(z) := \gamma F_1(z) + (z - v^k)$
(iii) updates $z^{k+1} = u^k + \gamma \cdot (F(z^k) - F_1(z^k) - F(u^k) + F_1(u^k))$ and broadcasts z^{k+1} to the workers
4: **end for**

tion 2.4.2. It accommodates either exact solutions of the strongly monotone subproblems (2.23) (corresponding to $e = 0$) or inexact ones (up to tolerance $e > 0$)–the latter can be computed, e.g., using Extragradient method [85], which is optimal in this case.

The communication complexity of the method is proved in the next theorem, which certifies that the proposed algorithm is optimal, i.e., achieves the lower bound (2.20) on the number of required communications. We refer to Appendix A.2.4 in the supplementary material for a detailed description of the algorithmic tuning, as well as a study of the computational complexity when the Extragradient method is employed to solve subproblems (2.23) (up to a suitably chosen tolerance).

Theorem 2.4.5. Consider the problem (2.1)+(2.7) under Assumptions 2.0.4,2.0.5,2.4.1. Let $\{z^k\}$ be the sequence generated by Algorithm 9 with tuning as described in Appendix A.2.4. Then, given $\varepsilon > 0$, the number of communication rounds for $\|z^k - z^*\|^2 \leq \varepsilon$ is

$$\mathcal{O}\left(\left(1 + \frac{\delta}{\mu}\right) \log\left(\frac{1}{\varepsilon}\right)\right)$$

.

2.4.5 Experiments

We simulate the robust linear regression problem ((2.6)) defined as

$$\min_{x \in \mathbb{R}^d} \max_{\|r_i\| \leq R} \frac{1}{2n} \sum_{i=1}^{n} (x^T(a_i + r_i) - b_i)^2 + \frac{\lambda}{2}\|x\|^2 - \frac{\beta}{2}\|r_i\|^2. \tag{2.24}$$

We consider a network with 25 workers. We test Algorithm 9 wherein the subproblems (2.23) at the first node are solved with high accuracy using the Extragradient method.

Our first experiment uses synthetic data, which allows us to control the factor δ, measuring

statistical similarity of functions over different nodes. Specifically, we assume all local datasets of size $n = 100$. The dataset $\{\hat{a}_i, \hat{b}_i\}_{i=1}^n$ at the first node is generated randomly, with each entry of $\hat{a}_i$ and $\hat{b}_i$, $i = 1, \ldots, n$ drawn from the Standard Gaussian distribution. The datasets at the other workers' sides, $m = 2, \ldots, M$, are obtained by perturbing $\{\hat{a}_i, \hat{b}_i\}_{i=1}^n$ with random noise ξ_i of controlled variance.

Figure 2.7 compares the performance of Algorithm 9 and the Centralized Extragradient method (Section 2.1) applied to the problem (2.24), under different levels of noise added to local datasets (level of similarity). It can be seen that Algorithm 9 consistently outperforms the Extragradient method in terms of the number of communications—the smaller the noise (the more similar the local functions are), the larger the gap between the two algorithms (in favor of Algorithm 9). On the other hand, at high noise (amplitude 10.0) the performance of Extragradient and Algorithm 9 becomes comparable.

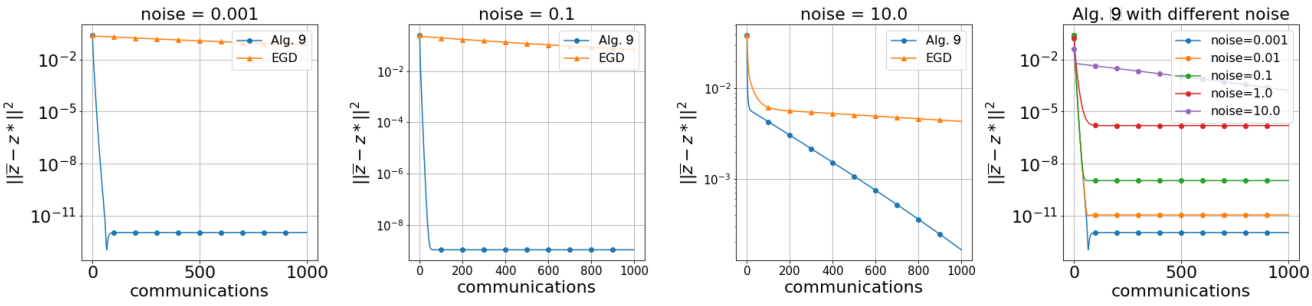


Figure 2.7: Simulated data, 25 workers, ambient dimension = 40

Our second experiment uses real data, specifically LIBSVM datasets [171]. In this scenario, we do not use additional noise, but we can still control the data similarity by choosing the number of workers. The larger the number of workers, the less similar the local functions (less data at each node). Figure 2.8 compares Algorithm 9 and the Extragradient method: we plot the distance of the iterates from the solution versus the number of communications. Interestingly, Algorithm 9 compares favorably even when the number of workers becomes large.

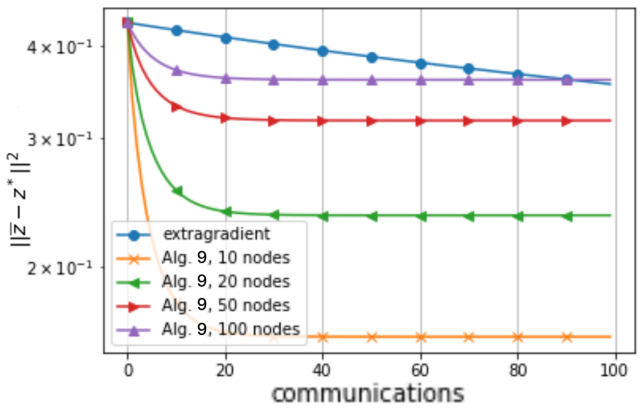


Figure 2.8: a9a dataset

2.5 Compressed Communications

We continue to make communications during the solving of (2.1)+(2.7) more effective. Note that in this section we place $g \equiv 0$.

In this section, we address another popular technique for managing communication bottlenecks—communication compression. The underlying principle of this technique is based on the observation that, in practical implementations, it is often advantageous to communicate messages that are compressed using (often randomized) lossy compression techniques instead of sending the full messages [44, 43]. When the number of parallel workers is sufficiently large, the noise introduced by compression diminishes, and training with compressed communication frequently yields comparable test errors while reducing the number of communicated bits. This results in faster training, both in theory and practice [60, 212].

Let us give the main definitions. We say that a (possibly) stochastic mapping $Q : \mathbb{R}^d \to \mathbb{R}^d$ is an *unbiased compression operator* if there exists a constant $q \geq 1$ such that

$$\mathbb{E}Q(z) = z, \quad \mathbb{E}\|Q(z)\|^2 \leq q\|z\|^2, \quad \forall z \in \mathbb{R}^d. \tag{2.25}$$

Further, we say that a stochastic mapping $C : \mathbb{R}^d \to \mathbb{R}^d$ is a *contractive compression operator* if there exists a constant $\delta \geq 1$ such that

$$\mathbb{E}\|C(z) - z\|^2 \leq (1 - 1/\delta)\|z\|^2, \quad \forall z \in \mathbb{R}^d. \tag{2.26}$$

If b is the number of bits needed to represent a single float (e.g., $b = 32$ or $b = 64$), then the number of bits needed to represent a generic vector $z \in \mathbb{R}^d$ is $\|z\|_{\text{bits}} := bd$. To describe how much a compression operator reduces its input vector on average, we define the notion of expected density, denoted via $\beta^{-1} := \frac{1}{bd}\mathbb{E}\|Q(z)\|_{\text{bits}}$, where $\|Q(z)\|_{\text{bits}}$ is the number of bits needed to represent the quantized vector $Q(z)$. Note that $\beta \geq 1$. For the Rand$k$ operator, we have $q = \beta = d/k$ (see Chapter 5).

2.5.1 Related Works and Contributions

Despite the fact that the use of compression is one of the most popular communication-efficient approaches for distributed minimization problems, no work has yet paid attention to the compression technique neither for distributed saddle point problems nor for variational inequalities, with the exception of the work [213], which relies on rounding to the nearest integer multiple of a certain quantity. This compression mechanism does not offer theoretical benefits and does not even lead to convergence to the solution since the errors introduced through rounding persist and prevent the method from solving the problem.

In this section, we investigate whether it is possible to design communication-efficient algorithms for solving distributed variational inequalities by borrowing generic communication compression techniques (2.25) and (2.26) from the optimization literature [44, 43, 60, 212, 45] and embedding

them into established, efficient methods for solving variational inequalities [78, 80, 85, 214]. Our contribution can be summarized as follows:

• **Two types of compressors.** We develop two approaches for distributed variational inequalities depending on whether we use unbiased (2.25) or contractive (2.26) compressors, as each type of compressor demands a different algorithmic design and a different analysis. In particular, contractive compressors are notoriously difficult to analyze even for optimization problems [215, 45]. Our method based on unbiased compressors is called `MASHA1` (Algorithm 11), and our method based on contractive compressors is called `MASHA2` (Algorithm 12).

• **Theoretical complexity results.** We establish a number of theoretical complexity results for our methods, which we summarize in Table 2.4. We consider the strongly monotone, monotone regimes, as well as the more general non-monotone/Minty regime. In the strongly monotone case we obtain linear convergence results ($O(\log 1/\epsilon)$) in terms of the distance to solution, in the monotone we obtain fast sublinear convergence results ($O(1/\epsilon)$) in terms of the gap function, and in the non-monotone case we have sublinear convergence results ($O(1/\epsilon^2)$) in terms of the Euclidean norm of the operator. To get an estimate for the number of information transmitted, one need to multiply the estimates from Table 2.4 by $1/\beta$. Then we get that from the point of view of the transmitted information (and also time for communications), `MASHA1` is better by a factor $\sqrt{1/\beta + 1/M}$ (M – number of workers) in comparison with the classical Extragradient. It means that we get an acceleration of $\min\{\sqrt{\beta}; \sqrt{M}\}$ times. For example, ADIANA from [216](the theoretical SOTA method with unbiased compressions for strongly convex minimization) has the same accelaration. The same situation is with `MASHA2`. The method has the same compression dependent multiplier as ECLK from [217] (the theoretical SOTA with contractive compression for minimization).

• **Stochastic case and variance reduction.** `MASHA1` and `MASHA2` are designed to handle the *deterministic* setting. However, in practice, computation of the full operators/gradients is expensive, necessitating consideration of *stochastic* realizations. In particular, a popular case is when each operator/gradient has a finite-sum structure on its own, e.g. , finite-sum of batches. To address this issue, we consider two modifications: `VR-MASHA1` (Algorithm 13) and `VR-MASHA2` (Algorithm 14). Both modifications are enhanced with bespoke *variance-reduction* techniques for improved theoretical and practical performance. These results are relevant in the non-distributed case. To the best of our knowledge, we are the first to consider variance reduction for non-monotone variational inequalities. We found only one paper addressing non-convex-concave saddle point problems [218] under the PL condition.

Table 2.4: Summary of our iteration complexity results for finding an ε-solution for problem (2.1) in the deterministic (i.e., (2.7)) with only device compression, deterministic with bidirectional (device-server) compression, stochastic (i.e., (2.7)+(2.28)) and federated learning/partial participation (i.e., (2.7)+(2.29)) setups. In the strongly monotone (strongly convex - strongly convex) case, convergence is measured by the distance to the solution. In the monotone(convex-concave) case, convergence is measured in terms of the gap function (2.8). In non-monotone (non-convex-non-concave) case convergence is measured in terms of the norm of the operator.

		Strongly monotone	Monotone	Non monotone
Deter. (Device)	MASHA1 Alg 11	$\tilde{\mathcal{O}}\left(\beta+\sqrt{\beta+\frac{q\beta}{M}}\cdot\frac{L}{\mu}\right)$	$\mathcal{O}\left(\sqrt{\beta+\frac{q\beta}{M}}\cdot\frac{LR^2}{\varepsilon}\right)$	$\mathcal{O}\left(\left(\beta+\frac{q\beta}{M}\right)\cdot\frac{L^2R^2}{\varepsilon^2}\right)$
	MASHA2 Alg 12	$\tilde{\mathcal{O}}\left(\beta+\delta\sqrt{\beta}\cdot\frac{L}{\mu}\right)$	$\mathcal{O}\left(\delta\sqrt{\beta}\cdot\frac{LR^2}{\varepsilon}\right)$	$\mathcal{O}\left(\delta^2\beta\cdot\frac{L^2R^2}{\varepsilon^2}\right)$
Deter. (Bidirec.)	MASHA1 Alg 11	$\tilde{\mathcal{O}}\left(\beta+\sqrt{q\beta+\frac{q^2\beta}{M}}\cdot\frac{L}{\mu}\right)$	$\mathcal{O}\left(\sqrt{q\beta+\frac{q^2\beta}{M}}\cdot\frac{LR^2}{\varepsilon}\right)$	$\mathcal{O}\left(\left(q\beta+\frac{q^2\beta}{M}\right)\cdot\frac{L^2R^2}{\varepsilon^2}\right)$
	MASHA2 Alg 12	$\tilde{\mathcal{O}}\left(\beta+\delta^2\sqrt{\beta}\cdot\frac{L}{\mu}\right)$	$\mathcal{O}\left(\delta^2\sqrt{\beta}\cdot\frac{LR^2}{\varepsilon}\right)$	$\mathcal{O}\left(\delta^4\beta\cdot\frac{L^2R^2}{\varepsilon^2}\right)$
Stoch. (F-S) (2.28)	VR-MASHA1 Alg 13	$\tilde{\mathcal{O}}\left(\beta+n+\max\{\sqrt{\beta};\sqrt{n}\}\sqrt{1+\frac{q}{M}}\cdot\frac{L}{\mu}\right)$	$\mathcal{O}\left(\max\{\sqrt{\beta};\sqrt{n}\}\sqrt{1+\frac{q}{M}}\cdot\frac{LR^2}{\varepsilon}\right)$	$\mathcal{O}\left(\left(\max\{\beta;n\}\left(1+\frac{q}{M}\right)\right)\cdot\frac{L^2R^2}{\varepsilon^2}\right)$
	VR-MASHA2 Alg 14	$\tilde{\mathcal{O}}\left(\beta+n+\max\{\sqrt{\beta};\sqrt{n}\}\delta\cdot\frac{L}{\mu}\right)$	$\mathcal{O}\left(\max\{\sqrt{\beta};\sqrt{n}\}\delta\cdot\frac{LR^2}{\varepsilon}\right)$	$\mathcal{O}\left(\max\{\beta;n\}\delta^2\cdot\frac{L^2R^2}{\varepsilon^2}\right)$
FL (PP) (2.29)	PP-MASHA1 Alg 15	$\tilde{\mathcal{O}}\left(\frac{\beta M}{b}+\sqrt{\frac{\beta M}{b}+\frac{q\beta M}{b}}\cdot\frac{L}{\mu}\right)$	$\mathcal{O}\left(\sqrt{\frac{\beta M}{b}+\frac{q\beta M}{b}}\cdot\frac{LR^2}{\varepsilon}\right)$	$\mathcal{O}\left(\left(\frac{\beta M}{b}+\frac{q\beta M}{b}\right)\cdot\frac{L^2R^2}{\varepsilon^2}\right)$
	PP-MASHA2 Alg 16	$\tilde{\mathcal{O}}\left(\frac{\beta M}{b}+\delta\sqrt{\frac{\beta M^3}{b^3}}\cdot\frac{L}{\mu}\right)$	$\mathcal{O}\left(\delta\sqrt{\frac{\beta M^3}{b^3}}\cdot\frac{LR^2}{\varepsilon}\right)$	$\mathcal{O}\left(\delta^2\beta\frac{M^3}{b^3}\cdot\frac{L^2R^2}{\varepsilon^2}\right)$

Notation: μ = constant of strong monotonicity of the operator F, L = maximum of local Lipschitz constants L_m, R = diameter (in Euclidean norm) of the optimization set, R_0 = initial distance to the solution, q = the variance parameter associated with an unbiased compressor (see (2.25)); δ = the variance parameter associated with a contractive compressor (see (2.26)); β, β = expected density (the number of times the operator compresses information); M = the number of parallel clients/nodes; n = the size of the local dataset (see (2.28)); b = the number of clients in Partial Participation (FL) setup. We have results with bidirectional compression also in stochastic and federated setups, but to simplify the bounds, we present bidirectional results only in the deterministic setup.

• **Federated learning and partial participation.** As mentioned, federated learning [105, 219] is an important and popular branch of distributed methods. Therefore, a good bonus for the algorithm is that it can be easily adapted for it. In a federated setup where the computing devices are mobile phones, tablets, and personal computers, the importance of the communication bottleneck is even higher. In such circumstances, devices can have weak and slow connections, or they can even disconnect for a while. At such moments, it is not necessary to interrupt the learning process; only available devices can be used. Therefore, we introduce two modifications: `PP-MASHA1` (Algorithm 15) and `PP-MASHA2` (Algorithm 16), that support the mode of *partial participation* of devices in the learning process. For minimization problems, a combination of quantization and

partial participation occurs in [220, 221, 212].

• **Bidirectional compression.** Most methods, especially those utilizing contractive compressors, only apply compression when transferring information from devices to the server. Meanwhile, quite often in practical situations, the transfer of information from the server to the device is also expensive [222, 223, 221]. In such cases, it is beneficial to compress the information when sending it from the server to the agents. We can highlight some works on bidirectional unbiased [221] and contractive compressors [224, 223, 225, 226] for distributed minimization problems. However, most of these methods exhibit minor shortcomings in theoretical analysis, such as being limited to a deterministic setting and the homogeneity of local functions. All our methods `MASHA1`, `MASHA2` and their modifications support bidirectional compression.

• **Experiments.** Toy experiments on bilinear problems demonstrate that methods with compression for minimization problems may not converge for saddle point problems. Additionally, we verify that `MASHA1` and `MASHA2` significantly outperform the classical ExtraGradient with added unbiased compression. Experiments on adversarial training of large-scale transformer (ALBERT) show the practical importance of compression in distributed methods for large saddle point problems.

2.5.2 Algorithms

In this section, we present new algorithms and their convergence. Section 2.5.2 is devoted to the algorithm (`MASHA1`) with unbiased compression. Section 2.5.2 is dedicated to the algorithm (`MASHA2`) with contractive compression. We also give modifications for the stochastic case – Section 2.5.3, and for the federated learning – Section 2.5.4.

`MASHA1`: Handling Unbiased Compressors

Before presenting our algorithm, let us discuss which approaches can be used to construct it. Most existing methods with compression are modifications of the classical Gradient Descent. For instance, the authors of [43] compress stochastic gradients. Therefore, it is a natural idea to use Gradient Descent-type methods for variational inequalities as well. But it is a well-known fact that GD-type methods can give bad convergence estimates (see Section B.1 from [227]) or do not converge at all (see Section 7.2 and 8.2 from [228]) even on the simplest saddle point problems. From a practical point of view, this approach can also fail (see QSGD and EF in Section 2.5.5). In the non-distributed case, this problem has long been solved and the Extragradient method [78, 80] is used instead of Gradient Descent:

$$z^{k+1/2} = z^k - \gamma F(z^k), \quad z^{k+1} = z^k - \gamma F(z^{k+1/2}). \tag{2.27}$$

This method is optimal for both variational inequalities and saddle point problems and has an estimate of convergence $\tilde{\mathcal{O}}(L/\mu)$ in the strongly monotone case. Therefore, the second idea for the compressed method is to add compression operators to the method (2.27), e.g. use $Q_k(F(z^k))$ and

$Q_{k+1/2}(F(z^{k+1/2}))$ instead of $F(z^k)$ and $F(z^{k+1/2})$. We implement this idea in Algorithm 10.

Algorithm 10 `CEG` : Compressed Extra Gradient

Parameters: stepsize $\gamma > 0$, number of iterations K
Initialization: choose $z^0 \in \mathcal{Z}$

Server sends to devices z^0
for $k = 0, 1, 2, \ldots, K-1$ **do**
  for each device m in parallel **do**
    Compute $F_m(z^k)$ & send $Q_m^{\text{dev}}(F_m(z^k))$ to server
  end for
  for server **do**
    Compute $\frac{1}{M}\sum\limits_{m=1}^{M} Q_m^{\text{dev}}(F_m(z^k))$ & send to devices
  end for
  for each device m in parallel **do**
    $z^{k+1/2} = z^k - \gamma \cdot \frac{1}{M}\sum\limits_{m=1}^{M} Q_m^{\text{dev}}(F_m(z^k))$
    Compute $F_m(z^{k+1/2})$ & send $Q_m^{\text{dev}}(F_m(z^{k+1/2}))$ to server
  end for
  for server **do**
    Compute $\frac{1}{M}\sum\limits_{m=1}^{M} Q_m^{\text{dev}}(F_m(z^{k+1/2}))$ & send to devices
  end for
  for each device m in parallel **do**
    $z^{k+1} = z^k - \gamma \cdot \frac{1}{M}\sum\limits_{m=1}^{M} Q_m^{\text{dev}}(F_m(z^{k+1/2}))$
  end for
end for

Theorem 2.5.1. Let the distributed variational inequality (2.1) + (2.7) be solved by Algorithm 10 with unbiased compressor operators (2.25) on devices with $\{q_m^{\text{dev}} = q\}$. Let Assumptions 2.0.4 and 2.0.5 (SM) are satisfied. Then the following estimates hold with $\gamma \leq \min\left[\frac{\mu}{48L^2}\cdot(1+q/M)^{-1}; \frac{1}{4\mu}\right]$:

$$\mathbb{E}\|z^K - z^*\|^2 \leq \left(1 - \frac{\mu\gamma}{2}\right)^K \cdot \|z^0 - z^*\|^2 + \frac{16q\gamma^2}{M^2}\sum_{m=1}^{M}\mathbb{E}\left[\|F_m(z^*)\|^2\right];$$

With $F_m(z^*) = 0$ we obtain the following estimates on iteration and bit complexities:

$$\mathcal{O}\left(\left[\left(1+\frac{q}{M}\right)\frac{L^2}{\mu^2}\right]\log\frac{1}{\varepsilon}\right), \quad \mathcal{O}\left(\left[\left(\frac{1}{\beta}+\frac{q}{M\beta}\right)\frac{L^2}{\mu^2}\right]\log\frac{1}{\varepsilon}\right).$$

We have an estimate $\tilde{\mathcal{O}}\left(1+q/M\right)\cdot {}^{L^2}/_{\mu^2}$), which is considerably worse in terms of ${}^{L}/_{\mu}$ than the original Extragradient method without compression. The key problem is that in the analysis one has to deal with $\|Q_k(F(z^{k+1/2})) - Q_{k+1/2}(F(z^k))\|^2$. Without compression operators, such difference is easily evaluated using Assumption 2.0.4. But when the compression operators are different (in fact the same, but have different randomness) we cannot make a good estimate for this term. The idea arises to use the same randomness in both steps of the method (2.27), namely to substitute $Q_k(F(z^k))$ and $Q_k(F(z^{k+1/2}))$. But then $z^{k+1/2}$ depends on the randomness Q_k, and

hence $Q_k(F(z^{k+1/2}))$ is biased, which further complicates the analysis. For exactly the same reasons, the various optimistic/single call modifications [79, 81, 229, 230] of the Extra Gradient method did not work for us either. We have also tested `CEG` in practice, and it turns out to be worse than the method we will present below. In the end, the use of variance reduction and negative momentum techniques [214] is key in creating our algorithm. These tricks are not in themselves relevant to distributed problems, but, in our case, they help in creating `MASHA1` and `MASHA2`.

Algorithm 11 `MASHA1`

Parameters: stepsize $\gamma > 0$, parameter τ, number of iterations K
Initialization: choose $z^0 = w^0 \in \mathcal{Z}$

Server sends to devices $z^0 = w^0$ and devices compute $F_m(w^0)$ and send to server and get $F(w^0)$
for $k = 0, 1, 2, \ldots, K-1$ **do**
  for each device m in parallel **do**
    $\bar{z}^k = \tau z^k + (1-\tau)w^k$
    $z^{k+1/2} = \bar{z}_k - \gamma F(w^k)$
    Compute $F_m(z^{k+1/2})$ & send $Q_m^{\text{dev}}(F_m(z^{k+1/2}) - F_m(w^k))$ to server
  end for
  for server **do**
    Compute $Q^{\text{serv}}\left[\frac{1}{M}\sum_{m=1}^{M} Q_m^{\text{dev}}(F_m(z^{k+1/2}) - F_m(w^k))\right]$ & send to devices
    Sends to devices one bit b_k: 1 with probability $1-\tau$, 0 with with probability τ
  end for
  for each device m in parallel **do**
    $z^{k+1} = z^{k+1/2} - \gamma Q^{\text{serv}}\left[\frac{1}{M}\sum_{m=1}^{M} Q_m^{\text{dev}}(F_m(z^{k+1/2}) - F_m(w^k))\right]$
    if $b_k = 1$ **then**
      $w^{k+1} = z^k$
      Compute $F_m(w^{k+1})$ & send it to server; and get $F(w^{k+1})$ as a response from server
    else
      $w^{k+1} = w^k$
    end if
  end for
end for

At the beginning of each `MASHA1` iteration, all devices know the value of $F(w^k)$; hence, they can calculate the value of $z^{k+1/2}$ locally without communications. Further, each device sends the compressed version of the difference $F_m(z^{k+1/2}) - F_m(w^k)$ to the server. The compression on these transfers is done by their local $\{Q_m^{\text{dev}}\}$ operators. The server aggregates the information from devices, averages it, compresses it by the Q^{serv} operator, and broadcasts it to all devices. As a result, an unbiased estimate of $F(z^{k+1/2}) - F(w^k)$ appears at each node. Also, the nodes receive one bit of information b_k. This bit is generated randomly on the server and is equal to 1 with probability $1-\tau$ (where $1-\tau$ is small). Note that b_k can be generated locally; it is sufficient to use the same random generator and set the same seed on all devices. Next, the devices locally make a final update on z^{k+1}. The final step is an update of w^{k+1}: if $b_k = 1$, then $w^{k+1} = z^k$; otherwise, $w^{k+1} = w^k$. In the case when $w^{k+1} = z^k$, we need to exchange the uncompressed values of $F_m(w^{k+1})$ to ensure that at the beginning of the next iteration, the value of $F(w^{k+1})$ is known

to all agents. We use a possibly different compressor on each device and also on the server. To distinguish between them, we denote the following notation: Q_m^{dev}, q_m^{dev}, β_m^{dev} and Q^{serv}, q^{serv}, β^{serv}.

Theorem 2.5.2. Let distributed variational inequality (2.1) + (2.7) is solved by Algorithm 11 with unbiased compressor operators (2.25): on server with q^{serv} parameter, on devices with $\{q_m^{\text{dev}}\}$. Let Assumption 2.0.4 and one case of Assumption 2.0.5 are satisfied. Then the following estimates holds

- in the strongly monotone case with $\gamma \leq \min\left[\frac{\sqrt{1-\tau}}{2C_q}; \frac{1-\tau}{2\mu}\right]$

(where $C_q = \sqrt{\frac{q^{\text{serv}}}{M^2}\sum_{m=1}^{M}(q_m^{\text{dev}}L_m^2 + (M-1)\tilde{L}^2)}$):

$$\mathbb{E}\left[\|z^K - z^*\|^2\right] \leq \left(1 - \frac{\mu\gamma}{2}\right)^K \cdot 2\|z^0 - z^*\|^2;$$

- in the monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2C_q + 4\tilde{L}}$:

$$\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1/2}\right)\right] \leq \frac{2\max_{z\in\mathcal{C}}\left[\|z^0 - z\|^2\right] + 6\|z^0 - z^*\|^2}{\gamma K};$$

- in the non-monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2C_q}$:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\|F(w^k)\|^2\right] \leq \frac{16\|z^0 - z^*\|^2}{\gamma^2 K}.$$

The bounds in Theorem 2.5.2 are related to τ. Let us find an optimal way to choose it. Note that (in average) once per $1/(1-\tau)$ iterations (when $b_k = 1$), we send uncompressed information. Based on this observation, we can find the best option for τ. Let us analyze the case of compressions only on the devices' side ($q^{\text{serv}} = 1$). For simplicity, we put $Q_m^{\text{dev}} = Q$ with $q_m^{\text{dev}} = q$ and $\beta_m^{\text{dev}} = \beta$, also $L_m = \tilde{L} = L$. Since compression is done only on devices, we assume that the server's broadcast is cheap, and we only care about devices. Then at each iteration, the device sends $\mathcal{O}\left(1/\beta + 1 - \tau\right)$ bits – each time information compressed by β times, and with probability $1 - \tau$ we send the full package. From this, we immediately get the optimal choice for τ:

Corollary 2.5.3. Let Assumption 2.0.4 and one case of Assumption 2.0.5 are satisfied. Then for some step γ and $1 - \tau = 1/\beta$ the following estimates on `MASHA1` number of iterations to achieve ε-solution holds

- in the strongly monotone case: $\mathcal{O}([\beta + \sqrt{\frac{q\beta}{M} + \beta} \cdot \frac{L}{\mu}]\log\frac{1}{\varepsilon})$;
- in the monotone case: $\mathcal{O}(\sqrt{\frac{q\beta}{M} + \beta} \cdot \frac{L\|z^0 - z^*\|^2}{\varepsilon})$;
- in the non-monotone case: $\mathcal{O}([\frac{q\beta}{M} + \beta]\frac{L^2\|z^0 - z^*\|^2}{\varepsilon^2})$.

We can see that `MASHA1` can outperform the uncompressed Extra Gradient method. Let us compare

them in the strongly monotone case. The communication complexity of the Extra Gradient method is $\tilde{\mathcal{O}}(L/\mu)$. MASHA1 has communication complexity $\tilde{\mathcal{O}}(\sqrt{q/\beta M + 1/\beta} \cdot L/\mu)$. For practical compressors (Chapter 5), $\beta \geq q$. Then, one can note that the communication complexity of MASHA1 differs from the complexity of the uncompressed method by an additional factor $(\sqrt{1/M + 1/\beta})$. It is easy to see that even for a small number of devices M and expected density β, this factor is less than 1; hence, MASHA1 outperforms the uncompressed method. We think that this factor $(\sqrt{1/M + 1/\beta})$ is theoretically unimprovable and optimal .

One can also consider the case of bidirectional compression ($q^{\text{serv}} \neq 1$). Table 2.4 (line 3) shows the result for $q^{\text{serv}} = q_m^{\text{dev}} = q$, $\beta^{\text{serv}} = \beta_m^{\text{dev}} = \beta$ and $1 - \tau = 1/\beta$.

MASHA2: Handling Contractive Compressors

The use of contractive compressions is a more complex issue. In particular, it is known that if one simply puts a contractive compressor instead of an unbiased one, the method may diverge even for quadratic problems (see Chapter 5). To address this, an error compensation technique [231, 215, 232] is used. The point of this approach is to retain untransmitted information and add it to a new package at the next iteration. This is the main difference between MASHA2 and MASHA1. MASHA2 introduces additional sequences e^k, e_m^k for the server's and devices' error. To define contractive operators on devices and on the server, we introduce the following notation: $C_m^{\text{dev}}, \delta^{\text{dev}}, \beta^{\text{dev}}$ and $C_m^{\text{serv}}, \delta^{\text{serv}}, \beta^{\text{serv}}$.

Similarly to MASHA 1, MASHA2, locally each device stores three vectors: a current point z^k, a reference point w^k, and a values $F(w^k)$ at this point. At each iteration, compressed communications are performed from devices to the server (line 6) and from the server to devices (line 10). There is also one bit b_k forwarding from the server (line 12). Additionally, communication can occur when b_k is equal to 1 (with a small probability of $1 - \tau$) – in this case, each device m updates point $w^{k+1} = z^k$, computes F_m at this point, and sends $F_m(w^{k+1})$ to the server without compression. The server calculates $F(w^{k+1})$ and sends it to devices also without compression. MASHA 2, similarly to MASHA 1, uses communications without compression, but very rarely (about once every $\frac{1}{1-\tau}$ iterations). Because, when $b_k = 0$, $w^{k+1} = w^k$ and all devices have locally the value $F(w^{k+1}) = F(w^k)$ obtained sometime in previous communications (when $b = 1$).

In the case of MASHA1, the key theoretical issue was the choice of a basic method. MASHA2 raises another problem for theoretical analysis: how to combine MASHA1 and the error feedback technique. The analysis of methods with error compensation for the minimization problem $\min_x f(x)$ is entirely tied to the existence of the function f [232, 217, 45]. In particular, the differences $(f(\cdot) - f(x^*))$ appear in the entire analysis and are key in the technical lemmas. As a result, $(f(\cdot) - f(x^*))$ is used as a convergence criterion even in the strongly convex case. However, for variational inequalities, there is no function f, only the operator F (the existence of $g(x, y)$ in saddle point problems setup does not alleviate the situation). This problem is resolved in the proof of Theorem 2.5.4 by utilizing an additional sequence $\|z^{k+1/2} - w^k\|$.

Algorithm 12 `MASHA2`

Parameters: stepsize $\gamma > 0$, parameter τ, number of iterations K
Initialization: choose $z^0 = w^0 \in \mathcal{Z}$, $e_m^0 = 0$, $e^0 = 0$

Server sends to devices $z^0 = w^0$ and devices compute $F_m(w^0)$ and send to server and get $F(w^0)$
for $k = 0, 1, 2, \ldots, K-1$ **do**
  for each device m in parallel **do**
    $\bar{z}^k = \tau z^k + (1-\tau) w^k$
    $z^{k+1/2} = \bar{z}_k - \gamma F(w^k)$
    Compute $F_m(z^{k+1/2})$ and send to server $C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k)$
    $e_m^{k+1} = e_m^k + \gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) - C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k)$
  end for
  for server **do**
    Compute $g^k = C^{\text{serv}}\left[\frac{1}{M}\sum\limits_{m=1}^{M} C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k) + e^k\right]$ & send to devices
    $e^{k+1} = e^k + \frac{1}{M}\sum\limits_{m=1}^{M} C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k) - g^k$
    Sends to devices one bit b_k: 1 with probability $1-\tau$, 0 with with probability τ
  end for
  for each device m in parallel **do**
    $z^{k+1} = z^{k+1/2} - C^{\text{serv}}\left[\frac{1}{M}\sum\limits_{m=1}^{M} C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k) + e^k\right]$
    if $b_k = 1$ **then**
      $w^{k+1} = z^k$
      Compute $F_m(w^{k+1})$ and it send to server; and get $F(w^{k+1})$
    else
      $w^{k+1} = w^k$
    end if
  end for
end for

Theorem 2.5.4. Let the distributed variational inequality (2.1) + (2.7) be solved by Algorithm 12 with $\tau \geq \frac{3}{4}$ and biased compressor operators (2.26): on the server with δ^{serv} parameter, and on devices with δ^{dev}. Let Assumption 2.0.4 and one case of Assumption 2.0.5 be satisfied. Then the following estimates hold:

- in the strongly monotone case with $\gamma \leq \min\left[\frac{1-\tau}{8\mu}; \frac{\sqrt{1-\tau}}{2L+165\delta^{\text{serv}}\delta^{\text{dev}}\tilde{L}}\right]$:

$$\mathbb{E}\left[\|\hat{z}^K - z^*\|^2\right] \leq \left(1 - \frac{\mu\gamma}{2}\right)^K \cdot 2\|z^0 - z^*\|^2;$$

- in the monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2L+165\delta^{\text{serv}}\delta^{\text{dev}}\tilde{L}}$:

$$\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1/2}\right)\right] \leq \frac{2\max_{z\in\mathcal{C}}\|z^0 - z\|^2 + 4\|z^0 - z^*\|^2}{\gamma K};$$

- in the non-monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2L+165\delta^{\text{serv}}\delta^{\text{dev}}\tilde{L}}$:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\|F(w^k)\|^2\right] \leq \frac{32\mathbb{E}\|z^0 - z^*\|^2}{\gamma^2 K}.$$

The proof can be found in Appendix A.2.5.

The same way as for unbiased compression, we can consider only devices' or bidirectional compression. In particular, in line 2 of Table 2.4 we present results for $\delta^{\text{serv}} = 1$, $\delta^{\text{dev}} = \delta$, $L_m = \tilde{L} = L$, and $1 - \tau = \beta$. In line 4 of Table 2.4 there are results for $\delta^{\text{serv}} = \delta^{\text{dev}} = \delta$, $L_m = \tilde{L} = L$, and $1 - \tau = \beta$.

2.5.3 Extension to Stochastic Case

In this subsection, we assume that the local operators on each node have either a finite-sum form:

$$F_m(z) := \frac{1}{n}\sum_{i=1}^{n} F_{m,i}(z). \tag{2.28}$$

This is exactly the same setting that we examined above in the section. However, for the sake of completeness, we introduce it .

This case corresponds to the stochastic setting, where it is expensive to calculate the full operator F_m, and it is cheaper to calculate the value $F_{m,i}$, one of the terms (batches). For this setup, we additionally assume that

Assumption 2.5.5. Each operator $F_{m,i}$ is $L_{m,i}$-Lipschitz continuous, i.e. for all $z_1, z_2 \in \mathbb{R}^d$ it holds

$$\|F_{m,i}(z_1) - F_{m,i}(z_2)\|^2 \leq L_{m,i}\|z_1 - z_2\|.$$

Let $\tilde{L}_m^2 = \frac{1}{r}\sum\limits_{i=1}^{r} L_{m,i}^2$ and $\hat{L}^2 = \frac{1}{M}\sum\limits_{m=1}^{M}\frac{1}{r}\sum\limits_{i=1}^{r} L_{m,i}^2$.

Next, we modify `MASHA1` and `MASHA2` for this setup. Modifications of the other steps (computing g^k, z^{k+1}, e_m^k, e^k etc.) in `VR-MASHA1` and `VR-MASHA2` occur according to the new g_m^k.

`VR-MASHA1`: stochastic and batch version

We provide information about `VR-MASHA1`. This is a modification of `MASHA1` for the stochastic case of a finite sum. Note that without compression, `VR-MASHA1` is an analogue of methods from [214].

The following theorem provides the convergence of `VR-MASHA1`.

Algorithm 13 `VR-MASHA1`

Parameters: stepsize $\gamma > 0$, parameter τ, number of iterations K
Initialization: choose $z^0 = w^0 \in \mathbb{R}^d$

Server sends to devices $z^0 = w^0$ and devices compute $F_m(w^0)$ and send to server and get $F(w^0)$
for $k = 0, 1, 2, \ldots, K-1$ **do**
  for each device m in parallel **do**
    $\bar{z}^k = \tau z^k + (1-\tau) w^k$
    $z^{k+1/2} = \bar{z}_k - \gamma F(w^k)$,
    Generate i_m^k from $\{1, \ldots, r\}$ independently
    Compute $F_{m,i_m^k}(z^{k+1/2})$ & send $Q_m^{\text{dev}}(F_{m,i_m^k}(z^{k+1/2}) - F_{m,i_m^k}(w^k))$ to server
  end for
  for server **do**
    Compute $Q^{\text{serv}}\left[\frac{1}{M}\sum\limits_{m=1}^{M} Q_m^{\text{dev}}(F_{m,i_m^k}(z^{k+1/2}) - F_{m,i_m^k}(w^k))\right]$ & send to devices
    Sends to devices one bit b_k: 1 with probability $1-\tau$, 0 with with probability τ
  end for
  for each device m in parallel **do**
    $z^{k+1} = z^{k+1/2} - \gamma Q^{\text{serv}}\left[\frac{1}{M}\sum\limits_{m=1}^{M} Q_m^{\text{dev}}(F_{m,i_m^k}(z^{k+1/2}) - F_{m,i_m^k}(w^k))\right]$
    if $b_k = 1$ **then**
      $w^{k+1} = z^{k+1}$
      Compute $F_m(w^{k+1})$ & send it to server; and get $F(w^{k+1})$ as a response from server
    else
      $w^{k+1} = w^k$
    end if
  end for
end for

Theorem 2.5.6. Let distributed variational inequality (2.1) + (2.7) + (2.28) is solved by Algorithm 13 with unbiased compressor operators (2.25): on server with q^{serv} parameter, on devices with $\{q_m^{\text{dev}}\}$. Let Assumptions 2.0.4, 2.5.5 and one case of Assumption 2.0.5 are satisfied. Then the following estimates holds

- in the strongly monotone case with $\gamma \leq \min\left[\frac{\sqrt{1-\tau}}{2\tilde{C}_q}; \frac{1-\tau}{2\mu}\right]$

(where $\tilde{C}_q = \sqrt{\frac{q^{\text{serv}}}{M^2} \cdot \sum_{m=1}^{M}(q_m^{\text{dev}}\tilde{L}_m^2 + (M-1)\tilde{L}^2)}$):

$$\mathbb{E}\left[\|z^K - z^*\|^2\right] \leq \left(1 - \frac{\mu\gamma}{2}\right)^K \cdot 2\|z^0 - z^*\|^2;$$

- in the monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2\tilde{C}_q + 4\tilde{L}}$:

$$\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1/2}\right)\right] \leq \frac{2\max_{z\in\mathcal{C}}\left[\|z^0 - z\|^2\right] + 6\|z^0 - z^*\|^2}{\gamma K};$$

- in the non-monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2\tilde{C}_q}$:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\|F(w^k)\|^2\right] \leq \frac{16\|z^0 - z^*\|^2}{\gamma^2 K}.$$

For `VR-MASHA1` we consider the case of only device compression. For simplicity, we set $Q_m^{\text{dev}} = Q$ with $q_m^{\text{dev}} = q$ and $\beta_m^{\text{dev}} = \beta$, along with $\tilde{L}_m = \tilde{L} = L$. Let us discuss the difference (with `MASHA1`) in choosing τ. When $b_k = 1$, we need not only to send uncompressed information to the server but also to compute the full F_m, which in the stochastic case is r times more expensive than computing one batch $F_{m,i}$. Then, at each iteration, we send $\mathcal{O}\left(\frac{1}{\beta} + 1 - \tau\right)$ bits of information and also count $\mathcal{O}\left(1 + n(1-\tau)\right)$ batches. Therefore, the optimal choice of τ depends on two factors and $1 - \tau = \frac{1}{\max\{\beta, n\}}$. In line 5 of Table 2.4 we present complexities to achieve ε-solution.

`VR-MASHA2`: stochastic and batch version

We provide information about `VR-MASHA2`. This is a modification of `MASHA2` for the stochastic case of a finite sum.

The following theorem provides the convergence of `VR-MASHA2`.

Theorem 2.5.7. Let distributed variational inequality (2.1) + (2.7) + (2.28) is solved by Algorithm 14 with $\tau \geq \frac{3}{4}$ and biased compressor operators (2.26): on server with δ^{serv} parameter, on devices with δ^{dev}. Let Assumption 2.10 and one case of Assumption 2.0.5 are satisfied. Then the following estimates holds

- in strongly monotone case with $\gamma \leq \min\left[\frac{1-\tau}{8\mu}; \frac{\sqrt{1-\tau}}{2L + 165\delta^{\text{serv}}\delta^{\text{dev}}\hat{L}}\right]$:

$$\mathbb{E}\left[\|\hat{z}^K - z^*\|^2\right] \leq \left(1 - \frac{\mu\gamma}{2}\right)^K \cdot 2\|z^0 - z^*\|^2;$$

- in monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2L + 165\delta^{\text{serv}}\delta^{\text{dev}}\hat{L}}$:

$$\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1/2}\right)\right] \leq \frac{2\max_{z\in\mathcal{C}}\|z^0 - z\|^2 + 6\|z^0 - z^*\|^2}{\gamma K};$$

- in non-monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2L + 165\delta^{\text{serv}}\delta^{\text{dev}}\hat{L}}$:

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|F(w^k)\|^2 \leq \frac{32\mathbb{E}\|z^0 - z^*\|^2}{\gamma^2 K}.$$

We consider only devices compression. For simplicity, we put $\tilde{L} = \hat{L} = L$. We use the same reasoning as in Section 2.5.3. The optimal choice is $1 - \tau = \frac{1}{\max\{\beta, n\}}$. In line 6 of Table 2.4, we provide complexities to achieve the ε-solution.

Algorithm 14 `VR-MASHA2`

Parameters: stepsize $\gamma > 0$, parameter τ, number of iterations K
Initialization: choose $z^0 = w^0 \in \mathcal{Z}$, $e_m^0 = 0$, $e^0 = 0$

Server sends to devices $z^0 = w^0$ and devices compute $F_m(w^0)$ and send to server and get $F(w^0)$
for $k = 0, 1, 2, \ldots, K-1$ **do**
  for each device m in parallel **do**
    $\bar{z}^k = \tau z^k + (1-\tau) w^k$
    $z^{k+1/2} = \bar{z}_k - \gamma F(w^k)$
    Generate i_m^k from $\{1, \ldots, r\}$ independently
    Compute $F_{m,i_m^k}(z^{k+1/2})$ and send $C_m^{\text{dev}}(\gamma F_{m,i_m^k}(z^{k+1/2}) - \gamma F_{m,i_m^k}(w^k) + e_m^k)$
    $e_m^{k+1} = e_m^k + \gamma F_{m,i_m^k}(z^{k+1/2}) - \gamma F_{m,i_m^k}(w^k) - C_m^{\text{dev}}(\gamma F_{m,i_m^k}(z^{k+1/2}) - \gamma F_{m,i_m^k}(w^k) + e_m^k)$
  end for
  for server **do**
    Compute $g^k = C^{\text{serv}}\left[\frac{1}{M}\sum_{m=1}^{M} C_m^{\text{dev}}(\gamma F_{m,i_m^k}(z^{k+1/2}) - \gamma F_{m,i_m^k}(w^k) + e_m^k) + e^k\right]$ & send to devices
    $e^{k+1} = e^k + \frac{1}{M}\sum_{m=1}^{M} C_m^{\text{dev}}(\gamma F_{m,i_m^k}(z^{k+1/2}) - \gamma F_{m,i_m^k}(w^k) + e_m^k) - g^k$
    Sends to devices one bit b_k: 1 with probability $1-\tau$, 0 with with probability τ
  end for
  for each device m in parallel **do**
    $z^{k+1} = z^{k+1/2} - C^{\text{serv}}\left[\frac{1}{M}\sum_{m=1}^{M} C_m^{\text{dev}}(\gamma F_{m,i_m^k}(z^{k+1/2}) - \gamma F_{m,i_m^k}(w^k) + e_m^k) + e^k\right]$
    if $b_k = 1$ **then**
      $w^{k+1} = z^k$
      Compute $F_m(w^{k+1})$ and it send to server; and get $F(w^{k+1})$
    else
      $w^{k+1} = w^k$
    end if
  end for
end for

2.5.4 Extension to Federated Learning Case

Here we consider a popular federated learning feature: partial participation. We model it as follows. At each iteration, only b random devices send information to the server. The remaining devices neither compute nor communicate. More formally, at each iteration, we

$$\text{generate subset } \{\xi_i^k\}_{i=1}^b \text{ of } \{1, \ldots, M\} \tag{2.29}$$

devices, which takes part in the current iteration. Next, we show how to modify `MASHA1` and `MASHA2` for partial participation.

`PP-MASHA1`: federated learning version

We provide information about `PP-MASHA1`. This is a modification of `MASHA1` for the federated learning case.

Algorithm 15 `PP-MASHA1`

Parameters: stepsize $\gamma > 0$, parameters τ and b, number of iterations K.
Initialization: choose $z^0 = w^0 \in \mathcal{Z}$

Server sends to devices $z^0 = w^0$ and devices compute $F_m(w^0)$ and send to server and get $F(w^0)$
for $k = 0, 1, 2, \ldots, K-1$ **do**
  Generate subset $\{\xi_i^k\}_{i=1}^b$ of $\{1, \ldots, M\}$ independently
  for each device m from $\{\xi_i^k\}_{i=1}^b$ in parallel **do**
    $\bar{z}^k = \tau z^k + (1-\tau) w^k$
    $z^{k+1/2} = \bar{z}_k - \gamma F(w^k)$,
    Compute $F_m(z^{k+1/2})$ & send $Q_m^{\text{dev}}(F_m(z^{k+1/2}) - F_m(w^k))$ to server
  end for
  for server **do**
    Compute $Q^{\text{serv}}\left[\frac{1}{b}\sum_{i=1}^b Q_{\xi_i^k}^{\text{dev}}(F_{\xi_i^k}(z^{k+1/2}) - F_{\xi_i^k}(w^k))\right]$ & send to devices
    Sends to devices one bit b_k: 1 with probability $1-\tau$, 0 with with probability τ
  end for
  for each device m in parallel **do**
    $z^{k+1} = z^{k+1/2} - \gamma Q^{\text{serv}}\left[\frac{1}{b}\sum_{i=1}^b Q_{\xi_i^k}^{\text{dev}}(F_{\xi_i^k}(z^{k+1/2}) - F_{\xi_i^k}(w^k))\right]$
    if $b_k = 1$ **then**
      $w^{k+1} = z^{k+1}$
      Compute $F_m(w^{k+1})$ & send it to server; and get $F(w^{k+1})$ as a response from server
    else
      $w^{k+1} = w^k$
    end if
  end for
end for

The following theorem provides the convergence of `PP-MASHA1`.

Theorem 2.5.8. Let distributed variational inequality (2.1) + (2.7) + (2.29) is solved by Algorithm 15 with unbiased compressor operators (2.25): on server with q^{serv} parameter, on devices with $\{q_m^{\text{dev}}\}$. Let Assumption 2.0.4 and one case of Assumption 2.0.5 are satisfied. Then the following estimates holds

- in strongly monotone case with $\gamma \leq \min\left[\frac{\sqrt{1-\tau}}{2C_q^b}; \frac{1-\tau}{2\mu}\right]$ (where $C_q^b = \sqrt{\frac{q^{\text{serv}}}{bM} \cdot \sum_{m=1}^M (q_m^{\text{dev}} \tilde{L}_m^2 + (b-1)\tilde{L}^2)}$):

$$\mathbb{E}\left(\|z^K - z^*\|^2 + \|w^K - z^*\|^2\right) \leq \left(1 - \frac{\mu\gamma}{2}\right)^K \cdot 2\|z^0 - z^*\|^2;$$

- in monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2C_q^b + 4\tilde{L}}$:

$$\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1/2}\right)\right] \leq \frac{2\max_{z \in \mathcal{C}}\left[\|z^0 - z\|^2\right] + 6\|z^0 - z^*\|^2}{\gamma K};$$

- in non-monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2C_q^b}$:

$$\mathbb{E}\left(\frac{1}{K}\sum_{k=0}^{K-1}\|F(w^k)\|^2\right) \leq \frac{16\|z^0 - z^*\|^2}{\gamma^2 K}.$$

For `PP-MASHA1` we consider the case of only devices compression. For simplicity, we set $Q_m^{\text{dev}} = Q$ with $q_m^{\text{dev}} = q$ and $\beta_m^{\text{dev}} = \beta$, also $\tilde{L}_m = \tilde{L} = L$. Let us discuss the difference (with `MASHA1`) in choosing τ. When $b_k = 1$, all devices send uncompressed information to the server, but only b devices send compressed information (line 9 `PP-MASHA1`). Then, at each iteration, we send $\mathcal{O}\left(\frac{b}{\beta} + M(1-\tau)\right)$ bits of information. Therefore, the optimal choice of τ is $1 - \tau = \frac{b}{\beta M}$. In line 7 of Table 2.4 we present complexities to achieve ε-solution.

`PP-MASHA2`: federated learning version

We provide information about `PP-MASHA2` from Section 2.5.4. This is a modification of `MASHA2` for the federated learning case.

The following theorem presents the convergence of `PP-MASHA2`.

Theorem 2.5.9. Let distributed variational inequality (2.1) + (2.7) is solved by Algorithm 16 with $\tau \geq \frac{3}{4}$ and biased compressor operators (2.26): on server with δ^{serv} parameter, on devices with δ^{dev}. Let Assumption 2.0.4 and one case of Assumption 2.0.5 are satisfied. Then the following estimates holds

- in strongly monotone case with $\gamma \leq \min\left[\frac{1-\tau}{8\mu}; \frac{\sqrt{1-\tau}}{(30\delta^{\text{serv}} + 10\delta^{\text{dev}}\frac{M}{b} + 165\delta^{\text{dev}}\delta^{\text{serv}}\sqrt{\frac{M}{b}})\tilde{L}}\right]$:

$$\mathbb{E}\left(\|\hat{z}^K - z^*\|^2 + \|w^K - z^*\|^2\right) \leq \left(1 - \frac{\mu\gamma}{2}\right)^K \cdot 2\|z^0 - z^*\|^2;$$

- in monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{(30\delta^{\text{serv}} + 10\delta^{\text{dev}}\frac{M}{b} + 165\delta^{\text{dev}}\delta^{\text{serv}}\sqrt{\frac{M}{b}})\tilde{L}}$:

$$\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1/2}\right)\right] \leq \frac{2\max_{z\in\mathcal{C}}\|z^0 - z\|^2 + 6\|z^0 - z^*\|^2}{\gamma K};$$

- in non-monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{(30\delta^{\text{serv}} + 10\delta^{\text{dev}}\frac{M}{b} + 165\delta^{\text{dev}}\delta^{\text{serv}}\sqrt{\frac{M}{b}})\tilde{L}}$:

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|F(w^k)\|^2 \leq \frac{32\mathbb{E}\|z^0 - z^*\|^2}{\gamma^2 K}.$$

We consider only devices compression. For simplicity, we put $\tilde{L} = \hat{L} = L$. We use the same reasoning as in Section 2.5.4. The optimal choice is $1 - \tau = \frac{b}{\beta M}$. In line 8 of Table 2.4 we present the complexities to achieve the ε-solution.

Algorithm 16 PP-MASHA2

Parameters: stepsize $\gamma > 0$, parameters τ and b, number of iterations K
Initialization: choose $z^0 = w^0 \in \mathcal{Z}$, $e_m^0 = 0$, $e^0 = 0$

Server sends to devices $z^0 = w^0$ and devices compute $F_m(w^0)$ and send to server and get $F(w^0)$
for $k = 0, 1, 2, \ldots, K-1$ **do**
  Generate subset $\{\xi_i^k\}_{i=1}^b$ of $\{1, \ldots, M\}$ independently
  for each device m from $\{\xi_i^k\}_{i=1}^b$ in parallel **do**
    $\bar{z}^k = \tau z^k + (1-\tau)w^k$
    $z^{k+1/2} = \bar{z}_k - \gamma F(w^k)$
    Compute $F_m(z^{k+1/2})$ and send to server $C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k)$
    $e_m^{k+1} = e_m^k + \gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) - C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k)$
  end for
  for devices not from $\{\xi_i^k\}_{i=1}^b$ in parallel **do**
    $e_m^{k+1} = e_m^k$
  end for
  for server **do**
    Compute $g^k = C^{\text{serv}}\left[\frac{1}{b}\sum_{i=1}^b C_{\xi_i^k}^{\text{dev}}(\gamma F_{\xi_i^k}(z^{k+1/2}) - \gamma F_{\xi_i^k}(w^k) + e_{\xi_i^k}^k) + e^k\right]$ & send to devices
    $e^{k+1} = e^k + \frac{1}{b}\sum_{i=1}^b C_{\xi_i^k}^{\text{dev}}(\gamma F_{\xi_i^k}(z^{k+1/2}) - \gamma F_{\xi_i^k}(w^k) + e_{\xi_i^k}^k) - g^k$.
    Sends to devices one bit b_k: 1 with probability $1-\tau$, 0 with with probability τ
  end for
  for each device m in parallel **do**
    $z^{k+1} = z^{k+1/2} - C^{\text{serv}}\left[\frac{1}{b}\sum_{i=1}^b C_{\xi_i^k}^{\text{dev}}(\gamma F_{\xi_i^k}(z^{k+1/2}) - \gamma F_{\xi_i^k}(w^k) + e_{\xi_i^k}^k) + e^k\right]$
    if $b_k = 1$ **then**
      $w^{k+1} = z^k$
      Compute $F_m(w^{k+1})$ and it send to server; and get $F(w^{k+1})$
    else
      $w^{k+1} = w^k$
    end if
  end for
end for

2.5.5 Experiments

Bilinear Saddle Point Problem

We start our experiments with a distributed bilinear problem, i.e. the problem (2.3) with

$$f_m(x, y) := x^T A_m y + a_m^T x + b_m^T y + \frac{\lambda}{2}\|x\|^2 - \frac{\lambda}{2}\|y\|^2, \tag{2.30}$$

where $A_m \in \mathbb{R}^{d\times d}$, $a_m, b_m \in \mathbb{R}^d$. This problem is λ-strongly convex–strongly concave and, moreover, all functions g_m are $\|A_m\|_2$-smooth. Therefore, such a distributed problem is well suited for the primary comparison of our methods. We take $d = 100$ and generate positive definite matrices A_m and vectors a_m, b_m randomly. λ is chosen as $\max_m \|A_m\|_2/10^5$.

The purpose of the experiment is to understand whether the MASHA1 and MASHA2 methods are superior to those in the literature. As a comparison, we take QGD [43] with Random 30%, classical Error Feedback [231] with Top 30% compression, as well as CEG, each step of which employs Random 30%. In MASHA1 (Algorithm 11) we also used Random 30%, in MASHA2 (Algorithm 12) – Top 30%. See Figure 2.9. The stepsizes of all methods are chosen for the best convergence.

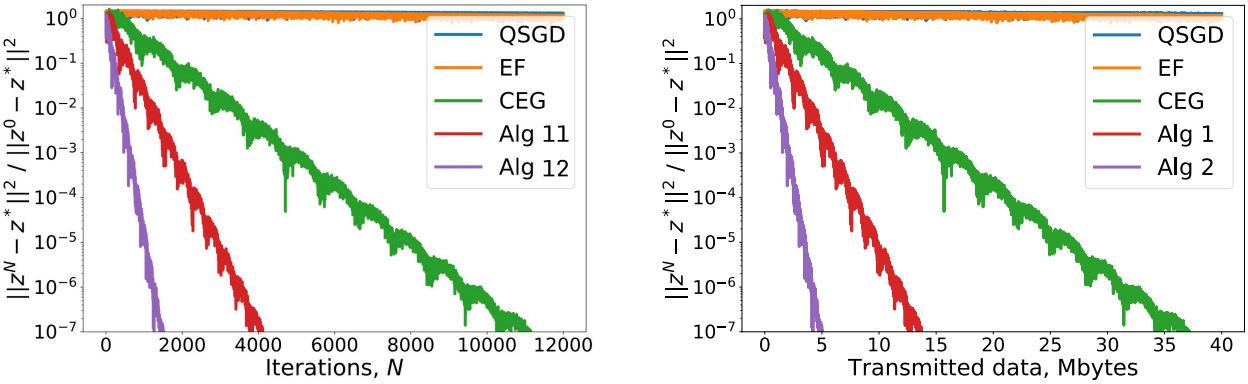


Figure 2.9: Comparison MASHA1 (Algorithm 11) and MASHA2 (Algorithm 12) with Error Feedback, QGD and Compressed Extra Gradient (CEG) in iterations and in Mbytes for (2.30).

We see on Figure 2.9 that methods based on Gradient Descent (QSGD and EF) converge slowly. This confirms the necessity of using methods specifically designed for saddle point problems, such as the extragradient method, rather than classical optimization methods. The significantly slower convergence of CEG highlights the efficiency of our approach, which compresses the differences $F_m(z^{k+1/2}) - F(w^k)$. MASHA2 outperforms MASHA1. This demonstrates that, in practice, a contractive compressor can achieve better performance than an unbiased one with identical parameters.

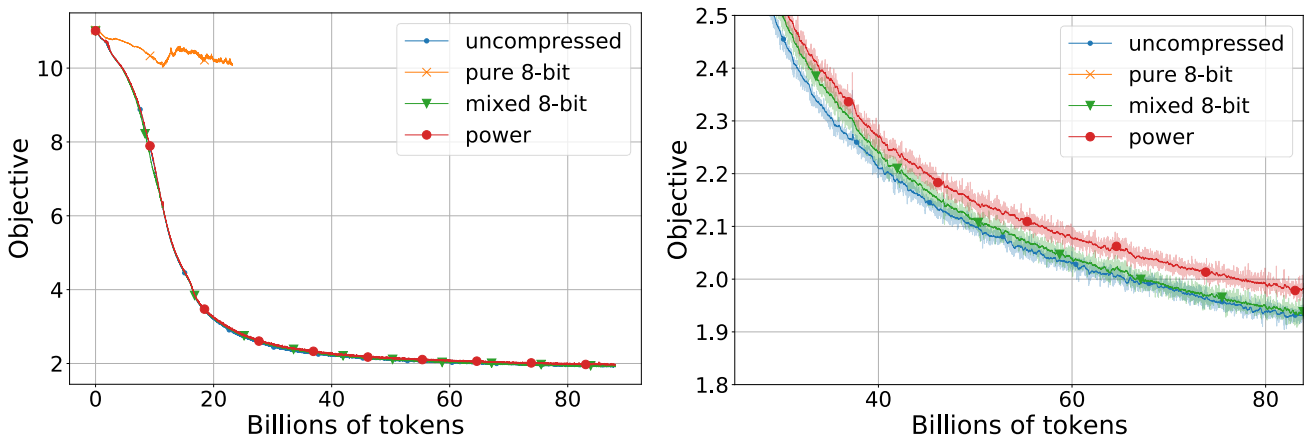
Adversarial Training of Transformers

We now evaluate how compression performs for variational inequalities (and for saddle point problems, as a special case) in a more practically motivated scenario. Indeed, saddle point problems (special case of variational inequalities) have sample applications in machine learning, including adversarial training. And our goal is to show that compression provides important improvements for such large-scale problems as well. We train a transformer-based masked language model [233, 234, 235] using a fleet of 16 low-cost preemptible workers with T4 GPU and low-bandwidth interconnect. For this task, we use the compute-efficient adversarial training regimen proposed for transformers by [236, 237]. The formal definition of this problem is given in (2.6). To make our setup more realistic, we train ALBERT-large with layer sharing [238], which was recently shown to be much more communication-efficient during training [239, 240]. We train our model on a combination of Bookcorpus and Wikipedia datasets with the same optimizer (LAMB) and parameters as in the original paper [238], use the adversarial training configuration of [236], and follow system design considerations for preemptible instances [239]. In LAMB optimizer we change

the original positive momentum to negative momentum, as in `MASHA`. This means that we do not exactly use `MASHA` in these experiments, but a combination of `MASHA` and LAMB. In fact this approach is typical, e.g., in papers [98, 81, 99, 100, 101], the theoretical methods are combined with Adam.

In terms of communication, we consider four different setups for gradient compression: the "baseline" strategy with uncompressed gradients, full 8-bit quantization [241, 242], mixed 8-bit quantization, and Power compression [42] with rank r=8. For mixed 8-bit quantization and Power, we only apply compression to gradient tensors with more than 2^{16} elements, while sending smaller ones uncompressed. These small tensors represent layer biases and LayerNorm scales [243] that collectively amount to $\leq 1\%$ of the total gradient but can be more difficult to compress than regular weight tensors. Finally, since Power is a biased compression algorithm, we use error feedback [215, 45] with a modified formulation proposed by [42]. For all experimental setups, we report learning curves in terms of the model training objective, similarly to [244, 239]. To quantify the differences in training loss better, we also evaluate the downstream performance for each model on several popular tasks from [245] after each model was trained on approximately 80 billion tokens. Finally, we measure the communication efficiency of each proposed strategy by assessing the average wall time per communication round when all 16 workers are active.

The learning curves in Figure 2.10 (upper) follow a predictable pattern, with more extreme compression techniques demonstrating slower per-iteration convergence. One curious exception to that is full 8-bit quantization, which was unable to achieve competitive training loss. The remaining three setups converge to similar loss values below 2. Both the baseline and mixed 8-bit compression show similar values in terms of downstream performance, with Power compression showing mild degradation. But in terms of information transfer time, methods using compression (especially Power) are significantly superior to the method without compression. This makes it possible to use such techniques to decrease the training time without sacrificing quality.



Setup	Avg time	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST2	STS-B	WNLI
Baseline	8.79 ± 0.03	45.2	81.1	83.0	88.3	**89.0**	**67.8**	85.5	**89.4**	**18.3**
Full 8-bit	4.42 ± 0.07	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Mixed 8-bit	4.61 ± 0.08	**48.8**	**81.3**	**88.7**	88.1	85.2	64.3	88.3	87.5	16.9
Power	**1.57 ± 0.05**	43.9	80.5	85.6	**88.6**	86.0	47.2	**88.5**	88.5	16.9

Figure 2.10: (upper left) ALBERT training objective convergence rate with different compression algorithms; (upper right) ALBERT training objective convergence rate with different compression algorithms (zoomed); (lower) Average wall time per communication round with standard deviation over 5 repetitions and downstream evaluation scores on GLUE benchmark tasks after at 80 billion training tokens ($\approx 10^4$ optimizer steps).

2.6 Combination

This section finalizes all the results obtained earlier in this chapter. We have investigated three techniques: local steps, similarity, and compression separately. We now combine them within a single technique.

2.6.1 Related Works and Contributions

A survey of all trends in the literature regarding local updates, similarity, and compression has been presented in previous sections. We summarize all results in Table 2.5. Several works attempt to combine these techniques for minimization.

• **Local methods with compression.** The combination of local updates and compression for minimization problems has been presented in [246, 247] (note that the theoretical rates are not optimal and used restrictive assumptions). Using the same local approach as in [39], in papers [248, 249], authors combine it with compression technique and get good convergence rates. For variational inequalities and saddle point problems, there are no methods with both local steps and compression.

• **Compression and similarity.** Recently, a new class of compressions called permutation compressors was introduced for the distributed minimization problem under the data similarity assumption [250]. However, due to not employing the local steps technique, the gain from combining these two approaches is slight.

Even for minimization problems, no work has combined all three techniques.

The contributions of this section can be summarized as follows.

• **Combination of two.** Our first algorithm `Optimistic MASHA` (Algorithm 17) combines two techniques for effective communication: compression and data similarity.

• **Combination of three.** We present a new method `Three Pillars Algorithm` (Algorithm 18) that combines three approaches: compression, similarity, and local steps for the efficient solution of distributed variational inequalities and saddle point problems.

• **The best communication complexity.** We analyze the convergence of the new distributed algorithms under a similarity assumption. As can be seen in Table 2.5, our algorithms have better communication complexity than all available competitors. Since $\delta \leq L$, it is easy to see that `Three Pillars Algorithm` always outperforms all methods except `Optimistic MASHA`. One can note that in the worst case with $\delta \sim L$ it reproduces the results of `Optimistic MASHA`. But if the data are uniformly distributed, then $\delta \sim \frac{L}{\sqrt{b}}$ or even $\frac{L}{b}$, where b is the size of the local sample. Assume that $N = bn$ is the size of the whole data, then `Three Pillars Algorithm` outperforms `Optimistic MASHA` when $L > \delta\sqrt{n}$, i.e. when $N > n^2$. Such a ratio is typical as classic modern datasets contain hundreds of thousands [251, 183] or even millions [252] of samples, and the number of devices rarely exceeds 100.

Table 2.5: Summary of the results on the number of transmitted information in different approaches to communication bottleneck for distributed variational inequalities and saddle point problems.

Method	Approach	Communication complexity
Extragradient [78, 85]		$\mathcal{O}\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
Local SGDA [157]	local steps	$\mathcal{O}\left(\frac{L^{2/5}M^{2/5}}{\mu^{2/5}\varepsilon^{1/5}}+\frac{L\zeta}{\mu^{3/2}\sqrt{\varepsilon}}\right)$
FedAvg-S [158]	local steps	$\mathcal{O}\left(\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}+\frac{L\zeta}{\mu^2\sqrt{\varepsilon}}\right)$
SCAFFOLD-S [158]	local steps	$\mathcal{O}\left(\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
SCAFFOLD-Catalyst-S [158]	local steps	$\mathcal{O}\left(\frac{L}{\mu}\log^2\frac{1}{\varepsilon}\right)$
`ESTVGM` (Section 2.3)	local steps	$\mathcal{O}\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}+\frac{L\zeta}{\mu^2\sqrt{\varepsilon}}\right)$
`SMMDS` (Section 2.4)	similarity local steps	$\mathcal{O}\left(\left[1+\frac{\delta}{\mu}\right]\log\frac{1}{\varepsilon}\right)$
`MASHA` (Section 2.5)	compression	$\mathcal{O}\left(\frac{L}{\sqrt{M}\mu}\log\frac{1}{\varepsilon}\right)$
Algorithm 17 (this section)	similarity compression	$\mathcal{O}\left(\left[\frac{L}{M\mu}+\frac{\delta}{\sqrt{M}\mu}\right]\log\frac{1}{\varepsilon}\right)$
Algorithm 18 (this section)	similarity local steps compression	$\mathcal{O}\left(\left[1+\frac{\delta}{\sqrt{M}\mu}\right]\log\frac{1}{\varepsilon}\right)$
Algorithm 19 (this section)	similarity local steps partial participation	$\mathcal{O}\left(\left[1+\frac{\delta}{\sqrt{M}\mu}\right]\log\frac{1}{\varepsilon}\right)$
Lower bound (Section 2.1)		$\Omega\left(\left[1+\frac{\delta}{\mu}\right]\log\frac{1}{\varepsilon}\right)$ [(1)]
Lower bound (this section)	similarity local steps compression	$\Omega\left(\left[1+\frac{\delta}{\sqrt{M}\mu}\right]\log\frac{1}{\varepsilon}\right)$
Lower bound (this section)	similarity local steps partial participation	$\Omega\left(\left[1+\frac{\delta}{\sqrt{M}\mu}\right]\log\frac{1}{\varepsilon}\right)$

Notation: μ = constant of strong monotonicity of the operator F (strong convexity – strong concavity of the corresponding saddle point problem), L = Lipschitz constant of F, δ = similarity parameter (typically, $\delta \ll L$), ζ = heterogeneity parameter, M = number of devices, ε = precision of the solution.

[(1)] lower bound is deterministic and does not take into account the possibility of compression and partial participation.

Further points of the contributions refer only to `Three Pillars Algorithm` as a more advanced variant.

• **Main calculations on server.** Many distributed approaches, such as one of the most popular from [253], utilize the power of all computing devices evenly. However, in terms of FL, we prefer to have the main computational workload on the server, rather than on the local devices of the users [16, 23]. Our `Three Pillars Algorithm` includes this feature.

• **Extension with partial participation.** We present a modified version of our first method – Algorithm 19. Instead of compression, one device is selected to send an uncompressed message. This reduction in local complexity on devices is achieved compared to Algorithm 18, while the overall communication complexity remains unchanged.

• **Lower bounds.** To demonstrate the optimality of our proposed approaches, we establish lower bounds for the communication complexities. These estimates enhance those from [16], which do not consider the potential for compression and partial participation. Our results are shown to be unsurpassable in both scenarios involving compression and partial participation (see Table 2.5).

• **Extension with stochastic local computations.** Motivated by applications where computation is expensive, we present a modification of Algorithm 18 that emphasizes the use of stochastic operators in local computations, termed Algorithm 20.

• **Non-monotone analysis.** Although we focus primarily on strongly monotone variational inequalities, many machine learning problems are non-monotone, and hence we extend the convergence analysis to this case. Our algorithm outperforms competitors in terms of communication complexity for both cases and is the first to gain acceleration in the non-monotone case.

• **Experimental confirmation.** Experimental results on adversarial training of regression models confirm the theoretical conclusion that both of our algorithms outperform their competitors.

2.6.2 Assumptions

In this section, we also put $g \equiv 0$ in the problem (2.1). Since we work with similarity, we introduce assumptions familiar to us from Section 2.4.

Assumption 2.6.1 (δ-relatedness)**.** Each operator F_m is δ-related. It means that each operator $F_m - F$ is δ-Lipschitz continuous, i.e. for all $u, v \in \mathbb{R}^d$ we have

$$\|F_m(u) - F(u) - F_m(v) + F(v)\| \leq \delta\|u - v\|.$$

Assumption 2.6.2 (δ-relatedness in mean)**.** The operators $\{F_m\}$ are δ-related in mean on $\mathbb{R}^d$. This means that for any j operators $\{F_m - F_l\}$ are δ-Lipschitz continuous in mean on $\mathbb{R}^d$.

Specifically, for all $u, v \in \mathbb{R}^d$, we have

$$\frac{1}{M}\sum_{m=1}^{M}\|F_m(u) - F_l(u) - F_m(v) + F_l(v)\|^2 \leq \delta^2\|u-v\|^2.$$

2.6.3 Algorithms

`Optimistic MASHA`

Our new algorithm `Optimistic MASHA`, as well as `MASHA` (Algorithm 11), is based on the concepts of negative momentum and variance reduction technique [214].

Algorithm 17 `Optimistic MASHA`

Parameters: stepsize $\gamma > 0$, parameter τ, number of iterations K
Initialization: choose $z^0 = w^0 \in \mathcal{Z}$
Server sends to devices $z^0 = w^0 = w^{-1}$ and devices compute $F_m(z^0)$ and send to server and get $F(z^0)$
for $k = 0, 1, 2, \ldots, K-1$ **do**
  for each device m in parallel **do**
    Compute $F_m(z^k)$
    $\delta_m^k = F_m(z^k) - F_m(w^{k-1}) + \alpha[F_m(z^k) - F_m(z^{k-1})]$
    Send $Q_m\left(\delta_m^k\right)$ to server
  end for
  for server **do**
    Compute $\frac{1}{M}\sum\limits_{m=1}^{M} Q_m(\delta_m^k)$ and send to devices
    Sends to devices b_k: 1 with probability γ, 0 with. probability $1-\gamma$
  end for
  for each device m in parallel **do**
    $\Delta^k = \frac{1}{M}\sum\limits_{m=1}^{M} Q_m^{\text{dev}}(\delta_m^k) + F(w^{k-1})$
    $z^{k+1} = \text{prox}_{\eta g}\left(z^k + \gamma(w^k - z^k) - \eta\Delta^k\right)$
    if $b_k = 1$ **then**
      $w^{k+1} = z^k$
      Compute $F_m(w^{k+1})$ and send it to server
      Get $F(w^{k+1})$ as a response from server
    else
      $w^{k+1} = w^k$
    end if
  end for
end for

For the sake of completeness, let us discuss the idea afresh. At the beginning of each iteration of `Optimistic MASHA`, each device sends the compressed version of δ_m^k to the server, and the server performs a reverse broadcast. It is also possible to compress the messages coming from the server to the devices; however, in practical cases, there is often no need for compression in this instance since the transfer process from the server takes less time than the sending from the devices to the server [43, 60]. Additionally, the workers receive a bit of information b_k. This bit is generated

randomly on the server and is equal to 1 with probability γ (where γ is small). Note that b_k can be generated locally; it is sufficient to use the same random generator and set the same seed on all devices. Then, all devices make a final update on z^{k+1} using Δ^k. One can notice that to compute Δ^k we need to know $F(w^{k-1})$ (the full operator over all nodes). Then, at first glance, it seems that we need to always send the uncompressed operators. But that is not the case. It is sufficient to look at the w^{k+1} update. We put $w^{k+1} = z^{k+1}$ if $b_k = 1$ or save it from the previous iteration $w^{k+1} = w^k$ if $b_k = 0$. In the case where $w^{k+1} = z^{k+1}$, we need to exchange the full values of $F_m(w^{k+1})$ to make the value $F(w^{k+1})$ known to all nodes at the next iteration; however, we do this rarely, with small probability γ.

Unlike `MASHA`, we do not use arbitrary Q_m compressors on the devices, but a specific set $\{Q_m\}$, the so-called Permutation compressors, introduced in [250].

Definition 2.6.3 (Permutation compressors [250])**.** There are two cases:
• **for** $d \geq M$**.** Assume that $d \geq M$ and $d = qM$, where $q \geq 1$ is an integer. Let $\pi = (\pi_1, \ldots, \pi_d)$ be a random permutation of $\{1, \ldots, d\}$. Then for all $u \in \mathbb{R}^d$ and each $m \in \{1, 2, \ldots, M\}$ we define

$$Q_m(u) := M \cdot \sum_{i=q(m-1)+1}^{qm} u_{\pi_i} e_{\pi_i}.$$

• **for** $d \leq M$**.** Assume that $M \geq d$, $M > 1$ and $M = qd$, where $q \geq 1$ is an integer. Define the multiset $S := \{1, \ldots, 1, 2, \ldots, 2, \ldots, d, \ldots, d\}$, where each number occurs precisely q times. Let $\pi = (\pi_1, \ldots, \pi_M)$ be a random permutation of S. Then for all $u \in \mathbb{R}^d$ and each $m \in \{1, 2, \ldots, M\}$ we define

$$Q_m(u) := d u_{\pi_m} e_{\pi_m}.$$

The essence of such compressors is that their behavior is related to each other. For example, in the case when $d \geq M$ and $d = Mq$, each device transmits only q components of the full gradient and, importantly, these components are unique to that device. To establish such a connection between compressors, one can set the same random seeds on the devices to generate permutations. The use of the Permutation compressors allows us to simultaneously benefit from both the data similarity and the compression of the transmitted information. Briefly, the idea can be described as follows. Since we have δ-related (δ-similar) operators $\{F_m\}$, in a rough approximation we can assume that $F_m \approx F$ and then $\delta_m^k \approx \delta^k = F(z^k) - F(w^{k-1}) + \alpha[F(z^k) - F(z^{k-1})]$. But when we compress $\{\delta_m^k\}$ with the Permutation compressors, we end up with $\frac{1}{M}\sum_{m=1}^M Q_m^{\text{dev}}(\delta_m^k)$ that is close to uncompressed δ^k. In the meantime, we transmit M times less information.

The following statements provide a formal convergence of Algorithm 17. To begin, we present the lemma regarding the compressors from [250].

Lemma 2.6.4 (see [250])**.** The permutation compressors from Definition 2.6.3 are unbiased and

satisfy

$$\mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M}Q_m(a_m)-\frac{1}{M}\sum_{m=1}^{M}a_m\right\|^2\leq\frac{1}{M}\sum_{m=1}^{M}\left\|a_m-\frac{1}{M}\sum_{i=1}^{M}a_i\right\|^2$$

for all $a_1,\ldots,a_M\in\mathbb{R}^d$.

Next, we present the main theorem.

Theorem 2.6.5. Consider the problem (2.1)+(2.7) under Assumptions 2.0.4, 2.0.5 and 2.6.1. Let $\{z^k\}$ be the sequence generated by Algorithm 17 with the compressors from Definition 2.6.3 and parameters

$$0<\gamma\leq\frac{1}{8},\quad \alpha=\frac{1}{2},\quad \eta=\min\left\{\frac{\sqrt{\alpha\gamma}}{2\delta},\frac{1}{8(L+\delta)}\right\}.$$

Then, given $\varepsilon>0$, the number of iterations for $\|z^k-z^*\|^2\leq\varepsilon$ is

$$O\left(\left[\frac{1}{\gamma}+\frac{L}{\mu}+\frac{\delta}{\sqrt{\gamma}\mu}\right]\log\frac{1}{\varepsilon}\right).$$

The resulting convergence estimate depends on the parameter γ. Let us find a way to choose it. On average, once per $\frac{1}{\gamma}$ iterations (when $b_k=1$), we send uncompressed information. Hence, we can identify the best option for γ. At each iteration, the device sends $\mathcal{O}\left(\frac{1}{M}+\gamma\right)$ bits – each time information compressed by $\frac{1}{M}$ times, and with probability γ, we send the full package. We get the optimal choice of γ:

Corollary 2.6.6. Under the conditions of Theorem 2.6.5, and with $\gamma=\frac{1}{M}$, `Optimistic MASHA` with the Permutation compressors has the following estimate on the total number of transmitted information to find ε-solution

$$O\left(\left[\frac{L}{M\mu}+\frac{\delta}{\sqrt{M}\mu}\right]\log\frac{1}{\varepsilon}\right).$$

Discussion of the results in terms of compression. As noted earlier, under conditions of uniformly distributed data, the parameter $\delta=\tilde{\mathcal{O}}\left(\frac{L}{\sqrt{b}}\right)$, where b is the number of local data points on each of the devices. Note that a typical situation arises when $b\geq M$. Then, the estimate from Corollary 2.6.6 can be rewritten as

$$O\left(\frac{L}{M\mu}\log\frac{1}{\varepsilon}\right).$$

State-of-the-art methods for solving variational inequalities [85], which are also optimal algorithms in terms of the number of communications (but not the number of transmitted information), provide the following estimate on the number of transmitted information.

$$O\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right).$$

MASHA can guarantee the following bound for the amount of transferred information:

$$O\left(\frac{L}{\sqrt{M}\mu}\log\frac{1}{\varepsilon}\right).$$

This shows that our result is M times better than the uncompressed methods, and better than MASHA (which does not use δ-relatedness) by $\sqrt{M}$ times.

Discussion of the results in terms of data similarity. The algorithm from Section 2.4 using similarity for variational inequalities (specifically, for saddle point problems) provides the following estimate for the number of information that must be forwarded:

$$O\left(\frac{\delta}{\mu}\log\frac{1}{\varepsilon}\right).$$

If $M \geq b$, the estimate from Corollary 2.6.6 transforms as follows

$$O\left(\left[\frac{L}{\sqrt{M}\sqrt{b}\mu}+\frac{\delta}{\sqrt{M}\mu}\right]\log\frac{1}{\varepsilon}\right) = O\left(\frac{\delta}{\sqrt{M}\mu}\log\frac{1}{\varepsilon}\right).$$

This result is $\sqrt{M}$ times better than that from Section 2.4.

`Three Pillars Algorithm`

We consider a centralized communication architecture where the main device is the server (node with number 1, which stores data associated with F_1). All other devices can send information to the server and receive responses from it.

This algorithm combines the concepts from the algorithms in Sections 2.4 and 2.5. For completeness, we repeat all parts and ideas again. We split the idea of our method `Three Pillars Algorithm` (Algorithm 18) into five parts.

• **Local problem.** The simplest local method involves devices making numerous independent gradient steps, utilizing only their local operators/gradients and occasionally averaging the current value of the optimization variables [107, 253]. With this approach, between communications, the devices minimize their local functions and can move quite far away if the number of local steps is large [176]. Therefore, the estimates for `Local SGDA`, `FedAvg-S`, `ESTVGM` from Table 2.5 depend on the data heterogeneity parameter. This problem can be solved by modifying original local operators/functions via regularization. For example, we can ask to locally minimize not the local $f_m(x)$, but $f_m(x) + \lambda\|x - x_s\|^2$, where x_s is the last point synchronized with all devices [40]. Statistical preconditioning, the popular trick for most methods under similarity conditions, is also a regularization of the local problem [108, 208]. It can also be considered as Mirror Descent with special Bregman divergence (see Section 1.1. of [41] for more details). The authors propose to rewrite the original distributed problem as a composite problem of two terms: $\frac{1}{M}\sum_{m=1}^{M} f_m(x) = r(x) + q(x) = [f_1(x)] + [\frac{1}{M}\sum_{m=1}^{M} f_m(x) - f_1(x)]$. The proximal method works well for composite problems with proximal-friendly functions (e.g, simple ℓ_2 regularizer)

Algorithm 18 `Three Pillars Algorithm`

Parameters: stepsizes γ and η, momentum τ, probability p, number of local steps H
Initialization: choose $z^0 = w^0 \in \mathbb{R}^d$

1: **for** $k = 0, 1, \ldots, K-1$ **do**
2: Server takes $u_0^k = z^k$;
3: **for** $t = 0, 1, \ldots, H-1$ **do**
4: Server computes
$$u_{t+1/2}^k = u_t^k - \eta(F_1(u_t^k) - F_1(w^k) + F(w^k) + \tfrac{1}{\gamma}(u_t^k - z^k - \tau(w^k - z^k)));$$
5: Server updates
$$u_{t+1}^k = u_t^k - \eta(F_1(u_{t+1/2}^k) - F_1(w^k) + F(w^k) + \tfrac{1}{\gamma}(u_{t+1/2}^k - z^k - \tau(w^k - z^k)));$$
6: **end for**
7: Server broadcasts u_H^k and $F_1(u_H^k)$ to devices;
8: Devices in parallel compute $Q_m(F_m(w^k) - F_1(w^k) - F_m(u_H^k) + F_1(u_H^k))$ and send to server;
9: Server updates $z^{k+1} = u_H^k + \gamma \cdot \frac{1}{M}\sum_{m=1}^M Q_m(F_m(w^k) - F_1(w^k) - F_m(u_H^k) + F_1(u_H^k))$;
10: Server updates $w^{k+1} = \begin{cases} z^k, & \text{with probability } p, \\ w^k, & \text{with probability } 1-p, \end{cases}$;
11: **if** $w^{k+1} = z^k$ **then**
12: Server broadcasts w^{k+1} to devices;
13: Devices in parallel compute $F_m(w^k)$ and send to server;
14: Server computes $F(w^{k+1}) = \frac{1}{M}\sum_{m=1}^M F_m(w^{k+1})$;
15: **end if**
16: **end for**

where the calculation of the proximal operator is for free, but, in the general case, The proximal operator is calculated using an auxiliary method that incorporates an additional iteration loop within the main algorithm. The key idea in [23, 16] is to use $r(x) = f_1(x)$ for the proximal operator. This implies the need of additional solving $\arg\min_x\{f_1(x) + \lambda\|x - x_s\|\}$, where x_s is some point. Note that no commutation is needed to calculate the proximal operator as to solution needs access only to f_1 and can be solved locally. This reasoning shows that the idea with composite reformulation is also related to the regularization of the local problem. To extend the idea with composite reformulation into variational inequalities (see (2.2) for connections between min. prob. and variational inequalities), we use the following proximal operator (see the loop in line 3 in Algorithm 18):

$$\text{Find } \hat{u}^k \in \mathbb{R}^d \text{ such that } G(\hat{u}^k) = 0.$$

with $G(z) = F_1(z) + \frac{1}{\gamma}(z - v^k)$ and $v^k = z^k + \tau(w^k - z^k) - \gamma \cdot (F(w^k) - F_1(w^k))$, where the essence of w^k will be explained later. An important issue to be addressed is the selection of the basic method that should be used for an inexact calculation of the proximal operator (we will discuss it in the next paragraphs). We also need to underline one more point. Our approach solves the local problem (proximal operator) only on the server that owns F_1. It is possible to solve different local problems on all devices in parallel as proposed in [108], but this leads to non-optimal convergence guarantees. The method from [108] was revised in [208], which proposed local steps on one device and improved convergence rates. In [40], an algorithm was constructed that uses local updates on all devices and achieves the same convergence results as in [208] (only in the quadratic case). If no

improvement is gained from running local updates on all devices, why should we do it?

• **Basic outer method.** We need a basic method for solving the composite variational inequalities. Gradient Descent is one option for minimization problems, but variational inequalities and saddle point problems have their own optimal methods. Therefore, the basic outer method for Algorithm 18 also needs to be specialized for variational inequalities. In [23, 16], the authors used Extragradient [78, 80, 85]. In our work, we select the Tseng's method [86] for this task.

• **Local method.** Since the subproblem in the inner loop from line 3 in Algorithm 18 is a variational inequality with a strongly monotone and Lipschitz continuous operator, then it can be solved using the Extragradient method [78]. It is optimal for this class of problems.

• **Use of compression.** In order to use compression in the method, an additional technique is also required. Following [217, 212], we take the idea of basing on the variance reduction technique from [214]. We introduce a reference sequence $\{w^k\}_{k\geq 0}$ as in the classical variance reduction method [52]. At point w^k, we need to know the full values of operator F. When w^k is updated (line 10), we transfer the full operators without compression (lines 12 and 13). If probability p is small, w^k is rarely modified and hence condition 11 is satisfied with low probability. It turns out that in Algorithm 18 at each iteration, all nodes send a compressed package and very rarely transmit a full operator. If p is chosen correctly, the effect of this is unnoticeable. However, simply introducing an additional w^k sequence is not enough, we also need to add a negative momentum: $\tau(w^k - z^k)$ (lines 4 and 5). This is dictated by the nature of the variance reduction method for variational inequalities.

• **Compression operators.** In Algorithm 18 (lines 8, 9) any unbiased compressor can be used, but to reveal the δ-similarity (δ-relatedness), we apply already known to us permutation compressors (Definition 2.6.3).

• **Summary.** We utilize the Tseng's method [86] for the composite variational inequality, inaccurately computing the resolvent (proximal operator) with the `Extra Gradient` method [78]. Our convergence analysis is novel for Tseng's method and can be regarded as a byproduct and secondary contribution. We use a variance reduction technique [214] for Tseng's method, remade into a compression technique in distributed settings [217, 212], and choose a special compression for similarity setting [250].

This section presents the convergence results of Algorithm 18 and its analysis.

Theorem 2.6.7. Let $\{z^k\}_{k\geq 0}$ denote the iterates of Algorithm 18 for solving problem (2.1)+(2.7), which satisfies Assumptions 2.0.4, 2.0.5(SM), 2.6.2. Then, if we choose the stepsizes γ, η and the momentum τ as follows:

$$\tau = p \leq \frac{1}{4}, \quad \gamma = \min\left\{\frac{p}{3\mu}, \frac{\sqrt{p}}{4\delta}, \frac{1}{L}\cdot\left(\frac{H}{4\log\frac{40L}{\mu p}} - 1\right)\right\}, \quad \eta = \frac{1}{4\left(L + \frac{1}{\gamma}\right)}, \quad H \geq 8\log\frac{40L}{\mu p},$$

we have the convergence:

$$\mathbb{E}\left\|z^K - z^*\right\|^2 \leq \left(1 - \frac{\gamma\mu}{2}\right)^K \cdot 2\left\|z^0 - z^*\right\|^2.$$

In Algorithm 18, one mandatory communication round with compression occurs (lines 7 and 8) and possibly one more (without compression) with probability p (lines 12 and 13). Permutation compressor compress by a factor of M; hence each iteration requires $\mathcal{O}\left(\frac{1}{M} + p\right)$ data transfers from devices to the server on average: in line 8, we use compression and forward $\frac{d}{M}$ coordinates from the devices to the server, in line 13, with probability p we transfer the full vector of d coordinates.

If p is close to 1, Theorem 2.6.7 gives faster convergence, but more data transfer is needed. If p tends to 0, the transmitted information complexity per iteration decreases but the iterative convergence rate drops. The optimal choice of p is $\frac{1}{M}$, as stated in the corollary.

Corollary 2.6.8. Under the conditions of Theorem 2.6.7, the following number of outer iterations is needed to achieve the accuracy ε (in terms of $\mathbb{E}\|z - z^*\|^2 \lesssim \varepsilon$) by Algorithm 18 with $p = \frac{1}{M}$:

$$\mathcal{O}\left(\left[M + \frac{\delta\sqrt{M}}{\mu} + \frac{L}{\mu H}\right] \log \frac{\|z^0 - z^*\|^2}{\varepsilon}\right). \tag{2.31}$$

Next, let us give a brief discussion of the previous estimate.

• **Communications and local calculations.** The iteration of Algorithm 18 has $(1 + p)$ communication rounds and forwards $\mathcal{O}\left(\frac{1}{M} + p\right)$ information on average. With $p = \frac{1}{M}$, the estimate in equation (2.31) is valid for both the number of iterations and communication rounds. To obtain an upper bound for the amount of information transmitted, divide (2.31) by M. The bound from Corollary 2.6.8 reflects the local complexity on all devices except the first (the server). To get an estimate for the first device, we need to multiply (2.31) by H.

• **Optimal choice of H.** It is clear from (2.31) that the number of local H steps can improve the number of communications, but there is a limit beyond which more iterations are not useful. Taking $H = \lceil \frac{L}{\delta\sqrt{M}} \rceil$ is optimal. If there is no data similarity ($\delta \sim L$), then one can choose $H = 1$, and Algorithm 18 communication complexity estimate becomes similar to those of `MASHA` and `Optimistic MASHA`. Choosing $H = 1$ implies that local updates are meaningless when $\delta \sim L$. Papers [40, 254] on distributed minimization problems show that collecting large batches works just as well as methods that perform many local updates. This thesis considers deterministic methods; thus, we collect full batches and need local steps only in the presence of data similarity.

• **"Break" lower bounds.** H from the previous paragraph yields estimates of $\tilde{\mathcal{O}}(n + \delta\sqrt{M}/\mu)$ and $\tilde{\mathcal{O}}(1 + \delta/\sqrt{M}\mu)$ for the number of communications and transmitted information, respectively. These results are better than any existing methods and even superior to lower bounds for deterministic methods. Note that our method utilizes randomized compression; hence, lower bounds for deterministic methods do not apply to Algorithm 18 (see Table 2.5). To close this gap in Section 2.6.3 we provide new lower bound estimates that demonstrate the optimality of our approach.

• **Variable server.** In Algorithm 18, the server can be reselected deterministically or randomly at each outer iteration. Communication with the server can be decentralized using reduce and gather procedures [255], for this kind of procedures compression also greatly speeds up communication both theoretically and in practice [256]. Although the server can change, and decentralized communication networks can be used, it must still receive data from all devices. Asynchronous communication is not supported, and the lack of communication with the device poses a problem for Algorithm 18. This is a potential issue that slows down the method. This issue is partially addressed by Algorithm 19 in the next section, which assumes communication with only 1 device per iteration.

Extension with partial participation

We also propose a modification of Algorithm 18 with partial participation (see Algorithm 19).

Algorithm 19 `Three Pillars Algorithm with Partial Participation`

Parameters: stepsizes γ and η, momentum τ, probability p, number of local steps H
Initialization: choose $z^0 \in \mathbb{R}^d$, $z_i^0 = z^0$, for all $i \in [M]$

for $k = 0, 1, \ldots, K-1$ **do**
  Server takes $u_0^k = z^0$;
  for $t = 0, 1, \ldots, H-1$ **do**
    Server computes
    $$u_{t+1/2}^k = u_t^k - \eta \left(F_1(u_t^k) - F_1(w^k) + F(w^k) + \tfrac{1}{\gamma}(u_t^k - z^k - \tau(w^k - z^k)) \right);$$
    Server updates
    $$u_{t+1}^k = u_t^k - \eta \left(F_1(u_{t+1/2}^k) - F_1(w^k) + F(w^k) + \tfrac{1}{\gamma}(u_{t+1/2}^k - z^k - \tau(w^k - z^k)) \right);$$
  end for
  Server broadcasts u_H^k to devices;
  Generate uniformly i_k from $[M]$;
  Device m_k computes $F_{m_k}(u_H^k)$ and sends to server;
  Server updates $z^{k+1} = u_H^k + \gamma \cdot (F_{m_k}(w^k) - F_1(w^k) - F_{m_k}(u_H^k) + F_1(u_H^k))$;
  Server updates $w^{k+1} = \begin{cases} z^k, & \text{with probability } p, \\ w^k, & \text{with probability } 1-p, \end{cases}$;
  if $w^{k+1} = z^k$ **then**
    Server broadcasts w^{k+1} to devices;
    All devices in parallel compute $F_m(w^k)$ and send to server;
    Server takes all $F_m(w^k)$, saves them and computes $F(w^{k+1}) = \frac{1}{M} \sum\limits_{m=1}^{M} F_m(w^{k+1})$;
  end if
end for

The following theorem is valid for Algorithm 19.

Theorem 2.6.9. Let $\{z^k\}_{k \geq 0}$ denote the iterates of Algorithm 19 for solving problem (2.1)+(2.7), which satisfies Assumptions 2.0.4, 2.0.5(SM), 2.6.2. Then, if we choose the stepsizes γ, η and the

momentum τ as follows:

$$\tau = p \leq \frac{1}{4}, \quad \gamma = \min\left\{\frac{p}{3\mu}, \frac{\sqrt{p}}{4\delta}, \frac{1}{L}\cdot\left(\frac{H}{4\log\frac{40L}{\mu p}} - 1\right)\right\}, \quad \eta = \frac{1}{4\left(L + \frac{1}{\gamma}\right)}, \quad H \geq 8\log\frac{40L}{\mu p},$$

we have the convergence:

$$\mathbb{E}\left\|z^K - z^*\right\|^2 \leq \left(1 - \frac{\gamma\mu}{2}\right)^K \cdot 2\left\|z^0 - z^*\right\|^2.$$

While in lines 8 and 9 of Algorithm 18 all devices compute the full local operators but forward only $\mathcal{O}\left(\frac{1}{M}\right)$ information; in Algorithm 19, only one local device computes its operator and forwards this full operator. Hence, the total amount of transmitted information from all devices is equivalent for Algorithms 18 and 19. Therefore, the optimal choice of p for Algorithm 19 is the same as for Algorithm 18.

Corollary 2.6.10. Under the conditions of Theorem 2.6.9, $\mathcal{O}([n + \frac{\delta\sqrt{M}}{\mu} + \frac{L}{\mu H}]\log\frac{\|z^0 - z^*\|^2}{\varepsilon})$ outer iterations are required for the accuracy ε (in terms of $\mathbb{E}\|z - z^*\|^2 \lesssim \varepsilon$) by Algorithm 19 with $p = \frac{1}{M}$.

Despite the fact that the estimates from Corollary 2.6.8 and 2.6.10 are the same, Algorithm 19 is better than Algorithm 18 in terms of local devices (but not the server) complexity. In line 8 of Algorithm 18 all devices collect the full operator, while in Algorithm 19 only one node does so. Consequently, the local complexity (excluding the first node) of Algorithm 19 is M times better. But while Algorithm 18 and Algorithm 19 are the same in terms of the number of transmitted information, Algorithm 19 is M times worse in terms of communication time. This is due to the fact that in line 8 of Algorithm 18, all devices in parallel send information using $\mathcal{O}\left(\frac{1}{M}\right)$ units of time, while in Algorithm 19, one node needs $\mathcal{O}(1)$ units of time.

Lower bounds

To derive lower bounds, we define a "full package" as the amount of information transmitted in one complete communication round from all devices to the server without compression. We start with the lower bounds for algorithms with partial participation, as discussed in the previous section.

Theorem 2.6.11. For any $\delta, \mu > 0$ and $M \in \mathbb{N}$ and $K \in \mathbb{N}$, there exists a distributed variational inequality (satisfying Assumptions 2.0.5(SM) and 2.6.2) on $\mathbb{R}^d$ (where d is sufficiently large) with $z^* \neq 0$, such that for any output $\hat{z}$ of any distributed algorithm with partial participation after sending K full packages from the devices to the server, it holds that $\mathbb{E}[\|\hat{z} - z^*\|^2]$ is $\Omega(\exp(-16/(1 + \sqrt{\frac{\delta^2 M}{16\mu^2} + 1})\cdot K)\|z^0 - z^*\|^2)$.

The lower bounds confirm that results of the form $(1 + \delta/(\sqrt{M}\mu))$ are optimal for distributed methods with partial participation; thus, our algorithm achieves this optimality. Lower bounds serve to present a "worst-case" scenario for distributed variational inequalities, wherein any

algorithm with partial participation cannot outperform the established bounds. For the methods with compression, we can also show that our result is optimal. But here we have to restrict the class of compression operators and consider only random sparsifiers – random choice of coordinates for forwarding (Permutation compressors are suitable). How to obtain lower guarantees for arbitrary compressors remains an open question.

Extension with stochastic local computations

In this section, we consider a stochastic formulation of (2.1)+(2.7), namely, we assume that each local term from (2.1)+(2.7) also has the form of a finite sum: $F_m(z) = \frac{1}{n}\sum_{i=1}^n F_{m,i}(z)$. Although we can compute F_m locally on each mth device, we would like to avoid this due to computational cost. The stochastic finite sum setting is common for machine learning and refers to the Monte Carlo approach and empirical risk minimization [51]. Moreover, such a setup is particularly natural in the context of similarity. For example, if we initially have a large dataset $N = nMb$ and distribute it to M devices, then each device has nb samples: $F = \frac{1}{M}\sum_{m=1}^M F_m = \frac{1}{M}\sum_{m=1}^M[\frac{1}{n}\sum_{i=1}^n F_{m,i}] = \frac{1}{M}\sum_{m=1}^M[\frac{1}{n}\sum_{i=1}^n(\frac{1}{b}\sum_{j=1}^b F_{m,i,j})]$. The nature of similarity goes precisely that F_m has the form of a sub-sum taken from a whole large sum. As we noted in Section 2.4, the larger the sample size nb for F_m, the smaller the similarity parameter δ. Now in the stochastic case we consider $F_{m,i}$ and they also have similarity, but with a larger $\tilde{\delta}$, since nb is larger than b. To work with the new setting, we introduce two assumptions.

Assumption 2.6.12 (Lipschitzness in mean)**.** Each operator F_m is $\tilde{L}$-Lipschitz continuous in mean on $\mathbb{R}^d$, i.e. for all $u, v \in \mathbb{R}^d$ we have $\frac{1}{n}\sum_{i=1}^n \|F_{m,i}(u) - F_{m,i}(v)\|^2 \leq \tilde{L}^2\|u - v\|^2$.

Assumption 2.6.13. The operators $\{F_{m,i}\}$ is $\tilde{\delta}$-related in mean. For any m_1 operators $\{F_{m_1,i_1} - F_{m_2,i_2}\}$ is $\tilde{\delta}$-Lipschitz continuous in mean, i.e. for all $u, v \in \mathbb{R}^d$ we have
$\frac{1}{Mn^2}\sum_{m_2=1}^M\sum_{i_1,i_2=1}^n \|F_{m_1,i_1}(u) - F_{m_2,i_2}(u) - F_{m_1,i_1}(v) + F_{m_2,i_2}(v)\|^2 \leq \tilde{\delta}^2\|u - v\|^2$.

The modified version of Algorithm 18 for the new setting looks as follows (Algorithm 20):

To address stochasticity, we implement two modifications in Algorithm 18: in lines 2–5 we switch the local algorithm from classical deterministic `Extra Gradient` to stochastic `Extra Gradient` from [214] (Algorithm 1), in lines 7–9 we utilize stochastic operators for all $\{F_m\}$ instead of deterministic ones. The next theorem and corollary provide the convergence.

Theorem 2.6.14. Let $\{z^k\}_{k\geq 0}$ denote the iterates of Algorithm 20 for solving problem (2.1)+(2.7), which satisfies Assumptions 2.0.5(SM), 2.6.12, 2.6.13. Then, if we choose the stepsizes γ, η and the momentum τ as follows:

$$\tau = p \leq \frac{1}{4}, \quad \gamma = \min\left\{\frac{p}{3\mu}, \frac{\sqrt{p}}{4\tilde{\delta}}, \frac{1}{\tilde{L}}\cdot\left(\frac{H}{12\sqrt{n}\log\frac{40\tilde{L}}{\mu p}} - 1\right)\right\}, \quad \eta = \frac{1}{2\sqrt{n}\left(\tilde{L} + \frac{1}{\gamma}\right)},$$

Algorithm 20 `Three Pillars Algorithm with Variance Reduction`

Parameters: stepsizes γ and η, momentums τ and α, probabilities p and q , number of local steps H

Initialization: choose $z^0 = w^0 \in \mathbb{R}^d$

for $k = 0, 1, \ldots, K-1$ **do**
  Server takes $u_0^k = w_0^k = z^0$;
  for $t = 0, 1, \ldots, H-1$ **do**
    Server computes
    $u_{t+1/2}^k = \alpha u_t^k + (1-\alpha) w_t^k - \eta(F_1(w_t^k) - F_1(w^k) + F(w^k) + \frac{1}{\gamma}(w_t^k - z^k - \tau(w^k - z^k)))$;
    Server updates
    $u_{t+1}^k = \alpha u_t^k + (1-\alpha) w_t^k - \eta(F_{1,i_t}(u_{t+1/2}^k) - F_{1,i_t}(w_t^k) + F_1(w_t^k) - F_1(w^k) + F(w^k)$
    $+\frac{1}{\gamma}(u_{t+1/2}^k - z^k - \tau(w^k - z^k)))$, where i_t is generated uniformly from $[n]$;
    Server updates $w_{t+1}^k = \begin{cases} u_{t+1}^k, & \text{with probability } q, \\ w_t^k, & \text{with probability } 1-q, \end{cases}$;
  end for
  Server broadcasts u_H^k and $F_{1,i_k^1}(u_H^k) - F_{1,i_k^1}(w^k)$ to devices, where i_k^1 is generated uniformly from $[n]$;
  Devices in parallel compute $Q_m(F_{m,i_k^m}(w^k) - F_{1,i_k^1}(w^k) - F_{i,i_k^m}(u_H^k) + F_{1,i_k^1}(u_H^k))$ and send to server, where i_k^m is generated uniformly from $[n]$;
  Server updates
  $z^{k+1} = u_H^k + \gamma \cdot \frac{1}{M}\sum_{m=1}^M Q_m(F_{m,i_k^m}(w^k) - F_{1,i_k^1}(w^k) - F_{m,i_k^m}(u_H^k) + F_{1,i_k^1}(u_H^k))$;
  Server updates $w^{k+1} = \begin{cases} z^k, & \text{with probability } p, \\ w^k, & \text{with probability } 1-p, \end{cases}$;
  if $w^{k+1} = z^k$ **then**
    Server broadcasts w^{k+1} to devices;
    All devices in parallel compute $F_m(w^k)$ and send to server;
    Server takes all $F_m(w^k)$, saves them and computes $F(w^{k+1}) = \frac{1}{M}\sum\limits_{m=1}^M F_m(w^{k+1})$;
  end if
end for

$$q = \frac{1}{n}, \quad \alpha = 1 - \frac{1}{n}, \quad H \geq 16n \log \frac{40\tilde{L}}{\mu p},$$

we have the convergence:

$$\mathbb{E}\left\|z^K - z^*\right\|^2 \leq \left(1 - \frac{\gamma\mu}{2}\right)^K \cdot 2\left\|z^0 - z^*\right\|^2.$$

Corollary 2.6.15. Under the conditions of Theorem 2.6.14, $\mathcal{O}([M + \frac{\tilde{\delta}\sqrt{M}}{\mu} + \frac{\sqrt{n}\tilde{L}}{\mu H}] \log \frac{\|z^0 - z^*\|^2}{\varepsilon})$ outer iterations are needed for the accuracy ε (in terms of $\mathbb{E}\|z - z^*\|^2 \lesssim \varepsilon$) by Algorithm 20 with $p = \frac{1}{M}$.

Non-monotone case

In this section, we abandon the monotonicity assumption.

Theorem 2.6.16. Let $\{z^k\}_{k\geq 0}$ denote the iterates of Algorithm 18 for solving problem (2.1)+(2.7), which satisfies Assumptions 2.0.4, 2.0.5(NM), 2.6.2. Then, if we additionally assume that $\|z^k\|, \|u_H^k\|, \|z^*\| \leq \Omega$ and choose the stepsizes γ, η and the momentum τ as follows:

$$\tau = p \leq \frac{1}{2}, \quad \gamma = \frac{\sqrt{p}}{6\delta},$$

we have the convergence:

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|F(u_H^k)\|^2 \lesssim \frac{\delta^2\Omega^2}{pK}\left(\frac{1}{K} + \frac{1}{\sqrt{H}}\left(\frac{L\sqrt{p}}{6\delta}+1\right) + \frac{1}{H}\left(\frac{L\sqrt{p}}{6\delta}+1\right)^2\right).$$

By substituting $H \sim K^2(\frac{L\sqrt{p}}{\delta}+1)^2$, we can obtain that $\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|F(u_H^k)\|^2] \lesssim \frac{\delta^2\Omega^2}{pK}$. The discussion of the optimal choice of the parameter p, in this case, is the same as that following Theorem 2.6.7. Therefore, the following corollary is true.

Corollary 2.6.17. Under the conditions of Theorem 2.6.16, $\mathcal{O}(\frac{M\delta^2\Omega^2}{\varepsilon^2})$ outer iterations is needed to achieve the accuracy ε (in terms of $\mathbb{E}\|F(z)\|^2 \lesssim \varepsilon^2$) by Algorithm 18 with $p = \frac{1}{M}$.

One can note that the number of information transmitted is M times less than the number of iterations in Corollary 2.6.17. Therefore, we have the communication complexity is equal to $\mathcal{O}\left(\delta^2\Omega^2/\varepsilon^2\right)$. This is better than the estimate $\mathcal{O}\left(L^2\Omega^2/\varepsilon^2\right)$ of a basic method that does not use local steps, similarity, and compression [257]. The problem is that our estimate of communication complexity does not benefit from large M. In the strongly monotone case, however, we observe a more favorable situation (see Table 2.5). Unfortunately, this is a standard problem of the non-monotone case. For example, the methods from Section 2.5 with compression have the same complexity $\mathcal{O}\left(L^2\Omega^2/\varepsilon^2\right)$ as the basic method without compression. The methods from Section 2.3 converge only in the special case when all local operators are equal to each other. Other papers from Table 2.5 do not consider the non-monotone case or consider it under other assumptions; thus, their complexity results are hard to compare with ours. Finally, it turns out that our work is the first to achieve an acceleration in terms of communication complexity in the non-monotone case. On the negative side, we can note that the estimate on the number of local iterations H in Theorem 2.6.16 is rather strict and grows with the number of iterations K. The results for Algorithm 19 in the non-monotone case can be obtained in a similar way.

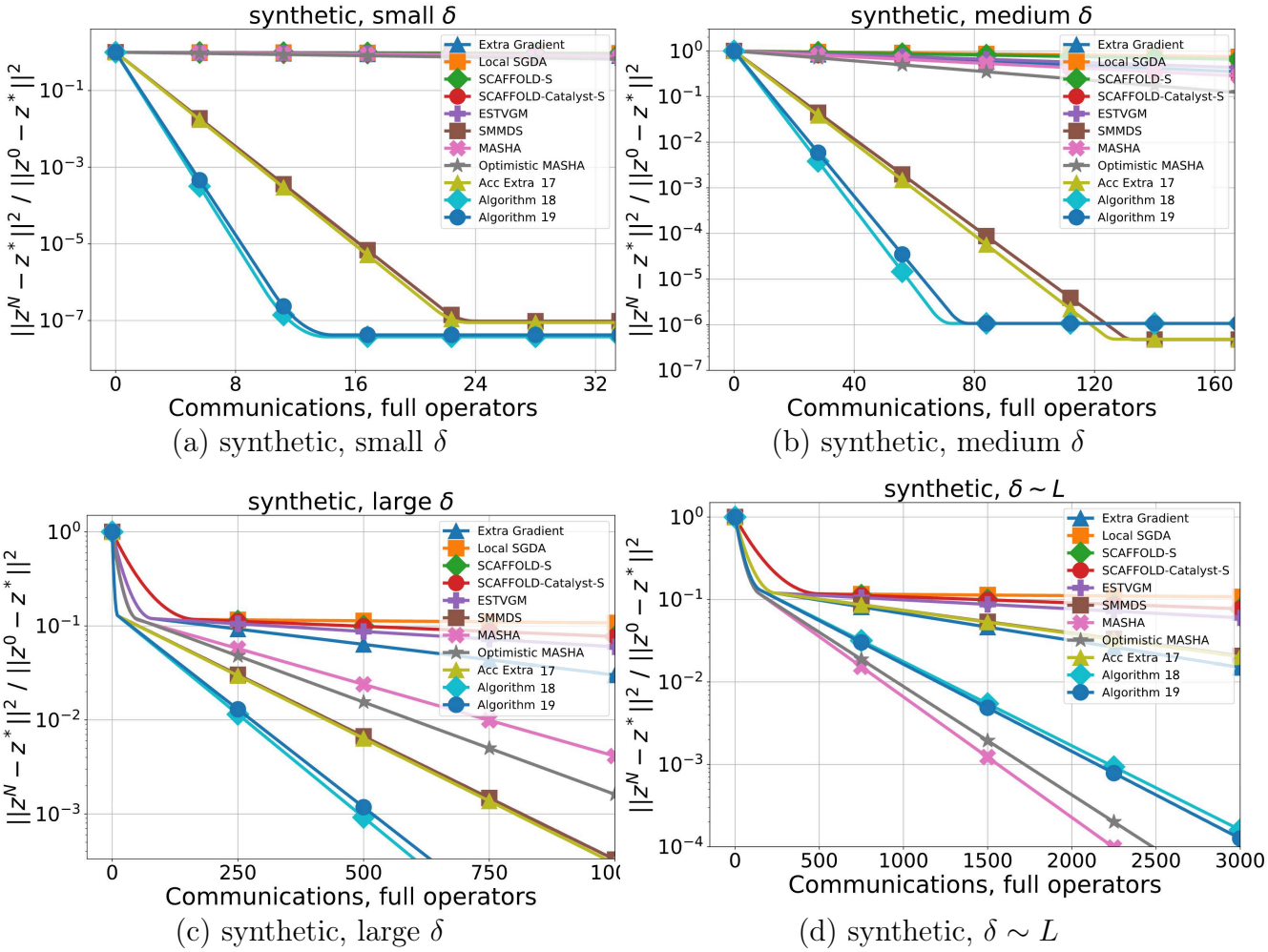
2.6.4 Experiments

We conduct experiments on the robust linear regression problem (see (2.6)). This problem is defined as

$$\min_{x\in\mathbb{R}^d}\max_{\|r_i\|\leq R}\frac{1}{2n}\sum_{i=1}^{n}(x^T(a_i+r_i)-b_i)^2 + \frac{\lambda}{2}\|x\|^2 - \frac{\beta}{2}\|r_i\|^2.$$

We consider a network with $n = 25$ devices and two types of datasets: synthetic and real. Synthetic data allows us to control the factor δ, which measures statistical similarity of functions over

different nodes. We take local datasets of size $b = 100$ and generate the dataset $\{\hat{a}_i, \hat{b}_i\}_{i=1}^n$ at the server randomly, with each entry drawn from the Standard Gaussian distribution. The datasets at other devices ($m = 2, \ldots, M$) are obtained by perturbing $\{\hat{a}_i, \hat{b}_i\}_{i=1}^n$ with normal noise ξ_i, having zero mean and controlled variance. Real datasets are taken from the LibSVM library [171]. We uniformly divide the whole dataset among the devices. The solution z^* is found using a large number of iterations of `Extra Gradient` [78, 85] method. Algorithms 17, 18 and 19 are compared with methods from Table 2.5 (`Extra Gradient` [78, 85], `Local SGDA` [157], `FedAvg-S` [158], `SCAFFOLD-S` [158], `SCAFFOLD-Catalyst-S` [158], `ESTVGM`, `SMMDS`, `MASHA`). All the methods are tuned as in the theory of the corresponding papers. The number of full operator transmissions by one of the arbitrary devices is used as the x-asis. The experiment results are shown in Figures 2.11, 2.12. It is evident that Algorithm 18 and 19 consistently outperform all other competing algorithms by a significant margin. Furthermore, as the value of δ decreases, the superiority over `Local SGDA`, `SCAFFOLD-S`, `SCAFFOLD-Catalyst-S`, `MASHA` becomes even more pronounced.



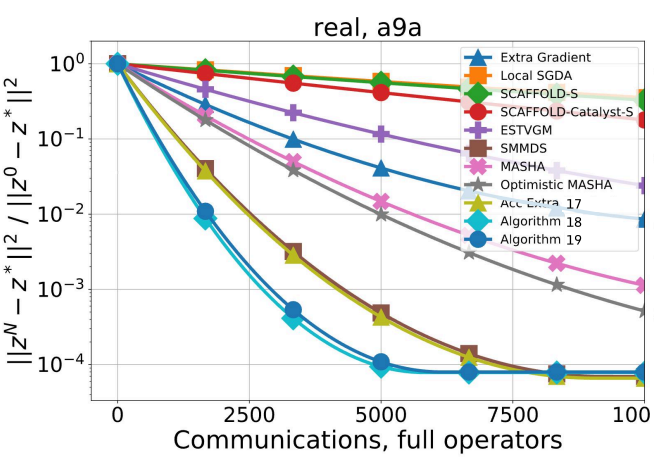
(a) synthetic, small δ

(b) synthetic, medium δ

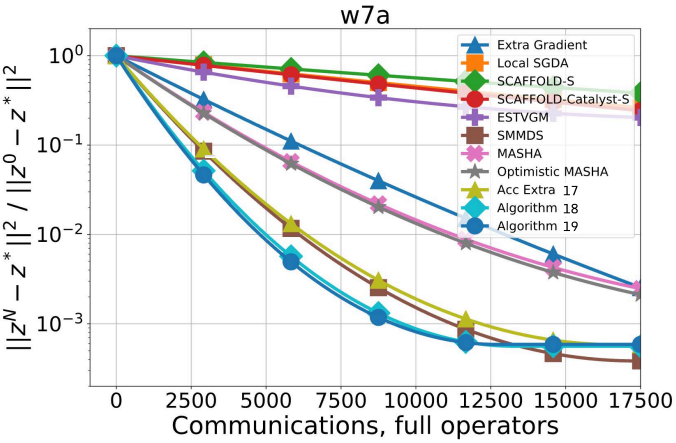
(c) synthetic, large δ

(d) synthetic, $\delta \sim L$

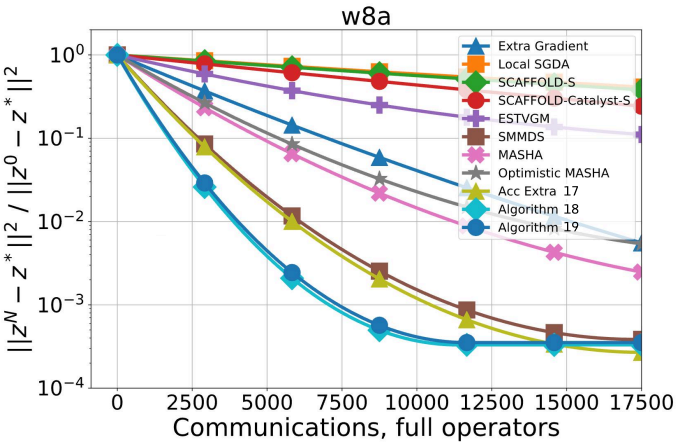
Figure 2.11: Comparison of state-of-the-art methods for distributed variational inequalities. The comparison is made on synthetic datasets with small, medium, and large δ. The x-axis denotes the number of full operators transmitted by one of the devices.



(e) real, a9a



(f) real, w7a



(g) real, w8a

Figure 2.12: Comparison of state-of-the-art methods for distributed variational inequalities. The comparison is made on the real datasets a9a, w7a, w8a from LibSVM. The x-axis denotes the number of full operators transmitted by one of the devices.

Chapter 3

Stochastic Saddle Point Problems and Variational Inequalities

In this chapter, we continue to discuss saddle point problems and variational inequalities. In particular, we focus on problems of the form

$$\langle F(z^*), z - z^* \rangle + g(z) - g(z^*) \geq 0 \quad \text{for all} \quad z \in \mathbb{R}^d, \tag{3.1}$$

where $F : \mathbb{R}^d \to \mathbb{R}^d$ is some operator and $g : \mathbb{R}^d \to \mathbb{R}$ is a regularization term (a proper lower semicontinuous convex function) that is assumed to have a simple structure.

The only aspect distinct in this chapter is the departure from distributed formulations. The focus remains solely on the stochasticity of the formulation. There are two sections featuring non-overlapping plots. The first section addresses cocoercive variational inequalities and the Descent-Ascent method related to them. The second section returns to classical variational inequalities with a Lipschitz operator and discusses the optimal method with batching.

Since we continue to work with variational inequalities in this chapter, we need the Gap convergence criterion already known from the previous chapter. Let us introduce it again in this chapter for completeness:

$$\text{Gap}(z) := \sup_{u \in \mathcal{C}} \left[\langle F(u), z - u \rangle + g(x) - g(u) \right]. \tag{3.2}$$

3.1 Cocoercive Variational Inequalities

As mentioned above, we are interested in stochastic formulations of problem (3.1). In this section, we try to include as many different plots as possible under a single umbrella. More precisely, we are interested in the situations when operator F is accessible through the calls of an unbiased stochastic oracle. This is natural when F has an expectation form $F(x) = \mathbb{E}_{\xi \sim \mathcal{D}}[F_\xi(z)]$ or a finite-sum form $F(z) = \frac{1}{n} \sum_{i=1}^n F_i(z)$. In the context of machine learning, $\mathcal{D}$ corresponds to some unknown distribution on the data, n corresponds to the number of samples, and F_ξ, F_i denote vector fields corresponding to the samples ξ, and i, respectively [81, 258]. We examined these examples in the previous chapter.

One of the most popular methods for solving (3.1) is Stochastic Gradient Descent-Ascent (SGDA) [259, 260]. However, besides its rich history, SGDA only recently was analyzed without using strong assumptions on the noise [258] such as uniformly bounded variance. In the last few years, several powerful algorithmic techniques like variance reduction [227, 218] and coordinate-wise randomization [261], were also combined with SGDA resulting in better algorithms. However, these methods were analyzed under different assumptions, using different analysis approaches, and required different intuitions. Moreover, to the best of our knowledge, fruitful directions such as communication compression for distributed versions of SGDA or linearly converging variants of coordinate-wise methods for regularized VIPs were never considered in the literature before.

All of these facts motivate the importance and necessity of a novel general analysis of SGDA unifying several special cases and providing the ability to design and analyze new SGDA-like methods filling existing gaps in the theoretical understanding of the method.

3.1.1 Assumptions

Throughout the section, we assume that (3.1) has at least one solution and operator F is *μ-quasi-strongly monotone* and *ℓ-star-cocoercive*: there exist constants $\mu \geq 0$ and $\ell > 0$ such that for all $z \in \mathbb{R}^d$

$$\langle F(z) - F(z^*), z - z^* \rangle \geq \mu \|z - z^*\|^2, \tag{3.3}$$

$$\|F(z) - F(z^*)\|^2 \leq \ell \langle F(z) - F(z^*), z - z^* \rangle, \tag{3.4}$$

where $z^* = \text{proj}_{\mathcal{Z}^*}(x) := \arg\min_{y \in Z^*} \|y - z\|$ is the projection of z on the solution set $\mathcal{Z}^*$ of (3.1). If $\mu = 0$, inequality (3.3) is known as the variational stability condition [114], which is weaker than standard monotonicity: $\langle F(z) - F(y), z - y \rangle \geq 0$ for all $z, y \in \mathbb{R}^d$. It is worth mentioning that there exist examples of non-monotone operators satisfying (3.3) with $\mu > 0$ [258]. Condition (3.4) is a relaxation of standard cocoercivity $\|F(z) - F(y)\|^2 \leq \ell \langle F(z) - F(y), z - y \rangle$. At this point, let us highlight that it is possible for an operator F to satisfy (3.4) and not be Lipschitz continuous [258]. This emphasizes the wider applicability of the *ℓ-star-cocoercivity* compared to ℓ-cocoercivity. We emphasize that in our convergence analysis, we do not assume ℓ-cocoercivity nor L-Lipschitzness of F.

We consider SGDA for solving (3.1) in its general form:

$$z^{k+1} = \text{prox}_{\gamma_k g}(z^k - \gamma_k g^k), \tag{3.5}$$

where g^k is an unbiased estimator of $F(z^k)$, $\gamma_k > 0$ is a stepsize at iteration k, and $\text{prox}_{\gamma g}(z) := \arg\min_{y \in \mathbb{R}^d} \{g(y) + {}^{\|y-z\|^2}/_{2\gamma}\}$ is a proximal operator defined for any $\gamma > 0$ and $z \in \mathbb{R}^d$. While g^k gives an information about operator F at step k, the proximal operator is needed to take into account regularization term g. We assume that function g is such that $\text{prox}_{\gamma g}(x)$ can be easily computed for all $z \in \mathbb{R}^d$. This is a standard assumption satisfied for many practically interesting regularizers [262]. By default, we assume that $\gamma_k \equiv \gamma > 0$ for all $k \geq 0$.

3.1.2 Contributions

• **Unified analysis of SGDA.** We propose a general assumption on the stochastic estimates and the problem (3.1) (Assumption 3.1.1) and show that several variants of SGDA (3.5) satisfy this assumption. In particular, through our approach, we cover SGDA with arbitrary sampling [258], variance reduction, coordinate randomization, and compressed communication. Under Assumption 3.1.1 we derive general convergence results for quasi-strongly monotone (Theorem 3.1.2), monotone star-cocoercive (Theorem 3.1.5) and cocoercive problems (Theorem 3.1.7).

• **Extensions of known methods and analysis.** As a by-product of the generality of our theoretical framework, we derive new results for the proximal extensions of several known methods such as *proximal* SGDA-AS [258] and *proximal* SGDA with coordinate randomization [261]. Moreover, we close some gaps on the convergence of known methods, e.g., we derive the first convergence guarantees in the monotone case for SGDA-AS [258] and SAGA-SGDA [227] and we obtain the first result on the convergence of SAGA-SGDA for (averaged star-)cocoercive operators.

• **Sharp rates for known special cases.** For the known methods fitting our framework, our general theorems either recover the best rates known for these methods (SGDA-AS) or tighten them (SGDA-SAGA, Coordinate SGDA).

• **New methods.** The flexibility of our approach allows us to develop and analyze several new variants of SGDA. Guided by algorithmic advances for solving minimization problems, we propose a new variance-reduced method (L-SVRGDA), new distributed methods with compression (QSGDA, DIANA-SGDA, VR-DIANA-SGDA), and a new method with coordinate randomization (SEGA-SGDA). We show that the proposed methods fit our theoretical framework, and using our general theorems, we obtain tight convergence guarantees for them. Although the analogs of these methods are known for solving minimization problems [263, 264, 43, 60, 265, 59], they were never considered for solving min-max and variational inequality problems. Therefore, by proposing and analyzing these new methods, we close several gaps in the literature on SGDA. For example, VR-DIANA-SGDA is the first SGDA-type *linearly converging* distributed stochastic method with compression, and SEGA-SGDA is the first *linearly converging* coordinate method for solving *regularized* VIPs.

• **Numerical evaluation.** In numerical experiments, we illustrate the most important properties of the new methods. The results corroborate our theoretical findings.

Throughout the section, we provide necessary comparisons with closely related work.

3.1.3 Main Theorems

Unified Assumption

Key assumption. We start by introducing the next parametric assumption – a central part of our approach.

Assumption 3.1.1. We assume that for all $k \geq 0$ the estimator g^k from (3.5) is unbiased: $\mathbb{E}_k\left[g^k\right] = F(z^k)$, where $\mathbb{E}_k[\cdot]$ denotes the expectation w.r.t. the randomness at iteration k. Next, we assume that there exist non-negative constants $A, B, C, D_1, D_1 \geq 0$, $\rho \in (0, 1]$ and a sequence of (possibly random) non-negative variables $\{\sigma_k\}_{k\geq 0}$ such that for all $k \geq 0$

$$\mathbb{E}_k\left[\|g^k - g^{*,k}\|^2\right] \leq 2A\langle F(z^k) - g^{*,k}, z^k - z^{*,k}\rangle + B\sigma_k^2 + D_1, \tag{3.6}$$

$$\mathbb{E}_k\left[\sigma_{k+1}^2\right] \leq 2C\langle F(z^k) - g^{*,k}, z^k - z^{*,k}\rangle + (1-\rho)\sigma_k^2 + D_2, \tag{3.7}$$

where $z^{*,k} = \text{proj}_{\mathcal{Z}^*}(z^k)$ and $g^{*,k} = F(z^{*,k})$.

While the unbiasedness of g^k is a standard assumption, inequalities (3.6)-(3.7) are new and require clarifications. For simplicity, assume that $\sigma_k^2 \equiv 0$, $F(z^*) = 0$ for all $z^* \in \mathcal{Z}^*$, and focus on (3.6). In this case, (3.6) gives an upper bound for the second moment of the stochastic estimate g^k. For example, such a bound follows from the expected cocoercivity assumption [258], where A denotes some expected/averaged (star-)cocoercivity constant and D_1 stands for the variance at the solution (see also Section 3.1.4). When F is not necessarily zero on $\mathcal{Z}^*$, the shift $g^{*,k}$ helps to take this fact into account. Finally, the sequence $\{\sigma_k^2\}_{k\geq 0}$ is typically needed to capture the variance reduction process, parameter B is typically some numerical constant, C is another constant related to (star-)cocoercivity[1], and D_2 is the remaining noise that is not handled by the variance reduction process. As we show in the next sections, inequalities (3.6)-(3.7) hold for various `SGDA`-type methods.

We point out that Assumption 3.1.1 is inspired by similar assumptions that appeared in [266, 267]. However, the difference between our assumption and the ones that appeared in these papers is significant. [266] focuses on solving the much simpler minimization problems using `SGD`. In particular, their Assumption 4.1 requires a function suboptimality (or Bregman divergence) for the upper bound, a concept that cannot be used in VI problems (there are no functions). Thus, the difference between the two notions does not solely lie in the norm bound but begins at a deeper, conceptual level. In addition, we also focus on monotone VIs (non-quasi-strongly monotone), while [266] considers only the class of quasi-strongly convex minimization problems. Next, [267] provides convergence guarantees for vanilla `SEG` under the arbitrary sampling paradigm. Their analysis is not able to capture `SEG` with variance reduction, quantization, and coordinate-wise randomization. In contrast, our approach covers variants of `SGDA` with variance reduction, quantization, and coordinate-wise randomization. We are able to capture these more advanced variants by using sequence $\{\sigma_k^2\}_{k\geq 0}$ (see (3.7)) in our key assumption, and this is a major difference between our approach and the approach of [267]. In addition, our analysis works for the case $g(z) \not\equiv 0$. Although the generalization of the analysis to the case of non-zero g might be trivial in the quasi-strongly monotone case, for the monotone case this is definitely not straightforward. Finally, for the monotone case, we do not require large batch sizes to achieve any predefined accuracy, while the

[1]Although Assumption 3.1.1 does not formally imply star-cocoercivity of F, but in all special cases, considered in this work, operator F is star-cocoercive.

analysis of SEG in [267] does (see Appendix B in their work).

Convergence Results

Quasi-strongly monotone case. Under Assumption 3.1.1 and the quasi-strong monotonicity of F, we derive the following general result.

Theorem 3.1.2. Let F be μ-quasi-strongly monotone ($\mu > 0$) and let Assumption 3.1.1 hold. Assume that $0 < \gamma \leq \min\{1/\mu, 1/2(A+CM)\}$ for some $M > B/\rho$ (when $B = 0$, we suppose $M = 0$ and $B/M := 0$ in all following expressions). Then the iterates of SGDA (3.5), satisfy:

$$\mathbb{E}[V_k] \leq \left(1 - \min\left\{\gamma\mu, \rho - \frac{B}{M}\right\}\right)^k V_0 + \frac{\gamma^2(D_1 + MD_2)}{\min\{\gamma\mu, \rho - B/M\}}.$$

where the Lyapunov function V_k is defined by $V_k = \|z^k - z^{*,k}\|^2 + M\gamma^2\sigma_k^2$ for all $k \geq 0$.

See the proof in Section A.2.6. The above theorem states that SGDA (3.5) converges linearly to the neighborhood of the solution. The size of the neighborhood is proportional to the noises D_1 and D_2. When $D_1 = D_2 = 0$, i.e., the method is variance reduced, it converges linearly to the exact solution in expectation. However, in general, to achieve any predefined accuracy, one needs to reduce the size of the neighborhood somehow. One possible way to do that is to use a proper stepsize schedule. We formalize this discussion in the following result.

Corollary 3.1.3. Let the assumptions of Theorem 3.1.2 hold. Consider two possible cases.

Case 1. Let $D_1 = D_2 = 0$. Then, for any $K \geq 0$, $M = 2B/\rho$, and $\gamma = \min\{1/\mu, 1/2(A+2BC/\rho)\}$, the iterates of SGDA, given by (3.5), satisfy: $\mathbb{E}[V_K] \leq V_0 \exp\left(-\min\left\{\frac{\mu}{2(A+2BC/\rho)}, \frac{\rho}{2}\right\} K\right)$.

Case 2. Let $D_1 + MD_2 > 0$. For any $K \geq 0$ and $M = 2B/\rho$ one can choose $\{\gamma_k\}_{k\geq 0}$ as follows:

$$\gamma_k = \frac{1}{h} \quad \text{if } K \leq \frac{h}{\mu} \text{ or } \left(K > \frac{h}{\mu} \text{ and } k < k_0\right),$$
$$\gamma_k = \frac{2}{\mu(\kappa + k - k_0)} \quad \text{if } K > \frac{h}{\mu} \text{ and } k \geq k_0,$$

where $h = \max\{2(A + 2BC/\rho), 2\mu/\rho\}$, $\kappa = 2h/\mu$ and $k_0 = \lceil K/2 \rceil$. For this choice of γ_k, the iterates of SGDA, given by (3.5), satisfy:

$$\mathbb{E}[V_K] \leq \frac{32hV_0}{\mu} \exp\left(-\frac{\mu}{h}K\right) + \frac{36(D_1 + 2BD_2/\rho)}{\mu^2 K}.$$

Monotone case. When $\mu = 0$, we additionally assume that F is monotone, i.e., for all $z, y \in \mathbb{R}^d$

$$\langle F(z) - F(y), z - y\rangle \geq 0.$$

Similar to minimization, in the case of $\mu = 0$, the squared distance to the solution is not a valid measure of convergence. To introduce an appropriate convergence measure, we make the following

assumption.

Assumption 3.1.4. There exists a compact convex set C (with the diameter $\Omega_C := \max_{z,y\in C} \|x-y\|$) such that $\mathcal{Z}^* \subset C$.

One can use Assumption 3.1.4 to correctly introduce $\text{Gap}(z)$ from (3.2).

Under these assumptions, Assumption 3.1.1, and star-cocoercivity, we derive the following general result.

Theorem 3.1.5. Let F be monotone, ℓ-star-cocoercive and let Assumptions 3.1.1, 3.1.4 hold. Assume that $0 < \gamma \leq 1/2(A+BC/\rho)$. Then, for all $K \geq 0$, the iterates of `SGDA`, given by (3.5), satisfy:

$$\begin{aligned}\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq & \frac{3\left[\max_{u\in\mathcal{C}} \|z^0-u\|^2\right]}{2\gamma K} + 9\gamma \max_{z^*\in\mathcal{Z}^*} \|F(z^*)\|^2 \\ & + \frac{8\gamma\ell^2\Omega_C^2}{K} + (4A+\ell+8BC/\rho)\cdot\frac{\|z^0-z^{*,0}\|^2}{K} \\ & + (4+(4A+\ell+8BC/\rho)\gamma)\frac{\gamma B\sigma_0^2}{\rho K} \\ & + \gamma(2+\gamma(4A+\ell+8BC/\rho))(D_1+2BD_2/\rho). \end{aligned} \tag{3.8}$$

See the proof in Appendix A.2.6. The above result establishes $\mathcal{O}(1/K)$ rate of convergence to the accuracy proportional to the stepsize γ multiplied by the noise term $D_1 + 2BD_2/\rho$ and $\max_{z^*\in\mathcal{Z}^*} \|F(z^*)\|^2$. We notice that if $g \equiv 0$ in (3.1), then $F(z^*) = 0$, meaning that in this case, the second term from (3.8) equals zero. Otherwise, even in the deterministic case, one needs to use small stepsizes to ensure convergence to any predefined accuracy – see the next corollary.

Corollary 3.1.6. Let the assumptions of Theorem 3.1.5 hold. Then, for all K one can choose γ as

$$\gamma = \min\left\{\frac{1}{4A+\ell+8BC/\rho}, \frac{\Omega_{0,C}\sqrt{\rho}}{\widehat{\sigma}_0\sqrt{B}}, \frac{\Omega_{0,C}}{\sqrt{K(D_1+2BD_2/\rho)}}, \frac{\Omega_{0,C}}{G_*\sqrt{K}}\right\},$$

where $\Omega_0 := \|z^0 - z^{*,0}\|^2$ and $\Omega_{0,C}$, $\widehat{\sigma}_0$, and G_* are some upper bounds for $\max_{u\in C}\|z^0-u\|$, σ_0, and $\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|$ respectively. This choice of γ implies $\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=1}^K z^k\right)\right]$ equals

$$\mathcal{O}\left(\frac{(A+\ell+BC/\rho)(\Omega_{0,C}^2+\Omega_0^2)+\ell\Omega_C^2}{K} + \frac{\Omega_{0,C}\widehat{\sigma}_0\sqrt{B}}{\sqrt{\rho}K} + \frac{\Omega_{0,C}(\sqrt{D_1+BD_2/\rho}+G_*)}{\sqrt{K}}\right).$$

Cocoercive case. The term proportional to $\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2$ can be removed if we assume that the operator F is not just monotone star-cocoercive (3.4), but also general cocoercive, i.e., it holds that for all $z, y \in \mathbb{R}^d$

$$\|F(z)-F(y)\|^2 \leq \ell\langle F(z)-F(y), z-y\rangle.$$

Theorem 3.1.7. Let F be ℓ-cocoercive and Assumptions 3.1.1, 3.1.4 hold. Assume that $0 < \gamma \leq \min\left\{1/\ell, 1/2(A+BC/\rho)\right\}$. Then for all $K \geq 0$ the iterates of SGDA, given by (3.5), satisfy:

$$\begin{aligned}\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq & \frac{3\max_{u\in\mathcal{C}}\|z^0-u\|^2}{2\gamma K} \\ & + (6A+3\ell+12BC/\rho)\cdot\frac{\|z^0-z^{*,0}\|^2}{K} \\ & + \left(6+(6A+3\ell+12BC/\rho)\,\gamma\right)\frac{\gamma B\sigma_0^2}{\rho K} \\ & + \gamma(3+\gamma\,(6A+3\ell+12BC/\rho))(D_1+2BD_2/\rho).\end{aligned}$$

See the proof in Appendix A.2.6. In contrast to Theorem 3.1.5, the above result implies $\mathcal{O}(1/K)$ convergence rate in the deterministic case. See Corollary 3.1.8 for the results of convergence with a selected stepsize.

Corollary 3.1.8. Let the assumptions of Theorem 3.1.7 hold. Then, for all K one can choose γ as

$$\gamma = \min\left\{\frac{1}{6A+3\ell+12BC/\rho}, \frac{\Omega_{0,C}\sqrt{\rho}}{\widehat{\sigma}_0\sqrt{B}}, \frac{\Omega_{0,C}}{\sqrt{K(D_1+2BD_2/\rho)}}\right\},$$

where $\Omega_0 := \|z^0 - z^{*,0}\|^2$ and $\Omega_{0,C}$, and $\widehat{\sigma}_0$ are some upper bounds for $\max_{u\in C}\|z^0-u\|$, and σ_0 respectively. This choice of γ implies $\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right]$ equals

$$\mathcal{O}\left(\frac{(A+\ell+BC/\rho)(\Omega_{0,C}^2+\Omega_0^2)}{K} + \frac{\Omega_{0,C}\widehat{\sigma}_0\sqrt{B}}{\sqrt{\rho}K} + \frac{\Omega_{0,C}\sqrt{D_1+BD_2/\rho}}{\sqrt{K}}\right).$$

3.1.4 Special Cases

SGDA with Arbitrary Sampling

We begin our consideration of special cases with a standard SGDA (3.5) with $g^k = F_{\xi^k}(z^k), \xi^k \sim \mathcal{D}$ under the so-called *expected cocoercivity* assumption from [258], which we properly adjust to the setting of regularized VIPs.

Algorithm 21 SGDA-AS: Stochastic Gradient Descent-Ascent with Arbitrary Sampling

Parameters: stepsize $\gamma > 0$
Initialization: choose $z^0 \in \mathbb{R}^d$, distribution $\mathcal{D}$

1: **for** $k = 0$ **to** $K-1$ **do**
2: Sample $\xi^k \sim \mathcal{D}$ independently from previous iterations and compute $g^k = F_{\xi^k}(z^k)$
3: $z^{k+1} = \text{prox}_{\gamma g}(z^k - \gamma g^k)$
4: **end for**

Assumption 3.1.9 (Expected Cocoercivity)**.** We assume that the stochastic operator $F_\xi(z), \xi \sim \mathcal{D}$ is such that for all $z \in \mathbb{R}^d$, $\mathbb{E}_{\mathcal{D}}\left[\|F_\xi(z) - F_\xi(z^*)\|^2\right] \leq \ell_{\mathcal{D}}\langle F(z) - F(z^*), z - z^*\rangle$, where $z^* =$

$\text{proj}_{\mathcal{Z}^*}(z)$.

When $g(z) \equiv 0$, this assumption recovers the original one from [258]. We also emphasize that for operator F Assumption 3.1.9 implies only star-cocoercivity.

Following [258], we mainly focus on the finite-sum case and its stochastic reformulation: we consider a random *sampling* vector $\xi = (\xi_1, \ldots, \xi_n)^\top \in \mathbb{R}^n$ having a distribution $\mathcal{D}$ such that $\mathbb{E}_{\mathcal{D}}[\xi_i] = 1$ for all $i \in [n]$. Using this, we can rewrite $F(z) = \frac{1}{n}\sum_{i=1}^n F_i(z)$ as

$$F(z) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\mathcal{D}}[\xi_i F_i(z)] = \mathbb{E}_{\mathcal{D}}\left[F_\xi(z)\right], \tag{3.9}$$

where $F_\xi(z) = \frac{1}{n}\sum_{i=1}^n \xi_i F_i(z)$. Such a reformulation allows handling a wide range of samplings: the only assumption on $\mathcal{D}$ is $\mathbb{E}_{\mathcal{D}}[\xi_i] = 1$ for all $i \in [n]$. Therefore, this setup is often referred to as *arbitrary sampling* [268, 269, 270, 271, 272, 273, 274, 275].

Next, we give the particular examples of arbitrary sampling fitting this setup. In all the examples below, we focus on a special case of stochastic reformulation from (3.9) and assume that for all $i \in [n]$ operator F_i is (ℓ_i, X^*)-cocoercive, i.e., for all $i \in [n]$ and $z \in \mathbb{R}^d$ we have

$$\|F_i(z) - F_i(z^*)\|^2 \le \ell_i \langle F_i(z) - F_i(z^*), z - z^* \rangle, \tag{3.10}$$

where z^* is the projection of z on $\mathcal{Z}^*$. Note that (3.10) holds whenever F_i are cocoercive.

Uniform Sampling. We start with the classical uniform sampling: let $\mathbb{P}\{\xi = ne_i\} = 1/n$ for all $i \in [n]$, where $e_i \in \mathbb{R}^n$ is the i-th coordinate vector from the standard basis in $\mathbb{R}^n$. Then, $\mathbb{E}[\xi_i] = 1$ for all $i \in [n]$ and Assumption 3.1.9 holds with $\ell_{\mathcal{D}} = \max_{i\in[n]} \ell_i$:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\|F_\xi(z) - F_\xi(z^*)\|^2\right] &= \frac{1}{n}\sum_{i=1}^n \|F_i(z) - F_i(z^*)\|^2 \\
&\overset{(3.10)}{\le} \frac{1}{n}\sum \left(\ell_i \langle F_i(z) - F_i(z^*), z - z^* \rangle\right) \\
&\le \max_{i\in[n]} \ell_i \langle F(z) - F(z^*), z - z^* \rangle
\end{aligned}$$

Importance Sampling. Next, we consider a non-uniform sampling strategy – importance sampling: let $\mathbb{P}\left\{\xi = e_i n\bar{\ell}/\ell_i\right\} = \ell_i/n\bar{\ell}$ for all $i \in [n]$, where $\bar{\ell} = \frac{1}{n}\sum_{i=1}^n \ell_i$. Then, $\mathbb{E}[\xi_i] = 1$ for all $i \in [n]$ and Assumption 3.1.9 holds with $\ell_{\mathcal{D}} = \bar{\ell}$:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\|F_\xi(z) - F_\xi(z^*)\|^2\right] &= \sum_{i=1}^n \frac{\ell_i}{n\bar{\ell}} \left\| \frac{\bar{\ell}}{\ell_i}\left(F_i(z) - F_i(z^*)\right)\right\|^2 \\
&= \sum_{i=1}^n \frac{\bar{\ell}}{n\ell_i} \|F_i(z) - F_i(z^*)\|^2 \\
&\overset{(3.10)}{\le} \frac{\bar{\ell}}{n}\sum \langle F_i(z) - F_i(z^*), z - z^* \rangle
\end{aligned}$$

$$\leq \quad \overline{\ell}\langle F(z) - F(z^*), z - z^*\rangle$$

We emphasize that $\overline{\ell} \leq \max_{i\in[n]} \ell_i$ and, in fact, $\overline{\ell}$ might be much smaller than $\max_{i\in[n]} \ell_i$. Moreover, $\sigma^2_{*,\text{IS}} \leq \sigma^2_{*,\text{US}}$, when $\max_{x^*\in X^*} \|F_i(x^*)\| \sim \ell_i$. Therefore, compared to `SGDA` with uniform sampling, `SGDA` with importance sampling can be better.

Minibatch Sampling With Replacement. Let $\xi = \frac{1}{b}\sum_{i=1}^{b} \xi^i$, where ξ^i are i.i.d. samples from some distribution $\mathcal{D}$ satisfying (3.9) and Assumption 3.1.9. Then, the distribution of ξ satisfies (3.9) and Assumption 3.1.9 as well with the same constant $\ell_{\mathcal{D}}$. Therefore, minibatched versions of uniform sampling and importance sampling fit the framework as well with $\ell_{\mathcal{D}} = \max_{i\in[n]} \ell_i, \sigma^2_* = \frac{\sigma^2_{*,\text{US}}}{b}$ and $\ell_{\mathcal{D}} = \overline{\ell}, \sigma^2_* = \frac{\sigma^2_{*,\text{IS}}}{b}$.

Minibatch Sampling Without Replacement. For a given batch size $b \in [n]$, we consider the following sampling strategy: for each subset $S \subseteq [n]$ such that $|S| = b$, we have $\mathbb{P}\left\{\xi = \frac{n}{b}\sum_{i\in S} e_i\right\} = \frac{b!(n-b)!}{n!}$, i.e., S is chosen uniformly at random from all b-element subsets of $[n]$. In the special case when $g(z) \equiv 0$, [258] show that this sampling strategy satisfies (3.9) and Assumption 3.1.9 with

$$\ell_{\mathcal{D}} = \frac{n(b-1)}{b(n-1)}\ell + \frac{n-b}{b(n-1)}\max_{i\in[n]} \ell_i, \quad \sigma^2_* = \frac{n-b}{b(n-1)}\sigma^2_{*,\text{US}}.$$

Clearly, both parameters are smaller than the corresponding parameters for the minibatched version of uniform sampling with replacement, indicating the theoretical benefits of sampling without replacement.

Now we want to show that `SGDA` with Arbitrary Sampling (`SGDA-AS`) fits our unified framework.

Proposition 3.1.10. Let Assumption 3.1.9 hold. Then, `SGDA-AS` satisfies Assumption 3.1.1 with $A = \ell_{\mathcal{D}}$, $D_1 = 2\sigma^2_* := 2\max_{z^*\in\mathcal{Z}^*} \mathbb{E}_{\mathcal{D}}\left[\|F_\xi(z^*) - F(z^*)\|^2\right]$, $B = 0$, $\sigma^2_k \equiv 0$, $C = 0$, $\rho = 1$, $D_2 = 0$.

Plugging these parameters into Theorem 3.1.2 we recover the result from [258] when $g(z) \equiv 0$ and generalize it to the case of $g(z) \not\equiv 0$ without sacrificing the rate. Applying Corollary 3.1.3, we establish the rate of convergence to the exact solution.

Corollary 3.1.11. Let F be μ-quasi-strongly monotone, and let Assumption 3.1.9 hold. Then, for all $K > 0$, there exists a choice of γ for which the iterates of `SGDA-AS`, satisfy:

$$\mathbb{E}[\|z^K - z^{*,K}\|^2] = \mathcal{O}\left(\frac{\ell_{\mathcal{D}}\Omega_0^2}{\mu}\exp\left(-\frac{\mu}{\ell_{\mathcal{D}}}K\right) + \frac{\sigma^2_*}{\mu^2 K}\right),$$

where $\Omega_0^2 = \|z^0 - z^{*,0}\|^2$.

For the different stepsize schedule, [258] derive the convergence rate $\mathcal{O}(1/K + 1/K^2)$ which is inferior to our rate, especially when σ^2_* is small. In addition, [258] consider explicitly only uniform minibatch sampling without replacement as a special case of arbitrary sampling. In Section 3.1.6, we provide

numerical experiments that verify our theoretical findings and show the benefits of importance sampling over uniform sampling for SGDA.

SGDA with Variance Reduction

In this point, we focus on variance-reduced variants of SGDA for solving finite-sum problems $F(z) = \frac{1}{n}\sum_{i=1}^n F_i(z)$. We start with the Loopless Stochastic Variance Reduced Gradient Descent-Ascent (L-SVRGDA), which is a generalization of the L-SVRG algorithm proposed in [263, 264]. L-SVRGDA (see Algorithm 22) follows the update rule (3.5) with

Algorithm 22 L-SVRGDA: Loopless Stochastic Variance Reduced Gradient Descent-Ascent

Parameters: stepsize $\gamma > 0$, probability p
Initialization: choose $w^0 = z^0 \in \mathbb{R}^d$, distribution $\mathcal{D}$

1: **for** $k = 0$ **to** $K - 1$ **do**
2: Draw j_k from the uniform distribution on $[n]$ and compute $g^k = F_{j_k}(z^k) - F_{j_k}(w^k) + F(w^k)$
3: $w^{k+1} = \begin{cases} z^k, & \text{with probability } p, \\ w^k, & \text{with probability } 1-p, \end{cases}$
4: $x^{k+1} = \text{prox}_{\gamma g}(z^k - \gamma g^k)$
5: **end for**

Here, the full operator F is computed once w^k is updated, which occurs with probability p. Typically, p is chosen as $p \sim 1/n$ ensuring that the expected cost of 1 iteration equals $\mathcal{O}(1)$ oracle calls, i.e., computations of $F_i(z)$ for some $i \in [n]$.

We introduce the following assumption about operators F_i.

Assumption 3.1.12 (Averaged Star-Cocoercivity)**.** We assume that there exists a constant $\widehat{\ell} > 0$ such that for all $x \in \mathbb{R}^d$

$$\frac{1}{n}\sum_{i=1}^n \|F_i(z) - F_i(z^*)\|^2 \leq \widehat{\ell}\langle F(z) - F(z^*), z - z^*\rangle, \tag{3.11}$$

where $z^* = \text{proj}_{\mathcal{Z}^*}(z)$.

For example, if F_i is ℓ_i-cocoercive for $i \in [n]$, then (3.11) holds with $\widehat{\ell} \leq \max_{i\in[n]} \ell_i$. Next, if F_i is L_i-Lipschitz for all $i \in [n]$ and F is μ-quasi strongly monotone, then (3.11) is satisfied for $\widehat{\ell} \in [\overline{L}, \overline{L}^2/\mu]$, where $\overline{L}^2 = \frac{1}{n}\sum_{i=1}^n L_i^2$.

Moreover, for the analysis of variance-reduced variants of SGDA we also use the uniqueness of the solution.

Assumption 3.1.13 (Unique Solution)**.** We assume that the solution set $\mathcal{Z}^*$ of problem (3.1) is a singleton: $\mathcal{Z}^* = \{z^*\}$.

These assumptions are sufficient to derive the validity of Assumption 3.1.1 for L-SVRGDA estimator.

Proposition 3.1.14. Let Assumptions 3.1.12 and 3.1.13 hold. Then, `L-SVRGDA` satisfies Assumption 3.1.1 with $A = \widehat{\ell}$, $B = 2$, $\sigma_k^2 = \frac{1}{n}\sum_{i=1}^n \|F_i(w^k) - F_i(z^*)\|^2$, $C = p\widehat{\ell}/2$, $\rho = p$, $D_1 = D_2 = 0$.

Plugging these parameters into our general results on the convergence of `SGDA`-type algorithms, we derive the convergence results for `L-SVRGDA`, see Table 3.1.

Next, we demonstrate that `SAGA-SGDA` [227] (Algorithm 23) fits our framework, and by utilizing our general analysis, we tighten the convergence rates for this method (Table 3.1).

Algorithm 23 `SAGA-SGDA` [227]

Parameters: stepsize $\gamma > 0$
Initialization: choose $w_i^0 = z^0 \in \mathbb{R}^d$, distribution $\mathcal{D}$
for $k = 0$ **to** $K - 1$ **do**
  Draw j_k from the uniform distribution on $[n]$ and compute $g^k = F_{j_k}(z^k) - F_{j_k}(w_{j_k}^k) + \frac{1}{n}\sum_{i=1}^n F_i(w_i^k)$
  Set $w_{j_k}^{k+1} = x^k$ and $w_i^{k+1} = w_i^k$ for $i \neq j_k$
  $z^{k+1} = \text{prox}_{\gamma g}(z^k - \gamma g^k)$
end for

Proposition 3.1.15. Let Assumptions 3.1.12 and 3.1.13 hold. Then, `SAGA-SGDA` satisfies Assumption 3.1.1 with $A = \widehat{\ell}$, $B = 2$, $\sigma_k^2 = \frac{1}{n}\sum_{i=1}^n \|F_i(w_i^k) - F_i(z^*)\|^2$, $C = \widehat{\ell}/2n$, $\rho = 1/n$, $D_1 = D_2 = 0$.

We compare our convergence guarantees with known results in Table 3.1. We note that by neglecting the importance sampling scenario, in the worst case, our convergence results match the best-known results for `SGDA`-type methods, i.e., ones derived in [227]. Indeed, this follows from $\widehat{\ell} \in [\overline{L}, \overline{L}^2/\mu]$. Next, when the difference between $\overline{\ell}$ and $\widehat{\ell}$ is not significant, our complexity results match the one derived in [100] for SVRE, which is EG-type method. Although in general, $\overline{\ell}$ might be smaller than $\widehat{\ell}$, our analysis does not require cocoercivity of each F_i and it works for $g(z) \not\equiv 0$. Finally, [214] derives a better rate (when $n = \mathcal{O}(\overline{L}^2/\mu^2)$), but their method is based on EG. Therefore, our results match the best-known ones in the literature on `SGDA`-type methods.

Distributed `SGDA` with Compression

In this point, we consider the distributed version of (3.1), i.e., we assume that $F(z) = \frac{1}{n}\sum_{i=1}^n F_i(z)$, where $\{F_i\}_{i=1}^n$ are distributed across n devices connected with a parameter-server in a centralized fashion. Each device i has access to the computation of the unbiased estimate of F_i at the given point. Typically, in these settings, communication is a bottleneck, especially when n and d are huge. This means that in the naive distributed implementations of `SGDA`, communication rounds take much more time than local computations on the clients. Various approaches are used to circumvent this issue.

One of them is based on the usage of compressed communications. We focus on unbiased compression operators.

Table 3.1: Summary of the complexity results for variance reduced methods for solving (3.1). By complexity we mean the number of oracle calls required for the method to find z such that $\mathbb{E}[\|z - z^*\|^2] \leq \varepsilon$. Dependencies on numerical and logarithmic factors are hidden. By default, operator F is assumed to be μ-strongly monotone and, as the result, the solution is unique. Our results rely on μ-quasi strong monotonicity of F (3.3), but we also assume uniqueness of the solution.

Method	Assumptions	Complexity
SVRE [100] (1)	F_i is ℓ_i-cocoer.	$n + \frac{\overline{\ell}}{\mu}$
EG-VR [214] *(1)	F_i is L_i-Lip.	$n + \sqrt{n}\frac{\overline{L}}{\mu}$
SVRGDA [227] *	F_i is L_i-Lip.	$n + \frac{\overline{L}^2}{\mu^2}$
SAGA-SGDA [227] *	F_i is L_i-Lip.	$n + \frac{\overline{L}^2}{\mu^2}$
VR-AGDA [218]	F_i is $L_{\max}$-Lip.(2)	$\min\left\{n + \frac{L_{\max}^9}{\mu^9}, n^{2/3}\frac{L_{\max}^3}{\mu^3}\right\}$
`L-SVRGDA` (this thesis) *	As. 3.1.12	$n + \frac{\widehat{\ell}}{\mu}$
`SAGA-SGDA` (this thesis) *	As. 3.1.12	$n + \frac{\widehat{\ell}}{\mu}$

Methods supporting $g(z) \not\equiv 0$ are highlighted with *.
Notation: $\overline{\ell}$, $\overline{L}$ = averaged cocoercivity/Lipschitz constants depending on the sampling strategy, e.g., for uniform sampling $\overline{\ell}^2 = \frac{1}{n}\sum_{i=1}^n \ell_i^2$, $\overline{L}^2 = \frac{1}{n}\sum_{i=1}^n L_i^2$ and for importance sampling $\overline{\ell} = \frac{1}{n}\sum_{i=1}^n \ell_i$, $\overline{L} = \frac{1}{n}\sum_{i=1}^n L_i$; $\widehat{\ell}$ = averaged star-cocoercivity constant from Assumption 3.1.12.
(1) The method is based on Extragradient update rule.
(2) [218] consider saddle point problems satisfying so-called two-sided PŁ condition, which is weaker than strong-convexity-strong-concavity of the objective function.

Definition 3.1.16. An operator $Q : \mathbb{R}^d \to \mathbb{R}^d$ (possibly randomized) is called *unbiased compressor/quantization* if there exists a constant $\omega \geq 0$ such that for all $x \in \mathbb{R}^d$

$$\mathbb{E}[Q(x)] = x, \quad \mathbb{E}[\|Q(x) - x\|^2] \leq \omega\|x\|^2. \tag{3.12}$$

In this section, we consider compressed communications in the direction from clients to the server. The simplest method with compression – `QSGDA` (Algorithm 24) – can be described as `SGDA` (3.5) with $g^k = \frac{1}{n}\sum_{i=1}^n Q(g_i^k)$. Here g_i^k are stochastic estimators satisfying the following assumption.

Assumption 3.1.17 (Bounded variance)**.** All stochastic realizations g_i^k are unbiased and have

Algorithm 24 QSGDA: Quantized Stochastic Gradient Descent-Ascent

Parameters: stepsize $\gamma > 0$
Initialization: choose $z^0 \in \mathbb{R}^d$, distribution $\mathcal{D}$

for $k = 0$ **to** $K-1$ **do**
  Broadcast z^k to all workers
  for $i = 1, \ldots, n$ in parallel **do**
    Compute g_i^k and send $Q(g_i^k)$ to the server
  end for
  $g^k = \frac{1}{n}\sum_{i=1}^n Q(g_i^k)$
  $z^{k+1} = \text{prox}_{\gamma g}\left(z^k - \gamma g^k\right)$
end for

bounded variance, i.e., for all $i \in [n]$ and $k \geq 0$ the following holds:

$$\mathbb{E}[g_i^k] = F_i(z^k), \quad \mathbb{E}[\|g_i^k - F_i(z^k)\|^2] \leq \sigma_i^2.$$

Despite its simplicity, QSGDA was never considered in the literature on solving min-max problems and VIPs. It turns out that under such assumptions QSGDA satisfies our Assumption 3.1.1.

Proposition 3.1.18. Let F be ℓ-star-cocoercive and Assumptions 3.1.12, 3.1.17 hold. Then, QSGDA satisfies Assumption 3.1.1 with $A = \frac{3\ell}{2} + \frac{9\omega\widehat{\ell}}{2n}$, $B = 0$, $\sigma_k^2 \equiv 0$, $D_1 = \frac{3(1+3\omega)\sigma^2 + 9\omega\zeta_*^2}{n}$, $C = 0$, $\rho = 1$, $D_2 = 0$, where $\sigma^2 = \frac{1}{n}\sum_{i=1}^n \sigma_i^2, \zeta_*^2 := \frac{1}{n}\max_{z^* \in \mathcal{Z}^*}\sum_{i=1}^n \|F_i(z^*)\|^2$.

As for the other special cases, we derive the convergence results for QSGDA using our general theorems (see Table 3.2). The proposed method is simple, but has a significant drawback: even in the deterministic case ($\sigma = 0$), QSGDA does not converge linearly unless $\zeta_*^2 = 0$. However, when the data on clients is arbitrarily heterogeneous, the dissimilarity measure ζ_*^2 is strictly positive and can be large (even when $g(z) \equiv 0$).

To resolve this issue, we propose a more advanced scheme based on DIANA update [60, 265] – DIANA-SGDA (Algorithm 25). In a nutshell, DIANA-SGDA is SGDA (3.5) with g^k defined as follows: where the first two lines correspond to the local computations on the clients and the last two lines – to the server-side computations. Taking into account the update rule for h^{k+1}, one can notice that DIANA-SGDA requires workers to send only vectors $Q(\Delta_i^k)$ to the server at step k, i.e., the method uses only compressed workers-server communications.

As we show next, DIANA-SGDA fits our framework.

Proposition 3.1.19. Let Assumptions 3.1.12, 3.1.13, 3.1.17 hold. Suppose that $\alpha \leq 1/(1+\omega)$. Then, DIANA-SGDA with quantization (3.12) satisfies Assumption 3.1.1 with $\sigma_k^2 = \frac{1}{n}\sum_{i=1}^n \|h_i^k - F_i(z^*)\|^2$ and $A = \left(\frac{1}{2} + \frac{\omega}{n}\right)\widehat{\ell}$, $B = \frac{2\omega}{n}$, $D_1 = \frac{(1+\omega)\sigma^2}{n}$, $C = \frac{\alpha\widehat{\ell}}{2}$, $\rho = \alpha$, $D_2 = \alpha\sigma^2$, where $\sigma^2 = \frac{1}{n}\sum_{i=1}^n \sigma_i^2$.

DIANA-SGDA can be considered as a variance-reduced method since it reduces the term proportional to $\omega\zeta_*^2$ that the bound for QSGDA contains (see Table 3.2). As a result, when $\sigma = 0$, i.e., workers compute $F_i(z)$ at each step, DIANA-SGDA enjoys linear convergence to the exact solution.

Algorithm 25 `DIANA-SGDA`: `DIANA` Stochastic Gradient Descent-Ascent [60, 265]

Parameters: stepsizes $\gamma, \alpha > 0$
Initialization: choose $z^0 \in \mathbb{R}^d$, $h_i^0 = 0$, $h^0 = \frac{1}{n}\sum_{i=1}^n h_i^0$

for $k = 0$ **to** $K-1$ **do**
  Broadcast z^k to all workers
  for $i = 1, \ldots, n$ in parallel **do**
    Compute g_i^k and $\Delta_i^k = g_i^k - h_i^k$
    Send $Q(\Delta_i^k)$ to the server
    $h_i^{k+1} = h_i^k + \alpha Q(\Delta_i^k)$
  end for
  $g^k = h^k + \frac{1}{n}\sum\limits_{i=1}^n Q(\Delta_i^k) = \frac{1}{n}\sum\limits_{i=1}^n (h_i^k + Q(\Delta_i^k))$
  $z^{k+1} = \text{prox}_{\gamma g}\left(z^k - \gamma g^k\right)$
  $h^{k+1} = h^k + \alpha\frac{1}{n}\sum\limits_{i=1}^n Q(\Delta_i^k) = \frac{1}{n}\sum\limits_{i=1}^n h_i^k$
end for

Next, when local operators F_i have a finite-sum form $F_i(z) = \frac{1}{m}\sum_{j=1}^m F_{ij}(z)$, one can combine `L-SVRGDA` and `DIANA-SGDA` as follows:

Algorithm 26 `VR-DIANA-SGDA`: `VR-DIANA` Stochastic Gradient Descent-Ascent [265]

Parameters: stepsizes $\gamma, \alpha > 0$, probability p
Initialization: choose $z^0 \in \mathbb{R}^d$, $h_i^0 = 0$, $h^0 = \frac{1}{n}\sum_{i=1}^n h_i^0$

for $k = 0$ **to** $K-1$ **do**
  Broadcast z^k to all workers
  for $i = 1, \ldots, n$ in parallel **do**
    Draw j_i^k from the uniform distribution on $[m]$ and compute $g_i^k = F_{ij_i^k}(z^k) - F_{ij_i^k}(w_i^k) + F_i(w_i^k)$
    $w_i^{k+1} = \begin{cases} x^k, & \text{with probability } p, \\ w_i^k, & \text{with probability } 1-p, \end{cases}$
    $\Delta_i^k = g_i^k - h_i^k$
    Send $Q(\Delta_i^k)$ to the server
    $h_i^{k+1} = h_i^k + \alpha Q(\Delta_i^k)$
  end for
  $g^k = h^k + \frac{1}{n}\sum\limits_{i=1}^n Q(\Delta_i^k) = \frac{1}{n}\sum\limits_{i=1}^n (h_i^k + Q(\Delta_i^k))$
  $x^{k+1} = \text{prox}_{\gamma g}\left(z^k - \gamma g^k\right)$
  $h^{k+1} = h^k + \alpha\frac{1}{n}\sum\limits_{i=1}^n Q(\Delta_i^k) = \frac{1}{n}\sum\limits_{i=1}^n h_i^k$
end for

We call the resulting method `VR-DIANA-SGDA` (Algorithm 26) and we note that its analog for solving minimization problems (`VR-DIANA`) was proposed and analyzed in [265].

To cast `VR-DIANA-SGDA` as a special case of our general framework, we need to make the following assumption.

Assumption 3.1.20. We assume that there exists a constant $\widetilde{\ell} > 0$ such that for all $z \in \mathbb{R}^d$

$$\frac{1}{nm} \sum_{i,j=1,1}^{n,m} \|F_{ij}(z) - F_{ij}(z^*)\|^2 \leq \widetilde{\ell} \langle F(z) - F(z^*), z - z^* \rangle,$$

where $z^* = \text{proj}_{\mathcal{Z}^*}(z)$.

Using Assumption 3.1.20 and the previously introduced conditions, we obtain the following result.

Proposition 3.1.21. Let F be ℓ-star-cocoercive and Assumptions 3.1.12, 3.1.13, 3.1.20 hold. Suppose that $\alpha \leq \min\left\{\frac{p}{3}, \frac{1}{1+\omega}\right\}$. Then, `VR-DIANA-SGDA` satisfies Assumption 3.1.1 with $A = \frac{\ell}{2} + \frac{\widetilde{\ell}}{n} + \frac{\omega(\widehat{\ell}+\widetilde{\ell})}{n}$, $B = \frac{2(\omega+1)}{n}, \sigma_k^2 = \frac{1}{n}\sum_{i=1}^n \|h_i^k - F_i(z^*)\|^2 + \frac{1}{nm}\sum_{i=1}^n \sum_{j=1}^m \|F_{ij}(w_i^k) - F_{ij}(z^*)\|^2$, $C = \frac{p\widetilde{l}}{2} + \alpha(\widetilde{\ell} + \widehat{\ell})$, $\rho = \alpha$, $D_1 = D_2 = 0$.

Since $D_1 = D_2 = 0$, our general results imply linear convergence of `VR-DIANA-SGDA` when $\mu > 0$. That is, `VR-DIANA-SGDA` is the *first linearly converging distributed* `SGDA`*-type method with compression*. We compare it with `MASHA1` from Chapter 2 in Table 3.2. Firstly, let us note that `MASHA1` is a method based on `EG`, and its convergence guarantees depend on the Lipschitz constants. In addition, we note that the complexity of `MASHA1` could be better than the one of `VR-DIANA-SGDA` when cocoercivity constants are large compared to Lipschitz ones. However, our complexity bound has better dependency on the quantization parameter ω, number of clients n, and the size of the local dataset m. These parameters can be large, indicating that the improvement is noticeable.

Coordinate `SGDA`

In this point, we focus on the coordinate versions of `SGDA`. To denote the i-th component of the vector z, we use $[z]_i$. We start from the simple `CSGDA`

Algorithm 27 `CSGDA`: Coordinate Stochastic Gradient Descent-Ascent

Parameters: stepsizes $\gamma > 0$
Initialization: choose $z^0 \in \mathbb{R}^d$
for $k = 0$ **to** $K - 1$ **do**
  Sample uniformly at random $j \in [d]$
  $g^k = d e_j [F(z^k)]_j$
  $z^{k+1} = \text{prox}_{\gamma g}\left(z^k - \gamma g^k\right)$
end for

Proposition 3.1.22. Let F be ℓ-star-cocoercive. Then, `CSGDA` satisfies Assumption 3.1.1 with $A = d\ell$, $D_1 = 2d \max_{z* \in \mathcal{Z}^*}\left[\|F(z^*)\|^2\right]$, $\sigma_k^2 = 0$, $B = 0$, $C = 0$, $\rho = 1$, $D_2 = 0$.

Next, we consider a modification of `SEGA` [59] – the linearly converging coordinate method for composite optimization problems that works even for non-separable regularizers.

Table 3.2: Summary of the complexity results for distributed methods with unbiased compression for solving distributed (3.1) with $F(z) = \frac{1}{n}\sum_{i=1}^n F_i(z)$. By complexity we mean the number of communication rounds required for the method to find z such that $\mathbb{E}[\|z - z^*\|^2] \leq \varepsilon$. Dependencies on numerical and logarithmic factors are hidden. Our results rely on μ-quasi strong monotonicity of F (3.3), but we also assume the uniqueness of the solution.

Setup	Method	Assumptions	Complexity
$\mathbb{E}$	`QSGDA` (this thesis) *	As. 3.1.12, 3.1.17	$\frac{\ell}{\mu} + \frac{\omega\widehat{\ell}}{n\mu} + \frac{(1+\omega)\sigma^2 + \omega\zeta_*^2}{n\mu^2\varepsilon}$
	`DIANA-SGDA` (this thesis) *	As. 3.1.12, 3.1.17	$\omega + \frac{\ell}{\mu} + \frac{\omega\widehat{\ell}}{n\mu} + \frac{(1+\omega)\sigma^2}{n\mu^2\varepsilon}$
Σ	`MASHA1` (this thesis, Ch. 2) *(1)	F_i is L_i-Avg. Lip. (2)	$m + \omega + \frac{L_{\max}\sqrt{(m+\omega)\left(1+\frac{\omega}{n}\right)}}{\mu}$
	`VR-DIANA-SGDA` (this thesis) *	As. 3.1.12, 3.1.20	$m + \omega + \frac{\ell}{\mu} + \frac{(1+\omega)(\widehat{\ell}+\widetilde{\ell})}{n\mu} + \frac{(1+\omega)\max\{m,\omega\}\widetilde{\ell}}{nm\mu}$

Notation: $\mathbb{E}$ stands for the setup, when $F_i(z) = \mathbb{E}_{\xi_i}[F_{\xi_i}(z)]$; Σ denotes the case, when $F_i(z) = \frac{1}{m}\sum_{j=1}^m F_{ij}(z)$. Methods supporting $g(z) \not\equiv 0$ are highlighted with *.
$\sigma^2 = \frac{1}{n}\sum_{i=1}^n \sigma_i^2$ – averaged upper bound for the variance (see Ass. 3.1.17 for the definition of σ_i^2); ω = quantization parameter (see Def. 3.1.16); $\zeta_*^2 = \frac{1}{n}\max_{z^*\in\mathcal{Z}^*}\sum_{i=1}^n \|F_i(z^*)\|^2$; $L_{\max} = \max_{i\in[n]} L_i$; $\widetilde{\ell}$ = averaged star-cocoercivity constant from Ass. 3.1.20.
(1) The method is based on Extragradient update rule.
(2) This means that for all $z, y \in \mathbb{R}^d$ and $i \in [n]$ the following inequality holds: $\frac{1}{m}\sum_{j=1}^m \|F_{ij}(z) - F_{ij}(y)\|^2 \leq L_i^2\|z - y\|^2$.

Algorithm 28 `SEGA-SGDA`: `SEGA` Stochastic Gradient Descent-Ascent [59]

Parameters: stepsizes $\gamma, \alpha > 0$, probability p
Initialization: choose $z^0 \in \mathbb{R}^d$, $h^0 = 0$

1: **for** $k = 0$ **to** $K - 1$ **do**
2: Sample uniformly at random $j \in [d]$
3: $h^{k+1} = h^k + e_j([F(z^k)]_j - h_j^k)$
4: $g^k = de_j([F(x^k)]_j - h_j^k) + h^k$
5: $z^{k+1} = \text{prox}_{\gamma g}\left(z^k - \gamma g^k\right)$
6: **end for**

Proposition 3.1.23. Let F be ℓ-star-cocoercive and Assumption 3.1.13 holds. Then, `SEGA-SGDA` satisfies Assumption 3.1.1 with $\sigma_k^2 = \|h^k - F(x^*)\|^2$ and $A = d\ell$, $B = 2d$, $D_1 = 0$, $C = \frac{\ell}{2d}$, $\rho = \frac{1}{d}$, $D_2 = 0$.

The summary of rates in the (quasi-) strongly monotone case is provided in Table 3.3. First of all, our results are the first convergence for solving *regularized* VIPs via coordinate methods. In particular, `SEGA-SGDA` is the first linearly converging coordinate method for solving regularized VIPs. Next, when $q = 2$ in `zoVIA` from [261], i.e., Euclidean proximal setup is used, our rate for `SEGA-SGDA` is better than the one derived for `zoVIA` in [261] since $\ell \leq L^2/\mu$. Finally, `zoscESVIA` might achieve a better rate, but it is based on `EG` and it uses approximation of each component of

the operator F at each iteration, which makes one iteration of the method costly.

Table 3.3: Summary of the complexity results for zeroth-order methods with two-points feedback oracles for solving (3.1). By complexity we mean the number of oracle calls required for the method to find z such that $\mathbb{E}[\|z - z^*\|^2] \leq \varepsilon$. By default, operator F is assumed to be μ-strongly monotone and, as the result, the solution is unique. Our results rely on μ-quasi strong monotonicity of F (3.3).

Method	Assumptions	Complexity
`zoscESVIA` [261] [(1)]	F is L-Lip.[(2)]	$\widetilde{\mathcal{O}}\left(d\frac{L}{\mu}\right)$
`zoVIA` [261]	F is L-Lip.[(2)]	$\widetilde{\mathcal{O}}\left(d^{2/q}\frac{L^2}{\mu^2}\right)$
`CSGDA` (this thesis) *	F is ℓ-cocoer.	$\widetilde{\mathcal{O}}\left(d\frac{\ell}{\mu} + \frac{dG_*^2}{\mu^2\varepsilon}\right)$
`SEGA-SGDA` (this thesis) *	F is ℓ-cocoer. As. 3.1.13	$\widetilde{\mathcal{O}}\left(d + d\frac{\ell}{\mu}\right)$

Notation: Methods supporting $g(z) \not\equiv 0$ are highlighted with *.
q = the parameter depending on the proximal setup, $q = 2$ in Euclidean case and $q = +\infty$ in the ℓ_1-proximal setup; $G_* = \max_{z^* \in \mathcal{Z}^*} \|F(z^*)\|$, which is zero when $g(z) \equiv 0$.
[(1)] The method is based on Extragradient update rule. Moreover, at each step full operator is approximated.
[(2)] The problem is defined on a bounded set.

3.1.5 Beyond Unified Assumptions

The purpose of this subsection is to show that there are other methods beyond the analysis we was done above. This is because the unified assumption does not work for them. That said, they can also be analyzed in an elegant manner. We consider the simple version of (3.1) with $g(z) \equiv 0$:

$$\text{Find} \quad z^* \in \mathbb{R}^d \quad \text{such that} \quad F(z^*) = 0.$$

We already considered SAGA and SVRG in combination with `SGDA` above. For them, it is fulfilled that $\mathbb{E}_k\left[g^k\right] = F(z^k)$. This is not valid for all variance-reduced methods and there are other methods [53, 276, 53, 277, 278, 54] that use v^k instead of g^k, for which $\mathbb{E}_k\left[g^k\right] \neq F(z^k)$. Thus, these methods are often better in practice than analogues with unbiased g^k. In particular, we base our method on the original SARAH [53].

Algorithm 29 `SARAH-SGDA` [53]

Parameters: stepsize $\gamma > 0$, number of inner iterations T
Initialization: choose $z_0^0 \in \mathbb{R}^d$

for $k = 0, 1, 2, \ldots, K-1$ **do**
  $v_0^k = F(z_0^k)$
  $z_1^k = z_0^k - \gamma v_0^k$
  for $t = 1, 2, \ldots, T-1$ **do**
    Draw i_t^k from the uniform distribution on $[n]$
    $v_t^k = F_{i_t^k}(z_t^k) - F_{i_t^k}(z_{t-1}^k) + v_{t-1}^{k1}$
    $z_{t+1}^k = z_t^k - \gamma v_t^k$
  end for
  $z_0^{k+1} = z_T^k$
end for

Next, we analyze the convergence of this method. We use the following assumption, which differs slightly from (3.3) and (3.4)

Assumption 3.1.24 (Cocoercivity)**.** Each operator F_i is ℓ-cocoercive, i.e. for all $z, y \in \mathbb{R}^d$ we have

$$\|F_i(z) - F_i(y)\|^2 \leq \ell \langle F_i(z) - F_i(y), z - y \rangle.$$

Assumption 3.1.25 (Strong monotonicity)**.** The operator F is μ-strongly monotone, i.e. for all $u, v \in \mathbb{R}^d$ we have

$$\langle F(z) - F(y); z - y \rangle \geq \mu \|z - y\|^2.$$

Theorem 3.1.26. Suppose that Assumptions 3.1.24 and 3.1.25 hold. Consider Algorithm 29 with $\gamma = \frac{2}{9\ell}$ and $T = \frac{10\ell}{\mu}$. Then, we have

$$\mathbb{E}[\|F(z^k)\|^2] \leq \frac{1}{2}\mathbb{E}[\|F(z^{k-1})\|^2].$$

Since we need to find a point z such that $F(z) \approx F(z^*) = 0$, we can easily obtain an estimate on the oracle complexity (number of F_i calls) required to achieve precision ε.

Corollary 3.1.27. Suppose that Assumptions 3.1.24 and 3.1.25 hold. Consider SARAH (Algorithm 30) with $\gamma = \frac{2}{9\ell}$ and $T = \frac{10\ell}{\mu}$. Then, to achieve ε-solution ($\mathbb{E}\|F(z^K)\|^2 \sim \varepsilon^2$), we need

$$\mathcal{O}\left(\left[n + \frac{\ell}{\mu}\right] \log_2 \frac{\|F(z^0)\|^2}{\varepsilon^2}\right) \quad \text{oracle calls.}$$

Note that the obtained oracle complexity coincides with the similar complexity for `L-SVRGDA` from Table 3.1.

3.1.6 Experiments

We consider the special case of (3.1) with F and g defined as follows:

$$F(z) = \frac{1}{n}\sum_{i=1}^{n} F_i(z), \quad F_i(z) = A_i z + b_i,$$

$$g(z) = \lambda\|z\|_1 + \delta_{B_r(0)}(z) = \lambda\|z\|_1 + \begin{cases} 0, & \text{if } \|z\|_\infty \le r, \\ +\infty, & \text{if } \|z\|_\infty > r, \end{cases}$$

where each matrix $A_i \in \mathbb{R}^{d\times d}$ is non-symmetric with all eigenvalues with strictly positive real part, $b_i \in \mathbb{R}^d$, $r > 0$ is the radius of ℓ_∞-ball, and $\lambda \ge 0$ is regularization parameter. One can show (see Example 6.22 from [262]) that for the given $g(z)$ prox operator has an explicit formula:

$$\text{prox}_{\gamma g}(z) = \text{sign}\,(z)\min\left\{\max\left\{|z| - \gamma\lambda, 0\right\}, r\right\},$$

where $\text{sign}(\cdot)$ and $|\cdot|$ are component-wise operators. The considered problem generalizes the following quadratic game:

$$\min_{\|x_1\|_\infty \le r}\max_{\|x_2\|_\infty \le r} \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}x_1^\top A_{1,i}x_1 + x_1^\top A_{2,i}x_2 - \frac{1}{2}x_2^\top A_{3,i}x_2 + b_{1,i}^\top x_1 - b_{2,i}^\top x_2 + \lambda\|x_1\|_1 - \lambda\|x_2\|_1$$

with $\mu_i I \preccurlyeq A_{1,i} \preccurlyeq L_i I$ and $\mu_i I \preccurlyeq A_{3,i} \preccurlyeq L_i I$. Indeed, the above problem is a special case of (3.1)+(3.1.6) with

$$z = \begin{pmatrix} x_1 \\ \\ x_2 \end{pmatrix}, \quad A_i = \begin{pmatrix} A_{1,i} & A_{2,i} \\ \\ -A_{2,i} & A_{3,i} \end{pmatrix}, \quad b_i = \begin{pmatrix} b_{1,i} \\ \\ b_{2,i} \end{pmatrix},$$

$$g(z) = \lambda\|x_1\|_1 + \lambda\|x_2\|_1 + \delta_{B_r(0)}(x_1) + \delta_{B_r(0)}(x_2).$$

In our experiments, we generate non-symmetric matrices $A_i \in \mathbb{R}^{d\times d}$ with positive eigenvalues. It gives that $F(x)$ is strongly monotone and cocoercive. The $b_i \in \mathbb{R}^d$ are sampled from a normal distribution with variance $100/d$. For all the experiments, we choose $n = 1000$ and $d = 100$. In the distributed experiments, we simulate $m = 10$.

Uniform sampling (`US`) vs Important sampling (`IS`). We note that [258] which studies `SGDA-AS` does not consider `IS` explicitly. Although we show the theoretical benefits of `IS` in comparison to `US`, here we provide a numerical comparison to illustrate the superiority of `IS` (on both constrained and unconstrained quadratic games). We choose the matrices A_i such that $\ell_{\max} = \max_i \ell_i \gg \bar{\ell}$. In this case, our theory predicts that `IS` should perform better than `US`. We provide the results in Figure 3.1. We observe that indeed `SGDA` with `IS` converges faster and to a smaller neighborhood than `SGDA` with `US`. This observation perfectly corroborates our theory.

Comparison of variance reduced methods. In this experiment, we test the performance of our proposed `L-SVRGDA` (Algorithm 22) and compare it to other variance-reduced methods on quadratic games, see Figure 3.2. In particular, we compare it to `SVRG` [227], `SVRE` [100], `EG-VR` [214] and `VR-AGDA` [218]. In the constrained setting, we only compare `L-SVRGDA` to `SVRG` and `EG-VR`, as they are the only methods from this list that handle constrained settings. For loopless variants, we choose $p = \frac{1}{n}$ and for the non-loopless variants, we pick the number of inner-loop iterations to be n. We observe that all methods converge linearly and that `L-SVRGDA` is competitive with the other considered variance-reduced methods, converging slightly faster than all of them.

We point out that we plot the distance to optimality as a function of the number of oracle calls. When using variance-reduced methods, we sometimes have to compute the full-batch gradient, and thus have to make n oracle calls. This observation explains the presence of "steps" for variance-reduced methods in Figure 3.2: we observe a "step" occurs each time the full-batch gradient is computed.

Comparison of distributed methods. In our third experiment, we consider a distributed version of the quadratic game, in which we assume that $F(x) = \frac{1}{n}\sum_{i=1}^{n} F_i(x)$ with each $\{F_i\}_{i=1}^{n}$ being constructed similarly to the previous experiments. The information about operator F_i is stored on node i only. We compare the distributed methods proposed in the section: `QSGDA`, `DIANA-SGDA`, and `VR-DIANA-SGDA`. For the quantization, we use the `RandK` sparsification with $K = 5$. We show our findings in Figure 3.3, where the performance is measured in terms of the number of oracle calls and the number of bits communicated from workers to the server. In both figures, we can clearly see the advantage of using quantization in terms of reducing the communication cost compared to the baseline `SGDA`. We also observe that `VR-DIANA-SGDA` achieves linear convergence to the solution.

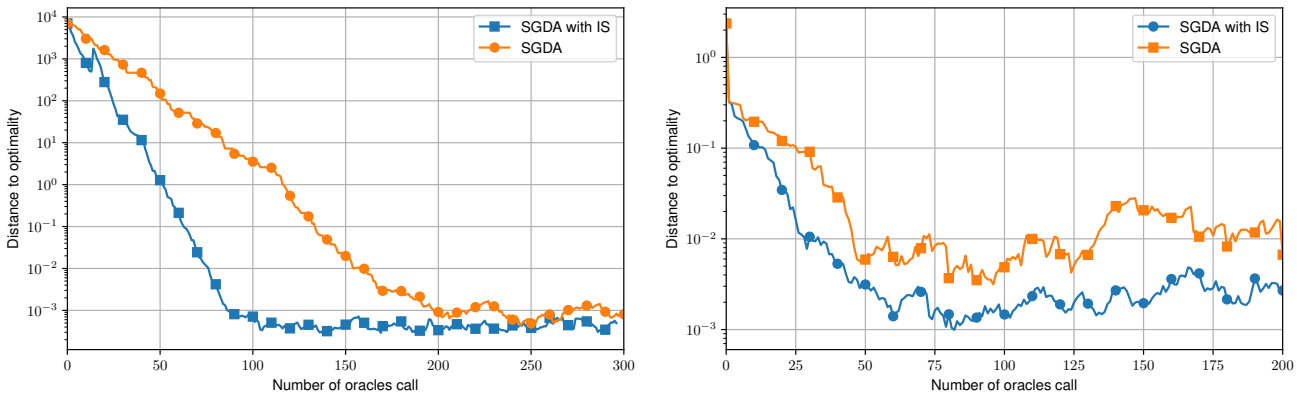


Figure 3.1: Comparison of Uniform Sampling (US) vs Importance Sampling (IS). Left: the result for the problem without constraints, right: with constraints. As expected by theory IS converges faster and to a smaller neighborhood than US.

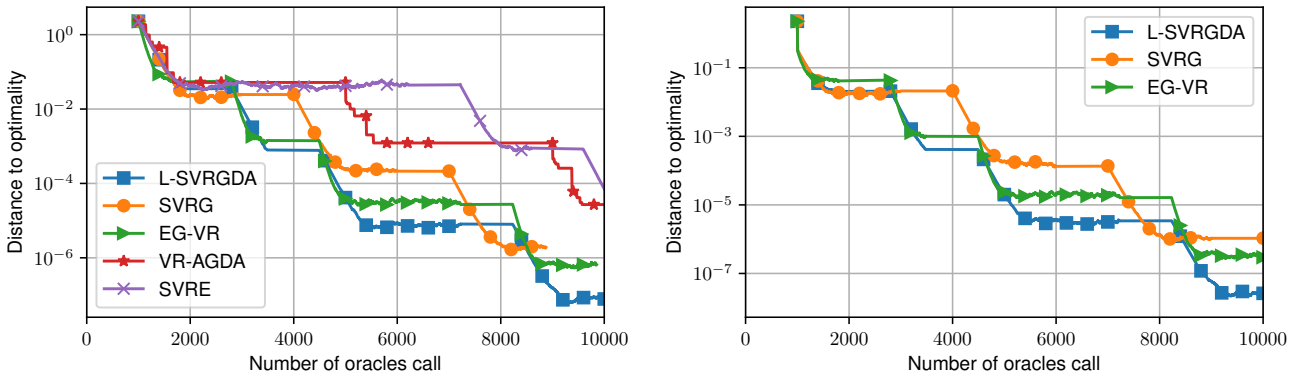


Figure 3.2: Comparison of variance-reduced methods. Left: the result for the problem without constraints, right: with constraints. Note that L-SVRGDA is very competitive, and outperforms all the other methods.

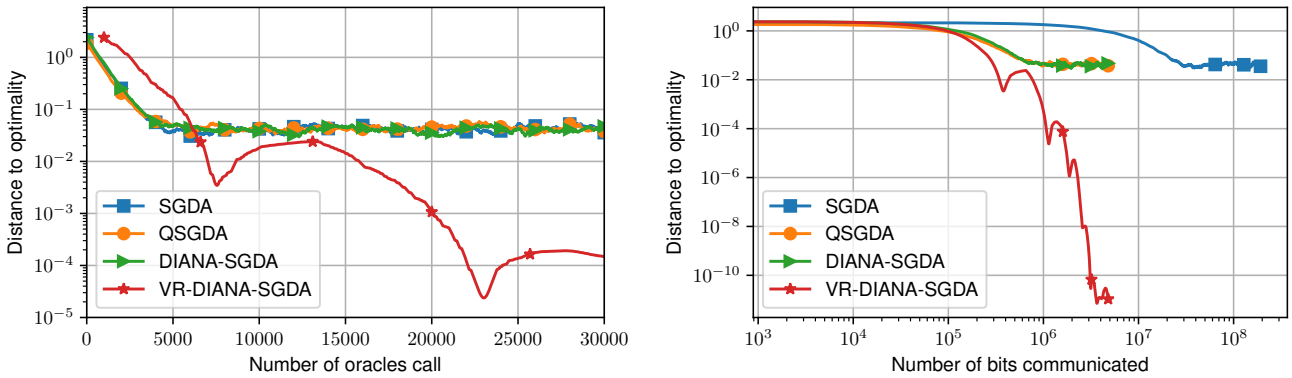


Figure 3.3: Comparison of algorithms in distributed setting. Left: number of oracle calls, right: number of bits communicated.

3.2 Optimal Batching

In this section, we continue with the problem (3.1), but now restrict ourselves to the formulation of the form:

$$F(x) = \frac{1}{n}\sum_{i=1}^{n} F_i(x). \tag{3.13}$$

In the last section, we considered a whole family of methods for completely different formulations. The goal now is to consider one single problem, obtaining not just the best method in the literature, but an optimal and unimprovable method for it.

3.2.1 Related Works and Contributions

In fact, we have discussed the problem of the form (3.1)+(3.13) several times, but it was not the focus of main attention. Therefore, for the completeness of the picture, we provide a comprehensive literature survey with an emphasis on the methods for the problem (3.1)+(3.13).

We start the review with deterministic methods, as there are no restrictions on using them for solving the problem of the sum form. The simplest method for solving variational inequalities is the classical Gradient Method. Unfortunately, it converges only for the cases of strongly monotone F, in the non-strongly monotone case convergence is lacking [279]. As we know from the introduction, a significant improvement in algorithms for (3.1) occurred when the Extragradient method was proposed [78]. Remarkably, it solved the convergence issue of the Gradient Method. Another major advancement was made in [79], where the Optimistic method was proposed. It is worth noting a significant difference between these two methods, as Extragradient required us to call the oracle twice, while the algorithm from [79] is a single-call scheme. Later, the exploitation of Bregman divergence for variational inequalities was proposed [80]. This approach allowed for consideration of generalized geometry that could be non-Euclidean. As a result, Mirror Prox algorithm was created. Next, there were a number of stochastic modifications of Extragradient and Mirror Prox.

[85] for the first time considered the stochastic version of Mirror Prox. This paper examines the general stochastic case with bounded variance. Another approach considers a frequently used finite-sum stochastic setup of the problem, where it is possible to achieve a better convergence rate due to its specifics. In particular, one of the earliest solutions for the saddle point problem in the Euclidean setup was considered by [227]. They based their method on the SVRG algorithm [52] and added the Catalyst envelope acceleration. As mentioned, the result was made for strongly convex-concave saddle point problems, which corresponds to strongly monotone variational inequalities. Other algorithms utilizing the variance reduction technique for solving finite-sum variational inequalities with strongly monotone operators with better convergence rate are L-SVRGDA [19] and SVRE [100]. The idea of SVRE was continued in [218] with the introduction of VR-AGDA. Here results were obtained under Polyak-Lojasiewicz conditions. Moreover, in [280], strongly convex-concave saddle point problems were considered. In this work, methods were designed with the Catalyst envelope acceleration. The achieved rate is comparable to [227]. In addition, it is worth mentioning the work [281]. This paper deals with the Bregman setup and adapts the variance reduction

technique. Unfortunately, this solution is limited by additional assumptions on the operator F and by considering the matrix games setup. Later, [214] applied variance reduction to Mirror Prox using the double loop SVRG technique [282].

The works mentioned above have one common problem the proposed algorithms are sensitive to the size of mini-batches, apart from paper [85]. However, convergence rate in [85] does not attain the lower bound [170], as it considers the general case.

We compare the mentioned algorithms and their convergence results in Table 3.4.

In this section, our goal is to construct methods that are optimal for solving the problem (3.1)+(3.13). It is important that the new algorithms remain insensitive to the size of the batch, unlike their competitors from Table 3.4. We present algorithms for both the classical Euclidean formulation and the more complicated and general formulation with Bregman geometry. Our methods are based on the following ideas: optimistic scheme [79] with negative momentum to deal with variance reduction and avoid double updates as seen in Extragradient; and a double-loop VR scheme that takes into account the Bregman divergence [214].

3.2.2 Assumptions

In this section, we address non-Euclidean geometry for the first time in this thesis. Prior to this, we have consistently utilized short notation $\|\cdot\| = \|\cdot\|_2$. In this section, we assume that $\mathcal{Z} \subseteq \mathbb{R}^d$ is a normed vector space with a dual space $\mathcal{Z}^*$ and a primal-dual norm pair $\|\cdot\|$ and $\|\cdot\|_*$. As $\|\cdot\|$, we employ the norm $\|\cdot\|_p$ for $p \in [1, 2]$.

Desiring to work in arbitrary geometry requires us to introduce an alternative way to measure distance in the space rather than the default Euclidean distance.

Definition 3.2.1. Let $h: \mathcal{Z} \to \mathbb{R} \cup \{+\infty\}$ be a convex lower semicontinuous function such that $\operatorname{dom} g \subseteq \operatorname{dom} h$ and h is differentiable on $\operatorname{dom} h$, h is 1-strongly convex on $\operatorname{dom} g$. We define the Bregman distance $V: \operatorname{dom} g \times \operatorname{dom} \partial h \to \mathbb{R}_+$ associated with h by

$$V(u, v) = h(u) - h(v) - \langle \nabla h(v), u - v \rangle.$$

Here, we used the following definition.

Definition 3.2.2. Function f is 1-strongly convex if it can be lower bounded by a quadratic function of the form:

$$f(y) \geq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{1}{2}\|y - x\|^2$$

for all x and y in $\operatorname{dom} f$.

We also provide a couple of examples of h and their corresponding V:

- For $h(x) = \frac{1}{2}\|x\|_2^2$ we have $V(x, y) = \frac{1}{2}\|x - y\|_2^2$.

Table 3.4: Summary complexities for finding an ε-solution for monotone stochastic (finite-sum) variational inequality (3.1)+(3.13) with Lipschitz operators. Convergence is measured by gap function.

Reference	Complexity	Non-Euclidean
Nemirovski et al. [80] (1)	$\mathcal{O}\left(n\frac{L}{\varepsilon}\right)$	✓
Juditsky et al. [85] (4)	$\mathcal{O}\left(\frac{L}{\varepsilon}+\frac{1}{\varepsilon^2}\right)$	✓
Palaniappan & Bach [227] (2,3,4)	$\tilde{\mathcal{O}}\left(b\frac{L_2^2}{\varepsilon^2}\right)$	
Palaniappan & Bach [227] (2,3,4)	$\tilde{\mathcal{O}}\left(\sqrt{bn}\frac{L_2}{\varepsilon}\right)$	
Chavdarova et al. [100] (3)	$\tilde{\mathcal{O}}\left(\frac{bL_2^2}{\varepsilon^2}\right)$	
Carmon et al. [281] (2,4)	$\tilde{\mathcal{O}}\left(\sqrt{bn}\frac{L}{\varepsilon}\right)$	✓
Yang et al. [218] (2,3,4)	$\tilde{\mathcal{O}}\left(b^{\frac{1}{3}}n^{\frac{2}{3}}\frac{L_2^3}{\varepsilon^3}\right)$	
Alacaoglu & Malitsky [214]	$\mathcal{O}\left(\sqrt{bn}\frac{L}{\varepsilon}\right)$	✓
Tominin et al. [280] (2,3,4)	$\tilde{\mathcal{O}}\left(\sqrt{bn}\frac{L_2}{\varepsilon}\right)$	
this thesis (Section 3.1) (3,4)	$\tilde{\mathcal{O}}\left(b\frac{L_2^2}{\varepsilon^2}\right)$	
this thesis (5)	$\tilde{\mathcal{O}}\left(\sqrt{n}\frac{L_2}{\varepsilon}\right)$	
this thesis (5)	$\mathcal{O}\left(\sqrt{n}\frac{L_2}{\varepsilon}\right)$	✓
this thesis (5)	$\tilde{\mathcal{O}}\left(\sqrt{n}\frac{L}{\varepsilon}\right)$	✓
Lower bounds [170]	$\Omega\left(\sqrt{n}\frac{L_2}{\varepsilon}\right)$	

Notation: L and L_2 are Lipschitz constants for F and F_m in terms of $\|\cdot\|_*$ and $\|\cdot\|_2$ norms, respectively (see Assumptions 3.2.5, 3.2.6), n = size of dataset, b = batch size per iteration.
(1) deterministic methods, similar results were also obtained in [86, 283],
(2) for saddle point problems only,
(3) only for μ-strongly monotone operators. To compare algorithms, we apply regularization trick,
(4) only for bounded domain,
(5) for $b \leq \sqrt{n}$.

- The entropy function

$$h(x) = \sum_{i=1}^{d} x_i \log x_i \tag{3.14}$$

in a probabilistic simplex

$$\Delta^d = \{x \in \mathbb{R}^d \,|x_i \geq 0 \; \sum_{i=1}^{d} x_i = 1\} \tag{3.15}$$

generates KL-divergence

$$V(x, y) = \sum_{i=1}^{d} x_i \log \frac{x_i}{y_i}.$$

To further analyze the problem, we introduce the following assumptions:

Assumption 3.2.3. The solution (maybe not unique) for the problem (3.1)+(3.13) exists.

Assumption 3.2.4. The operator F is monotone, i.e. for all $u, v \in \mathcal{Z}$ we have

$$\langle F(u) - F(v); u - v\rangle \geq 0.$$

Assumption 3.2.5. The operator F is L_2-Lipschitz and for all $i \in [1, n]$ F_i is $L_{2,i}$-Lipschitz, i.e. for all $u, v \in \mathcal{Z}$ we have

$$\|F(u) - F(v)\|_2 \leq L_2\|u - v\|_2,$$
$$\|F_i(u) - F_i(v)\|_2 \leq L_{2,i}\|u - v\|_2.$$

In addition, we define $\overline{L}_2$ such that $\overline{L}_2^2 = \frac{1}{n}\sum_{i=1}^{n} L_{2,i}^2$. With this notation, for all $u, v \in \mathcal{Z}$ we have

$$\frac{1}{n}\sum_{i=1}^{n} \|F_i(u) - F_i(v)\|_2^2 \leq \bar{L}_2^2\|u - v\|_2^2.$$

Assumption 3.2.6. The operator F is L-Lipschitz and for all $i \in [1, n]$ F_i is L-Lipschitz, i.e. for all $u, v \in \mathcal{Z}$ we have

$$\|F(u) - F(v)\|_* \leq L\|u - v\|,$$
$$\|F_i(u) - F_i(v)\|_* \leq L\|u - v\|.$$

Note that $\|\cdot\| \leq \|\cdot\|_2$ for all $p \in [1, 2]$. Thus, we can state that $L \leq \bar{L}_2$. In addition, we would like to point out that these meanings often differ dramatically.

3.2.3 Algorithms

Let us start with the Euclidean case and then move on to general Bregman geometry.

Euclidean case

In line 4 of Algorithm 30 we use the variance reduction technique in $\left(F_j(z^k) - F_j(w^{k-1}) + F(w^{k-1})\right)$ part. We also add batching in lines 3 and 4. In addition, in line 4 we use $\left(F_j\left(z^k\right) - F_j\left(z^{k-1}\right)\right)$ term to implement the so-called optimistic scheme, slightly different from the original option [79]. Our update is a modification of the Forward-Reflected-Backward approach [284], where we set $\alpha \equiv 1$ in $z^{k+1} = \eta F_j(z^k) + \eta\alpha\left[F_j(x^k) - F_j(x^{k-1})\right]$. While for minimization problems it is usual to apply positive (heavy-ball) momentum [285], the opposite approach is suitable for variational inequalities. This effect was noticed earlier [286, 287, 288] and appeared in the theory of stochastic methods for VIs. Hence, in line 5 we also apply the negative momentum in $\left(-\frac{1}{\eta}\gamma V(x, x_s^k) + \frac{1}{\eta}\cdot\gamma V(x, \overline{w}_s)\right)$ part.

Algorithm 30 Optimistic Method with Momentum and Batching

Parameters: stepsize $\gamma > 0$, momentum $\alpha > 0$, probability $q \in (0; 1)$, batchsize $b \in \{1, \ldots, n\}$
Initialization: choose $z^0 = w^0 = z^{-1} = w^{-1} \in \mathbb{R}^d$

1: **for** $k = 0, 1, \ldots, K-1$ **do**
2: Sample $j_1^k, \ldots, j_b^k$ independently from $\{1, ..., n\}$ uniformly at random
3: $S^k = \{j_1^k, ..., j_b^k\}$
4: $\Delta^k = \frac{1}{b}\sum\limits_{j\in S^k}(F_j(z^k) - F_j(w^{k-1}) + (F_j(z^k) - F_j(z^{k-1}))) + F(w^{k-1})$
5: $z^{k+1} = \text{prox}_{\gamma g}(z^k + \alpha(w^k - z^k) - \gamma\Delta^k)$
6: $w^{k+1} = \begin{cases} z^{k+1}, & \text{with probability } q \\ w^k, & \text{with probability } 1-q \end{cases}$
7: **end for**

The following theorem describes the convergence of the proposed method.

Theorem 3.2.7. Consider the problem (3.1)+(3.13) under Assumptions 3.2.3, 3.2.4, 3.2.5. Let $\{z^k\}$ be the sequence generated by Algorithm 30 with tuning of γ, α, q as follows:

$$0 < q = \alpha \leq \frac{1}{16}, \quad \gamma = \min\left\{\frac{\sqrt{\alpha b}}{8\bar{L}_2}, \frac{1}{8L_2}\right\}$$

Then, given $\varepsilon > 0$, the number of iterations for $\mathbb{E}[\text{Gap}(z^k)] \leq \varepsilon$ is

$$\mathcal{O}\left(\frac{1}{\sqrt{qb}}\frac{\bar{L}_2}{\varepsilon} + \frac{L_2}{\varepsilon}\right).$$

Theorem provides the iteration complexity of Algorithm 30, but the primary focus is on the oracle complexity (the number of calls to terms F_i). One can note that at each iteration we call $\mathcal{O}(b + qn)$ terms on average/expectation – each time we call the batch with a size of b, and with a

probability q, we compute the full operator. From this, we immediately obtain the optimal choice for q:

Corollary 3.2.8. Under the conditions of Theorem 3.2.7, if we choose $q = \frac{b}{n}$, then we obtain the following oracle complexities

$$\mathcal{O}\left(\sqrt{n}\frac{\bar{L}_2}{\varepsilon} + b\frac{L_2}{\varepsilon}\right).$$

With $b \leq \frac{\bar{L}\sqrt{n}}{L}$, we get $\mathcal{O}\left(\sqrt{n}\frac{\bar{L}_2}{\varepsilon}\right)$. This result is optimal and unimprovable [170].

Bregman case

The essence of Algorithm 31 for the Bregman setup is similar to Algorithm 30. Here, we also use variance reduction, batching, and an optimistic scheme in line 5. Negative momentum is also present, but in the form $(1-\alpha)V(z, z_s^k) + \alpha V(x, \overline{w}_s)$ in line 6. The only thing is that instead of the loopless version, which updates w using coin flip (6 of Algorithm 30), two loops are used in Algorithm 31 and w is updated deterministically. The snapshot point updates in line 8 similar to [214] and SVRG [52]. Due to the Bregman setup is that we have the additional point $\overline{w}_{s+1}$ that averages in the dual space (line 9).

Algorithm 31 Bregman Optimistic Method with Momentum and Batching

Parameters: stepsize $\gamma > 0$, momentum $\alpha > 0$, batch size $b \in \{1, \ldots, n\}$
Initialization: choose $z_0^{-1} = z_0^0 = w_0^0 = z_{-1}^k = w_{-1}^k \in \mathcal{Z}$ for all $k \in [-1, K-1]$

1: **for** $s = 0, 1, \ldots S-1$ **do**
2: **for** $k = 0, 1, \ldots, K-1$ **do**
3: Sample $j_1^k, \ldots, j_b^k$ independently from $\{1, ..., n\}$ uniformly at random
4: $B^k = \{j_1^k, ..., j_b^k\}$
5: $\Delta^k = \frac{1}{b}\sum\limits_{j \in B^k}(F_j(z_s^k) - F_j(w_s) + (F_j(z_s^k) - F_j(z_s^{k-1}))) + F(w_s)$
6: $z_s^{k+1} = \operatorname{argmin}\limits_{z \in \mathcal{Z}}\left(g(z) + \frac{1}{\gamma}(1-\alpha)V(z, z_s^k) + \frac{1}{\gamma}\cdot\alpha V(x, \overline{w}_s) + \langle\Delta^k, x\rangle\right)$
7: **end for**
8: $w_{s+1} = \frac{1}{K}\sum\limits_{k=1}^{K} z_s^k$
9: $\nabla h(\overline{w}_{s+1}) = \frac{1}{K}\sum\limits_{k=1}^{K}\nabla h(z_s^k)$
10: $z_{s+1}^0 = z_s^K$
11: $z_{s+1}^{-1} = z_s^{K-1}$
12: **end for**

Now we present the theorem to state the convergence of Algorithm 31.

Theorem 3.2.9. Consider the problem (3.1)+(3.13) under Assumptions 3.2.3, 3.2.4 and 3.2.5.

Let $\{z_B^k\}$ be the sequence generated by Algorithm 31 with tuning of α, γ as follows:

$$0 < \alpha \leq \frac{1}{16}, \quad \gamma = \min\left\{\frac{\sqrt{\alpha b}}{8\bar{L}_2}, \frac{1}{8L_2}\right\}.$$

Then for $z_S = \frac{1}{KS}\sum\limits_{s=0}^{S-1}\sum\limits_{k=0}^{K-1} z_B^k$ it holds that

$$\mathbb{E}\left[\mathrm{Gap}(z_S)\right] \leq \frac{(2+K\gamma)}{\eta K S}\max_{z\in\mathcal{C}}\left\{V(z, z_0)\right\}.$$

Now we are ready to obtain the complexity of Algorithm 31. We choose $\alpha = \frac{1}{K}$ to achieve $O(1)$ in the numerator of estimation. In addition, it is natural to take $K = \frac{n}{3b}$ due to the triple operator call in line 4.

Corollary 3.2.10. Let $K = \frac{n}{3b}$, $\gamma = \frac{1}{K} = \frac{3b}{M}$ and $\gamma = \min\left\{\frac{\sqrt{\alpha b}}{2\bar{L}_2}, \frac{1}{8L_2}\right\} = \min\left\{\frac{\sqrt{b}}{2\bar{L}_2}\cdot\sqrt{\frac{3b}{n}}, \frac{1}{8L_2}\right\}$ and $b \leq \frac{\sqrt{n}\bar{L}_2}{L_2}$. Then the total complexity of Algorithm 31 to reach ϵ-accuracy is $\mathcal{O}(n + \frac{L\sqrt{n}}{\epsilon})$.

Now let us consider the case when Lipschitzness of the operator is given in the form of Assumption 3.2.6.

Theorem 3.2.11. Consider the problem (3.1)+(3.13) under Assumptions 3.2.3, 3.2.4 and 3.2.6. Let $\{z_s^k\}$ be the sequence generated by Algorithm 31 with tuning of $\eta, \theta, \alpha, \beta, \gamma$ as follows:

$$0 < \alpha \leq \frac{1}{16}, \quad \gamma = \min\left\{\frac{\sqrt{\alpha b}}{8L\sqrt{1+\ln d}}, \frac{1}{8L\sqrt{1+\ln d}}\right\}.$$

Then for $z_S = \frac{1}{KS}\sum\limits_{s=0}^{S-1}\sum\limits_{k=0}^{K-1} z_s^k$ it holds that

$$\mathbb{E}\left[\mathrm{Gap}(z_S)\right] \leq \frac{(2+K\gamma)}{\eta K S}\max_{z\in\mathcal{C}}\left\{V(z, z_0)\right\}.$$

Now we proceed similarly to Corollary 3.2.10.

Corollary 3.2.12. Let $K = \frac{n}{3b}$, $\gamma = \frac{1}{K} = \frac{3b}{n}$ and $\gamma = \min\left\{\frac{\sqrt{\gamma b}}{2\sqrt{1+\ln d}L}, \frac{1}{8L\sqrt{1+\ln d}}\right\}$
$= \min\left\{\frac{b}{2\sqrt{1+\ln d}L}\cdot\sqrt{\frac{3}{M}}, \frac{1}{8L\sqrt{1+\ln d}}\right\}$ and $b \leq \sqrt{n}$. Then the total complexity of the Algorithm 31 to reach ϵ-accuracy is $\mathcal{O}(n + \frac{L\sqrt{n}}{\epsilon})$.

As mentioned before, $L \leq \bar{L}_2$. Thus, we can state $\frac{L}{\sqrt{n}S} \leq \frac{\bar{L}_2}{\sqrt{n}S}$.

3.2.4 Experiments

In this section, empirical performance of Algorithms 30 and 31 is demonstrated. We consider matrix games

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \langle Ax, y \rangle \tag{3.16}$$

with matrix $A \in \mathbb{R}^{d \times d}$ and use simplex constraints (3.15) in the entropic setup:

$$\mathcal{X} = \Delta^d, \quad \mathcal{Y} = \Delta^d.$$

For our experiments, we choose the policeman and burglar matrix [289] and the first test matrix from [260].

Note that the problem (3.16) does not have the finite-sum form as (3.1)+(3.13), but it can be rewritten as

$$A = \sum_{i=1}^{d} A_{i:} \text{ or } A = \sum_{j=1}^{d} A_{:j}.$$

where $A_{i:}$ is the i-th row of A and $A_{:j}$ is the j-th column of A – see details in Section 5.1.2 from [214]. In this formulation the problem already has the form of a finite sum.

Euclidean case

For comparison, we take Algorithm 30 and methods from Table 3.4. In particular, we choose Algorithms 1 and 2 from [214], Algorithm 1+2 from [281]. All algorithms are considered in the Euclidean setting with the projection to the simplex from [290]. The parameters of all methods are tuned for the best convergence. We run all methods with different batch sizes. We use duality gap (3.2) as the convergence measure, it can be simply computed as $\left[\max_i (A^T x)_i - \min_j (Ay)_j\right]$ for simplex constraints. The comparison criterion is the number of operations (one operation is computationally equal to calculations of Ay and $A^T x$).

The results are reflected in Figures 3.4 and 3.5. They show that Algorithm 30 outperforms all competitors.

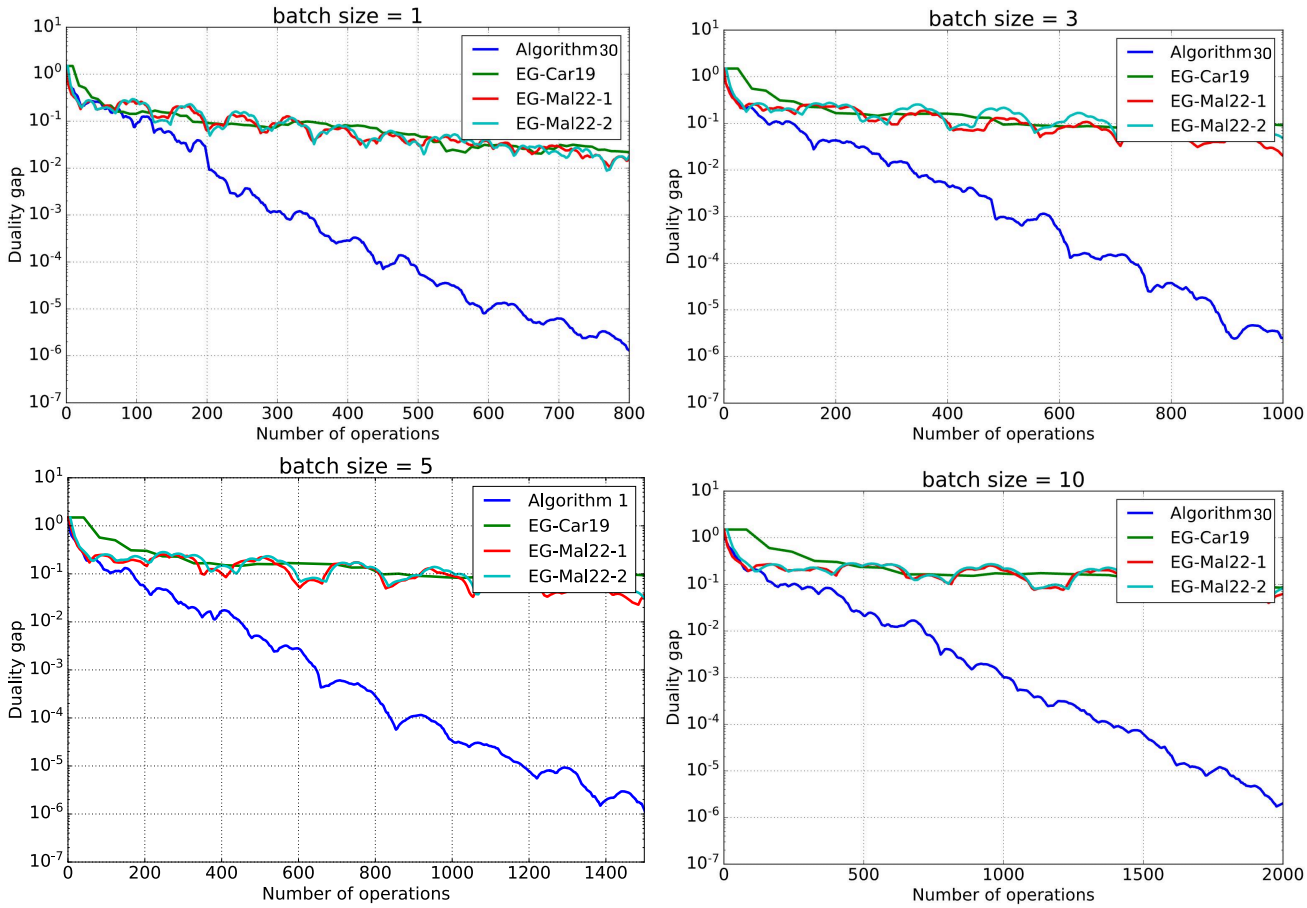


Figure 3.4: Comparison of computational complexities for Algorithm 30, EG-Mal22-1 (Algorithm 1 [214]), EG-Mal22-2 (Algorithm 2 [214]), EG-Car19 (Algorithm 1+2 from [281]) on (3.16) with policeman and burglar matrix from [289].

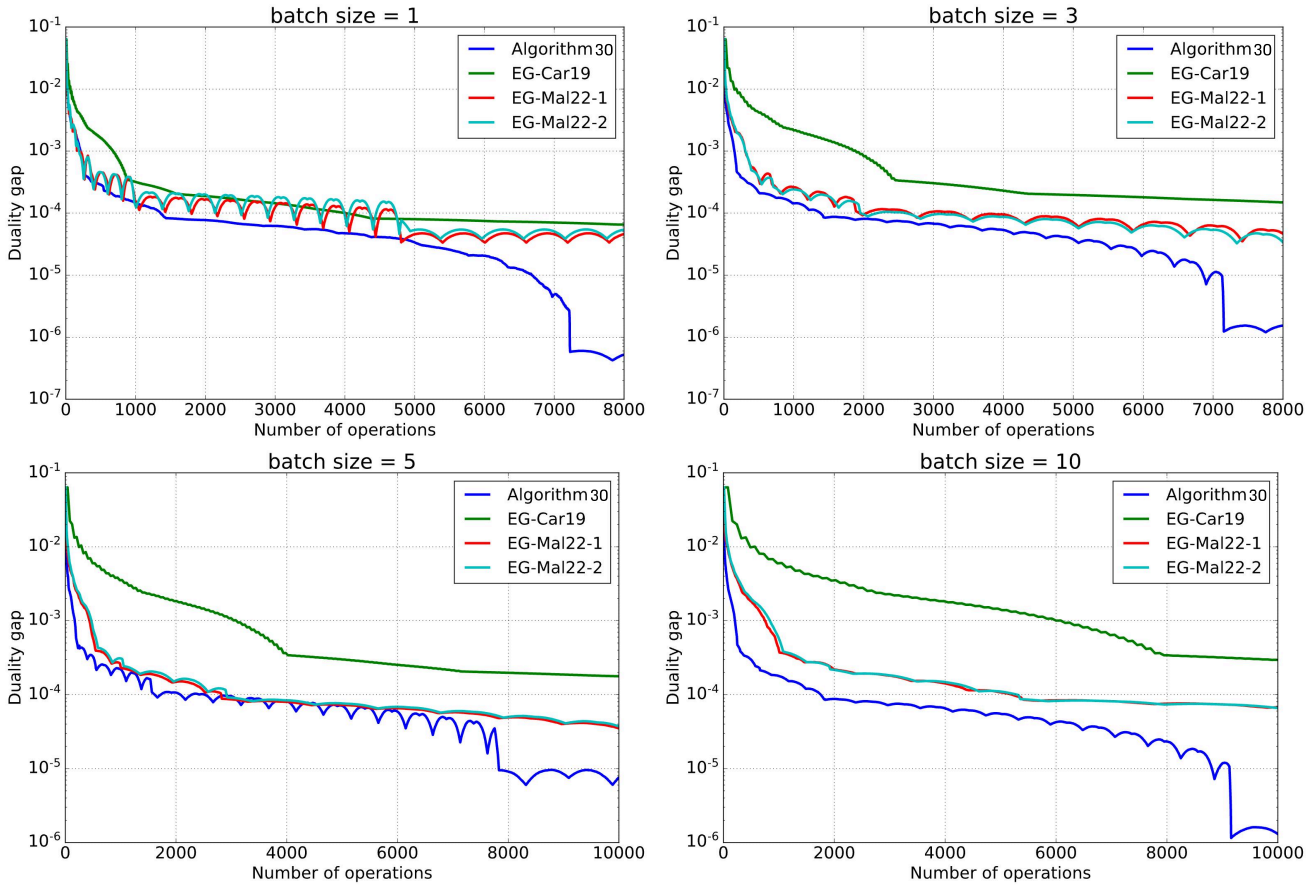


Figure 3.5: Comparison of computational complexities for Algorithm 30, EG-Mal22-1 (Algorithm 1 [214]), EG-Mal22-2 (Algorithm 2 [214]), EG-Car19 (Algorithm 1+2 from [281]) on (3.16) with policeman and burglar matrix from [289].

Bregman case

In order to calculate the update of Algorithm 31 (in particular, we need to compute ∇h in closed form) we introduce $u = (u^x, u^y)$ and $v = (v^x, v^y)$. With this notation, we can adapt (3.14) in a manner similar to previous works [288, 281] way:

$$\begin{aligned}\nabla h\left(x_s^{k+1}\right) =&(1-\gamma)\nabla h\left(x_s^k\right) + \gamma \nabla h\left(\overline{u}^s\right) \\ &- \eta\left(\frac{1}{b}\sum_{i\in B^k}\left[\frac{1}{r_i}A_{i:}\left(2y_{s,i}^k - v_{s,i} - y_{s,i}^{k-1}\right)\right] + A^\top v_s\right) \\ =&(1-\gamma)\nabla h\left(x_s^k\right) + \gamma \nabla h\left(\overline{u}^s\right) - \eta A^\top v_s \\ &- \frac{\eta}{b}\sum_{i\in B^k}\left[\frac{1}{r_i}A_{i:}\left\|2y_s^k - v_s - y_s^{k-1}\right\| \cdot \operatorname{sign}\left(2y_{s,i}^k - v_{s,i} - y_{s,i}^{k-1}\right)\right].\end{aligned}$$

Likewise, for the second component, one can write

$$\begin{aligned}\nabla h\left(y_s^{k+1}\right) =&(1-\gamma)\nabla h\left(y_s^k\right) + \gamma \nabla h\left(\overline{v}^s\right) + \eta A u_s \\ &+ \frac{\eta}{b}\sum_{j\in B^k}\left[\frac{1}{c_j}A_{:j}\left\|2x_s^k - u_s - x_s^{k-1}\right\| \cdot \operatorname{sign}\left(2x_{s,j}^k - u_{s,j} - x_{s,j}^{k-1}\right)\right].\end{aligned}$$

Now we are ready to implement Algorithm 31 and compare our method with other methods. In particular, we choose Algorithm 2 from [214], Algorithm 1+2 from [281]. The parameters of the algorithms are taken for the best convergence.

We run all methods with various batch sizes. The results are reflected in Figures 3.6 and 3.7. It is clearly shown that Algorithm 31 outperforms competitors, and its convergence rate is insensitive to the size of the batch (unlike the other methods).

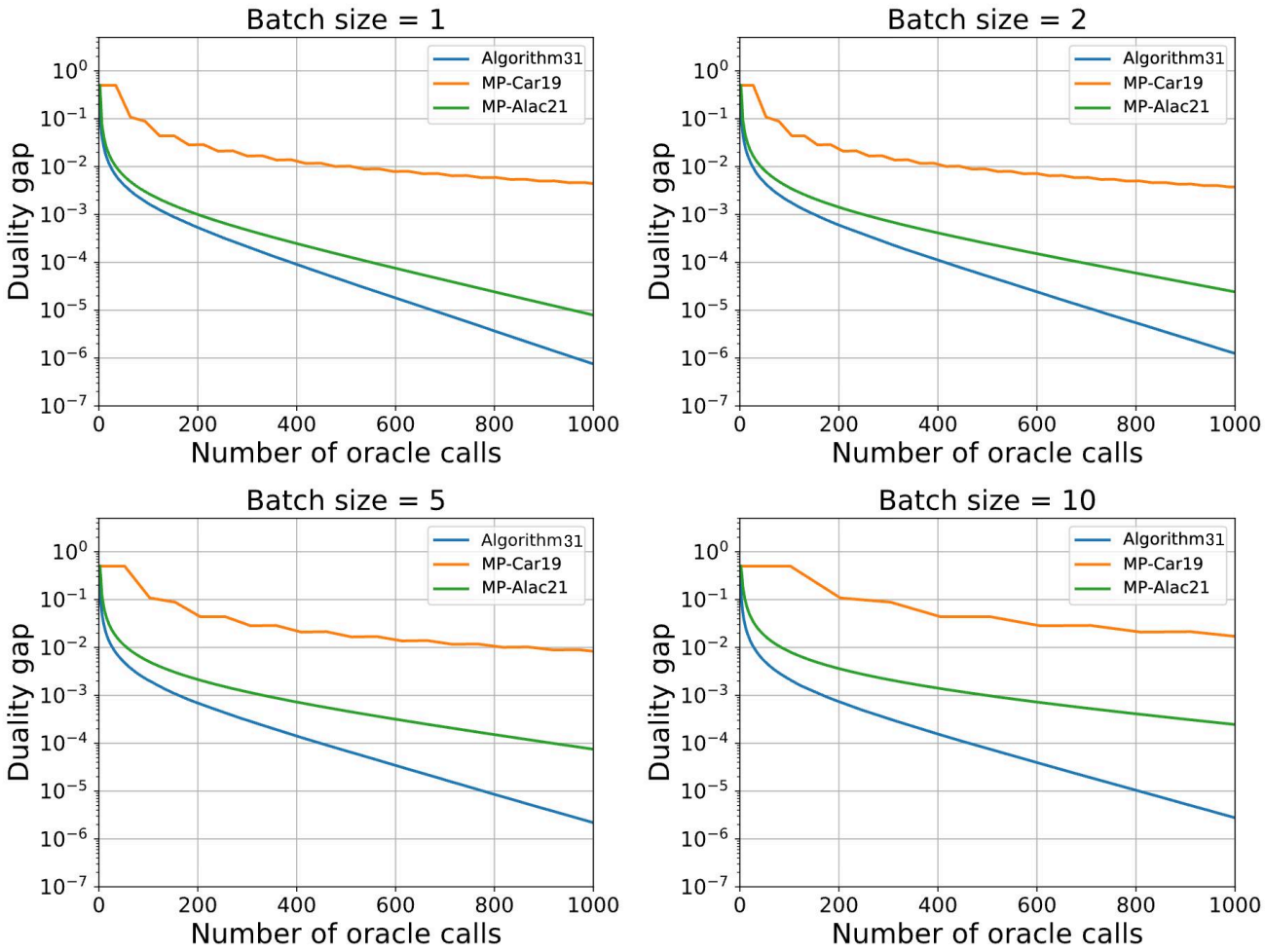


Figure 3.6: Comparison of computational complexities for Algorithm 31, MP-Car19 (Algorithm 1+2 from [281]), and MP-Alac21 (Algorithm 2 [214]) with different batch sizes on test matrix from [289].

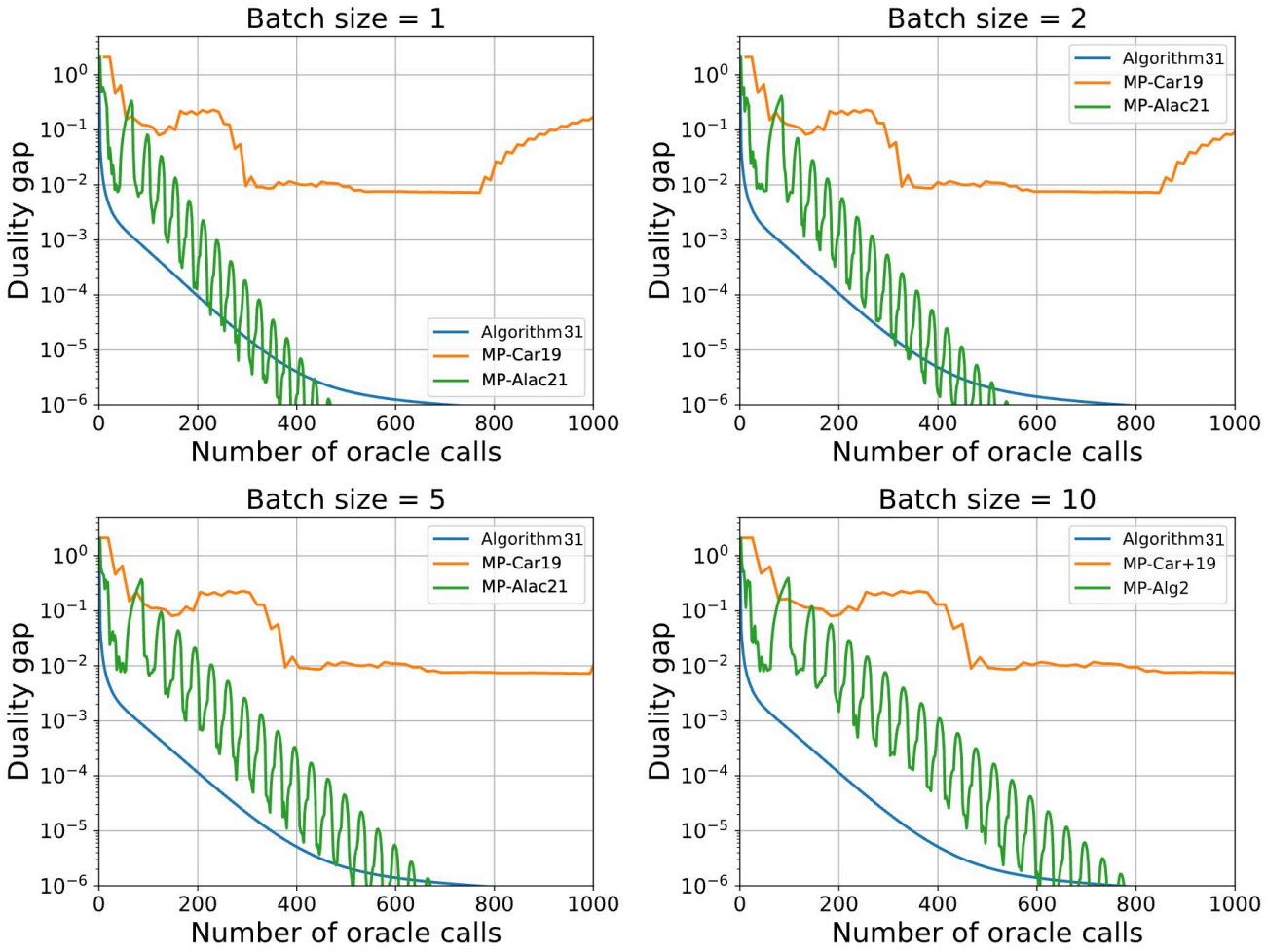


Figure 3.7: Comparison of computational complexities for Algorithm 31, MP-Car19 (Algorithm 1+2 from [281]), and MP-Alac21 (Algorithm 2 [214]) with different batch sizes on policeman and burglar matrix from [289].

Additionally, we notice that for the given test matrices, it is possible to improve convergence rates by adjusting parameters. Using grid search, we achieve the parameters that guarantee the best convergence. With these parameters, we run algorithms and plot Figures 3.8 and 3.9. The results of this experiment are similar: our algorithm provides better results than the competitors, and the convergence rate does not depend on the batch size.

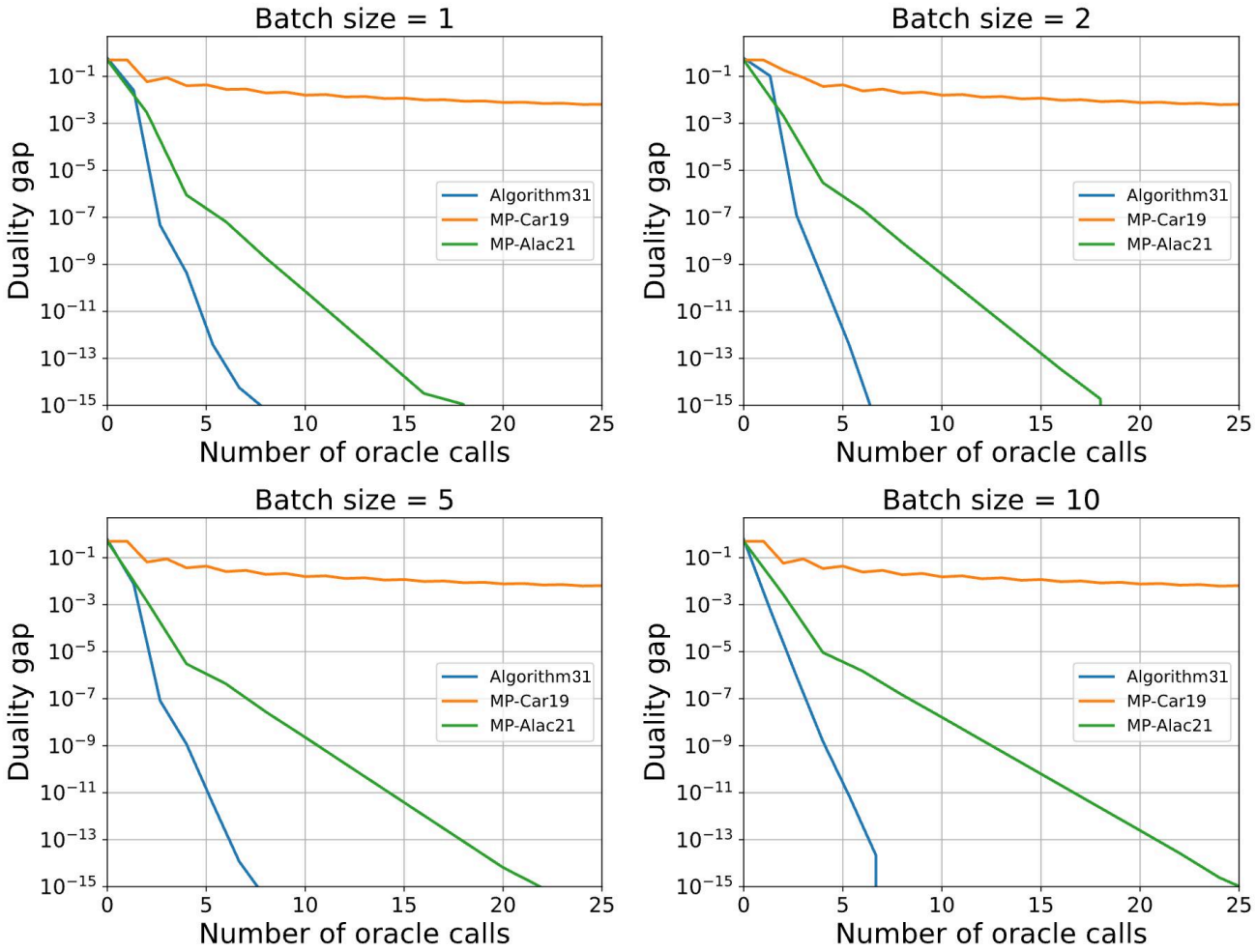


Figure 3.8: Comparison of computational complexities for Algorithm 31(with adjusted parameters), MP-Car19 (Algorithm 1+2 from [281]), and MP-Alac21 (Algorithm 2 [214]) with different batch sizes on first test matrix from [289].

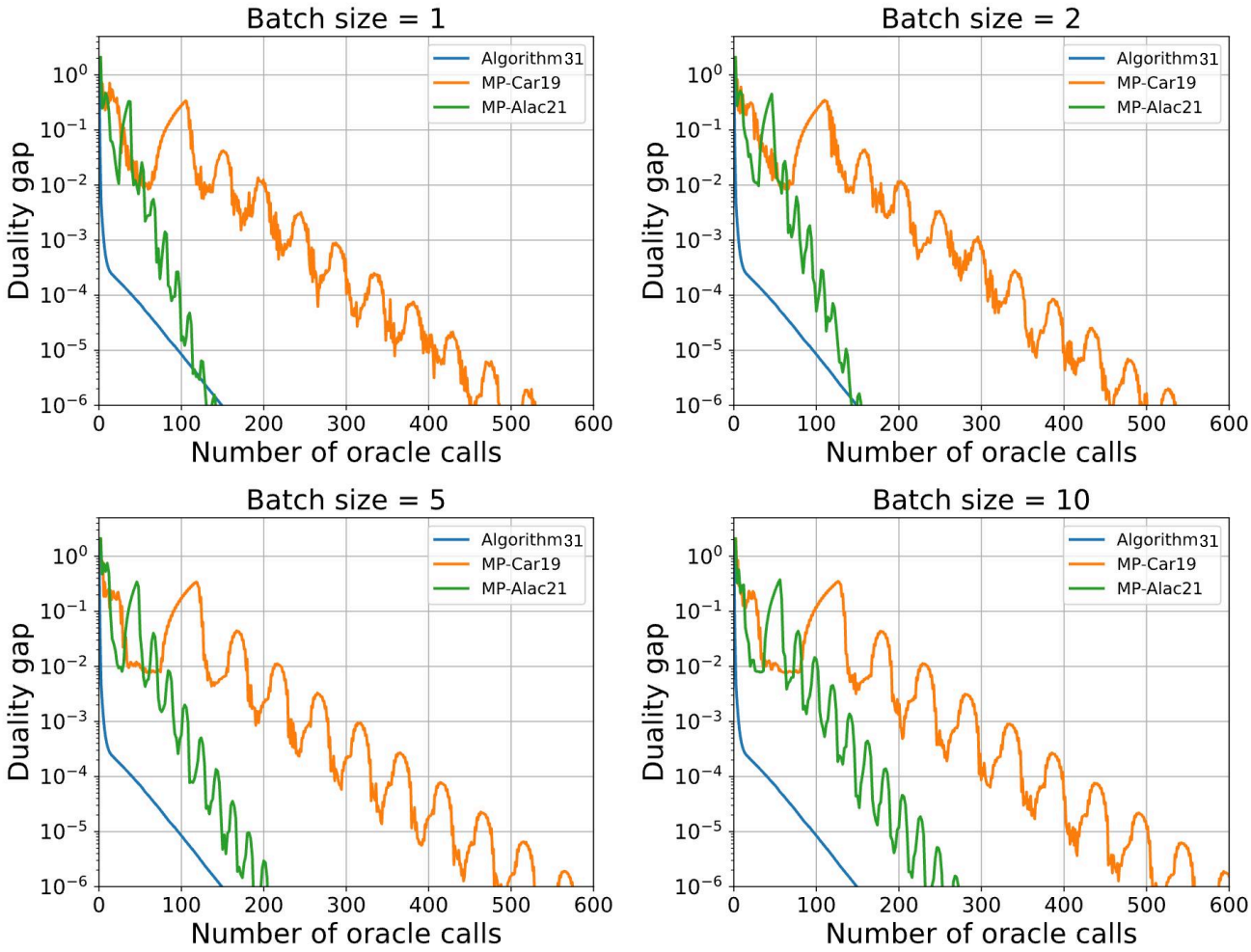


Figure 3.9: Comparison of computational complexities for Algorithm 31(with adjusted parameters), MP-Car19 (Algorithm 1+2 from [281]), and MP-Alac21 (Algorithm 2 [214]) with different batch sizes on policeman and burglar matrix from [289].

Chapter 4

Stochastic Frank-Wolfe

In this chapter, we move away from solving variational inequalities and consider a more widely used optimization formulation, the minimization problem:

$$\min_{x \in \mathcal{X} \subset \mathbb{R}^d} f(x)\,. \tag{4.1}$$

It is important to note that we are interested in the setting where $\mathcal{X} \neq \mathbb{R}^d$, it is significantly less studied compared to the unconstrained setting with $\mathcal{X} = \mathbb{R}^d$. Although it should be noted that the unconstrained setting also has its open questions (see Chapter 5).

There are many different algorithms that take into account different properties of $\mathcal{X}$. The simplest ones, for example, are projection methods. In cases when it is difficult or expensive to make a projection onto the set $\mathcal{X}$, another algorithms was invented. In particular, methods with a linear minimization oracle (LMO). This chapter focuses on these methods.

The study of algorithms for (4.1) that do not require projections has a history of more than half a century. Arguably, the most popular projection-free method is the Frank-Wolfe (also known as the Conditional Gradient algorithm [291]). This method maintains sparse iterates and only requires a linear minimization oracle that takes into account the specificity of the constraint set $\mathcal{X}$. In particular, the classical version of the method considers a linear approximation of the function at the current point x^k, and minimizes this approximation on the set $\mathcal{X}$:

$$\begin{aligned} s^k &= \arg\min_{s \in \mathcal{X}} \langle \nabla f(x^k), s - x^k \rangle, \\ x^{k+1} &= x^k + \eta_k (s^k - x^k), \end{aligned} \tag{4.2}$$

where η_k is parameter-free and equal to $\frac{2}{k+2}$.

After [291] proposed the FW algorithm, many works improved its theory and extended it to special cases [292, 293, 294, 295]. In the last decade, the Frank-Wolfe-type approaches have attracted increasing interest in the machine-learning community [296, 297] because of their good performance on sparse problems or on problems where the constraints are complex but structured (e.g., various ℓ_p balls, trace norms), having applications to submodular optimization [298], vision [299, 300], and variational inference [301]. See [302] for a detailed historical survey.

As stated in the introductory chapter, modern applications require the use of stochastic methods.

In this chapter, we examine various stories around the setting (4.1) and stochastic variations of the Frank-Wolfe method for it.

Before we start, we introduce the general assumptions needed for all sections. Special additional assumptions will be introduced already within the sections. We begin with the assumption that the function f is smooth. This assumption is widely used in the analysis of Frank-Wolfe-type methods.

Assumption 4.0.1. The function $f : \mathcal{X} \to \mathbb{R}$, is L-smooth on $\mathcal{X}$, i.e., there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall\, x, y \in \mathcal{X}.$$

The second assumption is the convexity of the function f.

Assumption 4.0.2. The function $f : \mathcal{X} \to \mathbb{R}$, is convex, i.e.,

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle, \quad \forall\, x, y \in \mathcal{X}.$$

Note that we consider both convex and non-convex cases of the function f. The next assumption is also typical and found in all works on projection-free methods.

Assumption 4.0.3. The set $\mathcal{X}$ is convex and compact with a diameter D, i.e., for any $x, y \in \mathcal{X}$,

$$\|x - y\| \leq D.$$

For convergence analysis in the non-convex case we need to introduce the *Frank-Wolfe gap* function [296] as a criterion for convergence:

$$\text{gap}(y) = \max_{x \in \mathcal{X}} \langle \nabla f(y), y - x \rangle. \tag{4.3}$$

Such a criterion is standard in the analysis of algorithms for the constrained problems with non-convex functions [303, 304]. It is easy to check that $\text{gap}(y) \geq 0$ for any $y \in \mathcal{X}$. Moreover, a point $y \in X$ is stationary for (4.1) if and only if $\text{gap}(y) = 0$. [303] notes that the Frank-Wolfe gap is a meaningful measure of non-stationarity, and also an affine invariant generalization of the more standard convergence criterion $\|\nabla f(y)\|$ that is used for unconstrained non-convex problems.

4.1 Stochastic Frank-Wolfe for Finite-Sum Problems

In this section, we consider a constrained problem of this type:

$$\min_{x \in \mathcal{X} \subset \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)\,. \tag{4.4}$$

Note that this problem is a special case of problem (4.1), where the target operator has the form of a finite sum. We have previously encountered such formulations in Chapters 2 and 3, but in

the general variational inequality form. The problem (4.4) leads us to the problems of empirical risk minimization, a cornerstone for training supervised machine learning models, such as various regression methods, support vector machines, and neural networks [51].

As mentioned earlier, this setting corresponds to the sum of the losses of the model with parameters x applied to a large number of data points, indexed by $i = 1, \ldots, n$. Because n is large, calculating the full gradient of f is expensive. Therefore, stochastic methods for (4.4) that rarely resort to calling ∇f (or avoid it altogether) are of particular importance.

4.1.1 Related Works and Contributions

If we discuss stochastic methods for solving problem (4.4), we should begin by mentioning the methods applicable to the unconstrained problem. The theory of stochastic methods for unconstrained (or projection-friendly) optimization problems has built upon the highly successful SGD method [46, 260] to obtain faster methods for finite sum-problems. Particularly, many so-called variance-reduced variants of SGD have been proposed, including SAG/SAGA [55, 305, 274], SVRG [52, 282, 306], MISO [307], SARAH [53, 276, 53, 277, 308], SPIDER [309], STORM [278], PAGE [54], and many others.

Extensive research in the theory of deterministic Frank-Wolfe-type methods and stochastic methods for unconstrained problems has led to the development of stochastic versions of projection-free algorithms. [310] proposed an algorithm for online stochastic optimization. [311] developed a projection-free version using sliding. [312, 304, 313, 314, 315, 316] proposed modifications of the Frank-Wolfe method using variance reduction techniques, namely SVRG, SAGA and SPIDER. [317, 318, 319] used the idea of momentum to deal with stochasticity. [320] and [321] explored stochastic methods for linear predictors. [322] extended the results of [304] from convex sets to manifolds. We summarize and compare the convergence rate of each method in Table 4.1. Note that for the SAGA-related methods, we report a slightly different result from the one reported by [304] because we believe that their proof contains a slight inaccuracy. We also do not include the approach from [316] in Table 4.1, since this method uses the hessian of the target function.

Next, we detail the contributions of this section, which can be divided into four parts.

• **The best rates in the convex case.** Our convergence guarantees are superior to those of classical deterministic methods [291, 311] as well as the stochastic methods from [320, 321, 322] regarding dataset size n. Moreover, the theoretical rates of our methods also surpasses the rest existing results from [310, 311, 312, 314, 315, 317, 318, 322] in terms of accuracy ε.

• **No need for full gradients.** Many stochastic methods, especially for finite-sum problems, require the calculation of some full gradients. This makes these techniques less practical because even the infrequent computation of the deterministic gradient can slow down the convergence.

Table 4.1: Summary of the results on projection free methods for stochastic constrained minimization problems.

Reference	Convex case complexity		Non-convex case complexity		No full gradients?	No big batches?
	SFO	LMO	SFO	LMO		
FW [291][(1)] [303][(1)]	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
OFW [310]	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗		✓	✗
CGS [311]	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$	✗		✓	✗
CGS [311] [(1)]	$\mathcal{O}\left(\frac{n}{\sqrt{\varepsilon}}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$	✗		✗	✗
NSFW [304]	✗		$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✓	✗
SVFW [304]	✗		$\mathcal{O}\left(n+\frac{n^{2/3}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
SAGA Fw [304]	✗		$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$ [(2)]	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
SVRF [312]	$\tilde{\mathcal{O}}\left(n+\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$	✗		✗	✗
SNCGS [313]	✗		$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$ [(3)]	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$ [(3)]	✓	✗
NCGS-VR [313]	✗		$\mathcal{O}\left(n+\frac{n^{2/3}}{\varepsilon^2}\right)$ [(3)]	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$ [(3)]	✗	✗
SPIDER-FW [314]	$\mathcal{O}\left(n+\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(n+\frac{\sqrt{n}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
FZFW [315]	✗		$\mathcal{O}\left(n+\frac{\sqrt{n}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
FZCGS [315]	✗		$\mathcal{O}\left(n+\frac{\sqrt{n}}{\varepsilon^2}\right)$ [(3)]	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$ [(3)]	✗	✗
SFW [317]	$\mathcal{O}\left(\frac{1}{\varepsilon^3}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^3}\right)$	✗		✓	✓
OS SFW [323]	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗		✓	✓
SFW [320][(4)]	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$ [(5)]	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$ [(5)]	convergence without rate		✓	✓
SGFW [321][(4)]	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$	✗		✓	✓
MSHFW [318]	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗		✓	✓
SRFW [322]	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✓	✗
SVR RFW [322]	$\mathcal{O}\left(n+\frac{n^{2/3}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(n+\frac{n^{2/3}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
SPIDER RFW [322]	$\mathcal{O}\left(\frac{1}{\varepsilon^3}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^3}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✓	✗
MFW [319]	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\exp\left(\frac{1}{\varepsilon}\right)\right)$	$\mathcal{O}\left(\exp\left(\frac{1}{\varepsilon}\right)\right)$	✓	✓
Alg. 32 (this thesis)	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(\sqrt{n}+\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(n+\frac{\sqrt{n}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
Alg. 32 (this thesis)	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	✗	✓
Alg. 33 (this thesis)	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(\sqrt{n}+\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(n+\frac{\sqrt{n}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✓	✗
Alg. 33 (this thesis)	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	✓	✓

[(1)] fully deterministic; [(2)] the authors give a different complexity, but it seems to us that their proof contains an error, we try to correct it; [(3)] in the original papers, the authors give better results, e.g. $\mathcal{O}(\sqrt{n}/\varepsilon)$ instead of $\mathcal{O}(\sqrt{n}/\varepsilon^2)$, but this results violate the lower bounds (see Table 1 from [54]]), this is due to the difference in the convergence criterion: in [54], the authors use $\|\nabla f\|^2 \sim \varepsilon^2$ and in [313, 315] – $\|\nabla f\|^2 \sim \varepsilon$; [(4)] only for linear models; [(5)] the authors give a rate in the form κ/ε, where κ is a special constant, which is equal to n in the worst case.
Notation: ε = accuracy of the solution, n = size of the dataset, SFO = stochastic first-order oracle, LMO = linear minimization oracle.

Some methods for constrained problems also have this disadvantage [304, 312, 313, 314, 315, 322]. While Algorithm 32 also requires the computation of the full gradient, Algorithm 33 removes this issue and uses only stochastic gradients. Note that this modification does not affect convergence:

Algorithm 32 and Algorithm 33 have the same theoretical guarantees.

• **Small batches.** Many methods that avoid the computation of the full gradient still use large fixed batch sizes [310, 311, 304, 313, 315, 322] or batch sizes that geometrically increase with iteration number [312, 314], which, like the collection of full gradients, is a significant limitation on the practical applicability of the method. Conversely, our algorithms are guaranteed to converge with all sizes of batches. Large batches are required to achieve better dependence in n for the non-convex case. In particular, it is possible to improve the LMO estimate by a factor of n, and the estimate on SFO by a factor of $\sqrt{n}$. Methods dealing with fixed small batches are either only analyzed for linear predictors [320, 321], or exhibit slower convergence rates than our approach [317, 318, 319].

• **Non-convex analysis.** We give convergence results not only for the convex problem, but also in the case where the target function f in (4.4) is non-convex. In this setting, our oracle complexity results are the first to be non-exponential (in ϵ) with small mini-batches, and are state-of-the-art with large mini-batches.

4.1.2 Assumptions

In this section, we use Assumptions 4.0.1–4.0.3, but additionally we add the following to take into account the specificity of the problem (4.4).

Assumption 4.1.1. Each function $f_i : \mathcal{X} \to \mathbb{R}$, $i \in [n]$, is L_i-smooth on $\mathcal{X}$, i.e., there exists a constant $L_i > 0$ such that,

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i \|x - y\|, \quad \forall\, x, y \in \mathcal{X}.$$

We also define the constant $\tilde{L}$ as $\tilde{L}^2 := \frac{1}{n}\sum_{i=1}^n L_i^2$. By convexity of $x \mapsto x^2$, it is easy to prove that $\tilde{L} \geq L$.

4.1.3 Algorithms

In this section, we present two new algorithms along with their convergence guarantees.

State-of-the-art complexity with Sarah Frank-Wolfe

Previously, [304, 312, 322] proposed to modify the classical Frank-Wolfe algorithm (4.2) using the SVRG technique [52]. The essence of these modifications is to change the deterministic gradient in the Conditional Gradient method to some stochastic gradient g^k, e.g., calculated according to the SVRG approach:

$$g^k = \nabla f_{i_k}(x^k) - \nabla f_{i_k}(w^k) + \nabla f(w^k),$$

where i_k is randomly generated from $[n]$, and w^k is rarely taken equal to x^k, much more often equal to w^{k-1}. Therefore, when we update $w^k = x^k$, we sometimes need to consider the full deterministic

Algorithm 32 `Sarah Frank-Wolfe`

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$, probability p, batch size b;
Initialization: choose $x^0 \in \mathcal{X}$; $g^0 = \nabla f(x^0)$;

1: **for** $k = 0, 1, 2, \ldots K-1$ **do**
2: Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle g^k, s\rangle$;
3: Update $x^{k+1} = x^k + \eta_k(s^k - x^k)$
4: Generate batch S_k with size b;
5: Update $g^{k+1} = \begin{cases} \nabla f(x^{k+1}), & \text{with probability } p, \\ g^k + \frac{1}{b}\sum\limits_{i\in S_k}[\nabla f_i(x^{k+1}) - \nabla f_i(x^k)], & \text{with probability } 1-p, \end{cases}$;
6: **end for**

gradient. The update rule for w^k can be deterministic (as in the original version) or randomized, known as the loopless approach [264]. Meanwhile, there are other variance-reduced methods, such as SARAH [53]:

$$g^k = \nabla f_{i_k}(x^k) - \nabla f_{i_k}(x^{k-1}) + g^{k-1}, \tag{4.5}$$

where i_k is also randomly generated from $[n]$, and g^k is rarely taken equal to $\nabla f(x^k)$ rather than (4.5). As noted in the original paper on SARAH, this method has better convergence guarantees and smoother convergence paths with less oscillations than SVRG, making SARAH preferred in both theory and practice. As a result, the SARAH update is also used in the Conditional Gradient method for non-convex problems [314, 315, 322]. In these works, the authors call (4.5) the SPIDER technique. We also use SARAH as a base for Algorithm 32, but unlike [314, 315, 322], its' loopless version [54]. First we give the convergence of Algorithm 32 in the convex case.

Theorem 4.1.2. Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 32 for solving problem (4.4), which satisfies Assumptions 4.0.1,4.0.2,4.0.3,4.1.1. Let x^* be the minimizer of f. Then for any K one can choose $\{\eta_k\}_{k\geq 0}$ as follows:

$$\begin{aligned}
&\text{if } K \leq \frac{2}{p}, && \eta_k = \frac{p}{2}, \\
&\text{if } K > \frac{2}{p} \quad \text{and} \quad k < \left\lceil \frac{K}{2} \right\rceil, && \eta_k = \frac{p}{2}, \\
&\text{if } K > \frac{2}{p} \quad \text{and} \quad k \geq \left\lceil \frac{K}{2} \right\rceil, && \eta_k = \frac{2}{\left(4/p + k - \lceil K/2 \rceil\right)}.
\end{aligned}$$

For this choice of η_k, we have the following convergence:

$$\mathbb{E} f(x^K) - f(x^*) = \mathcal{O}\left(\frac{f(x^0) - f(x^*)}{p} \exp\left(-\frac{Kp}{4}\right) + \left[1 + \frac{\tilde{L}}{L}\sqrt{\frac{1-p}{pb}}\right] \frac{LD^2}{K} \right).$$

We highlight an important detail that the convergence is proved not only in terms of $f(x^K) - f(x^*)$, but also for the Lyapunov function, which additionally includes $\|g^K - \nabla f(x^K)\|^2$. Therefore, Theorem 4.1.2 provides guarantees that $\|g^K - \nabla f(x^K)\|^2 \sim \frac{1}{K}$ and hence g^K becomes a good approximation of the full gradient $\nabla f(x^K)$.

To choose p, one can note that for each iteration, we on average compute the stochastic gradient

$(pn + (1-p)\cdot 2b)$ times: with probability p we need the full gradient, with probability $(1-p)$ – a batch of size b in two points x^{k+1} and x^k. If we take p close to 1, the guarantees in Theorem 4.1.2 gives faster convergence, but the oracle complexity per iteration increases. For example, if we take $p = 1$, we simply obtain a deterministic method, and the estimates for convergence and the number of gradient calculations reproduce the results for the classical Frank-Wolfe method. On the other hand, if p tends to 0, the number of stochastic gradient calls per iteration decreases, but the iterative convergence rate drops. It is optimal to choose p based on the condition: $pn = 2(1-p)b$, i.e. $p = \frac{2b}{n+2b}$. From Theorem 4.1.2 we can also obtain an estimate on the required number of linear minimizations (LMO complexity). It is equal to the number of iterations of Algorithm 32. Then, the following corollary holds.

Corollary 4.1.3. Under the conditions of Theorem 4.1.2, Algorithm 32 with $p = \frac{2b}{n+2b}$ achieves an ε suboptimality in expectation with

$$\mathcal{O}\left(\frac{n}{b}\log\frac{1}{\varepsilon} + \left[1 + \frac{\tilde{L}\sqrt{n}}{bL}\right]\frac{LD^2}{\varepsilon}\right) \text{ LMO calls, and}$$
$$\mathcal{O}\left(n\log\frac{1}{\varepsilon} + \left[b + \frac{\tilde{L}\sqrt{n}}{L}\right]\frac{LD^2}{\varepsilon}\right) \text{ stoch. oracle calls.}$$

For any $b \leq \frac{\tilde{L}\sqrt{n}}{L}$, the estimate of the number of calls for the stochastic oracles does not change. Given that $\tilde{L} \geq L$, the smallest batch size $b = 1$ is appropriate for us. In this setting, the required number of stochastic gradient computations is $\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}\tilde{L}D^2}{\varepsilon}\right)$. This result is the best in the literature on stochastic projection-free methods, especially since it does not require using large batches (see Table 4.1). Meanwhile, in the general case (not necessarily projection-free), these estimates can be explicitly improved up to $\tilde{\mathcal{O}}\left(n + \sqrt{\frac{n\tilde{L}D^2}{\varepsilon}}\right)$ [324]. Whether this is possible in the case of Frank-Wolfe type methods is an attractive question for future consideration. For $p = \frac{2b}{n+2b}$ and $b = \sqrt{n}$, the LMO complexity is $\tilde{\mathcal{O}}\left(\sqrt{n} + \frac{\tilde{L}D^2}{\varepsilon}\right)$, this result is optimal up to an additional factor $\sqrt{n}$ (see Section 2.1.2 from [302]). With $b = 1$, the LMO complexity equals $\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}\tilde{L}D^2}{\varepsilon}\right)$. Note that for many practical examples, the LMO complexity is not the computational bottleneck since the solution of linear minimization problems has a closed-form solution (see, e.g., Algorithm 2 from [325]). It is also important to notice that, based on the above choices for η_k, p, and b, both our method and the original Frank-Wolfe are independent of the objective function parameters (e.g., L or $\tilde{L}$).

Next, we prove the convergence of Algorithm 32 for the non-convex objective function f. We use the (4.3) as a criterion for convergence and have the following theorem .

Theorem 4.1.4. Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 32 for solving problem (4.4), which satisfies Assumptions 4.0.1,4.0.3,4.1.1. Let x^* be the global (may be not unique) minimizer

of f. Then, if we choose $\eta_k = \frac{1}{\sqrt{K}}$, we have the following convergence:

$$\mathbb{E}\left[\min_{0\leq k\leq K-1} \text{gap}(x^k)\right] = \mathcal{O}\left(\frac{f(x^0) - f(x^*)}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left[1 + \frac{\tilde{L}}{L}\sqrt{\frac{(1-p)}{pb}}\right]\right).$$

In this case, the optimal choice of the parameter p is the same as in Corollary 4.1.3 of Theorem 4.1.2.

Corollary 4.1.5. Under the conditions of Theorem 4.1.4, Algorithm 32 with $p = \frac{2b}{n+2b}$ achieves an ε suboptimality in expectation with

$$\mathcal{O}\left(\left[\frac{h^0}{\varepsilon}\right]^2 + \left[\frac{LD^2}{\varepsilon}\right]^2\left[1 + \frac{\tilde{L}^2 n}{L^2 b^2}\right]\right) \text{ LMO calls, and}$$
$$\mathcal{O}\left(b\left[\frac{h^0}{\varepsilon}\right]^2 + \left[\frac{LD^2}{\varepsilon}\right]^2\left[b + \frac{\tilde{L}^2 n}{L^2 b}\right]\right) \text{ stoch. oracle calls,}$$

where $h^0 := f(x^0) - f(x^*)$.

First, we substitute $b = 1$ in the previous result. This gives us an oracle complexity of $\mathcal{O}\left(\frac{n(\tilde{L}^2+L^2)D^4+(h^0)^2}{\varepsilon^2}\right)$, which corresponds to the Frank-Wolfe complexity. If we choose the batch size $b = \frac{\sqrt{n}\tilde{L}}{L}$ that minimizes the expression $\left[b + \frac{\tilde{L}^2 n}{L^2 b}\right]$, then we need $\mathcal{O}\left(n + \frac{\sqrt{n}\tilde{L}[L^2D^4+(h^0)^2]}{L\varepsilon^2}\right)$ stochastic gradient calls. If one wishes to avoid using $\tilde{L}$ and L constants when selecting b, it is possible to take $b = \sqrt{n}$, and the complexity then becomes $\mathcal{O}\left(n + \frac{\sqrt{n}[(\tilde{L}^2+L^2)D^4+(h^0)^2]}{\varepsilon^2}\right)$. As noted earlier, both of these estimates are the best result of the projection-free methods for the non-convex setup (see Table 4.1). Moreover, they are optimal according to the lower estimates from Table 1 of [54]. Additionally, with $p = \frac{2b}{n+2b}$ and $b = 1$, the LMO complexity is equal to $\mathcal{O}\left(\frac{n(\tilde{L}^2+L^2)D^4+(h^0)^2}{\varepsilon^2}\right)$, and with $b = \sqrt{n}$, it is $\mathcal{O}\left(\frac{(\tilde{L}^2+L^2)D^4+(h^0)^2}{\varepsilon^2}\right)$. In terms of lower bounds, only the already mentioned LMO estimates $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ for the convex setting are known. One important detail to note is that in both the convex (see the discussion after Corollary 4.1.3) and non-convex cases, LMO estimates is better when the batches are chosen quite large.

Avoiding full gradient computations with Saga Sarah Frank-Wolfe

The idea of Algorithm 33 is to use a combination [308] of the SARAH and SAGA [55] approaches. Both SAGA and SARAH are some of the main variance-reduced methods for finite-sum minimization problems. An important feature of SAGA is that it does not use full gradient calculations, but it has worse convergence guarantees than SARAH (see, e.g., Table 2 in [53]). The synergy of SARAH and SAGA brings together the strengths of both methods.

The essence of the SAGA method is similar to SVRG, but while SVRG collects the full gradients at some reference points, SAGA instead maintains a "sliding" version of the full gradient. The gradient $\nabla f(w)$ at the reference point w may become obsolete after a small number of iterations. However, in the course of the algorithm, we compute newer stochastic gradients for some $i \in [n]$,

Algorithm 33 `Saga Sarah Frank-Wolfe`

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$; momentum λ; batch size b;
Initialization: choose $x^0 \in \mathcal{X}$; $g^0 = \nabla f(x^0)$ or $g^0 = \nabla f_{i_0}(x^0)$; $y_i^0 = \nabla f_i(x^0)$ or $y_i^0 = 0$;

1: **for** $k = 0, 1, 2, \ldots K-1$ **do**
2: Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle g^k, s\rangle$;
3: Update $x^{k+1} = x^k + \eta_k(s^k - x^k)$
4: Generate batch S_k with size b
5: Update $g^{k+1} = \frac{1}{b}\sum\limits_{i\in S_k}[\nabla f_i(x^{k+1}) - \nabla f_i(x^k)] + (1-\lambda)g^k + \lambda\left(\frac{1}{b}\sum\limits_{i\in S_k}[\nabla f_i(x^k) - y_i^k] + \frac{1}{n}\sum\limits_{j=1}^{n} y_j^k\right)$;
6: Update $y_i^{k+1} = \begin{cases}\nabla f_i(x^{k+1}), & i \in S_k,\\ y_i^k, & i \notin S_k,\end{cases}$;
7: **end for**

allowing us to leverage them to calculate a more recent approximation of the full gradient. To achieve this, SAGA introduces additional vectors $\{y\}_{i=1}^n$; each such y_i keeps the latest version of the gradient ∇f_i (implemented in line 6). The $\frac{1}{n}\sum_{j=1}^n y_j^k$ term is the aforementioned approximation of the full gradient. The calculation of g_{SAGA}^k is $g_{\text{SAGA}}^k = \left(\frac{1}{b}\sum_{i\in S_k}[\nabla f_i(x^k) - y_i^k] + \frac{1}{n}\sum_{j=1}^n y_j^k\right)$ (similarly to SVRG and SARAH). Line 5 provides a combination of g_{SARAH}^k and g_{SAGA}^k.

Recall that the average number of stochastic oracle calls per iteration of Algorithm 32 is $(pn + (1-p)\cdot 2b)$, and that Algorithm 33 requires $2b$ computations of the stochastic gradients each iteration. In lines 5 and 6, one needs to calculate $\nabla f_i(x^{k+1})$, $\nabla f_i(x^k)$ for $i \in S_k$. Therefore, if $b \leq \frac{n}{2}$, then for any $p \neq 0$, the complexity of one iteration of Algorithm 33 is better than that of Algorithm 32.

In summary, Algorithm 33 does not collect full gradients and has slightly better iteration complexity, but it requires the use of n extra vectors $\{y_i\}_{i=1}^n$, leading to an additional $\mathcal{O}(nd)$ memory cost compared to Algorithm 32. One can note that the methods from [320, 321] also use an extra memory size of $\mathcal{O}(nd)$.

For the convex target function f, Algorithm 33 satisfies the following convergence theorem.

Theorem 4.1.6. Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 33 for solving problem (4.4), which satisfies Assumptions 4.0.1–4.0.3. Let x^* be the minimizer of f. Then for any K one can choose $\{\eta_k\}_{k\geq 0}$ as follows:

$$\begin{aligned}
&\text{if } K \leq \frac{4n}{b}, && \eta_k = \frac{b}{4n},\\
&\text{if } K > \frac{4n}{b} \quad \text{and} \quad k < \left\lceil \frac{K}{2} \right\rceil, && \eta_k = \frac{b}{4n},\\
&\text{if } K > \frac{4n}{b} \quad \text{and} \quad k \geq \left\lceil \frac{K}{2} \right\rceil, && \eta_k = \frac{2}{\left({}^{8n}/_{b} + k - \lceil {}^{K}/_{2} \rceil\right)},
\end{aligned}$$

and $\lambda = \frac{b}{2n}$. For this choice of η_k and λ, we obtain the following convergence:

$$\mathbb{E}f(x^K) - f(x^*) = \mathcal{O}\left(\frac{n\left[f(x^0) - f(x^*)\right]}{b}\exp\left(-\frac{bK}{8n}\right) + \left[1 + \frac{\tilde{L}\sqrt{n}}{Lb}\right]\frac{LD^2}{K}\right).$$

As in the case of Theorem 4.1.2, here the proof is also obtained in terms of the Lyapunov function containing $\|g^K - \nabla f(x^K)\|^2$, we can guarantee that g^K tends to $\nabla f(x^K)$. The guarantees of Theorem 4.1.6 depend on $\|g^0 - \nabla f(x^0)\|^2$ and $\sum_{i=1}^n \|y_i^0 - \nabla f_i(x^0)\|$, but because of initialization we put them equal to 0 again. Since there is no need to choose p for Algorithm 33 we proceed directly to the corollary on the oracle complexity.

Corollary 4.1.7. Under the conditions of Theorem 4.1.6, Algorithm 33 achieves an ε suboptimality in expectation with

$$\mathcal{O}\left(\frac{n}{b}\log\frac{1}{\varepsilon} + \left[1 + \frac{\tilde{L}\sqrt{n}}{bL}\right]\frac{LD^2}{\varepsilon}\right) \text{ LMO calls, and}$$
$$\mathcal{O}\left(n\log\frac{1}{\varepsilon} + \left[b + \frac{\tilde{L}\sqrt{n}}{L}\right]\frac{LD^2}{\varepsilon}\right) \text{ stoch. oracle calls.}$$

This result is exactly the same as Corollary 4.1.3, so we obtain the same conclusions for choosing the size of b as that Corollary 4.1.3. In particular, in this case with $b = 1$, the method also have oracle complexity $\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}\tilde{L}D^2}{\varepsilon}\right)$ – the best among the works on stochastic projection-free methods. The findings on the LMO complexity is also consistent with Theorem 4.1.2. The reasoning around optimality is also consistent with that given after Corollary 4.1.3. Note also that Algorithm 33, just the same as Algorithm 32 and the classical Frank-Wolfe method, is free of the target function's parameters.

In the following theorem for the non-convex case of f, as in Theorem 4.1.4, we use (4.3) to estimate convergence.

Theorem 4.1.8. Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 33 for solving problem (4.4), which satisfies Assumptions 4.0.1,4.0.3. Let x^* be the global (may not be unique) minimizer of f on $\mathcal{X}$. Then, if we choose $\eta_k = \frac{1}{\sqrt{K}}$ and $\lambda = \frac{b}{2n}$, we have the following convergence:

$$\mathbb{E}\left[\min_{0\leq k\leq K-1}\text{gap}(x^k)\right] = \mathcal{O}\left(\frac{f(x^0) - f(x^*)}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left[1 + \frac{\tilde{L}\sqrt{n}}{Lb}\right]\right).$$

Corollary 4.1.9. Under the conditions of Theorem 4.1.8, Algorithm 33 achieves an ε suboptimality in expectation with

$$\mathcal{O}\left(\left[\frac{h^0}{\varepsilon}\right]^2 + \left[\frac{LD^2}{\varepsilon}\right]^2\left[1 + \frac{\tilde{L}^2 n}{L^2 b^2}\right]\right) \text{ LMO calls, and}$$

$$\mathcal{O}\left(b\left[\frac{h^0}{\varepsilon}\right]^2 + \left[\frac{LD^2}{\varepsilon}\right]^2\left[b + \frac{\tilde{L}^2 n}{L^2 b}\right]\right) \text{ stoch. oracle calls,}$$

where $h^0 := f(x^0) - f(x^*)$.

We once again have the same results as for Algorithm 32 in Corollary 4.1.5, resulting in the same analysis for choosing the batch sizes and for the LMO complexity.

4.1.4 Experiments

We conduct our experiments on the constrained empirical risk for a linear model with weights x and on training samples $\{a_i, b_i\}_{i=1}^n$. Then, the logistic regression problem is

$$\min_{x\in\mathcal{X}} f(x) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-b_i x^T a_i)), \tag{4.6}$$

where $y_i \in \{-1, 1\}$, and the non-linear least squares loss is

$$\min_{x\in\mathcal{X}} f(x) = \frac{1}{n}\sum_{i=1}^{n} (b_i - 1/(1 + \exp(x^T a_i)))^2, \tag{4.7}$$

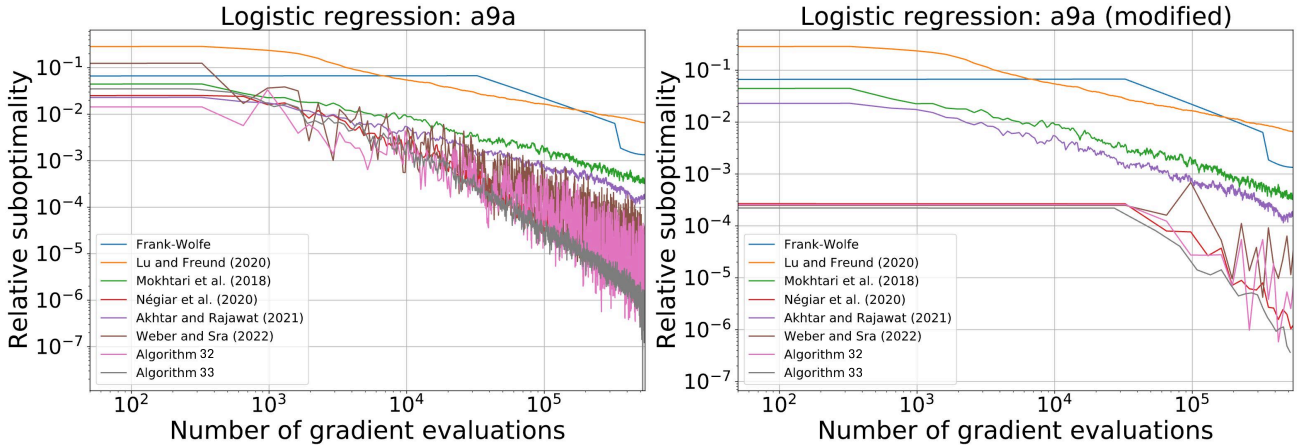
with $b_i \in \{0, 1\}$. We consider two different loss functions to test our algorithms on both convex and nonconvex settings. We choose $\mathcal{X}$ as the ℓ_1 norm ball with radius $R = 2 \cdot 10^3, 200, 20, 2$. The LMO for such a constraint set can be computed in a closed-form solution. We take LibSVM [171] datasets (see Table 4.2).

Table 4.2: Datasets from LibSVM in experiments.

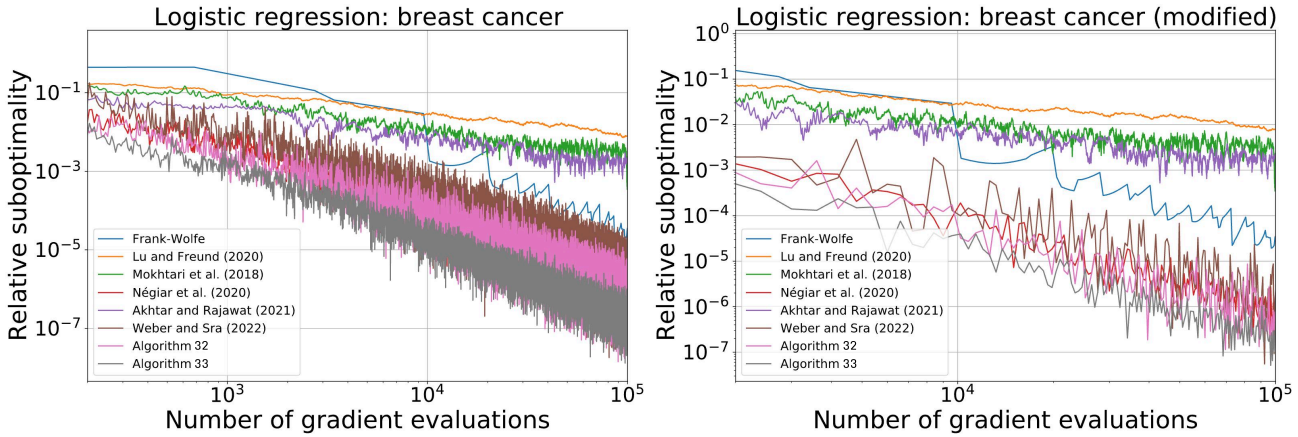
Dataset	d	n
`a9a`	123	22696
`breast cancer`	10	683
`madelon`	500	2000
`mushrooms`	112	8124
`w8a`	300	49749

For comparison, we consider the methods from Table 4.1, which do not use large batches: [317, 320, 321, 318, 322]. Our method is tuned according to the theory (see Sections 4.1.3 and 4.1.3), but we take the batchsize $b = \left\lceil \frac{n}{100} \right\rceil$ (similarly as the baselines). For the baselines, we use the implementation of [320] and tune each method accordingly.

In Figures, we plot the relative suboptimality, which is defined as $^{(f(x_t)-f_{\min})}/_{(f_{\max}-f_{\min})}$, where $f_{\max}$ is the largest value observed during the optimization and $f_{\min}$ is obtained by running the best algorithm slightly longer than what is plotted. From our results, it is clear that our algorithms are superior or comparable to the baselines, despite the fact that some methods were specifically designed for linear models (e.g. [320, 321]), which is not the case for Algorithm 32 and 33.

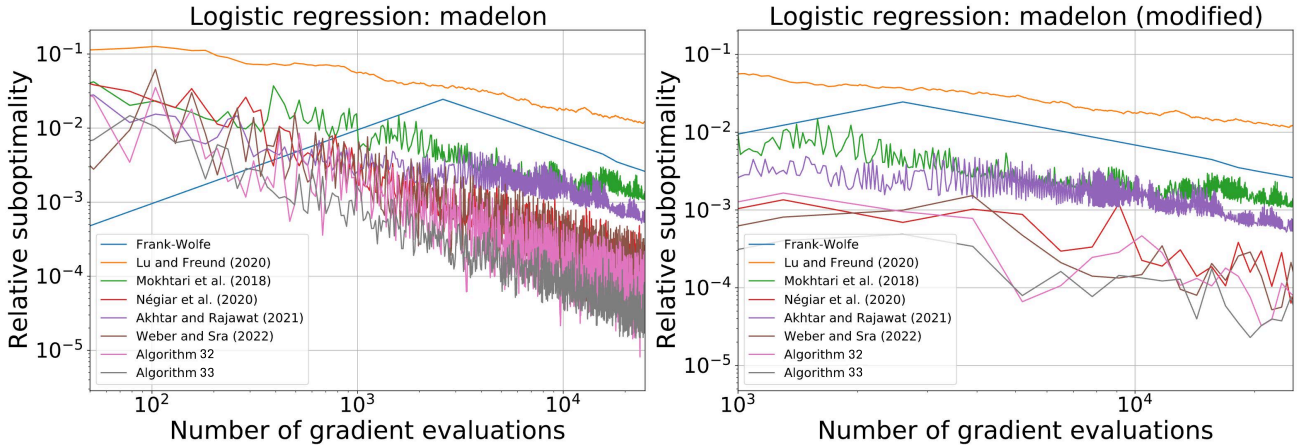


(a) a9a

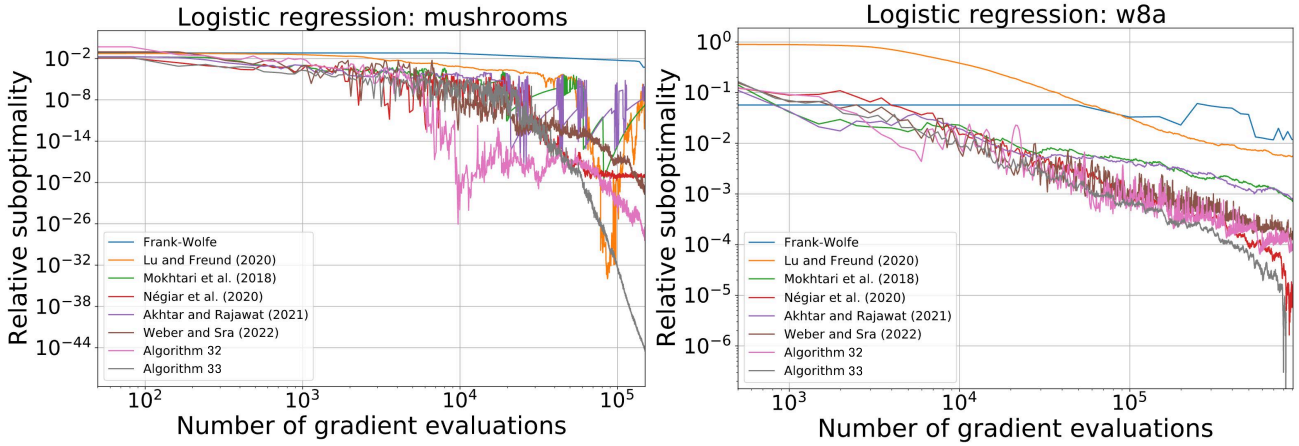


(b) `breast cancer`

Figure 4.1: Comparison of state-of-the-art projection free methods with small batches for (4.6) with $R = 2000$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. In the modified plots (the right plots in the first three lines), we left only every 100th point for [320], [322], Algorithm 32 and Algorithm 33.

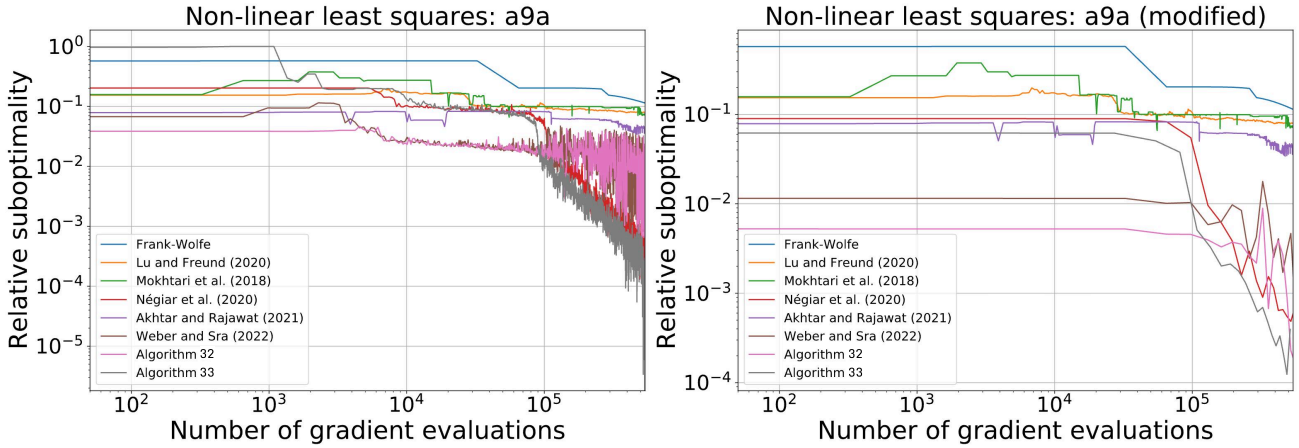


(c) `madelon`

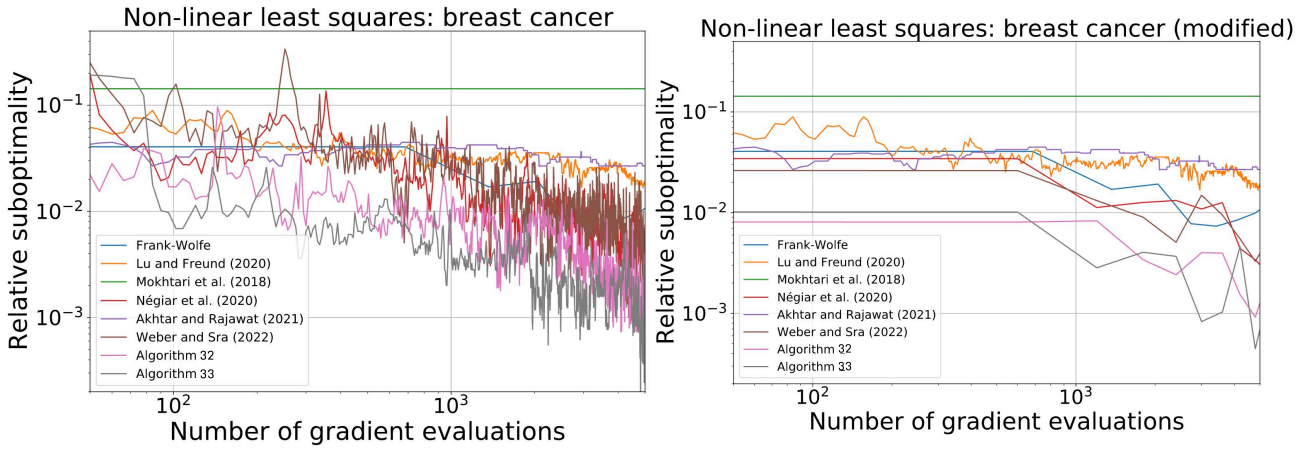


(d) `mushrooms` (e) `w8a`

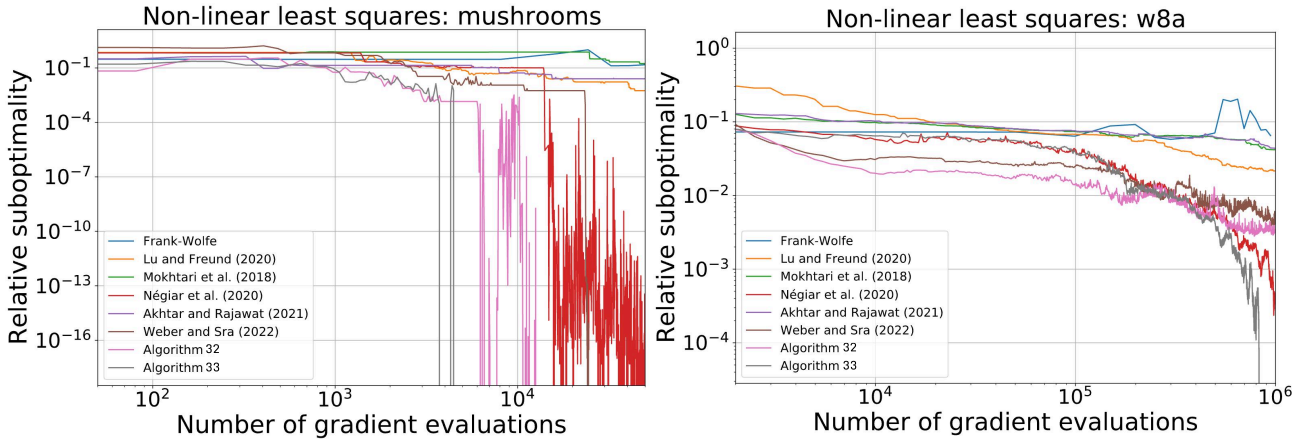
Figure 4.2: Comparison of state-of-the-art projection free methods with small batches for (4.6) with $R = 2000$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. In the modified plots (the right plots in the first three lines), we left only every 100th point for [320], [322], Algorithm 32 and Algorithm 33.



(a) a9a

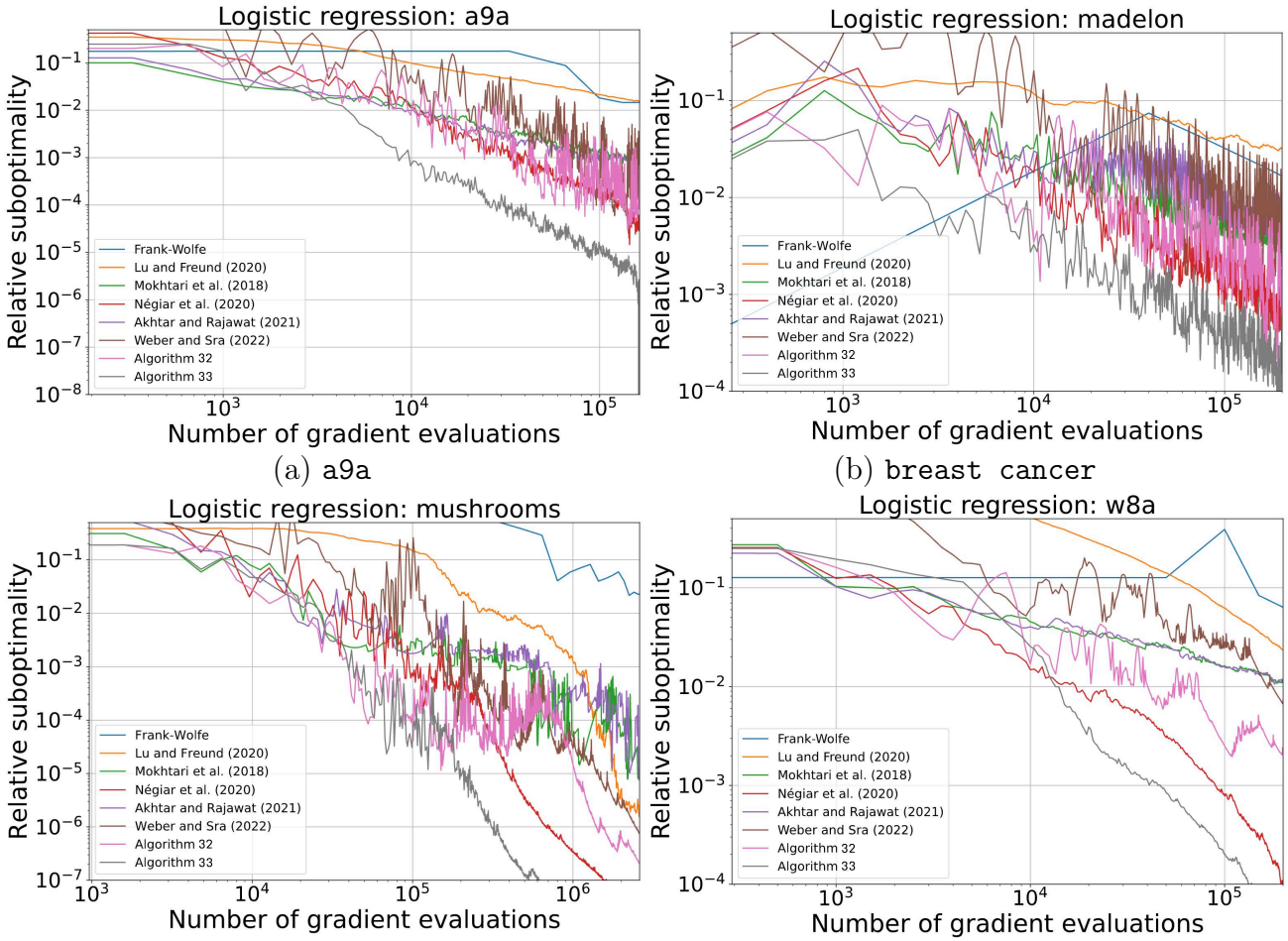


(b) `breast cancer`



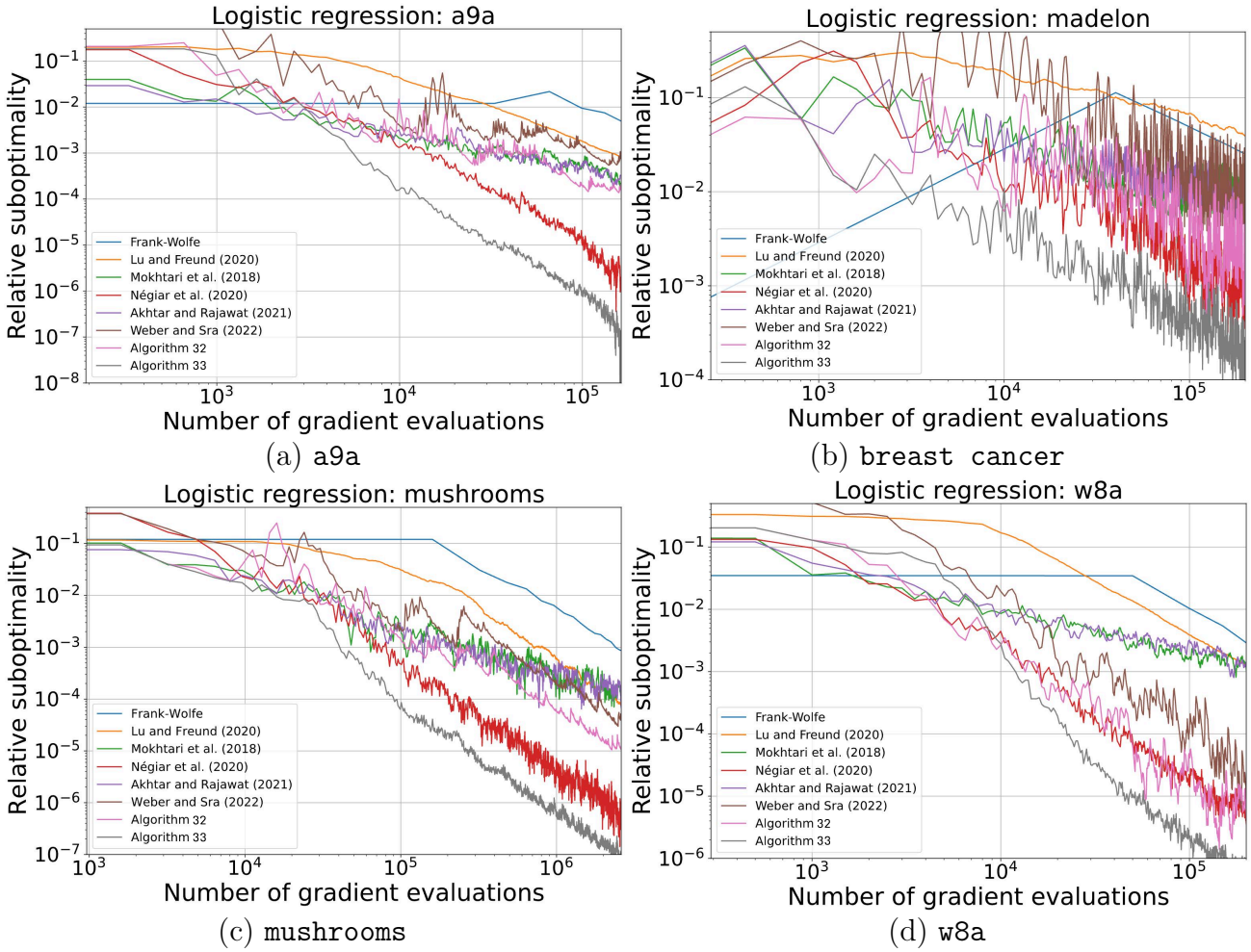
(c) `mushrooms` (d) `w8a`

Figure 4.3: Comparison of state-of-the-art projection free methods with small batches for (4.7) with $R = 2000$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. In the modified plots (the right plots in the first three lines), we left only every 100th point for [320], [322], Algorithm 32 and Algorithm 33.



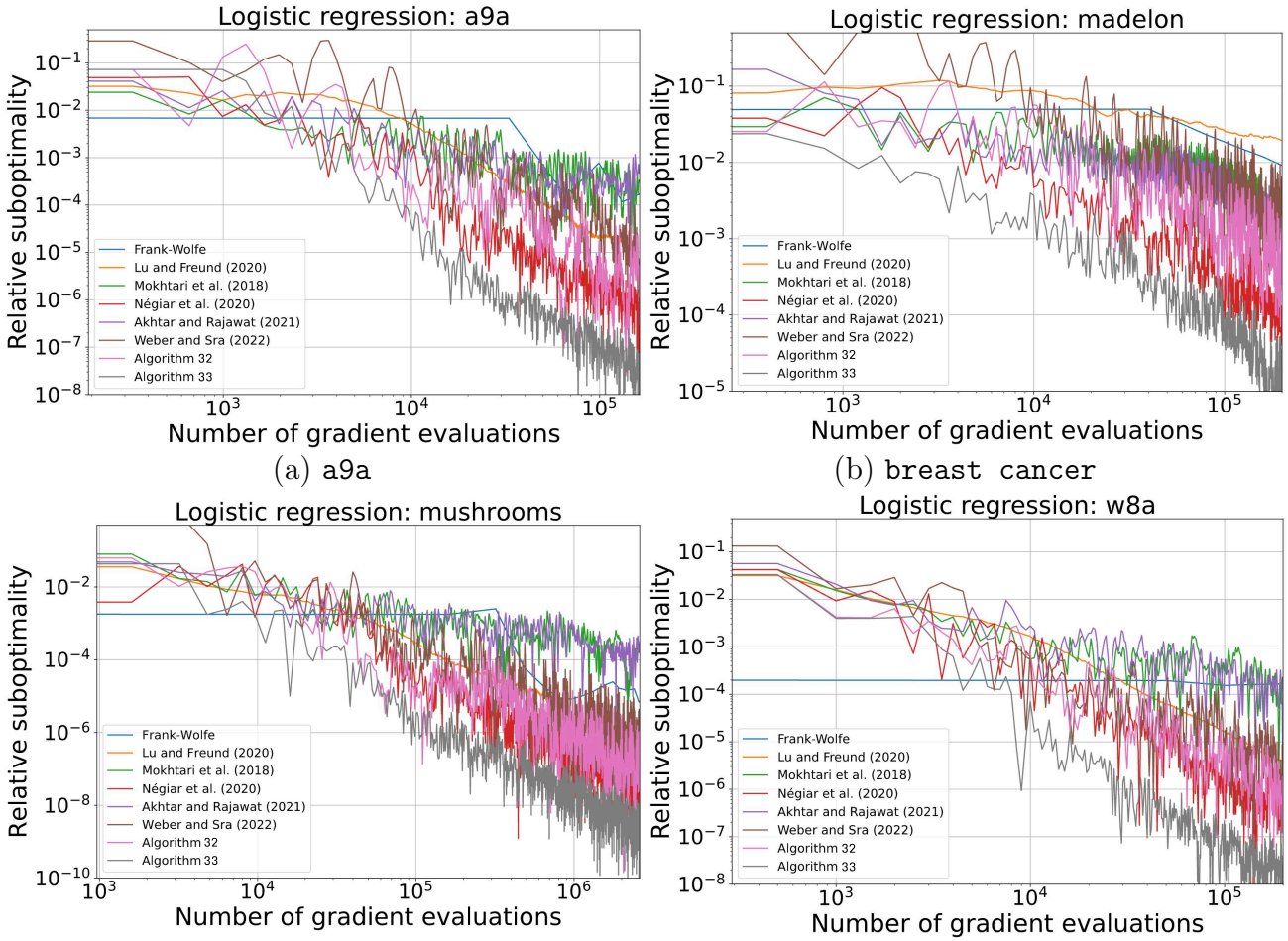
(a) a9a (b) `breast cancer` (c) `mushrooms` (d) `w8a`

Figure 4.4: Comparison of state-of-the-art projection free methods with small batches for (4.6) with $R = 200$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations.



(a) a9a (b) `breast cancer` (c) `mushrooms` (d) `w8a`

Figure 4.5: Comparison of state-of-the-art projection free methods with small batches for (4.6) with $R = 20$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations.



(a) a9a (b) `breast cancer`

(c) `mushrooms` (d) `w8a`

Figure 4.6: Comparison of state-of-the-art projection free methods with small batches for (4.6) with $R = 2$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations.

4.2 Zero-Order Frank-Wolfe with Memory

In the original Frank-Wolfe paper [291], the authors used the true gradient in their algorithm, but modern problems require the use of other gradient estimators. In particular, sometimes there are complicated setups where the gradient cannot be computed generally due to various reasons, such as the target function being non-differentiable or the computational difficulty of gradient evaluation [326, 327, 69]. This setting is called black-box optimization [328], and in this case, we are forced to use zero-order (ZO) gradient estimation methods through finite differences of the objective function (sometimes with additional noise) to approximate the gradient [67, 329]. This scenario frequently arises in machine learning tasks, such as attacks on deep neural networks [327, 330, 331] or reinforcement learning [332, 333].

Over the last few years of research on the topic of black-box optimization, we can outline two main methods for approximating the gradient using finite differences. The first estimates the gradient in m coordinates [334, 335, 336]:

$$\frac{d}{m}\sum_{i\in I}\frac{f(x+\tau e_i)-f(x-\tau e_i)}{2\tau}e_i., \tag{4.8}$$

where $I \subset [d] : |I| = m$ and τ is a smoothing parameter.

This finite difference approximates the gradient in m coordinates and requires $\mathcal{O}(m)$ oracle calls. If m is small, this estimation becomes inaccurate; if m is large then we need to make many zero-order oracle calls at each iteration. In the case of $m = d$, we call this method *full-approximation*.

Another finite difference does not use the standard basis, but random vectors e [329, 69, 337, 70, 338]:

$$d\frac{f(x+\tau e)-f(x-\tau e)}{2\tau}e, \tag{4.9}$$

where e can be uniformly distributed on a l_p-sphere $RS_p^d(1)$, then this scheme is called l_p-*smoothing*. In recent papers, authors usually use $p = 1$ [339, 340] or $p = 2$ [33, 341, 342]. Alternatively, e could be sampled from a normal distribution with zero mean and a unit covariance matrix [69].

The approximations (4.8) and (4.9) exhibit a very large variance or require numerous calls to the zero-order oracle. Therefore, there is a necessity to reduce the approximation error without increasing the number of oracle calls. In stochastic optimization, the method of remembering information from previous iterations is widely used. For example, in SEGA [59], authors suggest memorizing the gradient from previous iterations to improve the convergence of the method. We decided to employ this technique in a black-box optimization problem, memorizing the gradient approximations from previous iterations to reduce batch size without significant loss of accuracy.

4.2.1 Related Works and Contributions

To start with, we compare the problem statements and approximation methods in the literature on zero-order methods in the Frank-Wolfe-based algorithms. Some authors consider coordinate methods [343] these are also gradient approximations, but these methods use the true gradient of the observed function f; therefore, we cannot use them directly in black-box optimization. In general, the l_p-smoothing technique can approximate the gradient using $\mathcal{O}(1)$ oracle calls [344], but it may not be robust in the Frank-Wolfe setting, since in [345] authors have to collect large batches of directions e to achieve convergence. Note that in [345] the non-stochastic noise is taken into account. The full-approximation is also used in the literature [346, 315, 347], but at each iteration we need to make $\mathcal{O}(d)$ oracle calls, and since d is huge in modern applications, this can be a problem. Also, this method requires the smoothness of the objective function f. We summarize and compare the problem statements, approximation methods, and results for them in Table 4.3.

Table 4.3: Adjustment of the various zero-order and coordinate FW methods.

Method	Setting		Noise		Batch size	Approximation
	Smooth	Zero-order	Stochastic	Deterministic		
ZO-SCGS [345]	✗	✓	✗	✓	$\mathcal{O}\left(1/\varepsilon^2\right)$	l_2-smoothing (4.9)
FZFW [315]	✓	✓	✗	✗	$\mathcal{O}\left(\sqrt{d}\right)$	Full-approximation (4.8)
DZOFW [346]	✓	✓	✗	✗	$\mathcal{O}\left(d\right)$	Full-approximation (4.8)
MOST-FW [347]	✓	✓	✗	✗	$\mathcal{O}\left(d\right)$	Full-approximation (4.8)
BCFW [343]	✓	✗	✗	✗	$\mathcal{O}\left(1\right)$	Coordinate
SSFW [4]	✓	✗	✗	✗	$\mathcal{O}\left(1\right)$	Coordinate
`FW via JAGUAR` (this thesis)	✓	✓	✓	✓	$\mathcal{O}\left(1\right)$	`JAGUAR` (Algorithms 34 and 38)

Our contributions can be summarized as follows:

• **New approximation.** We present the `JAGUAR` method, which approximates the true gradient of the objective function $\nabla f(x)$ at the point x. By using the memory of previous iterations, we achieve an accuracy close to the full-approximation (4.8), but `JAGUAR` does not require $\mathcal{O}(d)$, but $\mathcal{O}(1)$ calls to the zero-order oracle. l_p-smoothing (4.9) also requires $\mathcal{O}(1)$ oracle calls; however, since there is no memory technique in this method, it has large variance and is not robust.

• **Theory of convergence.** We prove theoretical estimates for this method. We consider both deterministic and stochastic noises in the zero-order oracle. While the first setting is somehow obtained in the literature [345, 348], the second is rarely considered by authors. Therefore, our method is suitable for various black-box optimization problems.

• **Stochastic case.** We implement the `JAGUAR` approximation in the Frank-Wolfe algorithm for stochastic and deterministic minimization problems and prove its convergence in both cases.

• **Extension to Gradient Descent.** We also implement `JAGUAR` in the Gradient Descent method and prove its convergence in the non-convex and PL-condition cases (see Section 4.2.2).

• **Experiments.** We perform several computational experiments, comparing the JAGUAR approximation with l_2-smoothing (4.9) and full-approximation (4.8) on different minimization problems.

4.2.2 Algorithms

We assume that we only have access to the zero-order oracle, which returns the noisy value of the function $f(x)$: $f_\delta(x) := f(x) + \delta(x)$. Therefore, we make a common assumption regarding this noise [344, 348].

Assumption 4.2.1 (Bounded oracle noise)**.** The noise in the zero-order oracle is bounded by a constant $\Delta > 0$, i.e. there exists $\Delta > 0$ such that for all $x \in \mathcal{X}$ it holds $|\delta(x)|^2 \leq \Delta^2$.

JAGUAR gradient approximation. Deterministic case

Algorithm 34 JAGUAR gradient approximation. Deterministic case

Initialization: $x, h \in \mathbb{R}^d$

1: Sample $i \in [d]$ independently and uniform
2: Compute $\widetilde{\nabla}_i f_\delta(x) = \frac{f_\delta(x+\tau e_i) - f_\delta(x-\tau e_i)}{2\tau} e_i$
3: $h = h - \langle h, e_i \rangle e_i + \widetilde{\nabla}_i f_\delta(x)$

Above, we reviewed the gradient approximation techniques using finite differences (4.8) and (4.9). In this section, we introduce a new gradient estimation technique JAGUAR (Algorithm 34), which is based on the already investigated methods and utilizes the memory of previous iterations.

The idea behind the JAGUAR method is similar to well-known variance reduction techniques such as SAGA [55] and SVRG [52]. However, in zero-order optimization, we need to approximate the gradient; therefore, we need to apply the variance reduction technique to the coordinates [59]. Consequently, the JAGUAR method uses the memory of some coordinates of the previous gradients instead of memorizing gradients by batches at past points.

There are already works in the literature that combine zero-order optimization and variance reduction. However, the essence of these papers lies in modifying the gradient calculation to a gradient-free approximation (4.8) in batch variance-reduced algorithms such as SVRG or SPIDER [349], instead of applying the variance reduction technique for coordinates as in Algorithm 34.

JAGUAR approximation algorithm can be used with any iterative scheme that returns a new point x^k at each step k. Using these points, we obtain the sequence h^k in line 3, which in a sense serves as a memory of the gradient components from past moments. Therefore, it is reasonable to use h^k as an estimator of the true gradient $\nabla f(x^k)$ in incremental optimization methods. Using the following unified scheme, we can describe such an iterative algorithm that solves (4.1) (Algorithm 35).

Algorithm 35 Iterative algorithm using gradient estimator via `JAGUAR`

Initialization: choose h^0 and for `Proc`

Input: same as for `Proc` and h^0

for $k = 0, 1, 2, ..., K-1$ **do**

  $h^{k+1} =$ `JAGUAR`(x^k, h^k)

  $x^{k+1} =$ `Proc`$(x^k,$ `grad_est` $= h^{k+1})$

end for

In Algorithm 35, `Proc`$(x^k,$ `grad_est`$)$ is a sequence of actions that translates x^k to x^{k+1} using `grad_estimator` as the true gradient. Now we start to analyze `JAGUAR` gradient approximation (Algorithm 34). Our goal is to estimate the closeness of the true gradient $\nabla f(x^k)$ and the output of the `JAGUAR` algorithm h^k at step k.

Lemma 4.2.2. For x^k and h^k, generated by Algorithm 35, the following inequality holds:

$$H_{k+1} \leq \left(1 - \frac{1}{2d}\right) H_k + 2d\mathbb{E}\left[\left\|\nabla f(x^{k+1}) - \nabla f(x^k)\right\|^2\right] + \mathbb{E}\left[\left\|\widetilde{\nabla} f_\delta(x^k) - \nabla f(x^k)\right\|^2\right], \quad (4.10)$$

where we use notations $H_k := \mathbb{E}\left[\left\|h^k - \nabla f(x^k)\right\|^2\right]$ and

$$\widetilde{\nabla} f_\delta(x) := \sum_{i=1}^{d} \frac{f_\delta(x + \tau e_i) - f_\delta(x - \tau e_i)}{2\tau} e_i. \quad (4.11)$$

Discussion. We do not need to make any assumptions to satisfy Lemma 4.2.2, since in its proof we only use the form of Algorithm 34. This means that the performance of the `JAGUAR` approximation depends only on the quality of the full-approximation $\widetilde{\nabla} f_\delta(x)$ and the closeness of the points x^{k+1} and x^k, generated by Algorithm 35.

Let us analyze the formula (4.10), and show that using `JAGUAR` gradient approximation (Algorithm 34) gives us the same estimates as using $\widetilde{\nabla} f_\delta(x)$. Many optimization algorithms use a sufficiently small step size γ; therefore, `Proc`$(x^k,$`grad_est`$) \approx x^k$.. We can assume that $\nabla f(x^{k+1}) \approx \nabla f(x^k)$ (for a specific choice of γ, see Theorem 4.2.3). We can unroll (4.10) and, if we consider $k \gg d$, then we can obtain that $\mathbb{E}[\|h^k - \nabla f(x^k)\|^2] = \mathcal{O}(\|\widetilde{\nabla} f_\delta(x^k) - \nabla f(x^k)\|^2)$, i.e. it is the same estimate as for the full-approximation (4.11). However, now we make $\mathcal{O}(1)$ oracle calls at each iteration.

Next, we implement the `JAGUAR` approximation in the Frank-Wolfe algorithm and in Gradient Descent.

Frank-Wolfe via `JAGUAR`

In this section, we introduce the Frank-Wolfe algorithm, which solves the problem (4.1) using the `JAGUAR` gradient approximation (Algorithm 36).

Algorithm 36 `FW via JAGUAR`. Deterministic case

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$, $\tau > 0$;
Initialization: choose $x^0 \in \mathcal{X}$; $h^0 = \widetilde{\nabla} f_\delta(x^0)$;
for $k = 0, 1, 2, ..., K-1$ **do**
  $h^{k+1} =$ `JAGUAR` $\left(x^k, h^k\right)$
  $s^k = \arg\min_{s\in\mathcal{X}} \langle s, h^{k+1}\rangle$
  $x^{k+1} = x^k + \eta_k(s^k - x^k)$
end for

Using a certain form of the `Proc` function in Algorithm 36, we can unroll the results of Lemma 4.2.2 to carefully tune the step size η_k.

Theorem 4.2.3 (Step tuning for `FW via JAGUAR` (Algorithm 36). Deterministic case)**.** Consider Assumptions 4.0.1 and 4.0.3. For h^k, generated by Algorithm 36, we can take $\eta_k = 4/(k+8d)$, then the following inequality holds:

$$\mathbb{E}\left[\|h^k - \nabla f(x^k)\|^2\right] = \mathcal{O}\left(\|\widetilde{\nabla} f_\delta(x^k) - \nabla f(x^k)\|^2 + \frac{d^2L^2D^2}{(k+8d)^2}\right).$$

From Theorem 4.2.3 we can conclude that after $\mathcal{O}\left(\sqrt{d}D/\tau\right)$ steps we get the same estimate as in the full-approximation (4.11). We now examine the convergence of Algorithm 36.

Theorem 4.2.4 (Convergence rate of `FW via JAGUAR` (Algorithm 36))**.** Consider Assumptions 4.0.1, 4.0.2 and 4.0.3, 4.2.1. If we take $\eta_k = 4/(k+8d)$, then `FW via JAGUAR` (Algorithm 36) has the following convergence rate

$$\mathbb{E}\left[f(x^K) - f(x^*)\right] = \mathcal{O}\left(\frac{d\max\{LD^2, F_0\}}{K+8d} + \sqrt{d}LD\tau + \frac{\sqrt{d}\Delta D}{\tau}\right),$$

The results of Theorem 4.2.4 can be rewritten as an upper complexity bound on a number of iterations of Algorithm 36, using the appropriate smoothing parameter τ and noise bound Δ.

Corollary 4.2.5. Under the conditions of Theorem 4.2.4, choosing $\eta_k, \tau(\varepsilon), \Delta(\varepsilon)$ in a some right way in order to achieve an ε-approximate solution (in terms of $\mathbb{E}[f(x^K) - f(x^*)] \leq \varepsilon$) it takes

$$\mathcal{O}\left(\frac{d\max\{LD^2, f(x^0) - f(x^*)\}}{\varepsilon}\right) \text{ iterations of Algorithm 36.}$$

Discussion. The results of Theorem 4.2.4 coincide with results [291, 296] in which the authors employed a true gradient and obtained the result of the form

$$\mathbb{E}\left[f(x^K) - f^*\right] = \mathcal{O}\left(\frac{\max\{LD^2; f(x^0) - f(x^*)\}}{K}\right).$$

In the zero-order case, terms of the form $\mathcal{O}\left(\text{poly}(\tau) + \text{poly}(\Delta/\tau)\right)$ are inevitable, as they are crucial

for the approximation of the true gradient and always influence the convergence of zero-order methods [350, 351, 352]. The factor d, which appears in our theoretical estimates compared to the first-order result, relates to the zero-order structure of the method.

In Corollary 4.2.5, the dependence $\Delta(\varepsilon)$ was obtained, which may seem incorrect, because usually the maximum noise is given by nature, and it cannot be reduced. In this case, we should rewrite the dependence in the form $\varepsilon = \varepsilon(\Delta)$ and accordingly τ and N start to depend on Δ, not on ε. In the rest of the corollaries in this section, we will write the dependence as $\Delta(\varepsilon)$ for ease of presentation, but they can always be rewritten in terms of $\varepsilon(\Delta)$.

Gradient Descent via JAGUAR

Algorithm 37 `GD via JAGUAR`

Parameters: stepsize γ, $\tau > 0$;
Initialization: choose $x^0 \in \mathbb{R}^d$; $h^0 = \widetilde{\nabla} f_\delta(x^0)$;

1: **for** $k = 0, 1, 2, ..., K-1$ **do**
2: $\quad h^{k+1} = \texttt{JAGUAR}\left(x^k, h^k\right)$
3: $\quad x^{k+1} = x^k - \gamma h^{k+1}$
4: **end for**

In this part, we consider a problem of the form

$$\min_{x \in \mathbb{R}^d} f(x).$$

We cannot use FW-type algorithms in this setting because $Q \equiv \mathbb{R}^d$ is not a bounded set; therefore, we consider Gradient Descent with `JAGUAR` approximation. We now examine the convergence of Algorithm 37.

Theorem 4.2.6 (Convergence rate of `GD via JAGUAR` (Algorithm 37). Non-convex case)**.** Consider Assumptions 4.0.1 and 4.2.1. If we take $\gamma \equiv 1/(4dL)$, then `GD via JAGUAR` (Algorithm 37) in the non-convex case has the following convergence rate

$$\mathbb{E}\left[\left\|\nabla f(\widehat{x}^K)\right\|^2\right] = \mathcal{O}\left(\frac{dL\Phi_0}{K+1} + dL^2\tau^2 + \frac{d\Delta^2}{\tau^2}\right),$$

where $\widehat{x}^K$ is chosen uniformly from $\{x^k\}_{k=0}^{K-1}$ and $\Phi_0 := f(x^0) - f^* + d\gamma \left\|h^0 - \nabla f(x^0)\right\|^2$.

Corollary 4.2.7. Under the conditions of Theorem 4.2.6, choosing γ, τ, Δ in a some right way in order to achieve an ε-approximate solution (in terms of $\mathbb{E}\left[\|\nabla f(\widehat{x}^K)\|^2\right] \leq \varepsilon^2$, where $\widehat{x}^K$ is chosen uniformly from $\{x^k\}_{k=0}^{K-1}$) it takes

$$\mathcal{O}\left(\frac{dL\Phi_0}{\varepsilon^2}\right) \text{ iterations of Algorithm 37.}$$

Under the PL-condition we explore a better convergence rate.

Assumption 4.2.8 (PL condition)**.** The function $f(x)$ satisfies the PL-condition if there exists $\mu > 0$ such that for all $x \in \mathbb{R}^d$ it holds $\|\nabla f(x)\|_2^2 \geq 2\mu(f(x) - f^*)$.

Theorem 4.2.9 (Convergence rate of `GD via JAGUAR` (Algorithm 37) PL-case)**.** Consider Assumptions 4.0.1, 4.2.1 and 4.2.8. If we take $\gamma \equiv 1/(4dL)$, then `GD via JAGUAR` (Algorithm 37) in the PL-condition case (Assumption 4.2.8) has the following convergence rate:

$$\mathbb{E}\left[f(x^K) - f(x^*)\right] = \mathcal{O}\left(\left(f(x^0) - f(x^*)\right)\exp\left[-\frac{\mu K}{4dL}\right] + \frac{dL^2\tau^2 + d\Delta^2/\tau^2}{\mu}\right),$$

Corollary 4.2.10. Under the conditions of Theorem 4.2.9, choosing γ, τ, Δ in a some right way in order to achieve an ε-approximate solution (in terms of $\mathbb{E}[f(x^K) - f(x^*)] \leq \varepsilon$) it takes

$$\mathcal{O}\left(\frac{dL}{\mu}\log\left[\frac{(f(x^0) - f(x^*))}{\varepsilon}\right]\right) \text{ iterations of Algorithm 37.}$$

Discussion. The results of Theorems 4.2.6 and 4.2.9 match with first-order results, but in such results $\gamma \equiv 1/L$, therefore, in the zero-order case the factor d reappears in theoretical estimates. We also obtain a smoothing term of the form $\mathcal{O}\left(\text{poly}(\tau) + \text{poly}(\Delta/\tau)\right)$. The same estimates arise in previous works when studying Gradient Descent in the zero-order oracle setting [353, 354, 344].

Frank-Wolfe via JAGUAR. Stochastic case

In this part, we consider the stochastic version of the problem (4.1):

$$f(x) := \mathbb{E}_{\xi\sim\pi}[f(x,\xi)], \tag{4.12}$$

where ξ is a random vector from a mostly unknown distribution π. For this problem, we cannot use the values of the function $f(x)$ in the difference schemes, since only $f(x,\xi)$ is available. Again, we assume that we do not have access to the true value of the gradient $\nabla f(x,\xi)$, and the zero-order oracle returns the noisy value of the function $f(x,\xi)$: $f_\delta(x,\xi) := f(x,\xi) + \delta(x,\xi)$.

In the stochastic setup (4.12), there are two versions of the differences of the scheme (4.8) appear. The first one is called Two Point Feedback (TPF) [355, 341, 342, 356, 337]. In this case, we define gradient approximations of the function $f(x)$ as follows:

$$\widetilde{\nabla}_i f_\delta(x,\xi) := \frac{f_\delta(x+\tau e_i,\xi) - f_\delta(x-\tau e_i,\xi)}{2\tau}e_i. \tag{4.13}$$

The second one is called One Point Feedback (OPF) [33, 357, 358, 359, 360]. In this case, we define a slightly different gradient approximation of the function $f(x)$:

$$\widetilde{\nabla}_i f_\delta(x,\xi^\pm) := \frac{f_\delta(x+\tau e_i,\xi^+) - f_\delta(x-\tau e_i,\xi^-)}{2\tau}e_i. \tag{4.14}$$

The main difference between the approximations (4.13) and (4.14) is that the scheme (4.13) is more accurate, but difficult to implement in practice, because we have to get the same realization of ξ at two different points $x + \tau e$ and $x - \tau e$, hence the scheme (4.14) is more interesting from a practical point of view. To further simplify, we consider that in the case of two-point feedback (4.13) we have the same inscription as for the one-point feedback (4.14), but only $\xi^+ = \xi^- = \xi$. We provide several assumptions required for the analysis.

Assumption 4.2.11 (Smoothness)**.** The functions $f(x, \xi)$ are $L(\xi)$-smooth on the set $\mathcal{X}$, i.e. for all $x, y \in \mathcal{X}$ it holds $\|\nabla f(x, \xi) - \nabla f(y, \xi)\| \leq L(\xi)\|x - y\|$. We also assume that exists constant L such that $L^2 := \mathbb{E}[L(\xi)^2]$. Then function $f(x)$ is L-smooth on the set $\mathcal{X}$.

Assumption 4.2.12 (Bounded oracle noise)**.** The noise in the oracle is bounded by some constant $\Delta > 0$, i.e. there exists $\Delta > 0$ such that for all $x \in \mathcal{X}$ it holds $\mathbb{E}[|\delta(x, \xi)|^2] \leq \Delta^2$. If we define $\delta(x) := \mathbb{E}[\delta(x, \xi)]$, then it holds that $|\delta(x)|^2 \leq \Delta^2$.

Now we present two assumptions that are necessary only in the stochastic case.

Assumption 4.2.13 (Bounded variance of the gradient)**.** The variance of the $\nabla f(x, \xi)$ is bounded, i.e. there exists $\sigma_\nabla \geq 0$ such that for all $x \in \mathcal{X}$ it holds $\mathbb{E}[\|\nabla f(x, \xi) - \nabla f(x)\|^2] \leq \sigma_\nabla^2$.

Assumption 4.2.14 (Bounded variance of the function)**.** The variance of the $f(x, \xi)$ is bounded, i.e. there exists $\sigma_f \geq 0$ such that for all $x \in \mathcal{X}$ it holds $\mathbb{E}[\|f(x, \xi) - f(x)\|^2] \leq \sigma_f^2$.

Assumptions 4.2.13 and 4.2.14 are classical in the stochastic optimization literature [361, 362, 359, 363]. In the case of two-point feedback (4.13), we do not require Assumption 4.2.14. Therefore, for the sake of simplicity, we will assume that in this case, Assumption 4.2.14 is satisfied with $\sigma_f = 0$. Now we can present the Frank-Wolfe algorithm, which solves the problem (4.1) + (4.12) using `JAGUAR` gradient approximation.

Algorithm 38 `FW via JAGUAR`. Stochastic case

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$, momentums $\{\alpha_k\}_{k\geq 0}$, $\tau > 0$;
Initialization: choose $x^0 \in \mathcal{X}$; $h^0 = g^0 = 1/(2\tau)\sum_{i=1}^d (f_\delta(x^0 + \tau e_i, \xi_i^+) - f_\delta(x^0 - \tau e_i, \xi_i^-))e_i$;

1: **for** $k = 0, 1, 2, ..., K-1$ **do**
2: Sample $i_k \in [d]$ independently and uniform
3: Sample $\xi_k^+, \xi_k^- \sim \pi$ independently (in TPF $\xi_k^+ = \xi_k^-$)
4: Compute $\widetilde{\nabla}_{i_k} f_\delta(x^k, \xi_k^\pm) = 1/(2\tau)(f_\delta(x^k + \tau e_{i_k}, \xi_k^+) - f_\delta(x^k - \tau e_{i_k}, \xi_k^-))e_{i_k}$
5: $h^{k+1} = h^k - \left\langle h^k, e_{i_k} \right\rangle e_{i_k} + \widetilde{\nabla}_{i_k} f_\delta(x^k, \xi_k^+, \xi_k^-)$
6: $\rho^k = h^k - d \cdot \left\langle h^k, e_{i_k} \right\rangle e_{i_k} + d \cdot \widetilde{\nabla}_{i_k} f_\delta(x^k, \xi_k^+, \xi_k^-)$
7: $g^{k+1} = (1 - \alpha_k)g^k + \alpha_k \rho^k$
8: $s^k = \arg\min_{s \in \mathcal{X}} \langle s, g^{k+1} \rangle$
9: $x^{k+1} = x^k + \eta_k(s^k - x^k)$
10: **end for**

Algorithm 38 is similar to `FW via JAGUAR` in the deterministic case (Algorithm 36), but in lines

6 and 7 we utilize SEGA [59] and momentum [317] parts to converge in the stochastic case.

• We need the SEGA part [59] ρ_k in Algorithm 38, because in the stochastic case we care about the "unbiased" property, i.e.

$$\mathbb{E}_k[\rho^k] = \widetilde{\nabla} f_\delta(x^k) := \sum_{i=1}^{d} \frac{f_\delta(x+\tau e_i) - f_\delta(x-\tau e_i)}{2\tau} e_i,$$

where $\mathbb{E}_k[\cdot]$ is a conditional mathematical expectation at step k. Using the SEGA part ρ^k degrades our estimates by a factor of d compared to using h^k as a gradient approximation; however, we have to accept this factor.

• We need the momentum part [317] α_k in Algorithm 38, because when evaluating the expression of $\mathbb{E}[\|\widetilde{\nabla} f_\delta(x, \xi^{\pm}_{[d]}) - \nabla f(x)\|^2]$ in the stochastic case, terms containing σ_∇^2 appear and σ_f^2 and interfere with convergence. This is a common problem in the stochastic Frank-Wolfe-based methods (see [317]). We can provide a theorem similar to the one in Section 4.2.2 (Theorem 4.2.3), where we carefully choose the step sizes γ_k and η_k.

Theorem 4.2.15 (Step tuning for `FW via JAGUAR`. Stochastic case)**.** Consider Assumptions 4.0.3, 4.2.11, 4.2.12, 4.2.13 and 4.2.14 in the one-point feedback case. For x^k generated by Algorithm 38, we can take $\eta_k = 4/(k + 8d^{3/2})$ and $\alpha_k = 4/(k + 8d^{3/2})^{2/3}$, then, the following inequality holds:

$$G_k = \mathcal{O}\left(\left\|\widetilde{\nabla} f_\delta(x^k) - \nabla f(x^k)\right\|^2 + \frac{L^2 D^2 + d^2\sigma_f^2/\tau^2 + d^2\sigma_\nabla^2}{(k + 8d^{3/2})^{2/3}}\right),$$

where we use the notation $G_k := \mathbb{E}[\|g^k - \nabla f(x^k)\|^2]$. In the two-point feedback case, $\sigma_f^2 = 0$.

We obtain worse estimates compared to the deterministic case in Theorem 4.2.3 because we consider a more complicated setup. We now examine the convergence of Algorithm 38.

Theorem 4.2.16 (Convergence rate of `FW via JAGUAR` (Algorithm 38). Stochastic case)**.** Consider Assumptions 4.0.3 4.2.11, 4.2.12, 4.2.13 and 4.2.14 in the one point feedback case. We can take $\eta_k = 4/(k + 8d^{3/2})$ and $\alpha_k = 4/(k + 8d^{3/2})^{2/3}$, then we `FW via JAGUAR` (Algorithm 38) has the following convergence rate

$$\mathbb{E}[f(x^K) - f^*] = \mathcal{O}\left(\frac{LD^2 + d\sigma_f D/\tau + d\sigma_\nabla D + \sqrt{d}(f(x^0) - f(x^*)))}{(N + 8d^{3/2})^{1/3}} + \sqrt{d}LD\tau + \frac{\sqrt{d}\Delta D}{\tau}\right).$$

In the two point feedback case, $\sigma_f^2 = 0$.

Corollary 4.2.17. Under the conditions of Theorem 4.2.16, choosing $\eta_k, \alpha_k, \tau, \Delta$ in a some right way in order to achieve an ε-approximate solution (in terms of $\mathbb{E}[f(x^K) - f(x^*)] \leq \varepsilon$) it takes

$$\mathcal{O}\left(\max\left\{\left[\frac{LD^2+d\sigma_\nabla D+\sqrt{d}(f(x^0)-f(x^*))}{\varepsilon}\right]^3;\frac{d^{9/2}\sigma_f^3L^3D^6}{\varepsilon^6}\right\}\right) \text{ iterations of Algorithm 38.}$$

In the two point feedback case, $\sigma_f^2=0$ and the last equation takes form

$$\mathcal{O}\left(\left[\frac{LD^2+d\sigma_\nabla D+\sqrt{d}(f(x^0)-f(x^*))}{\varepsilon}\right]^3\right) \text{ iterations of Algorithm 38.}$$

Discussion. Since we used SEGA and momentum parts in the `JAGUAR` approximation algorithm (Algorithm 38), we do not obtain the same convergence rate as in Theorems 4.2.3 and 4.2.4, even if we switch from stochastic to deterministic setups, i.e., when we set $\sigma_\Delta=\sigma_f=0$ in Theorems 4.2.15 and 4.2.16. The same problems arise in the first-order case [317, 323], due to the difficulty of implementing the stochastic gradient in FW-type algorithms.

We can apply the deterministic `JAGUAR` method (Algorithm 34) to the stochastic problem (4.12) and obtain the same estimates as in Theorems 4.2.3 and 4.2.4, only the smoothed term of the form $\mathcal{O}\left(\text{poly}(\tau)+\text{poly}(\Delta/\tau)\right)$ will contain summands of the form $\mathcal{O}\left(\text{poly}(\sigma_\Delta^2)+\text{poly}(\sigma_f^2/\tau)\right)$. Therefore, if $\sigma_\Delta^2,\sigma_f^2\sim\Delta$, then deterministic Algorithm 34 is suitable for the stochastic problem (4.12). However, this means that we need to use big batches, therefore, we forced to use SEGA and momentum parts in the `JAGUAR` approximation.

4.2.3 Experiments

In this subsection, we present and discuss the results of our experiments on the application of `JAGUAR` gradient approximation to a variety of black-box optimization problems. Our results include optimization using Frank-Wolfe (FW) and Gradient Descent (GD) algorithms. In FW, we consider both deterministic and stochastic cases; in GD, we consider only the deterministic case.

We perform optimization on classification tasks with the SVM model on the set $\mathcal{X}$ of the form:

$$\min_{x\in\mathcal{X},w\in\mathbb{R}}\left\{f(x,w)=\frac{1}{n}\sum_{i=1}^n\left(1-b_i[a_i^Tx-w]\right)_+ +\frac{1}{2C}\|x\|^2\right\}. \tag{4.15}$$

We also consider a logistic regression model on the set Q of the form:

$$\min_{x\in\mathcal{X}}\left\{f(x)=\frac{1}{n}\sum_{i=1}^n\log\left(1+\exp\left[-b_ia_i^Tx\right]\right)+\frac{1}{2C}\|x\|^2\right\}. \tag{4.16}$$

In both problems, we use the regularization term $C=10$. As minimization set $\mathcal{X}$, we consider the simplex Δ_d and l_2-ball in the Frank-Wolfe algorithm and $\mathbb{R}^d$ in the Gradient Descent algorithm.

For the classification problem we use the classical datasets MNIST [364] and mushrooms [171]. We incorporate different approximation methods into optimization algorithms, solving (4.15) and (4.16) problems and show that the algorithm that uses the `JAGUAR` method (Algorithms 34 and 38) performs best. We consider l_2-smoothing (4.9) and full-approximation (4.8) as baseline estimators of the gradients.

Deterministic Frank-Wolfe

In this section, we consider deterministic noise of the form $f_\delta(x) = \text{round}(f(x), 5)$, i.e. we round all values of the function f to the fifth decimal place. Figure 4.7 shows the convergence over zero-order oracle calls of the deterministic FW algorithm. `FW via JAGUAR` (Algorithm 36) demonstrates improved results compared to the baseline algorithms. This observation confirms our theoretical findings.

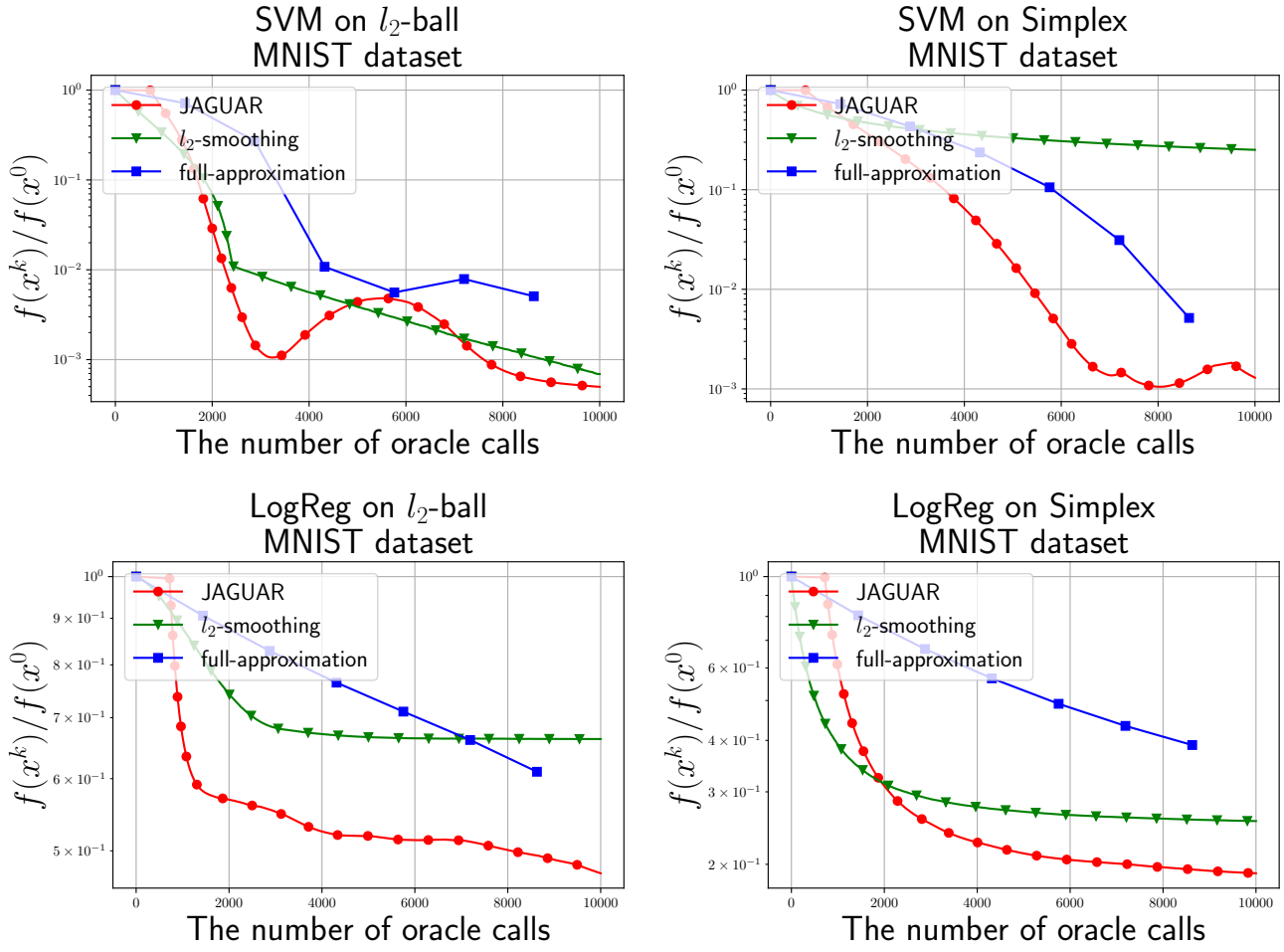


Figure 4.7: Deterministic FW Algorithm.

Stochastic Frank-Wolfe

In this section, we consider stochastic noise of the form $f_\delta(x, \xi) = f(x) + \xi; \xi \sim \mathcal{N}(0, 0.1)$. Figure 4.8 shows the convergence of the stochastic FW algorithm over zero-order oracle calls. Our theoretical findings are supported by observations. `FW via JAGUAR` (Algorithm 38) is robust to

noise and outperforms the baseline algorithms.

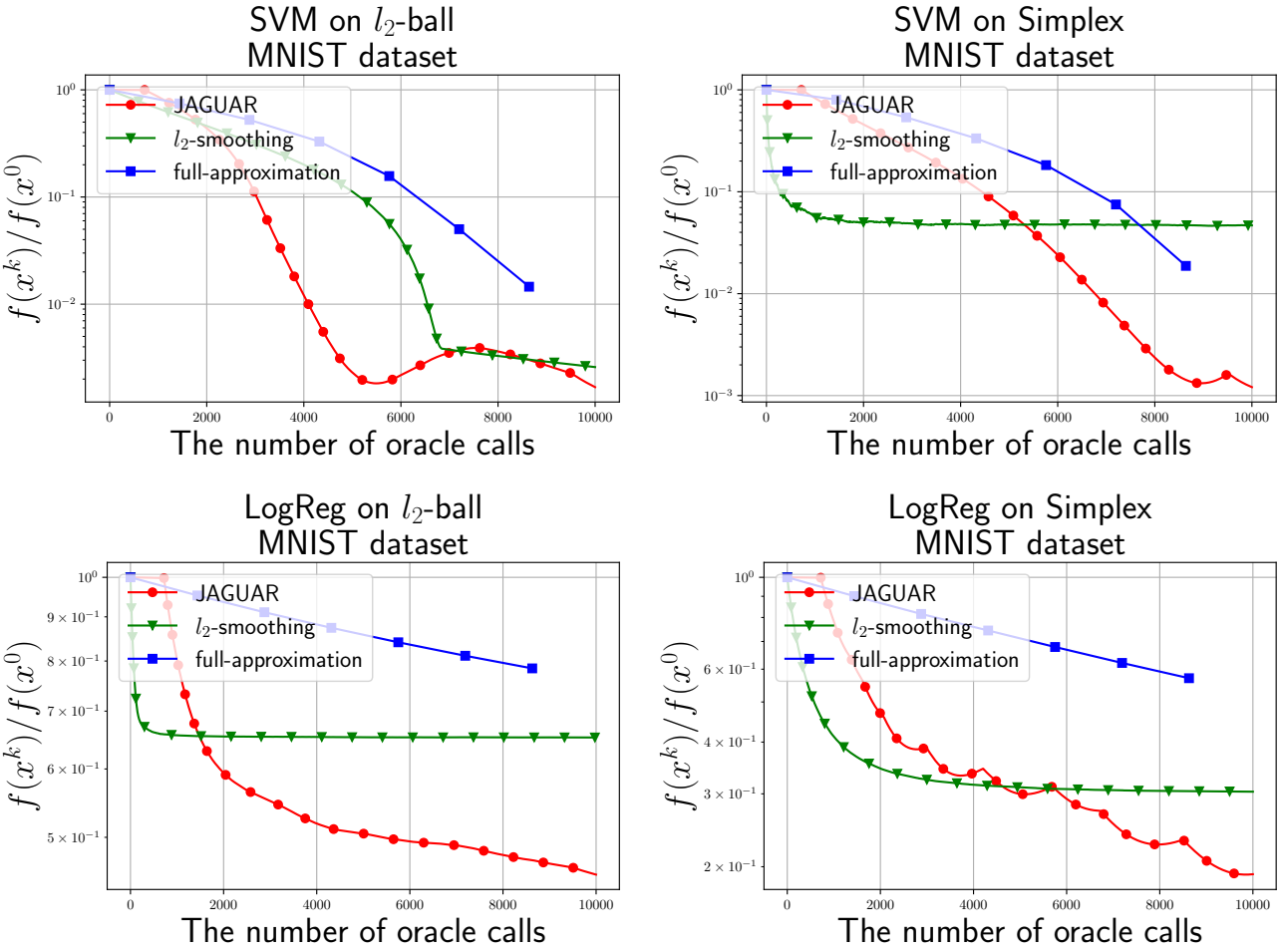


Figure 4.8: Stochastic FW Algorithm.

Gradient descent

In this section, we again consider deterministic noise of the form $f_\delta(x) = \text{round}(f(x), 5)$. Figure 4.9 shows convergence over zero-order oracle calls of the deterministic GD algorithm. `GD via JAGUAR` (algorithm 37) outperforms the baseline algorithms, albeit by a small margin, and this observation confirms our theoretical findings.

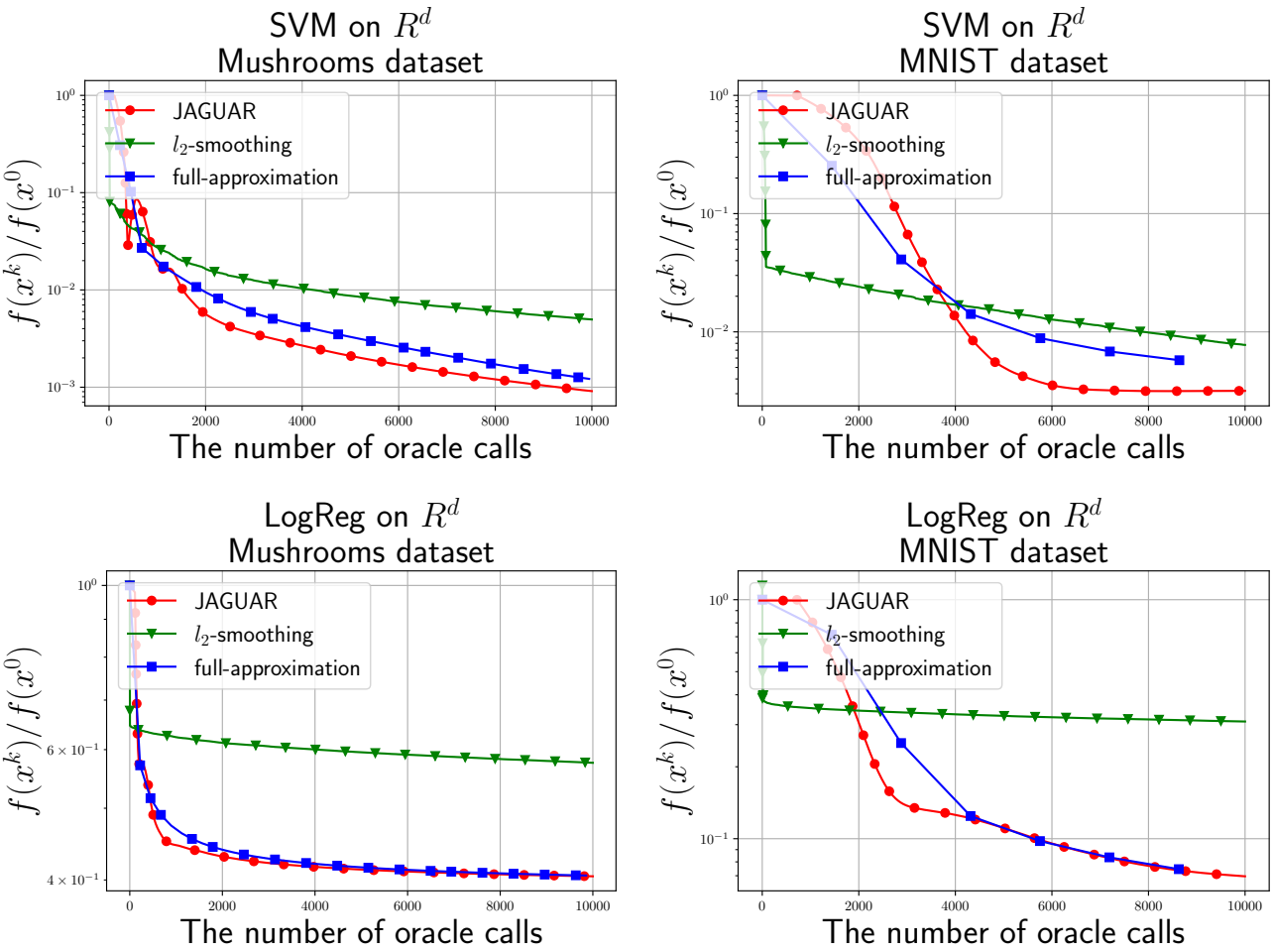


Figure 4.9: GD Algorithm.

4.3 Unified Frank-Wolfe

The purpose of this section is to incorporate the majority of existing results concerning the stochastic Frank-Wolfe method into a unified analysis. In particular, we aim to integrate both established methods and those proposed in the previous two sections. Additionally, we seek to develop new algorithms that have not been adapted to the Frank-Wolfe iteration before.

We consider a stochastic version of the Conditional Gradient method:

$$\begin{aligned} s^k &= \arg\min_{s \in \mathcal{X}} \langle g^k, s - x^k \rangle, \\ x^{k+1} &= (1 - \eta_k) x^k + \eta_k s^k, \end{aligned} \tag{4.17}$$

where g^k is some stochastic estimator of the real gradient $\nabla f(x^k)$. We specifically do not provide details on the structure of the estimator g^k. Various stochastic strategies can be implemented in it. Over its long history, the Frank-Wolfe method has received a significant number of different modifications in the form (4.17). Here, one can note the already known modifications related to variance reduction [304, 312, 313, 314, 315, 320, 321, 322] (see Section 4.1) and coordinate randomization [343, 346] (see Section 4.2). However, there are other approaches, for example, from distributed optimization [325, 365, 319].

However, all of these separate practical variants of the Conditional Gradient method have different intuitions of convergence, various formal techniques of proving, and do not always cover the same assumptions regarding the target function f. Moreover, a considerable gap remains in the development of Frank-Wolfe's modifications. Specifically, advanced SGD-type methods have not yet been adapted for use within the Frank-Wolfe iteration. These include some new coordinate approaches and approaches to finite-sum problems. Meanwhile, these two techiques help to address the main bottleneck of distributed algorithms – expensive communications. All of mentioned issues lead us to the contribution of this chapter.

4.3.1 Contributions

• **Unified analysis of Stochastic Frank-Wolfe.** We propose a general assumption on the stochastic estimator g^k from Stochastic Frank-Wolfe (4.17) – see Assumption 4.3.1. Below we note in more details that our assumption is broad and encompasses many special cases, in particular, those that could not be analyzed in a unified way before. Under Assumption 4.3.1 we present general convergence results for the problem (4.1).

• **Convex and non-convex cases.** Motivated by various applications primarily from machine learning, we provide a unified analysis in the convex (Theorem 4.3.2) and non-convex (Theorem 4.3.3) cases of the target function f. This is also relevant for special cases, as the authors of some papers do not provide an analysis in both setups.

• **Without assumptions of unbiasedness.** In our key Assumption 4.3.1, we bound the variance $\mathbb{E}[\|g^k - \nabla f(x^k)\|^2]$ using universal letter constants and an additional auxiliary sequence.

Similar assumptions are made in papers on the analysis of the SGD family of methods [66, 366, 367]. However, these works also assume that the stochastic gradient g^k is unbiased, i.e., $\mathbb{E}[g^k \mid x^k] = \nabla f(x^k)$. We avoid this assumption, extending the class of methods that can be considered under our assumptions compared to the works around the SGD-type methods. For example, it allows us to prove the convergence of distributed methods with biased/greedy compression [232, 45] or SARAH-based variance reduced methods [53].

• **Vast number of new special cases.** The previous point already provides an indication of the breadth and flexibility of the approach. Our general theoretical framework allows for the analysis of different variants of the classical Frank-Wolfe method. Guided by algorithmic advances for solving unconstrained minimization problems, we present a new method with coordinate randomization (`SEGA FW`), a new variance-reduced method (`L-SVRG FW`, `SARAH FW`, `SAGA FW`), new distributed methods with unbiased compression (`DIANA FW`, `MARINA FW`) and biased compression (`EF21 FW`), and others. Although the SGD-type analogs of these methods are known for solving primarily unconstrained minimization problems [59, 264, 212, 45], they have not been integrated into the Frank-Wolfe iteration for solving projection-free constrained problems. We also demonstrate that our general theorems allow for obtaining convergence for methods that are combinations of the two basic approaches, i.e., `SAGA SARAH FW` and others. Moreover, we present absolutely new methods that are not found in the literature on SGD. This algorithm employs special coordinate randomization. `JAGUAR` is a new coordinate method. `Q-L-SVRG FW` is a new distributed method with unbiased compression. This method is based on non-distributed `L-SVRG FW`: instead of the randomness of choosing a batch/term number, randomness from compression is utilized. `PP-L-SVRG FW` is a new distributed method with client sampling, also based on non-distributed `L-SVRG FW`.

• **Sharp rates for known special cases.** For the known methods fitting our framework, our general theorems recover the best rates known for these methods. These methods include `SARAH FW`, `SAGA SARAH FW` from Section 4.1.

• **Numerical experiments.** In numerical experiments, we illustrate the most important properties of the new methods. The results corroborate our theoretical findings.

Throughout the section, we provide necessary comparisons with closely related work.

4.3.2 Main Theorems

In this section, we first present the central part of our approach that allows us to conduct a general analysis of the algorithms (Assumption 4.3.1). We then provide convergence analysis for both convex and non-convex cases.

Unified Assumption

First, we introduce the central part of our approach. All subsequent analysis is based on the following assumption on the stochastic gradients g^k :

Assumption 4.3.1. Let $\{x^k\}_{k=0}^K$ be the iterates produced by Stochastic Frank-Wolfe (see (4.17)). Let there exist constants $A, B, C, E \geq 0$, $\rho_1, \rho_2 \in (0; 1]$ and a (possibly) random sequence $\{\sigma_k\}_{k\geq 0}$ such that the following inequalities hold

$$\mathbb{E}[\|g^k - \nabla f(x^k)\|^2 \,|\, x^k] \leq (1-\rho_1)\|g^{k-1} - \nabla f(x^{k-1})\|^2 + A\sigma_{k-1}^2 + \eta_{k-1}^2 BD^2 + C, \tag{4.18}$$

$$\mathbb{E}[\sigma_k^2 \mid x^k] \leq (1-\rho_2)\sigma_{k-1}^2 + \eta_{k-1}^2 ED^2.$$

The inequality (4.18) bounds the second moment of stochastic estimation g^k. The sequence $\{\sigma_k^2\}_{k\geq 0}$ is needed to capture the variance, which can be reduced during the algorithm's work process. Constants ρ_1, ρ_2 show how quick this reduction is regarding the previous iteration. B, E provide the information on convergence, depending on the previous step size and the set's geometry (Assumption 4.0.3). Finally, constant C represents the remaining noise that cannot be reduced as σ_k.

The proposed assumption takes into account the specificity of the Frank-Wolfe analysis in terms of an upper bound that contains the component D^2 as an irremovable part of such types of inequalities.

Convergence Results

Convex case. The following theorem describes the convergence rate of stochastic Frank-Wolfe (4.17) based methods under the convexity of f:

Theorem 4.3.2. Let Assumptions 4.0.1, 4.0.2, 4.0.3 and 4.3.1 be satisfied. For any K choose step sizes $\{\eta_k\}_{k\geq 1}$ as follows:

$$\begin{aligned}
&\text{if } K \leq d, && \eta_k = \tfrac{1}{d},\\
&\text{if } K > d \text{ and } k < k_0, && \eta_k = \tfrac{1}{d},\\
&\text{if } K > d \text{ and } k \geq k_0, && \eta_k = \tfrac{2}{2d+k-k_0},
\end{aligned}$$

where $d = \frac{2}{\min(\rho_1,\rho_2)}$, $k_0 = \left\lceil \frac{K}{2} \right\rceil$. Then the output of Stochastic Frank-Wolfe after K iterations satisfies

$$\mathbb{E}[r_{K+1}] = \mathcal{O}\left(r_0 \exp\left(-\frac{K}{2d}\right) + \frac{LD^2}{K+d} + \frac{D^2}{K+d}\sqrt{\frac{B\rho_2 + AE}{\rho_1\rho_2}} + \sqrt{\frac{KD^2}{K+d}\frac{C\rho_2}{\rho_1\rho_2}}\right),$$

where the Lyapunov function r_k is defined by $r_k := f(x^k) - f^* + M_1\|\nabla f(x^k) - g^k\|^2 + M_2\sigma_k^2$ with $M_1, M_2 > 0$.

Note that with zero noise ($C = 0$ in Assumption 4.3.1) this theorem reflects sublinear convergence $\mathcal{O}\left(\frac{1}{K}\right)$.

Non-convex case. To obtain our convergence results for the non-convex objective function,

we use the already known gap (4.3) and derive the following general convergence result:

Theorem 4.3.3. Let the Assumptions 4.0.1, 4.0.3 and 4.3.1 be satisfied. Then, there exist constants M_1, M_2 such that for any K there exist constant $\{\eta_k\}_{k\geq 1} \equiv \frac{1}{\sqrt{K}}$ for (4.17), thus

$$\mathbb{E}\Big[\min_{0\leq k\leq K-1} gap(x^k)\Big] = \mathcal{O}\left(\frac{r_0}{\sqrt{K}} + \frac{D^2}{\sqrt{K}}\Big[L + \sqrt{\frac{B\rho_2 + AE}{\rho_1\rho_2}}\Big] + \sqrt{D^2\frac{C\rho_2}{\rho_1\rho_2}}\right),$$

where $r_0 := f(x^0) - f^* + M_1\|g^0 - \nabla f(x^0)\|^2 + M_2\sigma_0^2$ with $M_1, M_2 > 0$.

Note that in the case of zero noise ($C = 0$ in Assumption 4.3.1) this result establishes as $\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$. See the proofs in Appendix A.2.7.

4.3.3 Special Cases

In this section, we fulfill the promises made at the beginning of this section and show how many existing and new techniques fit our framework.

Stochastic Methods

We start from the stochastic of the finite-sum type (4.4).

L-SVRG FW. One of the most popular stochastic algorithms for (4.4) with $\mathcal{X} = \mathbb{R}^d$ is SVRG [52]. We consider its loopless variant [264] called L-SVRG that utilizes the SVRG concept but is more amenable to theoretical analysis. In more detail, we need to compute g^k as follows:

Algorithm 39 `L-SVRG Frank-Wolfe`

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$, probability p, batch size b
Initialization: choose $w^0 = x^0 \in \mathcal{X}$; $g^0 = \nabla f(x^0)$

1: **for** $k = 0, 1, 2, \ldots K-1$ **do**
2: Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle s, g^k\rangle$
3: Update $x^{k+1} = (1-\eta_k)x^k + \eta_k s^k$
4: Update $w^{k+1} = \begin{cases} x^k, & \text{with probability } p \\ w^k, & \text{with probability } 1-p \end{cases}$
5: Generate batch S_k with size b
6: $g^{k+1} = \frac{1}{b}\sum\limits_{i\in S_k}\big[\nabla f_i(x^{k+1}) - \nabla f_i(w^{k+1})\big] + \nabla f(w^{k+1})$
7: **end for**

The essence of this approach is close to Algorithm 32. The probability p is taken close to zero, then the full gradient at the point w^{k+1} are computed quite rarely and in most cases we use the approximation g^{k+1} via stochastic gradients on mini-batches of random indexes S_k size of b.

Lemma 4.3.4. Under Assumptions 4.0.1, 4.0.2, 4.1.1 Algorithm 39 satisfies Assumption 4.3.1 with: $\rho_1 = 1$, $A = \frac{\widetilde{L}^2}{b}\left(1 - \frac{p}{2}\right)$, $B = \frac{8\widetilde{L}^2}{pb}$, $C = 0$, $\sigma_k^2 = \|x^k - w^k\|^2$, $\rho_2 = \frac{p}{2}$, $E = \frac{8}{p}$.

Using this lemma, one can obtain the convergence of Algorithm (39) in both convex and non-convex cases.

Corollary 4.3.5. For Algorithm (39) in the convex and non-convex cases accordingly the following convergences take place:

$$\mathbb{E}[f(x^K) - f^*] = \mathcal{O}\left(\Delta_0 \exp\left(-\frac{Kp}{8}\right) + \frac{LD^2}{K}\left[1 + \frac{\widetilde{L}}{L}\frac{1}{p\sqrt{b}}\right]\right),$$
$$\mathbb{E}\left[\min_{0\leq k\leq K-1} gap(x^k)\right] = \mathcal{O}\left(\frac{\Delta_0}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left[1 + \frac{\widetilde{L}}{L}\frac{1}{p\sqrt{b}}\right]\right).$$

We proved results for `L-SVRG FW` depending on the parameters p and b to be tuned. Let us find the optimal choices of p and b for `L-SVRG FW`. To find the optimal choice of p, one can note that on average we call the stochastic gradients $(pn + 2b)$ times. In more details, at each iteration we compute a batch size of b in two points x^k and w^k and with probability p we call the full gradient in the new point w^k. From Corollary 4.3.5, we know the estimate on the number of iterations of `L-SVRG FW`, then we can obtain an estimate on the number of stochastic gradient calls by multiplying this result by $(pn + 2b)$. Then the new estimate can be optimized first by p (in the convex case, we need to minimize $(1 + \frac{1}{p\sqrt{b}})(pn + 2b)$) and obtain that the optimal $p \sim b^{1/4}/n^{1/2}$. Then with already optimized p, the estimate on the number of stochastic gradient calls can additionally be optimized by b (actually we need to minimize $b^{1/4}n^{1/2} + b + n/b^{1/2}$) and find the optimal $b \sim n^{2/3}$.

SARAH FW. Next, we consider Algorithm 32 from Section 4.1.

Lemma 4.3.6. Under Assumptions 4.0.1, 4.0.2, 4.1.1 Algorithm 32 satisfies Assumption 4.3.1 with: $\rho_1 = p$, $A = 0$, $B = \frac{1-p}{b}\widetilde{L}^2$, $C = 0$, $\sigma_k = 0$, $\rho_2 = 1$, $E = 0$.

Corollary 4.3.7. For Algorithm 32 in the convex and non-convex cases respectively the following convergences take place:

$$\mathbb{E}[f(x^K) - f^*] = \mathcal{O}\left(\Delta_0 \exp\left(-\frac{Kp}{4}\right) + \frac{LD^2}{K}\left[1 + \frac{\widetilde{L}}{L}\frac{1}{\sqrt{pb}}\right]\right),$$
$$\mathbb{E}\left[\min_{0\leq k\leq K-1} gap(x^k)\right] = \mathcal{O}\left(\frac{\Delta_0}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left[1 + \frac{\widetilde{L}}{L}\frac{1}{\sqrt{pb}}\right]\right).$$

It is easy to see that with the same p the results for `SARAH FW` is better than for `L-SVRG FW` (Corollary 4.3.5). The same results for `SARAH FW` were obtained in Section 4.1.

SAGA FW. The final algorithm of provided lineup for (4.4) is SAGA [55]:

Algorithm 40 `SAGA Frank-Wolfe`

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$, batch size b
Initialization: choose $x^0 \in \mathcal{X}$; $y_i^0 = \nabla f_i(x^0)$, $g^0 = \nabla f(x^0)$

1: **for** $k = 0, 1, 2, \ldots K-1$ **do**
2: Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle s, g^k\rangle$
3: Update $x^{k+1} = (1-\eta_k)x^k + \eta_k s^k$
4: Generate batch S_k with size b
5: Update $y_i^{k+1} = \begin{cases} \nabla f_i(x^k), & \text{for } i \in S_k \\ y_i^k, & \text{for } i \notin S_k \end{cases}$
6: Update $g^{k+1} = \frac{1}{b}\sum\limits_{i\in S_k}\left[\nabla f_i(x^{k+1}) - y_i^{k+1}\right] + \frac{1}{n}\sum\limits_{j=1}^{n} y_j^{k+1}$
7: **end for**

We use this algorithm as a part of Algorithm 33. The essence of the SAGA technique is different from SVRG and SARAH. In this case, we do not compute full gradients even rarely, but we need to store an additional set of vectors $\{y_i^k\}_{i=1}^n$. The vector y_i stores information about the last gradient of the function f_i that was computed during the operation of the algorithm. Thus one can state that we collect "delayed" full gradient in $\frac{1}{n}\sum_{j=1}^n y_k^j$.

Lemma 4.3.8. Under Assumptions 4.0.1, 4.0.2, 4.1.1 Algorithm 40 satisfies Assumption 4.3.1 with: $\rho_1 = 1$, $A = \frac{1}{b}\left(1 + \frac{b}{2n}\right)$, $B = \frac{2\widetilde{L}^2}{b}\left(1 + \frac{2n}{b}\right)$, $C = 0$, $\sigma_k^2 = \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^k) - y_j^{k+1}\|^2$, $\rho_2 = \frac{b}{2n}$, $E = \frac{2n}{b}\widetilde{L}^2$.

Corollary 4.3.9. For Algorithm 40 in the convex and non-convex cases the following convergences take place:

$$\mathbb{E}[f(x^K) - f^*] = \mathcal{O}\left(\Delta_0 \exp\left(-\frac{Kb}{8n}\right) + \frac{LD^2}{K}\left[1 + \frac{\widetilde{L}}{L}\frac{n}{b\sqrt{b}}\right]\right),$$
$$\mathbb{E}\left[\min_{0\leq k\leq K-1} gap(x^k)\right] = \mathcal{O}\left(\frac{\Delta_0}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left[1 + \frac{\widetilde{L}}{L}\frac{n}{b\sqrt{b}}\right]\right).$$

One more method (`SAGA SARAH FW`) for the stochastic setting will be analyzed in Section 4.3.3. A comparison of the presented and already existing methods is provided in Table 4.4. In particular, there we analyze Corollaries 4.3.5–4.3.9 with the substituted optimal parameter value of p.

Coordinate Methods

Previous approaches reduce the cost of gradient computing by selecting small batches, but there are other strategies. In particular, wide range of algorithms use random sampling of coordinates for gradient evaluation [67, 368, 369]. This technique can also significantly decrease the computational cost. Then, in this section, we focus on methods, where gradient estimator stochastically depend on function's partial derivatives.

SEGA FW. The original algorithm developed in [59] covers a general setting, instead of which we use a slightly more simplified version. Particularly, we update g^k as follows:

Table 4.4: Summary of complexity results for finding an ε-solution stochastic finite-sum non-distributed constrained minimization problems (4.4) with n terms by projection free methods. Convergence is measured by the functional distance to the solution in the convex case and by the gap function in the non-convex case. Complexities are given in terms of the number of stochastic gradient calls.

Method	Link	Reference	CVX	nCVX
`Frank-Wolfe`	Alg. 1	[291], [303]	$\mathcal{O}(\frac{n}{\varepsilon})$	$\mathcal{O}(\frac{n}{\varepsilon^2})$
`SVRG FW`	Alg. 3	[304]	✗	$\mathcal{O}(n+\frac{n^{2/3}}{\varepsilon^2})$
`SVRG FW`	Alg. 1	[312]	$\mathcal{O}\left(n+\frac{1}{\varepsilon^2}\right)$	✗
`SPIDER FW`	Alg. 2	[314]	$\mathcal{O}(\frac{1}{\varepsilon^2})$	$\mathcal{O}(n+\frac{\sqrt{n}}{\varepsilon^2})$
`SFW`	Alg. 1	[320]	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$	✗
`GSFW`	Alg. 1	[321]	$\mathcal{O}(\frac{n}{\varepsilon})$	✗
`SVRG FW`	Alg. 3	[322]	✗	$\mathcal{O}(n+\frac{n^{2/3}}{\varepsilon^2})$
`SAGA FW`	Alg. 40	[304]	$\widetilde{\mathcal{O}}(n+\frac{n^{2/3}}{\varepsilon})$	$\mathcal{O}(n+\frac{n^{2/3}}{\varepsilon^2})$
`SARAH FW`	Alg. 32	**NEW**, [53]	$\widetilde{\mathcal{O}}(n+\frac{\sqrt{n}}{\varepsilon})$	$\mathcal{O}(n+\frac{\sqrt{n}}{\varepsilon^2})$
`SAGA SARAH FW`	Alg. 33	**NEW**, [308]	$\widetilde{\mathcal{O}}(n+\frac{\sqrt{n}}{\varepsilon})$	$\mathcal{O}(n+\frac{\sqrt{n}}{\varepsilon^2})$
`L-SVRG FW`	Alg. 39	**NEW**, [264]	$\widetilde{\mathcal{O}}(n+\frac{n^{2/3}}{\varepsilon})$	$\mathcal{O}(n+\frac{n^{2/3}}{\varepsilon^2})$

Algorithm 41 `SEGA Frank-Wolfe`

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$
Initialization: choose $x^0 \in \mathcal{X}$; $h^0 = \nabla f(x^0)$, $g^0 = \nabla f(x^0)$

1: **for** $k = 0, 1, 2, \ldots K-1$ **do**
2: Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle s, g^k\rangle$
3: Update $x^{k+1} = (1-\eta_k)x^k + \eta_k s^k$
4: Sample $i_k \in [d]$ uniformly at random
5: Set $h^{k+1} = h^k + e_{i_k}(\nabla_{i_k} f(x^k) - h^k_{i_k})$
6: Update $g^{k+1} = de_{i_k}(\nabla_{i_k} f(x^{k+1}) - h^{k+1}_{i_k}) + h^{k+1}$
7: **end for**

The idea of this approach is in some sense close to SAGA. We also have a memory buffer, but unlike SAGA, where we save the last calculated gradient on ith batch, here in h_i we save the last partial derivative $\nabla_i f$ calculated for the ith coordinate.

Lemma 4.3.10. Under Assumptions 4.0.1, 4.0.2 Algorithm 41 satisfies Assumption 4.3.1 with:

$\rho_1 = 1$, $A = d$, $B = d^2L^2$, $C = 0$, $\sigma_k^2 = \|h^{k+1} - \nabla f(x^k)\|^2$, $\rho_2 = \frac{1}{2d}$, $E = 3L^2 d$.

Corollary 4.3.11. For Algorithm 41 in the convex and non-convex cases the following convergences take place:

$$\mathbb{E}[f(x^K) - f^*] = \mathcal{O}\Big(\Delta_0 \exp\Big(-\frac{K}{8d}\Big) + \frac{LD^2}{K} \cdot d\sqrt{d}\Big),$$
$$\mathbb{E}\Big[\min_{0\le k\le K-1} \mathbf{gap}(x^k)\Big] = \mathcal{O}\Big(\frac{\Delta_0}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}} \cdot d\sqrt{d}\Big).$$

Despite the fact that SEGA has established itself as an effective coordinate method, the theoretical estimation of the required steps for convergence includes an undesirable component $d\sqrt{d}$, while the original algorithm contains only d. Therefore, it is reasonable to introduce a new algorithm that has theoretically improved convergence.

JAGUAR. An important feature of SEGA is that it utilizes unbiased gradient estimation: $\mathbb{E}[g^k \mid x^k] = \nabla f(x^k)$. On one hand, this characteristic simplifies the theoretical analysis. However, experience demonstrates that stochastic methods with biased gradient approximations can outperform their unbiased counterparts. The comparison between SARAH (biased) and SVRG (unbiased) supports this notion. Therefore, we propose to consider the following form of g^k:

Algorithm 42 JAGUAR

Parameters: stepsizes $\{\eta_k\}_{k\ge 0}$
Initialization: choose $x^0 \in \mathcal{X}$ $g^0 = \nabla f(x^0)$
1: **for** $k = 0, 1, 2, \dots K-1$ **do**
2: Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle s, g^k\rangle$
3: Update $x^{k+1} = (1-\eta_k)x^k + \eta_k s^k$
4: Sample $i_{k+1} \in [d]$ uniformly at random
5: Update $g^{k+1} = e_{i_{k+1}}(\nabla_{i_{k+1}} f(x^k) - g^k_{i_{k+1}}) + g^k$
6: **end for**

This algorithm is a coordinate version of the gradient-free algorithms from Section 4.2.

Lemma 4.3.12. Under Assumptions 4.0.1, 4.0.2 Algorithm 42 satisfies Assumption 4.3.1 with: $\rho_1 = 1, A = 0,\ B = 3dL^2,\ C = 0,\ \sigma_k^2 = 0,\ \rho_2 = 1, E = 0.$

Corollary 4.3.13. For Algorithm 42 in the convex and non-convex cases the following convergences take place:

$$\mathbb{E}[f(x^K) - f^*] = \mathcal{O}\left(\Delta_0 \exp\left(-\frac{K}{8d}\right) + \frac{LD^2}{K} \cdot d\right),$$
$$\mathbb{E}\left[\min_{0\le k\le K-1} gap(x^k)\right] = \mathcal{O}\left(\frac{\Delta_0}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}} \cdot d\right).$$

Table 4.5: Summary of complexity results for finding an ε-solution constrained minimization problems with dimension d by coordinate projection free methods. Convergence is measured by the functional distance to the solution in the convex case and by the gap function in the non-convex case. Complexities are given in terms of the number of coordinates derivatives computed.

Method	Link	Reference	CVX	nCVX
BCFW	Alg. 3	[343]	$\mathcal{O}(d + \frac{d}{\varepsilon})$	✗
SGFFW	Alg. 1	[346](1)	$\mathcal{O}(d + \frac{d}{\varepsilon})$	✗
SGFFW	Alg. 2	[346](1)	$\mathcal{O}(d + \frac{d}{\varepsilon^3})$	$\mathcal{O}(d + \frac{d^{4/3}}{\varepsilon^4})$
SEGA FW	Alg. 41	**NEW**, [59]	$\widetilde{\mathcal{O}}(d + \frac{d\sqrt{d}}{\varepsilon})$	$\mathcal{O}(d + \frac{d^3}{\varepsilon^2})$
JAGUAR	Alg. 42	**NEW**	$\widetilde{\mathcal{O}}(d + \frac{d}{\varepsilon})$	$\mathcal{O}(d + \frac{d^2}{\varepsilon^2})$

(1) Zero-order methods.

Distributed Methods with Compression

In this section, we focus on distributed versions of the Frank-Wolfe algorithm for solving finite-sum problems (4.4), where $\{f_i\}_{i=1}^n$ are distributed across n devices connected with a parameter server in a centralized manner, and each device has access only to f_i. Here we allow different machines to have different data distributions, thus considering the heterogeneous data setting. For this type of problem, the bottleneck is commonly the communication cost [105], which motivates the use of compressed communication [44]. We have already met compression in Section 2.5. For the completeness of this section, we reintroduce the definition of the compression operator.

Definition 4.3.14. *Map* $\mathcal{Q} : \mathbb{R}^d \to \mathbb{R}^d$ is an unbiased compression operator, if there exist a constant $\omega \geq 0$ such that for all $x \in \mathbb{R}^d$

$$\mathbb{E}[\mathcal{Q}(x)] = x, \quad \mathbb{E}[\|\mathcal{Q}(x) - x\|^2] \leq \omega \|x\|^2.$$

As we know, the usage of unbiased compression has been extensively studied. The first and simplest idea that comes to mind is to apply compression directly on the gradient estimator when forwarding to the server [43]. However, this approach presents a problem; namely, for a fixed step, it guarantees convergence only to the neighborhood of the solution. Therefore, we propose to consider more advanced techniques that compress some difference that tends to zero during the course of the algorithm.

DIANA FW. We start with the DIANA technique [60]. Its essence lies in the fact that it maintains the "memory" variables h_i^k at each worker i and compresses gradient differences $\nabla f_i(x^k) - h_i^k$. In particular,

Algorithm 43 `DIANA Frank-Wolfe`

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$, α
Initialization: choose $x^0 \in \mathcal{X}$; $h_i^0 = \nabla f_i(x^0)$, $h^0 = \frac{1}{n}\sum_{i=1}^n h_i^0$

for $k = 0, 1, 2, \ldots K-1$ **do**
  Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle s, g^k\rangle$
  Update $x^{k+1} = (1-\eta_k)x^k + \eta_k s^k$
  Update $h^{k+1} = h^k + \alpha \cdot \frac{1}{n}\sum_{i=1}^n \Delta_i^k$
  for $i = 1, \ldots, n$ **do**
    Compress shifted local gradient $\Delta_i^{k+1} = \mathcal{Q}(\nabla f_i(x^{k+1}) - h_i^{k+1})$ and send Δ_i^{k+1} to the server
    Update local shift $h_i^{k+1} = h_i^k + \alpha \cdot \mathcal{Q}(\nabla f_i(x^{k+1}) - h_i^{k+1})$
  end for
  Aggregate received compressed gradient information $g^{k+1} = h^{k+1} + \frac{1}{n}\sum_{i=1}^n \Delta_i^{k+1}$
end for

It is important to highlight that we need only compressed differences Δ_i^k for the server updates.

Lemma 4.3.15. Under Assumptions 4.0.1, 4.0.2, 4.1.1 Algorithm 43 satisfies Assumption 4.3.1 with: $\rho_1 = 1$, $A = \frac{\omega}{n^2}$, $B = \frac{2\omega(\omega+1)\widetilde{L}^2}{n}$, $C = 0$, $\sigma_k^2 = \sum_{i=1}^n \|\nabla f_i(x^k) - h_i^k\|^2$, $\rho_2 = \frac{1}{2(1+\omega)}$, $E = 2(\omega+1)n\widetilde{L}^2$.

Corollary 4.3.16. For Algorithm 43 in the convex and non-convex cases the following convergences take place:

$$\mathbb{E}[f(x^K) - f^*] = \mathcal{O}\Big(\Delta_0 \exp\left(-\tfrac{K}{8\omega}\right) + \tfrac{LD^2}{K}\left(1 + \tfrac{\widetilde{L}}{L}\tfrac{\omega^{\frac{3}{2}}}{\sqrt{n}}\right)\Big),$$
$$\mathbb{E}\left[\min_{0\leq k\leq K-1} gap(x^k)\right] = \mathcal{O}\left(\tfrac{\Delta_0}{\sqrt{K}} + \tfrac{LD^2}{\sqrt{K}}\left(1 + \tfrac{\widetilde{L}}{L}\tfrac{\omega^{\frac{3}{2}}}{\sqrt{n}}\right)\right).$$

MARINA FW. Next, as a natural generalization of the idea of compressing gradient differences, we arrive at the fact that the gradient estimator could be biased. We have already seen this idea take place with the examples of `L-SVRG FW` vs `SARAH FW` (Section 4.3.3), as well as `SEGA FW` vs `JAGUAR` (Section 4.3.3). Therefore, we consider the work by [212], where the authors base their method on the SARAH technique but with compression stochasticity instead of (4.5). Our algorithm utilizes this idea and performs the update rule (4.17) with

Algorithm 44 `MARINA Frank-Wolfe`

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$, probability p
Initialization: choose $x^0 \in \mathcal{X}$; $g_i^0 = \nabla f_i(x^0)$, $g^0 = \nabla f(x^0)$

for $k = 0, 1, 2, \ldots K-1$ **do**
  Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle s, g^k\rangle$
  Update $x^{k+1} = (1-\eta_k)x^k + \eta_k s^k$
  Broadcast g^k to all workers
  for $i = 1, \ldots, n$ **do**
    Update $c_i^{k+1} = \begin{cases} \nabla f_i(x^{k+1}) - g_i^k, & \text{with probability } p \\ \mathcal{Q}(\nabla f_i(x^{k+1}) - \nabla f_i(x^k)), & \text{with probability } 1-p \end{cases}$
    Send c_i^{k+1} to the server
    Update $g_i^{k+1} = g_i^k + c_i^{k+1}$
  end for
  Set $g^{k+1} = g^k + \frac{1}{n}\sum_{i=1}^n c_i^{k+1}$
end for

where g_i^{k+1} are computed on the local devices and g^{k+1} are computed on the server side. In fact, with p close to zero, one can note that to compute g^{k+1} we typically need only compressed differences: $g^{k+1} = g^k + \frac{1}{n}\sum_{i=1}^n \mathcal{Q}(\nabla f_i(x^{k+1}) - \nabla f_i(x^k))$. However, rarely (with probability p) we need to send the full uncompressed gradients.

Lemma 4.3.17. Under Assumptions 4.0.1, 4.0.2, 4.1.1 Algorithm 44 satisfies Assumption 4.3.1 with: $\rho_1 = p$, $\rho_2 = 1$, $A = 0$, $B = \frac{(1-p)\omega L^2}{n}$, $C = 0$, $\sigma_k = 0$, $E = 0$.

Corollary 4.3.18. For Algorithm 44 in the convex and non-convex cases respectively the following convergences take place:

$$\mathbb{E}[\, f(x^K) - f^*] = \mathcal{O}\Big(\Delta_0 \exp\left(-\tfrac{pK}{4}\right) + \tfrac{LD^2}{K}\Big(1 + \sqrt{\tfrac{\omega}{pn}}\,\Big)\Big),$$
$$\mathbb{E}\Big[\min_{0\leq k\leq K-1} gap(x^k)\Big] = \mathcal{O}\Big(\tfrac{\Delta_0}{\sqrt{K}} + \tfrac{LD^2}{\sqrt{K}}\Big(1 + \sqrt{\tfrac{\omega}{pn}}\Big)\Big).$$

EF21 FW. To complement our results on distributed methods with compression, we again move to biased compressors:

Definition 4.3.19. Map $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is a biased compression operator, if there exist a constant $\delta \geq 1$ such that for all $x \in \mathbb{R}^d$

$$\mathbb{E}[\|\mathcal{C}(x) - x\|^2] \leq \left(1 - \frac{1}{\delta}\right)\|x\|^2.$$

As mentioned before in Section 2.5, this is a broader class than unbiased operators. Here, for example, one can find the greedy choice of coordinates [370], sparse decompositions [42] and other operators interesting in practice (see Chapter 5). Intuition suggests that using biased/greedy compressors may improve convergence over unbiased operators. However, biased compression is

less "suitable" in theory than unbiased ones. Indeed, one can construct a simple convex quadratic problem for which distributed SGD with Top1 compression diverges exponentially fast (see Chapter 5). This issue can be resolved using error compensation technique [44, 232, 217]. Then we consider one of the state-of-art algorithms with error compensation techique for biased compression [45]:

Algorithm 45 `EF21 Frank-Wolfe`

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$
Initialization: choose $x^0 \in \mathcal{X}$; $g_i^0 = \nabla f_i(x^0)$, $g^0 = \frac{1}{n}\sum_{i=1}^n g_i^0$
for $k = 0, 1, 2, \ldots K-1$ **do**
  Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle s, g^k\rangle$
  Update $x^{k+1} = (1-\eta_k)x^k + \eta_k s^k$
  Broadcast x^{k+1} to all workers
  for $i = 1, \ldots, n$ **do**
    Compress $c_i^{k+1} = \mathcal{C}(\nabla f_i(x^{k+1}) - g_i^k)$ and send to the server
    Update $g_i^{k+1} = g_i^k + c_i^{k+1}$
  end for
  Update $g^{k+1} = g^k + \frac{1}{n}\sum_{i=1}^n c_i^{k+1}$
end for

Here as in Algorithm 44 the computation of g_i^k takes place on the local devices and the computation of g^k – on the server and for that only compressed differences $\mathcal{C}(\nabla f_i(x^{k+1}) - g_i^k)$ are needed. Note that Algorithm 45 does not send uncompressed packages at all.

Lemma 4.3.20. Under Assumptions 4.0.1, 4.0.2, 4.1.1 Algorithm 45 satisfies Assumption 4.3.1 with: $\rho_1 = 1$, $A = 1$, $B = 0$, $C = 0$, $\sigma_k^2 = \frac{1}{n}\sum_{i=1}^n \|g_i^k - \nabla f_i(x^k)\|^2$, $\rho_2 = \frac{1}{2\delta}$, $E = 2\delta\widetilde{L}^2$.

Corollary 4.3.21. For Algorithm 45 in the convex and non-convex cases respectively the following convergences take place:

$$\mathbb{E}[f(x^K) - f^*] = \mathcal{O}\Big(\Delta_0 \exp(-\tfrac{K}{8\delta}) + \tfrac{LD^2}{K}\Big[1 + \tfrac{\widetilde{L}}{L}\delta\Big]\Big),$$
$$\mathbb{E}\Big[\min_{0\leq k\leq K-1} gap(x^k)\Big] = \mathcal{O}\Big(\tfrac{\Delta_0}{\sqrt{K}} + \tfrac{LD^2}{\sqrt{K}}\Big[1 + \tfrac{\widetilde{L}}{L}\delta\Big]\Big).$$

One more method with compression will be presented in the next section. A comparison of the presented and already existing methods is provided in Table 4.6.

Combinations of Different Approaches

In this section, we show that combinations of already presented methods can also be analyzed using Assumption 4.3.1.

SAGA SARAH FW. We start from Algorithm 33.

Lemma 4.3.22. Under Assumptions 4.0.1, 4.0.2, 4.1.1 Algorithm 33 satisfies Assumption 4.3.1 with: $\rho_1 = \frac{b}{2n}$, $A = \frac{b}{2n^2}$, $B = \frac{2\widetilde{L}^2}{b}$, $C = 0$, $\sigma_k^2 = \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^k) - y_j^{k+1}\|^2$, $\rho_2 = \frac{b}{2n}$, $E = \frac{2n\widetilde{L}^2}{b}$.

Table 4.6: Summary of complexity results for finding an ε-solution distributed constrained minimization problems (4.4) with n devices by projection free methods with compression with parameters ω and δ (see Definitions 4.3.14 and 4.3.19). Convergence is measured by the functional distance to the solution in the convex case and by the gap function in the non-convex case. Complexities are given in terms of the number of transmitted coordinates if we choose RandK and TopK as particular cases of compressors.

Method	Link	Reference	Any compression?	CVX	nCVX
`dFW`	Alg. 3	[325]	✗	$\mathcal{O}(\frac{1}{\varepsilon})$	✗
`DIANA FW`	Alg. 43	**NEW**, [60]	✓ (unbiased)	$\widetilde{\mathcal{O}}(1+\frac{1}{\varepsilon}(\frac{1}{\omega}+\frac{\sqrt{\omega}}{\sqrt{n}}))$	$\mathcal{O}(\frac{1}{\varepsilon^2}(\frac{1}{\omega}+\frac{\omega^2}{n}))$
`MARINA FW`	Alg. 44	**NEW**, [212]	✓ (unbiased)	$\widetilde{\mathcal{O}}(1+\frac{1}{\varepsilon}(\frac{1}{\omega}+\frac{1}{\sqrt{n}}))$	$\mathcal{O}(\frac{1}{\varepsilon^2}(\frac{1}{\omega}+\frac{\omega}{n}))$
`VR MARINA FW`	Alg. 47	**NEW**, [212]	✓ (unbiased)	$\widetilde{\mathcal{O}}(1+\frac{1}{\varepsilon}(\frac{1}{\omega}+\frac{1}{\sqrt{n}}))$	$\mathcal{O}(\frac{1}{\varepsilon^2}(\frac{1}{\omega}+\frac{\omega}{n}))$
`EF21 FW`	Alg. 45	**NEW**, [45]	✓	$\widetilde{\mathcal{O}}(1+\frac{1}{\varepsilon})$	$\mathcal{O}(\frac{\delta}{\varepsilon^2})$
`Q-L-SVRG FW`	Alg. 46	**NEW**	✓ (unbiased)	$\widetilde{\mathcal{O}}(1+\frac{1}{\varepsilon}(\frac{1}{\omega}+\frac{\sqrt{\omega}}{\sqrt{n}}))$	$\mathcal{O}(\frac{1}{\varepsilon^2}(\frac{1}{\omega}+\frac{\omega^2}{n}))$

Corollary 4.3.23. For Algorithm 33 in the convex and non-convex cases the following convergences take place:

$$\mathbb{E}[f(x^K)-f^*] = \mathcal{O}\left(\Delta_0 \exp\left(-\frac{bK}{8n}\right) + \frac{LD^2}{K}\left[1+\frac{\widetilde{L}}{L}\frac{\sqrt{n}}{b}\right]\right),$$
$$\mathbb{E}\left[\min_{0\leq k\leq K-1} gap(x^k)\right] = \mathcal{O}\left(\frac{\Delta_0}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left[1+\frac{\widetilde{L}}{L}\frac{\sqrt{n}}{b}\right]\right).$$

Also different combinations of techniques can be considered in the distributed case. In all approaches from Section 4.3.3, we considered that we have a full gradient available for each f_i. But for example we can assume that f_i also has the finite-sum form $f_i = \sum_{j=1}^m f_{ij}$. Thus one can add the techniques from Section 4.3.3. We consider two illustrations of this type of combination.

Q-L-SVRG FW. As noted above, MARINA (Algorithm 44) is based on SARAH. We can perform exactly the same manipulation with L-SVRG and convert it from a method for stochastic sum-type problems into a method with compression:

Algorithm 46 `Q-L-SVRG Frank-Wolfe`

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$, batch size b, probability p
Initialization: choose $w^0 = x^0 \in \mathcal{X}$

for $k = 0, 1, 2, \ldots K-1$ **do**
  Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle s, g^k\rangle$
  Update $x^{k+1} = (1-\eta_k)x^k + \eta_k g^k$
  Update $w^{k+1} = \begin{cases} x^k, & \text{with probability } p \\ w^k, & \text{with probability } 1-p \end{cases}$
  Broadcast x^{k+1} to all workers
  for $i = 1, \ldots, n$ **do**
    Compress $c_i^{k+1} = \mathcal{C}(\nabla f_i(x^{k+1}) - g_i^k)$ and send to the server
    Update $g_i^{k+1} = g_i^k + c_i^{k+1}$
  end for
  Update $g^{k+1} = \frac{1}{n}\sum_{i=1}^n \mathcal{Q}(\nabla f_i(x^{k+1}) - \nabla f_i(w^{k+1})) + \nabla f(w^{k+1})$
end for

The same way as in Algorithm 44, to compute g^{k+1} on the server, we typically need only to compress differences $\mathcal{Q}(\nabla f_i(x^{k+1}) - \nabla f_i(w^{k+1}))$. However, with a small probability p, we need to transfer full packages to the server and compute new $\nabla f(w^{k+1})$.

Lemma 4.3.24. Under Assumptions 4.0.1, 4.0.2, 4.1.1 Algorithm 46 satisfies Assumption 4.3.1 with: $\rho_1 = 1$, $A = \omega\widetilde{L}^2(1-\frac{p}{2})$, $B = \omega\widetilde{L}^2\left(1+\frac{8(1-p)}{p}\right)$, $C = 0$, $\sigma_k^2 = \|x^k - w^k\|^2$, $\rho_2 = \frac{p}{2}$, $E = 1 + \frac{8(1-p)}{p}$.

Corollary 4.3.25. For Algorithm 46 in the convex and non-convex cases accordingly the following convergences take place:

$$\mathbb{E}[f(x^K) - f^*] = \mathcal{O}\left(\Delta_0 \exp\left(-\frac{Kp}{8}\right) + \frac{\widetilde{L}D^2}{K+\frac{1}{p}}\left[1+\frac{\sqrt{\omega}}{p}\right]\right),$$
$$\mathbb{E}\left[\min_{0\leq k\leq K-1} gap(x^k)\right] = \mathcal{O}\left(\frac{\Delta_0}{\sqrt{K}} + \frac{\widetilde{L}D^2}{\sqrt{K}}\left[1+\frac{\sqrt{\omega}}{p}\right]\right).$$

VR-MARINA FW. Continuing to develop the idea of combining different approaches, we present a modification of `MARINA FW`, which is an adaptation from [212]. We first present the algorithm and then provide a detailed convergence result along with its proof.

Algorithm 47 `VR-MARINA Frank-Wolfe`

Parameters: stepsizes $\{\eta_k\}_{k\geq 0}$, batch size b, probability p
Initialization: choose $x^0 \in \mathcal{X}$, $g^0 = \nabla f(x^0)$

for $k = 0, 1, 2, \ldots, K-1$ **do**
  Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle s, g^k\rangle$
  Update $x^{k+1} = (1-\eta_k)x^k + \eta_k s^k$
  Broadcast g^k to all workers
  for $i = 1, \ldots, n$ **do**
    Generate batch $|S_k^i| = b$
    Update $c_i^{k+1} = \begin{cases} \nabla f_i(x^{k+1}) - g_i^k, & \text{with probability } p \\ \mathcal{Q}\left(\sum\limits_{j\in S_k^i} (\nabla f_{ij}(x^{k+1}) - \nabla f_{ij}(x^k))\right), & \text{with probability } 1-p \end{cases}$
    Send c_i^{k+1} to the server
    Update $g_i^{k+1} = g_i^k + c_i^{k+1}$
  end for
  Set $g^{k+1} = g^k + \frac{1}{n}\sum_{i=1}^n c_i^{k+1}$
end for

In this section, we assume that the local loss on each node has either a finite-sum form: $f_i(x) = \frac{1}{m}\sum\limits_{j=1}^{m} f_{ij}(x)$.

Assumption 4.3.26. (Average $\mathcal{L}$-smoothness)**.** For all $k \geq 0$ and $i \in [n]$ the minibatch stochastic gradients difference $\widetilde{\Delta}_i^k = \frac{1}{b}\sum\limits_{S_b^k}\left(\nabla f_{ij}(x^{k+1}) - \nabla f_{ij}(x^k)\right)$ computed on the i-th machine satisfies:

$$\mathbb{E}\left[\widetilde{\Delta}_i^k \mid x^k, x^{k+1}\right] = \Delta_i^k, \quad \mathbb{E}\left[\|\widetilde{\Delta}_i^k - \Delta_i^k\|^2 \mid x^k, x^{k+1}\right] \leq \frac{\mathcal{L}_i}{b}\|x^{k+1} - x^k\|^2,$$

with some $\mathcal{L}_i > 0$ and $\Delta_i^k = \nabla f_i(x^{k+1}) - \nabla f_i(x^k)$.

This assumption is satisfied in many cases. In particular, if $S_{i,k}^{'} = \{1\ldots, m\}$, then $\mathcal{L}_i = 0$, and if $S_{i,k}^{'}$ consists of $b^{'}$ i.i.d. samples from the uniform distributions on $\{1\ldots, m\}$ and f_{ij} are L_{ij}-smooth, then $\mathcal{L}_i \leq \max_{j\in[m]} L_{ij}$. Under this and the previously introduced assumptions, we derive the following result.

Lemma 4.3.27. Under Assumptions 4.1.1, 4.3.26 Algorithm 47 satisfies Assumption 4.3.1 with:

$$\rho_1 = p,\ A = 0,\ B = \frac{(1-p)}{n}(\omega L^2 + \frac{(1+\omega)\mathcal{L}^2}{b}),\ C = 0,\ \sigma_k = 0,\ \rho_2 = 1,\ E = 0.$$

Corollary 4.3.28. For Algorithm 47 in the convex and non-convex cases the following conver-

gences take place:

$$\mathbb{E}\left[r_{K+1}\right] = \mathcal{O}\left(\left(f(x^0) - f(x^*)\right)\exp\left(-\frac{pK}{4}\right) + \frac{D^2}{K+\frac{1}{p}}\left[L + \sqrt{\frac{(1-p)\left(\omega L^2 + \frac{1+\omega}{b}\mathcal{L}^2\right)}{np}}\right]\right).$$

$$\mathbb{E}\left[\min_{0\leq k\leq K-1} \mathbf{gap}(x^k)\right] = \mathcal{O}\left(\frac{f(x^0) - f(x^*)}{\sqrt{K}} + \frac{D^2}{\sqrt{K}}\left[L + \sqrt{\frac{(1-p)\left(\omega L^2 + \frac{1+\omega}{b}\mathcal{L}^2\right)}{np}}\right]\right).$$

The optimal choice of parameter p for `VR-MARINA FW` remains the same as in case of `MARINA FW`, hence $p = \frac{k}{d+k}$.

4.3.4 Experiments

In this section, we present experimental results to support our theoretical findings.

We consider the logistic regression problem:

$$\min_{x\in\mathcal{X}} f(x) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-b_i \cdot x^T a_i)), \tag{4.19}$$

where x are weights and $\{a_i, b_i\}_{i=1}^n$ are training data samples with $a_i \in \mathbb{R}^d$ and $b_i \in \{-1, 1\}$. We choose $\mathcal{X}$ as the l_1-ball with radius $r = 2\cdot 10^3$. The linear minimization oracle, i.e. $\arg\min_{s\in\mathcal{X}} \langle g, s\rangle$, for this constraint set can be computed in the closed-form: $s^* = -\text{sign}(g_i)e_i$ with $i = \arg\max_j |g_j|$. We take LibSVM datasets [171].

Stochastic methods. In this experiment, we test the performance of proposed stochastic Frank-Wolfe-based methods (`SAGA SARAH FW`, `L-SVRG FW`) and compare them to deterministic version of it. The performance is measured in number of full gradients computed. Hyperparameters of the methods are chosen according to theory and all methods starts from zero. The results are provided in Figure 4.10. One can note that `SAGA SARAH FW` outperforms competitors. It confirms our theoretical conclusions perfectly.

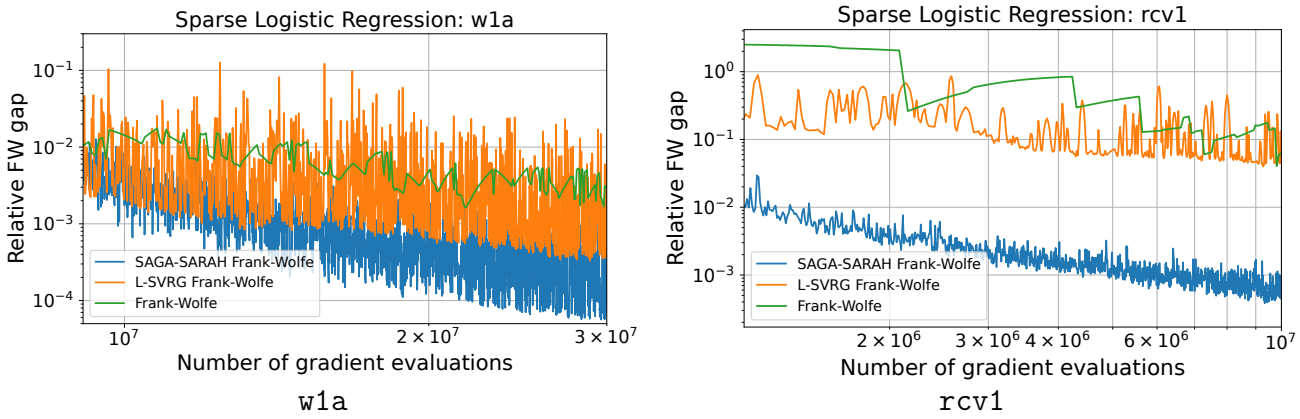


Figure 4.10: Comparison of methods for solving (4.19) in the stochastic case. SARAH FW, SAGA SARAH FW are considered. The comparison is made on LibSVM datasets w1a, rcv1.

In addition, we compared the methods with stochastic baselines chosen from [321], [317], [320]:

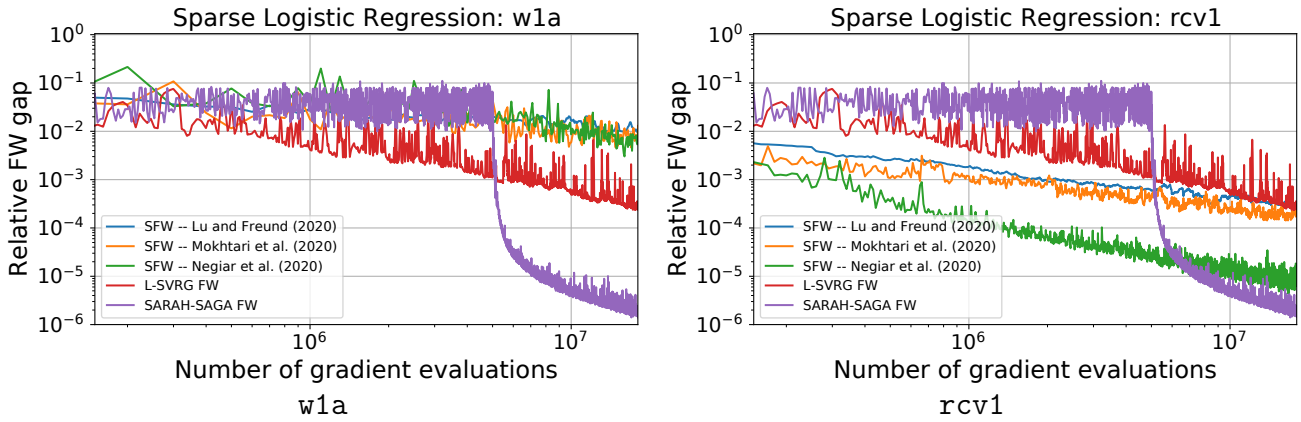


Figure 4.11: Comparison of stochastic methods with baselines. SARAH FW, SAGA SARAH FW are considered. The comparison is made on LibSVM datasets w1a, rcv1.

Based on the obtained results, one can see, `SAGA SARAH FW` overcomes all competitors on both datasets.

Finally, for this section of methods, we introduce a comparison with methods for constrained optimization that employ Euclidean projection instead of linear minimization. We consider `L-SVRG FW`, `SAGA SARAH FW`. As competitors, we take the effective $O(n)$ time point Euclidean projection method from [371] combined with SGD, as well as SVRG:

According to the conducted experiment (Figure 4.12), regardless of long-term convergence, the proposed algorithms `L-SVRG` and `SARAH-SAGA` are comparatively better than their direct competitors. Despite the modest success compared to effective projection algorithms, the results can be considered significant.

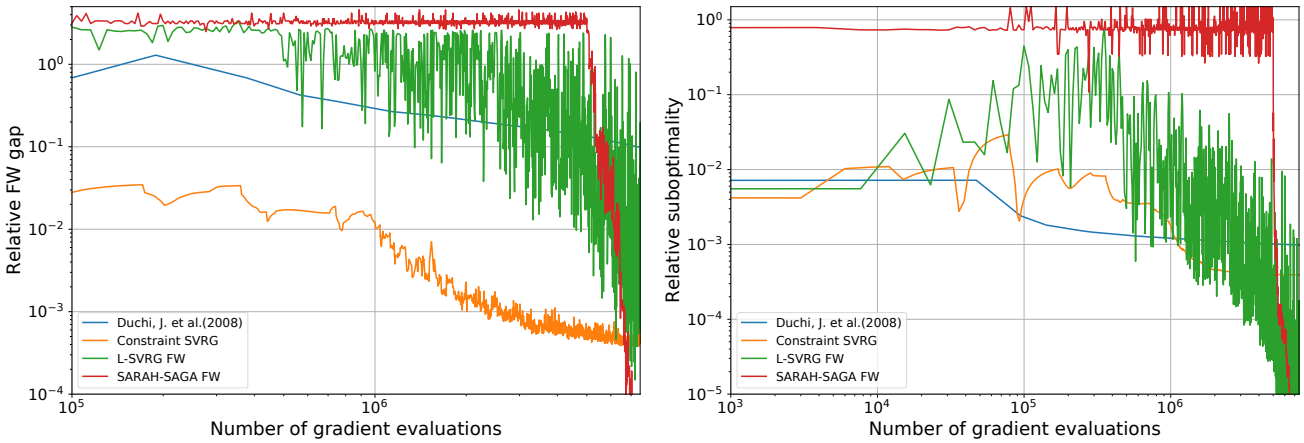


Figure 4.12: Comparison of Frank-Wolfe-based algorithms and methods with projection for the stochastic problem. L-SVRG FW, SAGA SARAH FW as well as SGD and SVRG with projection are considered. The comparison is made on LibSVM dataset mushrooms.

Coordinate methods. In the second experiment, we check the convergence of alleged coordinate methods: `SEGA FW`, `JAGUAR` and compare them to the original Frank-Wolfe method. The performance is measured by the number of full gradients computed. Hyperparameters of the

methods are chosen based on theoretical considerations, and all methods start from zero. The results are presented in Figure 4.13.

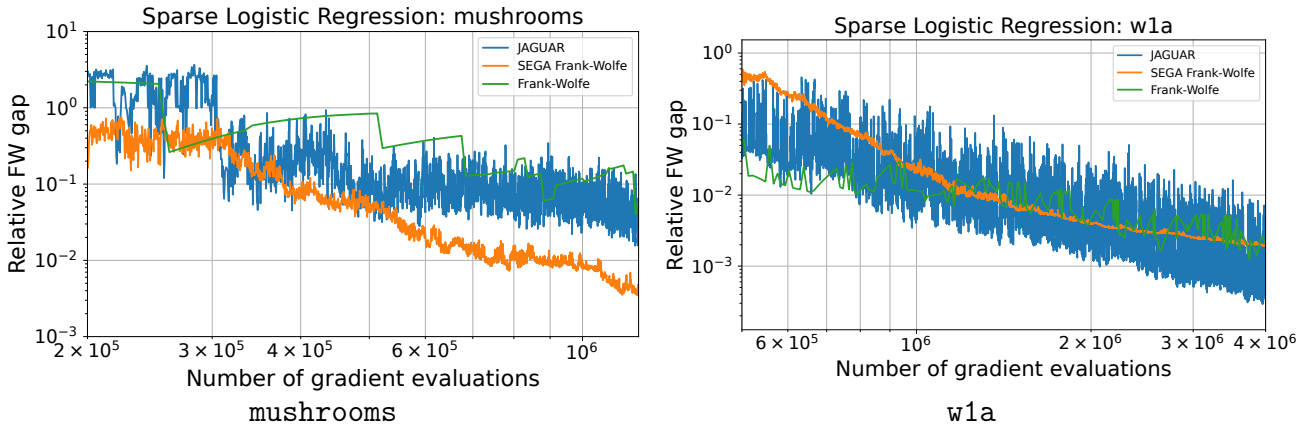


Figure 4.13: Comparison of methods for solving (4.19) in the coordinate case. SEGA FW, JAGUAR are considered. The comparison is made on LibSVM datasets mushrooms, w1a.

Distributed methods. In the last experiment, we consider the distributed setup of (4.19), where we assume that the information about f_i is available only to worker i. Here, we compare the distributed methods proposed in this section: `MARINA Frank-Wolfe`, `VR-MARINA Frank-Wolfe`, `DIANA Frank-Wolfe` and `EF21 Frank-Wolfe`. The performance is measured in number of bits communicated from workers to the server. Hyperparameters of the methods are chosen according to theory, and all methods start from zero. In `MARINA` and `DIANA` algorithms RandK (random sparsification) compression is used, while in `EF21` TopK ("greedy" sparsification) compression is implemented. Convergence performance is shown in Figure 4.14. The advantage of using compressed communication is clearly observable in every case.

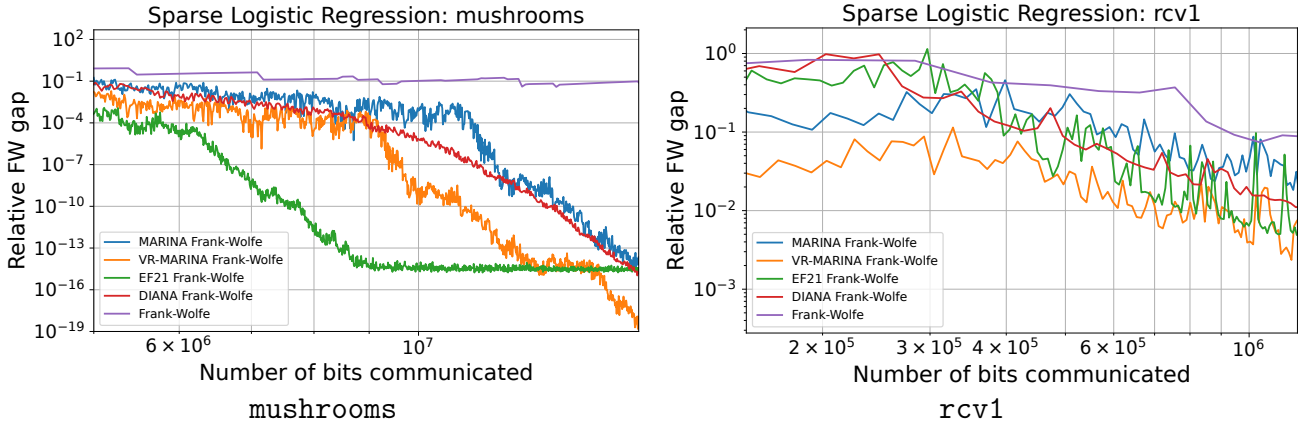


Figure 4.14: Comparison of methods for solving (4.19) in the distributed case. MARINA FW, VR-MARINA FW, EF21 FW, DIANA FW are considered. The comparison is made on LibSVM datasets mushrooms, rcv1.

Chapter 5

Beyond Unbiased Stochasticity

This chapter is a collection of several relatively distant subjects united by a single idea. In the previous chapters, we interacted extensively with stochastic methods. The theory of such methods often relies on the simple assumption that the stochastic gradient/operator approximation serves as an unbiased estimate of the true gradient/operator, with respect to the conditional expectation on the current randomness. However, in previous chapters, we occasionally departed from this setting and observed that more robust approaches could be achieved in this manner. We discussed biased compressions and advanced variance reduction methods. In this chapter, we continue these two subjects in more detail. Additionally, we introduce a third topic centered on Markovian randomness.

Since the three stories are weakly connected, each individual section can be perceived independently. All necessary details from the introduction to the experiments are provided within.

5.1 Variance Reduction without Full Gradients Computing via Shuffling

In this section, we address the problem of minimizing a finite-sum problem of the form

$$\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\right\}. \tag{5.1}$$

Of course, we have encountered such a task before in previous chapters. But still, for the sake of completeness, let us discuss it again.

Problems of the form (5.1) are very common in, for example, supervised learning [51]. Let a training dataset consist of n pairs, i.e., $\{(a_i, b_i)\}_{i=1}^n$, where $a_i \in \mathbb{R}^d$ is a feature vector for a datapoint i and b_i is the corresponding label. Then, for example, the least squares regression problem corresponds to (5.1) with $f_i(x) = \frac{1}{2}(a_i^T x - b_i)^2$. If $b_i \in \{-1, 1\}$ indicates a class, then logistic regression is obtained by choosing $f_i(x) = \log(1 + \exp(-b_i a_i^T x))$.

Recently, many algorithms have been proposed for solving (5.1). In this section, we are interested in a subclass of these algorithms that fall into a stochastic gradient descent (SGD) framework originating from the work of Robbins and Monro in '50s [46]. Let v^k be some sort of (possibly

stochastic and very rough) approximation of $\nabla f(x^k)$, then many SGD-type algorithms update the x as follows:

$$x^{k+1} = x^k - \gamma_k v^k. \tag{5.2}$$

The classical SGD defines $v^k = \nabla f_i(x^k)$, where $i \in [n]$ is chosen randomly [372] or its mini-batch version [373], where $v^k = \frac{1}{|S|}\sum_{i\in S} \nabla f_i(x^k)$, with $S \subset [n]$. Even with unbiased gradient estimates of SGD, where $\mathbb{E}[x^k|x^k] = \nabla f(x^k)$, the variance of x^k is the main source of slower convergence [374, 375, 376].

5.1.1 Related Works and Contributions

Recently, many variance-reduced variants of SGD have been proposed, including SAG/SAGA [305, 55, 274], SVRG [52, 282, 306], MISO [307], SARAH [53, 276, 53, 277], SPIDER [309], STORM [278], PAGE [54], and many others. Generally speaking, the variance-reduced variants of SGD aim to sample $\mathcal{O}(1)$ functions and use their gradients to update x^k. For example, SVRG [52] fixes a point $\tilde{w}$, in which a full gradient $\nabla f(\tilde{w})$ is computed and subsequently, the stochastic gradient is defined as $x^k = \nabla f_i(x^k) - \nabla f_i(\tilde{w}) + \nabla f(\tilde{w})$, where $i \in [n]$ is picked at random.

SARAH Algorithm. The SARAH algorithm [53], on the other hand, updates x^k recursively. It starts with a full gradient computation $v_0 = \nabla f(w_0)$, then taking a step (5.2) and updating the gradient estimate recursively as $v^k = \nabla f_i(x^k) - \nabla f_i(x^{k-1}) + x^{k-1}$. For a smooth and strongly convex problem, the procedure highlighted above converges, but not to the optimal solution x^*. Therefore, similar to SVRG, the process is restarted after i) a predefined number of iterations, ii) randomly [308, 377, 54], or iii) decided in run-time by computing the ratio $\|x^k\|/\|v^0\|$ (SARAH+ [53]), and a new full gradient estimate has to be computed. To address this issue, e.g., in [276], they proposed inexact SARAH (iSARAH), where the full gradient estimate is replaced by a mini-batch gradient estimate $v^0 = \frac{1}{|S|}\sum_{i\in S} f_i(x^0)$, where $S \subset [n]$. To find a point $\hat{x}$ such that $\mathbb{E}\nabla f(\hat{x}) \leq \epsilon$, the mini-batch size has to be chosen as $|S| \sim \mathcal{O}(\frac{1}{\epsilon})$, and the step-size will be $\gamma \sim \mathcal{O}(\frac{\epsilon}{L})$.

There are a few variants of SARAH that do not require any restart or full-gradient estimates. For example, the Hybrid Variance-Reduced variant [378] defines

$$x^k = \beta \nabla f_i(x^k) + (1-\beta)\left(\nabla f_i(x^k) - \nabla f_i(x^{k-1}) + x^{k-1}\right), \tag{5.3}$$

where $\beta \in (0, 1)$ is a hyperparameter. The STORM variant [278] uses (5.3) not with a fixed value of parameter β, but in STORM, the value of β_k diminishes to 0. The ZeroSARAH [308] is another variant, where the x^k is a combination of (5.3) with SAG/SAGA.

Random Sampling vs. Random Reshuffle. All the stochastic algorithms discussed so far sample functions f_i randomly. However, it is standard practice, for a finite-sum problem, not to choose functions f_i randomly with replacement, but rather to make a data permutation/shuffling and then choose the f_is in a cyclic fashion [379, 380].

This is primarily because methods involving data shuffling are more cache-friendly, leading to a significant acceleration in their implementation, as noted in [381]. From learning and optimization perspectives, a cyclic pass through the data appears more beneficial, as it ensures that each data point is utilized an equal number of times throughout the training process. Hence, shuffling strategies are frequently used for real-world practical problems, including those encountered in deep learning, as outlined in [382].

The basic shuffling approaches include

- **Random Reshuffling (RR)** - reshuffle data before each epoch;
- **Shuffle-Once (SO)** - shuffle data only once before optimizing;
- **Incremental Gradient (IG)** - access data in a cycling fashion over the given dataset.

In recent years, numerous studies have provided theoretical analyses of various shuffling techniques, both in the context of classical SGD [383, 384, 385, 386, 387, 388] and in variance reduction methods [389, 390, 391, 392, 393, 394, 395, 396].

In Table 5.1 we summarize the results from the literature on stochastic methods that involve independent generation of batch numbers and shuffling techniques. In the case of independent generation, the results do not claim to be complete and are provided rather for information and comparison.

The main contribution of this section is the modification of the SARAH algorithm to remove the requirement of computing a full gradient $\nabla f(x)$, while achieving linear convergence with a fixed step-size for strongly convex objectives. The crucial algorithmic modification needed to achieve this goal was to replace the random selection of functions with either of the shuffle options (**RR**, **SO**, **IG**) and to design a mechanism that can build a progressively better approximation of a full gradient $\nabla f(x^k)$ as $x^k \to x^*$.

One can note that, according to Table 5.1, our Algorithm 48 is one of the few methods with shuffling that simultaneously does not compute full gradients, does not use large amounts of additional memory, and converges linearly for smooth strongly convex finite-sum problems (5.1). Moreover, with a sufficiently small δ (see the discussion after Assumption 5.1.1 and at the end of Section 5.1.2), Algorithm 48 has state-of-the-art guarantees of convergence among methods using the shuffling technique.

Table 5.1: Summary of results on solving a smooth strongly convex finite-sum optimization problem. The results are given both for independent generation of batch numbers and for different shuffling techniques. In the case of independent generation, the results are given for reference and make no claim to completeness.

	Method	No Full Grad.?	Memory	Lin. Conv.?	Complexity
Independent	GD [285]	✗	$\mathcal{O}(d)$	✓	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
	AGD [397]	✗	$\mathcal{O}(d)$	✓	$\mathcal{O}\left(n\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$
	SGD [46, 47, 398, 194]	✓	$\mathcal{O}(d)$	✗	$\mathcal{O}\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}+\frac{\sigma_*^2}{\mu^2\varepsilon}\right)$
	ASGD [399, 65]	✓	$\mathcal{O}(d)$	✗	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}+\frac{\sigma^2}{\mu^2\varepsilon}\right)$
	SAGA [55]	✓	$\mathcal{O}(nd)$	✓	$\mathcal{O}\left(\left[n+\frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right)$
	SVRG [52]	✗	$\mathcal{O}(d)$	✓	$\mathcal{O}\left(\left[n+\frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right)$
	SARAH [53]	✗	$\mathcal{O}(d)$	✓	$\mathcal{O}\left(\left[n+\frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right)$
	Katyusha [56]	✗	$\mathcal{O}(d)$	✓	$\mathcal{O}\left(\left[n+\sqrt{n\frac{L}{\mu}}\right]\log\frac{1}{\varepsilon}\right)$
Shuffling	SGD [383, 384, 385, 386, 387]	✓	$\mathcal{O}(d)$	✗	$\mathcal{O}\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}+\sqrt{\frac{L\sigma_*^2}{\mu^3 n\varepsilon}}\right)$
	IAG [389, 390]	✓	$\mathcal{O}(nd)$	✓	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
	DIAG [391]	✓	$\mathcal{O}(nd)$	✓	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
	SAGA [392, 393, 394]	✓	$\mathcal{O}(nd)$	✓	$\mathcal{O}\left(n\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
	SVRG [392, 395]	✗	$\mathcal{O}(d)$	✓	$\mathcal{O}\left(n\frac{L^{3/2}}{\mu^{3/2}}\log\frac{1}{\varepsilon}\right)$
	AVRG [394]	✓	$\mathcal{O}(d)$	✓	$\mathcal{O}\left(n\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
	Prox-DFinito [396]	✓	$\mathcal{O}(nd)$	✓	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
	`Shuffled-SARAH` (this thesis)	✓	$\mathcal{O}(d)$	✓	$\mathcal{O}\left(\left[n\frac{L}{\mu}+n^2\frac{\delta}{\mu}\right]\log\frac{1}{\varepsilon}\right)$
	`RR-SARAH` (this thesis)	✗	$\mathcal{O}(d)$	✓	$\mathcal{O}\left(\left[n\frac{L}{\mu}+n^2\frac{\delta}{\mu}\right]\log\frac{1}{\varepsilon}\right)$

Columns: No Full Grad.? = whether the method computes full gradients, Memory = number of additional memory, Lin. Conv.? = whether the method converges linearly to the solution x^*, Complexity = oracle complexity on the terms f_i to find the solution (give the best result of the works listed in References).

Notation: μ = constant of strong monotonicity of f, L = Lipschitz constant of ∇f_i, δ = similarity parameter, $\sigma_*^2 = \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x^*)\|^2$ (in [386], the authors use a slightly different definition), σ^2 = upper bound for $\|\nabla f_i(x) - \nabla f(x)\|^2$, n = size of the dataset, d = dimension of the problem, ε = accuracy of the solution.

5.1.2 Algorithms

Building Gradient Estimate While Optimizing

An intuition. By accessing data in a cyclic order, employing any of the alternative methods mentioned earlier, we can approximate the full gradient $\tilde{v} \approx \nabla f$. In fact, if we set the stepsize γ_k in (5.2) to zero and let x^k represent a stochastic gradient ∇f_i, then by averaging all the stochastic gradients in a single pass, we would obtain the exact full gradient $\nabla f(x)$. As we increase the value of γ, the stochastic gradients would be computed at different points.

$$\tilde{v} = \frac{1}{n} \sum_{i=1}^{n} \nabla f_{\pi^i}(x^i), \tag{5.4}$$

and hence we would not obtain the exact full gradient of $f(x)$ but rather a rough estimate. However, is it solely γ that affects the quality of the $\tilde{v}$? Certainly not, as x^k is updated using (5.2). One can observe that the radius of a set of ws used to compute gradient estimates is dependent on x^k. Ideally, as we converge to x^*, then also $v^k \to \nabla f(x^*) = 0$ and hence $\tilde{v}$ get closer to $\nabla f(x^k)$.

Building the gradient estimate. Our proposed approach to eliminate the need to compute the full gradient is based on a simple recursive update. Let us initialize $\tilde{v}^0 = 0 \in \mathbb{R}^d$. Then while making a pass $i = \{1, 2, \ldots, n\}$ over the data, we will keep updating $\tilde{v}$ using the gradient estimates as follows

$$\tilde{v}^i = \frac{i-1}{i} \tilde{v}^{i-1} + \frac{1}{i} \nabla f_{\pi^i}(x^i), \quad \text{for } i \in \{1, 2, \ldots, n\}.$$

It is an easy exercise to see that $\tilde{v}^i$ will be the average of gradients seen so far, and moreover, after n updates, it will be exactly as in (5.4). Let us note that making the pass over the dataset is crucial to build a good estimate of the gradient and random selection of functions would not achieve this goal.

Algorithm. We are now ready to present the `Shuffled-SARAH` algorithm, as outlined in Algorithm 48. The random permutation in line 2 could have one of the three options mentioned in Section 5.1.1.

Theoretical Analysis

Before presenting our theoretical results we introduce some notations and assumptions.

Assumption 5.1.1. For problem (5.1) the following hold:

(i) Each $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex and twice differentiable, with L-Lipschitz gradient:

$$\|\nabla f_i(x_1) - \nabla f_i(x_2)\| \leq L \|x_1 - x_2\|,$$

for all $x_1, x_2 \in \mathbb{R}^d$;

Algorithm 48 `Shuffled-SARAH`

Parameters: stepsize γ
Initialization: choose $x_0^0 \in \mathbb{R}^d$; $\widetilde{v}_0^0 = 0, v_0 = 0$

for epochs $k = 0, 1, 2, \ldots, K$ **do**
  Obtain a permutation $\pi_k^1, \ldots, \pi_k^n$ of $[n]$
  $v_k^0 = v_k$
  $x_k^1 = x_k^0 - \gamma v_k^0$
  for $i = 1, 2, 3, \ldots, n$ **do**
    $\widetilde{v}_k^{i+1} = \frac{i-1}{i}\widetilde{v}_k^i + \frac{1}{i}\nabla f_{\pi_k^i}(x_k^i)$
    $v_k^i = \nabla f_{\pi_k^i}(x_k^i) - \nabla f_{\pi_k^i}(x_k^{i-1}) + v_k^{i-1}$
    $x_k^{i+1} = x_k^i - \gamma v_k^i$
  end for
  $x_{k+1}^0 = x_k^{n+1}$
  $\widetilde{v}_{k+1}^1 = 0$
  $v_{k+1} = \widetilde{v}_k^{n+1}$
end for

(ii) $f(x)$ is a μ-strongly convex function with minimizer x^* and optimal value $f(x^*)$;
(iii) Each f_i is δ-similar with f, i.e. for all $x \in \mathbb{R}^d$ it holds that

$$\|\nabla^2 f_i(x) - \nabla^2 f(x)\| \leq \delta/2,$$

where $\|\cdot\|$ is the operator norm between matrices.

We have known the last assumption since Chapter 2, but for completeness, let us recall that this assumption pertains to the Hessian similarity of $\{f_i\}_{i=1}^n$. This effect is prominently seen when the data with $n \cdot s$ samples is randomly partitioned across n batches $\{\{z_j^i\}_{j=1}^s\}_{i=1}^n$ and f_i is defined as $f_i(x) = \frac{1}{s}\sum_{j=1}^s \ell(x, z_j^i)$, where ℓ is a L-smooth and convex loss function. One can see that $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x) = \frac{1}{ns}\sum_{i=1}^n \sum_{j=1}^s \ell(x, z_j^i)$ forms the total loss function over the whole dataset.

Subsequently, one can use Hoeffding's inequality for matrices [203] and obtain that with probability $1 - p$ it holds

$$\|\nabla^2 f_i(x) - \nabla^2 f(x)\| \leq \sqrt{\frac{32L^2 \log(d/p)}{s}}.$$

It turns out that we can consider that $\delta \sim \frac{L}{\sqrt{s}}$. Moreover, in the case when ℓ is a quadratic function $\delta \sim \frac{L}{s}$ [41].

The following theorem presents the convergence guarantees of `Shuffled-SARAH`.

Theorem 5.1.2. Suppose that Assumption 5.1.1 holds. Consider `Shuffled-SARAH` (Algorithm 1) with the choice of γ such that

$$\gamma \leq \min\left[\frac{1}{8nL}; \frac{1}{8n^2\delta}\right]. \tag{5.5}$$

Then, we have

$$f(x^{k+1}) - f(x^*) + \frac{\gamma(n+1)}{16}\|v^k\|^2 \leq \left(1 - \frac{\gamma\mu(n+1)}{2}\right)\left(f(x^k) - f(x^*) + \frac{\gamma(n+1)}{16}\left\|v^{k-1}\right\|^2\right).$$

Hence, it is easy to obtain an estimate for the number of outer iterations in Algorithm 48.

Corollary 5.1.3. Fix ε, and let us run `Shuffled-SARAH` with γ from (5.5). Then we can obtain an ε-approximate solution (in terms of $f(x) - f^* \leq \varepsilon$) after

$$S = \mathcal{O}\left(\left[n \cdot \frac{L}{\mu} + n^2 \cdot \frac{\delta}{\mu}\right]\log\frac{1}{\varepsilon}\right) \quad \text{calls of terms } f_i.$$

Difference between RR, SO, IG. A crucial detail is that we achieve deterministic convergence estimates irrespective of the generation strategy π_k employed in line 2 of Algorithm 48. Concurrently, it is anticipated that the diverse characteristics of π_k may result in variations in theoretical convergence. Indeed, different shuffling techniques have a theoretical impact on the SGD method as discussed in the most recent research findings [386, 388]. However, a significant detail of these findings is that the differences only manifest in the portion that depends on the variance bound of the stochastic gradient σ^2. But SARAH, like other methods incorporating variance reduction, nullifies the effect of σ^2. The most credible results on the effect of variance reduction and shuffling techniques do not discern a considerable difference among **RR**, **SO**, and **IG** generally [395, 396]. Minor improvements are detectable in specific cases [395] or include only logarithmic factors [396]. Consequently, the question of distinguishing between **RR**, **SO**, and **IG** in variance reduction algorithms remains unanswered.

Optimal choice of s. In the case of $\delta \sim \frac{L}{\sqrt{s}}$, it is required to take $\frac{n}{\sqrt{s}} \sim 1$, i.e. $s = N^{2/3}$, where N is the total sample size: $N = ns$.

Algorithm 49 `RR-SARAH`

Parameters: stepsize γ
Initialization: choose $x_0^0 \in \mathbb{R}^d$; $\widetilde{v}_0^0 = 0, v_0 = 0$

1: **for** epochs $k = 0, 1, 2, \ldots, K$ **do**
2: Sample a permutation $\pi_k^1, \ldots, \pi_k^n$ of $[n]$
3: $v_k^0 = \nabla f(x_k^0)$
4: $x_k^1 = x_k^0 - \gamma v_k^0$
5: **for** $i = 1, 2, 3, \ldots, n$ **do**
6: $v_k^i = \nabla f_{\pi_k^i}(x_k^i) - \nabla f_{\pi_k^i}(x_k^{i-1}) + v_k^{i-1}$
7: $x_k^{i+1} = x_k^i - \gamma v_k^i$
8: **end for**
9: $x_{k+1}^0 = x_k^{n+1}$
10: **end for**

5.1.3 Experiments

Trajectory. We start with a toy experiment in $\mathbb{R}^2$ involving a quadratic function. We compare the trajectories of classical SARAH (two random and average), the average trajectory of the `RR-SARAH` (see Algorithm 49), and the random trajectory of `Shuffled-SARAH` with Random Reshuffling. One can observe that Algorithm `RR-SARAH` exhibits a stable trajectory. `Shuffled-SARAH` performs worse but demonstrates improvement over the random trajectory of the original SARAH.

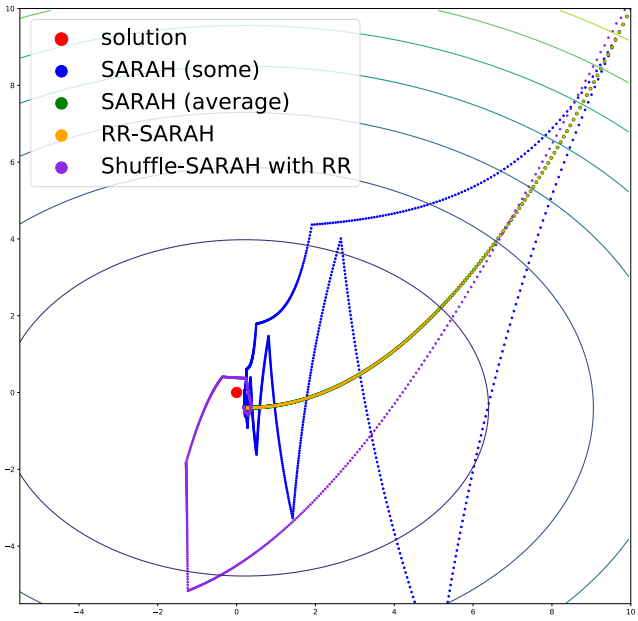


Figure 5.1: Trajectories on quadratic function.

Logistic regression. Next, we consider the logistic regression problem with ℓ_2-regularization for binary classification with

$$f_i(x) = \frac{1}{s}\sum_{i=1}^{s} \log\left(1 + \exp\left(-b_i x^T a_i\right)\right) + \frac{\lambda}{2}\|x\|^2,$$

where a_i are objects, $b_1, \ldots, b_s \in \{-1, 1\}$ are labels for these objects, s is the size of the datasets for f_i and $x \in \mathbb{R}^d$ is a vector of weights. We optimize this problem for `mushrooms`, `a9a`, `w8a` datasets from LIBSVM library[171]. Since we only have the estimate on s for our method (see the end of Section 5.1.2), it is not completely fair to use it for all methods, therefore we take as s the nearest power of two to $N/100$, where N is the full size of the data. More details on the dataset parameters can be found in Table 5.2. For each dataset, we take the regularization parameter $\lambda = 0.001 \cdot L$. One can note that the strong convexity constant of the problem $\mu = \lambda$. We compare the following method settings: 1) SARAH with theoretical parameters $m = 4.5 \cdot (L/\mu)$, $\gamma = 1/(2L)$ (m is the number of iterations of the inner loop – see [53]), 2) SARAH with optimal parameters (selected by brute force – see Table 5.3), 3) `RR-SARAH` with optimally tuned step-size, 4) `Shuffled-SARAH` (Random Reshuffling) with optimally tuned step-size, 5) `Shuffled-SARAH` (Shuffle Once) with optimally tuned step-size. All methods are run 20 times, and the convergence results are averaged. We are interested in how these methods converge in terms of the number of epochs (1 epoch is a

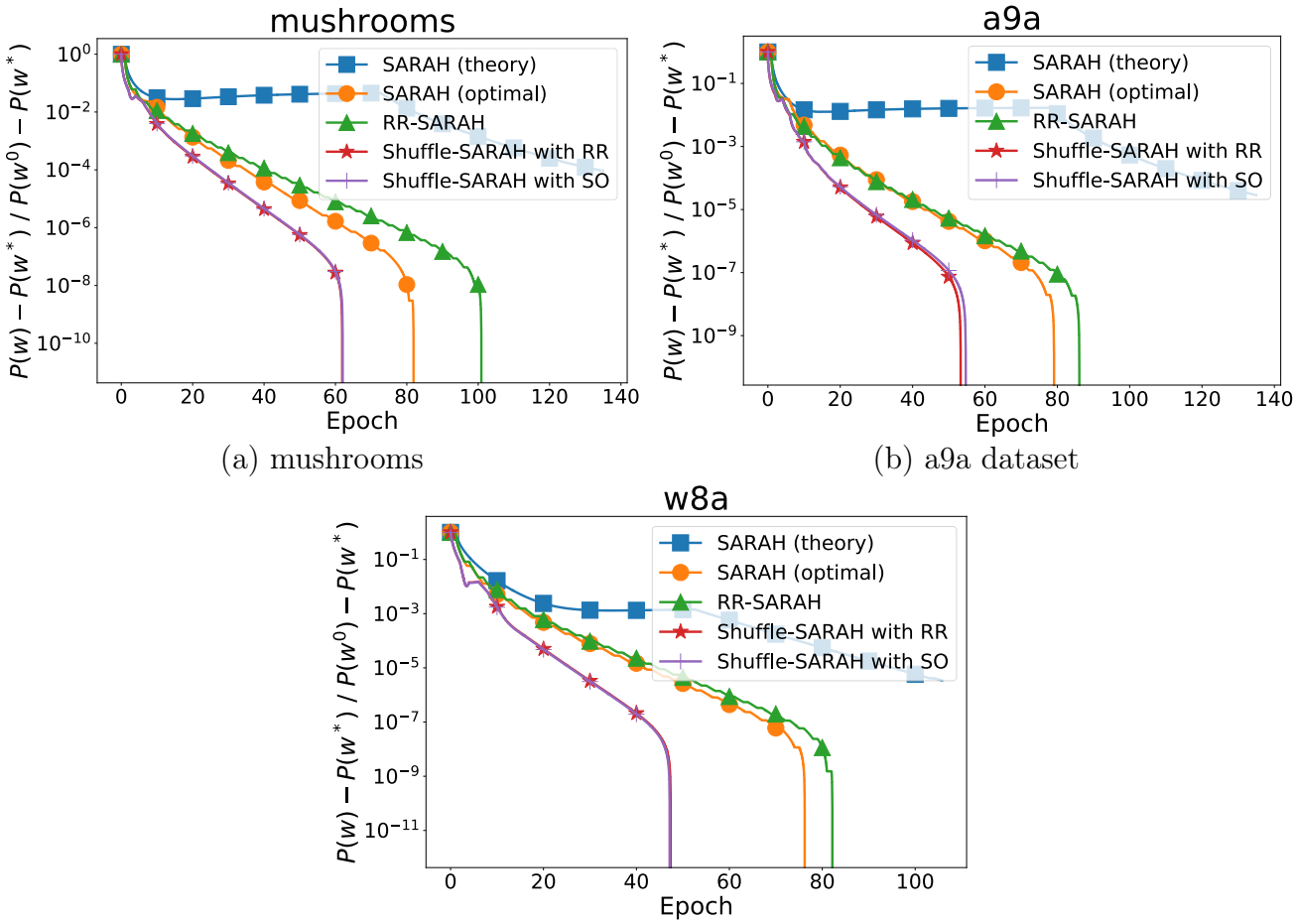
call of the full gradient of f). For results, see Figures 5.2, 5.3, 5.4. One can note that in these cases, our new methods are superior to the original SARAH.

Table 5.2: Summary of datasets.

	N	s	d	L
`mushrooms`	8124	64	112	5,3
`a9a`	32561	256	123	3,5
`w8a`	49749	256	300	28,5

Table 5.3: Optimal parameters for SARAH.

	m	γ
`mushrooms`	$0,5 \cdot (L/\mu)$	$1/L$
`a9a`	$0,25 \cdot (L/\mu)$	$1/L$
`w8a`	L/μ	$1/L$

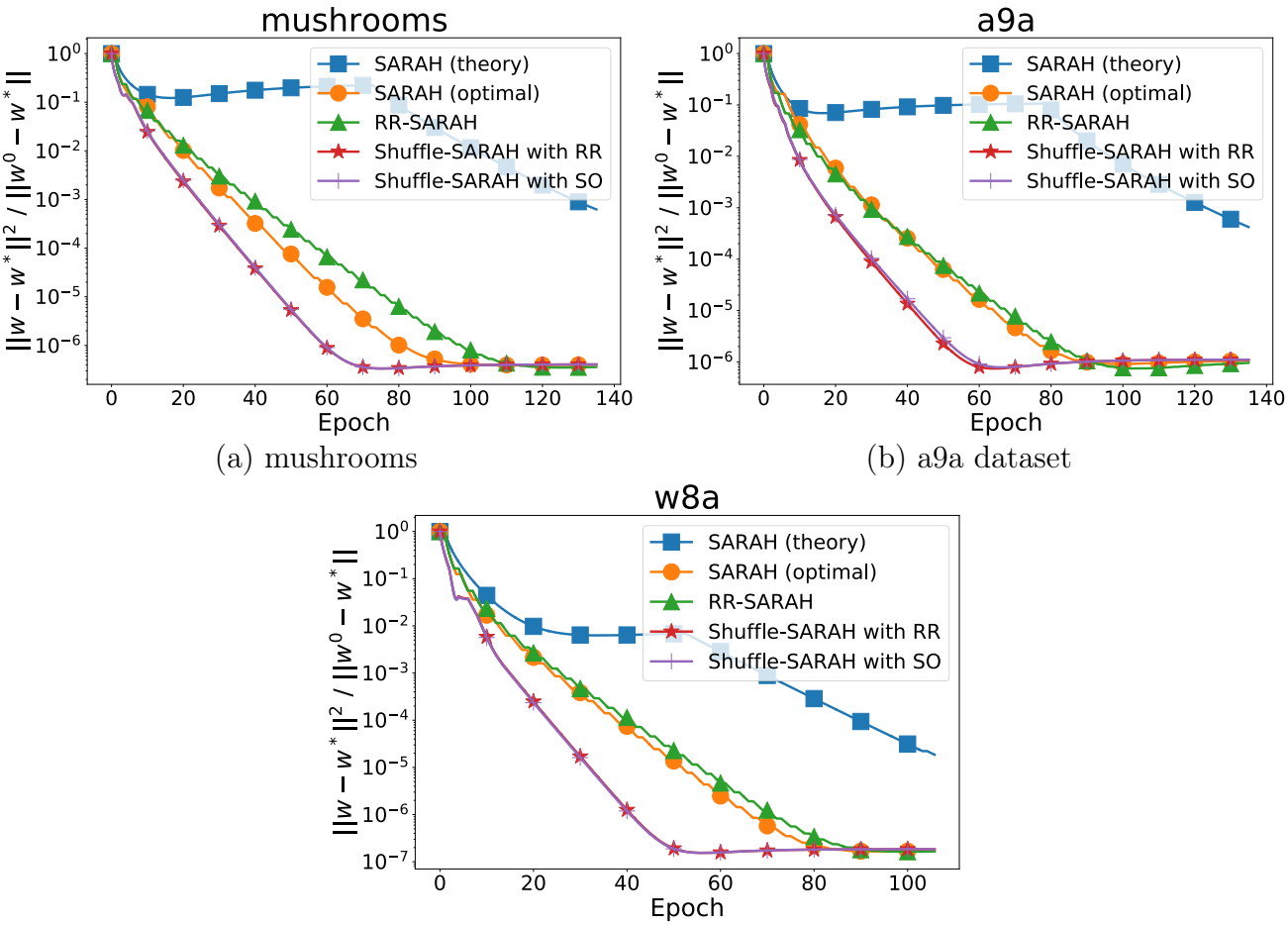


(a) mushrooms

(b) a9a dataset

(c) w8a dataset

Figure 5.2: Convergence of SARAH-type methods on various LiBSVM datasets. Convergence on the function.

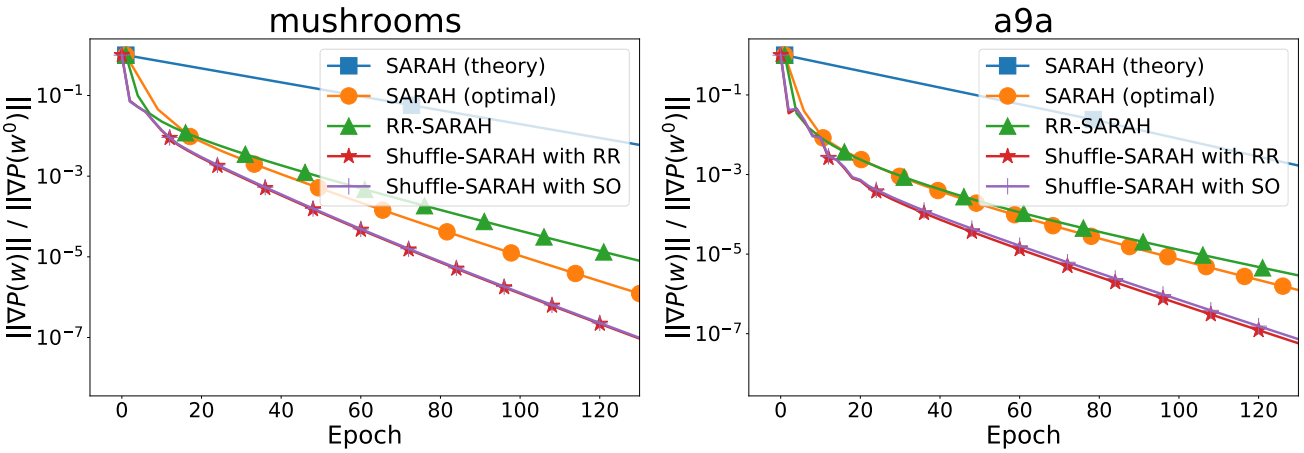


(a) mushrooms

(b) a9a dataset

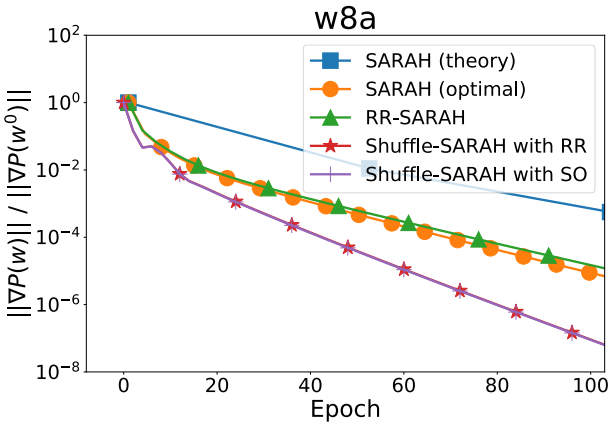
(c) w8a dataset

Figure 5.3: Convergence of SARAH-type methods on various LiBSVM datasets. Convergence on the distance to the solution.



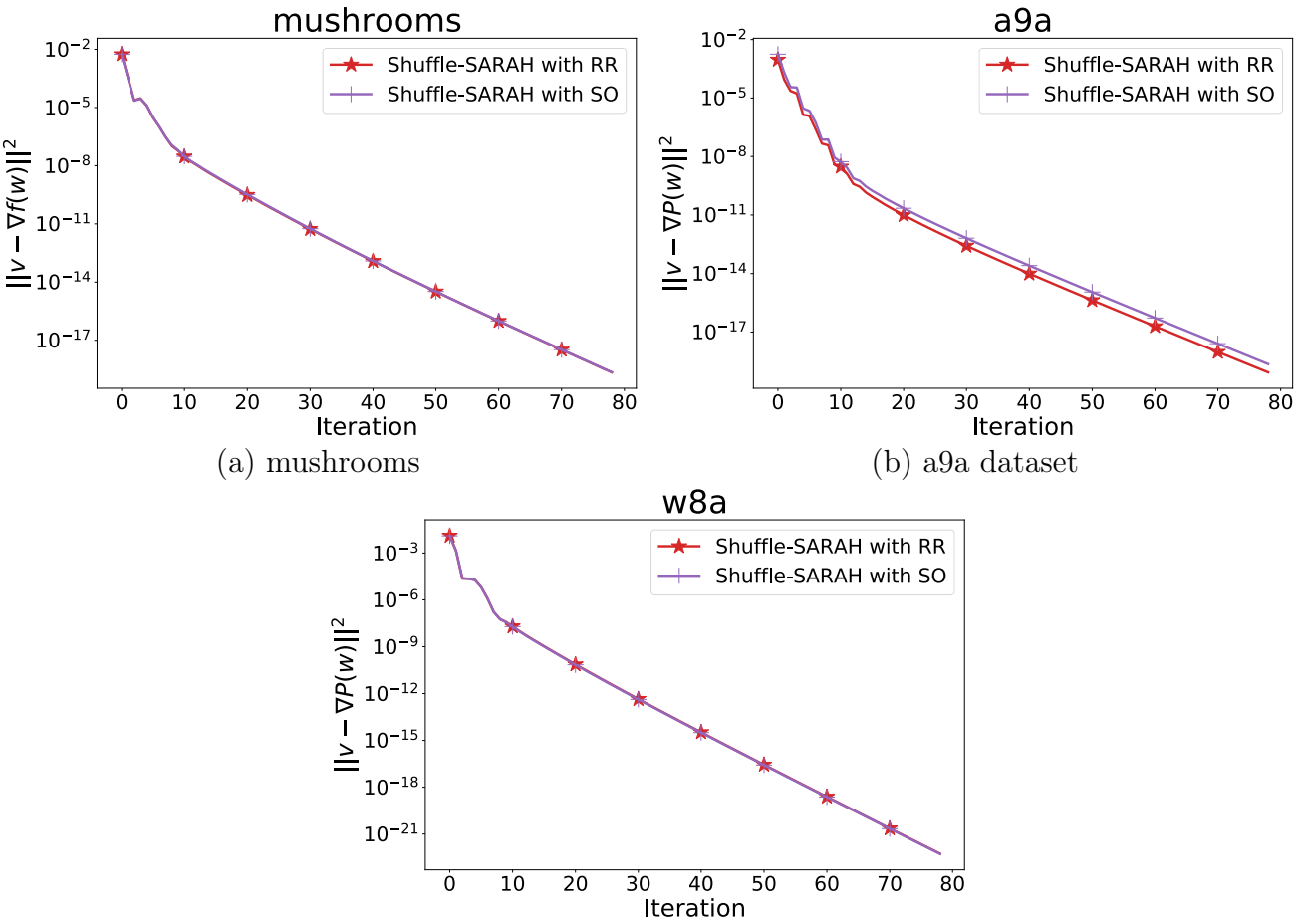
(a) mushrooms

(b) a9a dataset



(c) w8a dataset

Figure 5.4: Convergence of SARAH-type methods on various LIBSVM datasets. Convergence on the norm og the gradient.



(a) mushrooms

(b) a9a dataset

(c) w8a dataset

Figure 5.5: $\|v^k - \nabla f(x^k)\|^2$ changes.

The v^k is getting closer to $\nabla f(x^k)$. The goal of this experiment is to demonstrate that v provides a good approximation of ∇f and improves with each iteration. To achieve this, we analyze the changes of $\|v^k - \nabla f(x^k)\|^2$ on the logistic regression problem (refer to the previous paragraph). The results are shown in Figure 5.5. It is evident that the difference is decreasing $\|v^k - \nabla f(x^k)\|^2$.

5.2 Biased Compression for Distributed Learning

We already know about the importance of distributed computing in modern applications of optimization methods. We have extensively discussed distributed variational inequalities. Now, we focus on distributed minimization:

$$\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{M}\sum_{m=1}^{M} f_m(x)\right\}, \tag{5.6}$$

where $x \in \mathbb{R}^d$ collects the parameters of a statistical model to be trained, M represents the number of workers/devices, and $f_m(x)$ denotes the loss incurred by model x on data stored on worker m. The loss function $f_m : \mathbb{R}^d \to \mathbb{R}$ often assumes the form

$$f_m(x) := \mathbb{E}_{\xi_m\sim\pi_m}[f_{\xi_m}(x)],$$

with π_m being the distribution of training data owned by worker m. In federated learning applications, these local distributions can vary significantly, and no similarity assumption is imposed on them.

In this subsection, we contribute to a better understanding of the approach already known in this thesis for alleviating the communication bottleneck: communication compression. In particular, we study the theoretical properties of gradient-type methods that employ biased gradient compression operators, such as Top-k sparsification [370], or deterministic rounding [400]. Surprisingly, the current theoretical understanding of such methods is very limited. For instance, there is no general theory of such methods even in the $M = 1$ case. Only a handful of biased compression techniques have been proposed in the literature. We do not possess any theoretical understanding of why biased compression operators could outperform their unbiased counterparts and under what circumstances. More importantly, there is no good convergence theory for any gradient-type method with biased compression in the crucial $M > 1$ setting.

5.2.1 Related Works and Contributions

There has been extensive work related to communication compression, mostly focusing on unbiased compressions [43] as these are much easier to analyze. In particular, it was shown [66] that both the classical method with unbiased compression [43] and more advanced modifications [60, 265] can be considered as special versions of SGD. Subsequently, the results of [66] for strongly convex problems were transferred to general convex [367] and non-convex [366] target functions. In the meantime, works concerning biased compressions show stronger empirical results but with limited or no analysis [42, 242, 401]. There have been several attempts to address this issue; e.g., [402] provided analysis for quadratics in a distributed setting, [403] gave analysis for momentum SGD with a specific biased compression, but under unreasonable assumptions, i.e., bounded gradient norm and memory. The first result that obtained a linear rate of convergence for biased compression was achieved by [215], but only for one node and under the bounded gradient norm assumption,

which was later overcome by [232].

After the initial online appearance of our work, there have been several enhancements in the literature. In particular, [404] developed a theory for non-convex objectives in the single node setup. [405] designed a novel error-compensated SGD algorithm that converges linearly in a more relaxed setting with the help of an additional unbiased compressor. [220] proposed a simple trick to convert any biased compressor to the corresponding induced (unbiased) compressor, leading to improved theoretical guarantees. Recently, a new variant of the error feedback mechanism was introduced in [45, 226] showing improved rates for distributed non-convex problems.

Our main contributions are:

• **New general definitions.** We define and study three parametric classes of biased compression operators (see Section 5.2.3), which we denote $\mathbb{B}^1(\alpha,\beta)$, $\mathbb{B}^2(\gamma,\beta)$ and $\mathbb{B}^3(\delta)$, the first two of which are new. We prove that, despite they are alternative parameterization of the same collection of operators, the last two are more favorable than the first one, thus highlighting the importance of parametrization and providing further reductions. We show how the commonly used class of unbiased compression operators, which we denote $\mathbb{U}(\zeta)$, relates to these biased classes. We also study the scaling and compositions of such compressors.

• **New compressors.** We then proceed to provide a comprehensive list of both new and known biased (as well as some unbiased) compression operators that belong to the aforementioned classes in Section 5.2.3. A summary of all compressors considered is presented in Table 5.6.

• **Theory for $M = 1$.** In Section 5.2.4, we analyze compressed GD in the $M = 1$ case for compressors belonging to all three classes under the assumption of smoothness and strong convexity. Our theorems generalize existing results that hold for unbiased operators in a tight manner, and also recover the rate of GD in this regime. Our linear convergence results are summarized in Table 5.4.

Table 5.4: Complexity results for GD with biased compression. The identity compressor $C(x) \equiv x$ belongs to all classes with $\alpha = \beta = \gamma = \delta = 1$; all three results recover standard rate of Gradient Descent.

Compressor	$C \in \mathbb{B}^1(\alpha,\beta)$	$C \in \mathbb{B}^2(\gamma,\beta)$	$C \in \mathbb{B}^3(\delta)$
Theorem	Theorem 5.2.20	Theorem 5.2.21	Theorem 5.2.22
Complexity	$\mathcal{O}\left(\frac{\beta^2}{\alpha}\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$	$\mathcal{O}\left(\frac{\beta}{\gamma}\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$	$\mathcal{O}\left(\delta\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$

• **Biased better that unbiased?** We ask the question: do biased compressors outperform their unbiased counterparts in theory, and by how much? We answer this question by studying the performance of compressors under various synthetic and empirical statistical assumptions on the distribution of the entries of gradient vectors which need to be compressed. Particularly, we

quantify the gains of the Top-k sparsifier when compared against the unbiased Rand-k sparsifier in Section 5.2.3.

• **Theory for $M > 1$.** Finally, we study the important $M > 1$ setting in Section 5.2.5 and argue by giving a counterexample that a naive application of biased compression to distributed GD might diverge. We then show that the distributed SGD method equipped with an error-feedback mechanism [232] provably handles biased compressors. In our main result (Theorem 5.2.26; also see Table 5.5) we consider three learning schedules and iterate averaging schemes to provide three distinct convergence rates. Our analysis provides the first convergence guarantee for distributed gradient-type methods that provably converge for biased compressors, thus solving a major open problem in the literature.

Table 5.5: Ergodic convergence of distributed SGD with biased compression and error-feedback (Algorithm 50) for L-smooth and μ-strongly convex functions (K communications). Details are given in Theorem 5.2.26.

Stepsizes	Weights	Rate
$\mathcal{O}(\frac{1}{k})$	$\mathcal{O}(k)$	$\mathcal{O}\left(\frac{A_1}{K^2}+\frac{A_2}{K}\right)$
$\mathcal{O}(1)$	$\mathcal{O}(e^{-k})$	$\tilde{\mathcal{O}}\left(A_3\exp\left[-\frac{K}{A_4}\right]+\frac{A_2}{K}\right)$
$\mathcal{O}(1)$	1	$\mathcal{O}\left(\frac{A_3}{K}+\frac{A_5}{\sqrt{K}}\right)$

5.2.2 Assumptions

Assumption 5.2.1. Each function f_m is L-smooth on $\mathbb{R}^d$ with $L > 0$, i.e., it is differentiable and there is a constant $L > 0$ such that the following inequality holds for all $x, y \in \mathbb{R}^d$:

$$\|\nabla f_m(x) - \nabla f_m(y)\| \leq L\|x - y\|.$$

Assumption 5.2.2. The function f is μ-strongly convex on $\mathbb{R}^d$, i.e., it is continuously differentiable and there is a constant $\mu > 0$ such that the following inequality holds for all $x, y \in \mathbb{R}^d$:

$$\frac{\mu}{2}\|x - y\|^2 \leq f(x) - f(y) - \langle \nabla f(y), x - y\rangle.$$

In the following presentation, we assume that these assumptions are always satisfied.

5.2.3 Compression Operators

We are already familiar with some definitions of compressions (Chapter 2), but for the sake of completeness, we reintroduce them alongside new ones.

By a compression operator, we mean a (possibly random) mapping $C\colon \mathbb{R}^d \to \mathbb{R}^d$ with certain

constraints. Typically, literature considers *unbiased* compression operators C that have a bounded second moment, i.e.

Definition 5.2.3. Let $\zeta \geq 1$. We say that $C \in \mathbb{U}(\zeta)$ if C is unbiased (i.e., $\mathbb{E}C(x) = x$ for all x) and if the second moment is bounded as

$$\mathbb{E}\|C(x)\|^2 \leq \zeta \|x\|^2, \qquad \forall x \in \mathbb{R}^d.$$

Three classes of biased compressors

We focus on understanding *biased* compression operators, or "compressors" for short. We now introduce three classes of biased compressors; the first two are new and can be seen as natural extensions of unbiased compressors.

Definition 5.2.4. We say that $C \in \mathbb{B}^1(\alpha, \beta)$ for some $\alpha, \beta > 0$ if

$$\alpha \|x\|^2 \leq \mathbb{E}\|C(x)\|^2 \leq \beta \langle \mathbb{E}C(x), x\rangle, \qquad \forall x \in \mathbb{R}^d. \tag{5.7}$$

As we show next, the second inequality in (5.7) implies $\mathbb{E}\|C(x)\|^2 \leq \beta^2 \|x\|^2$.

Lemma 5.2.5. For any $x \in \mathbb{R}^d$, if $\mathbb{E}\|C(x)\|^2 \leq \beta \langle \mathbb{E}C(x), x\rangle$, then

$$\mathbb{E}\|C(x)\|^2 \leq \beta^2 \|x\|^2.$$

In the second class, we require the inner product between uncompressed x and compressed $C(x)$ vectors to dominate the squared norms of both vectors in expectation.

Definition 5.2.6. We say that $C \in \mathbb{B}^2(\gamma, \beta)$ for some $\gamma, \beta > 0$ if

$$\max\left\{\gamma \|x\|^2, \tfrac{1}{\beta}\mathbb{E}\|C(x)\|^2\right\} \leq \langle \mathbb{E}C(x), x\rangle \qquad \forall x \in \mathbb{R}^d.$$

Finally, in the third class, we require the compression error $\|C(x) - x\|^2$ to be strictly smaller than the squared norm $\|x\|^2$ of the input vector x in expectation.

Definition 5.2.7. We say $C \in \mathbb{B}^3(\delta)$ for some $\delta > 0$ if

$$\mathbb{E}\|C(x) - x\|^2 \leq \left(1 - \frac{1}{\delta}\right)\|x\|^2, \qquad \forall x \in \mathbb{R}^d.$$

This last definition was also considered by [231]. All three definitions require the compressed vector $C(x)$ to be in the neighborhood of the uncompressed vector x so that initial information is preserved with some accuracy. We now establish several basic properties and connections between the classes. We first show that the three classes of biased compressors defined above are equivalent in the following sense: a compressor from any of those three classes can be shown to belong to all three classes with different parameters and after possible scaling.

Theorem 5.2.8 (Equivalence between biased compressors)**.** Let $\lambda > 0$ be a free scaling parameter.

1. If $C \in \mathbb{B}^1(\alpha, \beta)$, then
   (i) $\beta^2 \geq \alpha$ and $\lambda C \in \mathbb{B}^1(\lambda^2\alpha, \lambda\beta)$, and
   (ii) $C \in \mathbb{B}^2(\alpha/\beta, \beta)$ and $\frac{1}{\beta}C \in \mathbb{B}^3(\beta^2/\alpha)$.
2. If $C \in \mathbb{B}^2(\gamma, \beta)$, then
   (i) $\beta \geq \gamma$ and $\lambda C \in \mathbb{B}^2(\lambda\gamma, \lambda\beta)$, and
   (ii) $C \in \mathbb{B}^1(\gamma^2, \beta)$ and $\frac{1}{\beta}C \in \mathbb{B}^3(\beta/\gamma)$.
3. If $C \in \mathbb{B}^3(\delta)$, then
   (i) $\delta \geq 1$, and
   (ii) $C \in \mathbb{B}^2\left(\frac{1}{2\delta}, 2\right) \subseteq \mathbb{B}^1\left(\frac{1}{4\delta^2}, 2\right)$.

Next, we show that, with a proper scaling, any unbiased compressor also belongs to all three classes of biased compressors.

Theorem 5.2.9 (From unbiased to biased with scaling)**.** If $C \in \mathbb{U}(\zeta)$, then for the scaled operator λC we have

(i) $\lambda C \in \mathbb{B}^1(\lambda^2, \lambda\zeta)$ for $\lambda > 0$,
(ii) $\lambda C \in \mathbb{B}^2(\lambda, \lambda\zeta)$ for $\lambda > 0$,
(iii) $\lambda C \in \mathbb{B}^3\left(\frac{1}{\lambda(2-\zeta\lambda)}\right)$ for $\zeta\lambda < 2$.

Examples of biased compressors: old and new

We now provide examples of compression operators that belong to the classes $\mathbb{B}^1$, $\mathbb{B}^2$, $\mathbb{B}^3$, and $\mathbb{U}$. Several of these operators are novel. For a summary, refer to Table 5.6.

Table 5.6: Compressors C described in Section 5.2.3 and their membership in $\mathbb{B}^1(\alpha,\beta)$, $\mathbb{B}^2(\gamma,\beta)$, $\mathbb{B}^3(\delta)$ and $\mathbb{U}(\zeta)$.

Compression Operator C	Unbiased?	α	β	γ	δ	ζ
Unbiased random sparsification	✓					d/k
Biased random sparsification **[NEW]**	✗	q	1	q	$1/q$	
Adaptive random sparsification **[NEW]**	✗	$1/d$	1	$1/d$	d	
Top-k sparsification [370]	✗	k/d	1	k/d	d/k	
General unbiased rounding **[NEW]**	✓					$\frac{1}{4}\sup\left(\frac{a_k}{a_{k+1}}+\frac{a_{k+1}}{a_k}+2\right)$
Unbiased exponential rounding **[NEW]**	✓					$\frac{1}{4}\left(b+\frac{1}{b}+2\right)$
Biased exponential rounding **[NEW]**	✗	$\left(\frac{2}{b+1}\right)^2$	$\frac{2b}{b+1}$	$\frac{2}{b+1}$	$\frac{(b+1)^2}{4b}$	
Natural compression [222]	✓					$9/8$
General exponential dithering **[NEW]**	✓					ζ_b
Natural dithering [222]	✓					ζ_2
Top-k + exponential dithering **[NEW]**	✗	k/d	ζ_b	k/d	$\zeta_b d/k$	

(a) For $k \in [d] := \{1,\dots,d\}$, the **unbiased random (aka Rand-k) sparsification** operator is defined via

$$C(x) := \frac{d}{k}\sum_{i\in S} x_i e_i, \tag{5.8}$$

where $S \subseteq [d]$ is the k-nice sampling; i.e., a subset of $[d]$ of cardinality k chosen uniformly at random, and $e_1,\dots,e_d$ are the standard unit basis vectors in $\mathbb{R}^d$.

Lemma 5.2.10. The Rand-k sparsifier (5.8) belongs to $\mathcal{U}(\frac{d}{k})$.

(b) Let $S \subseteq [d]$ be a random set, with probability vector $p := (p_1,\dots,p_d)$, where $p_i := \mathbb{P}(i \in S) > 0$ for all i (such a set is called a proper sampling [68]). Define **biased random sparsification** operator via

$$C(x) := \sum_{i\in S} x_i e_i. \tag{5.9}$$

Lemma 5.2.11. Letting $q := \min_i p_i$, the biased random sparsification operator (5.9) belongs to $\mathbb{B}^1(q,1)$, $\mathbb{B}^2(q,1)$, $\mathbb{B}^3(1/q)$.

(c) **Adaptive random sparsification** is defined via

$$C(x) := x_i e_i \quad \text{with probability} \quad \frac{|x_i|}{\|x\|_1}. \tag{5.10}$$

Lemma 5.2.12. Adaptive random sparsification operator (5.10) belongs to $\mathbb{B}^1(\frac{1}{d}, 1)$, $\mathbb{B}^2(\frac{1}{d}, 1)$, $\mathbb{B}^3(d)$.

(d) **Greedy (aka Top-k) sparsification** operator is defined via

$$C(x) := \sum_{i=d-k+1}^{d} x_{(i)} e_{(i)}, \tag{5.11}$$

where coordinates are ordered by their magnitudes so that $|x_{(1)}| \leq |x_{(2)}| \leq \cdots \leq |x_{(d)}|$.

Lemma 5.2.13. Top-k sparsification operator (5.11) belongs to $\mathbb{B}^1(\frac{k}{d}, 1)$, $\mathbb{B}^2(\frac{k}{d}, 1)$, and $\mathbb{B}^3(\frac{d}{k})$.

(e) Let $(a_k)_{k\in\mathcal{Z}}$ be an arbitrary increasing sequence of positive numbers such that $\inf a_k = 0$ and $\sup a_k = \infty$. Then **general unbiased rounding** C is defined as follows: if $a_k \leq |x_i| \leq a_{k+1}$ for some coordinate $i \in [d]$, then

$$C(x)_i = \begin{cases} \operatorname{sign}(x_i) a_k & \text{with probability } \frac{a_{k+1}-|x_i|}{a_{k+1}-a_k} \\ \operatorname{sign}(x_i) a_{k+1} & \text{with probability } \frac{|x_i|-a_k}{a_{k+1}-a_k} \end{cases} \tag{5.12}$$

Lemma 5.2.14. General unbiased rounding operator (5.12) belongs to $\mathcal{U}(\zeta)$, where

$$\zeta = \frac{1}{4} \sup_{k\in\mathcal{Z}} \left(\frac{a_k}{a_{k+1}} + \frac{a_{k+1}}{a_k} + 2 \right).$$

Notice that ζ is minimizing for exponential roundings $a_k = b^k$ with some basis $b > 1$, in which case $\zeta = \frac{1}{4}(b + 1/b + 2)$.

(f) Let $(a_k)_{k\in\mathcal{Z}}$ be an arbitrary increasing sequence of positive numbers such that $\inf a_k = 0$ and $\sup a_k = \infty$. Then **general biased rounding** is defined via

$$C(x)_i := \operatorname{sign}(x_i) \arg\min_{t\in(a_k)} |t - |x_i||, \qquad i \in [d]. \tag{5.13}$$

Lemma 5.2.15. General biased rounding operator (5.13) belongs to $\mathbb{B}^1(\alpha, \beta)$, $\mathbb{B}^2(\gamma, \beta)$, and $\mathbb{B}^3(\delta)$, where

$$\beta = \sup_{k\in\mathcal{Z}} \frac{2a_{k+1}}{a_k + a_{k+1}}, \qquad \gamma = \inf_{k\in\mathcal{Z}} \frac{2a_k}{a_k + a_{k+1}}, \qquad \alpha = \gamma^2, \qquad \delta = \sup_{k\in\mathcal{Z}} \frac{(a_k + a_{k+1})^2}{4a_k a_{k+1}}.$$

In the special case of exponential rounding $a_k = b^k$ with some base $b > 1$, we get

$$\alpha = \left(\frac{2}{b+1}\right)^2, \quad \beta = \frac{2b}{b+1}, \quad \gamma = \frac{2}{b+1}, \quad \delta = \frac{(b+1)^2}{4b}.$$

Remark 5.2.16. Plugging these parameters into the iteration complexities of Table 5.4, we find that the class $\mathbb{B}^3$ gives the best iteration complexity as $\frac{\beta^2}{\alpha} = b^2 > \frac{\beta}{\gamma} = b > \delta = \frac{(b+1)^2}{4b}$.

(g) **Natural compression** operator C_{nat} of [222] is the special case of general unbiased rounding operator (5.12) when $b = 2$. So,

$$C_{nat} \in \mathcal{U}\left(\frac{9}{8}\right).$$

(h) For $b > 1$, define **general exponential dithering** operator with respect to l_p-norm and with s exponential levels $0 < b^{1-s} < b^{2-s} < \cdots < b^{-1} < 1$ via

$$C(x) := \|x\|_p \times \operatorname{sign}(x) \times \xi\left(\frac{|x_i|}{\|x\|_p}\right), \tag{5.14}$$

where the random variable $\xi(t)$ for $t \in [b^{-u-1}, b^{-u}]$ is set to either b^{-u-1} or b^{-u} with probabilities proportional to $b^{-u} - t$ and $t - b^{-u-1}$, respectively.

Lemma 5.2.17. General exponential dithering operator (5.14) belongs to $\mathcal{U}(\zeta_b)$, where, letting $r = \min(p, 2)$,

$$\zeta_b = \frac{1}{4}\left(b + \frac{1}{b} + 2\right) + d^{\frac{1}{r}} b^{1-s} \min(1, d^{\frac{1}{r}} b^{1-s}). \tag{5.15}$$

(i) **Natural dithering** introduced by [222] without norm compression is the spacial case of general exponential dithering (5.14) when $b = 2$.

(j) **Ternary quantization** of [406] is the extreme case of general exponential dithering (5.14) with $s = 1$ levels and $b = 1$.

(k) **Top-k combined with exponential dithering.** Let C_{top} be the Top-k sparsification operator (5.11) and C_{dith} be general exponential dithering operator (5.14) with some base $b > 1$ and parameter ζ_b from (5.15). Define a new compression operator as the composition of these two:

$$C(x) := C_{dith}\left(C_{top}(x)\right). \tag{5.16}$$

Lemma 5.2.18. The composition operator (5.16) of Top-k sparsification and exponential dithering with base b belongs to $\mathbb{B}^1(\frac{k}{d}, \zeta_b)$, $\mathbb{B}^2(\frac{k}{d}, \zeta_b)$, $\mathbb{B}^3(\frac{d}{k}\zeta_b)$, where ζ_b is as in (5.15).

Superiority of Biased Compressors Under Statistical Assumptions

Here we highlight some advantages of biased compressors by comparing them with their unbiased counterparts. We evaluate compressors based on their average capacity to preserve gradient information, or, in other words, by the expected approximation error they produce. In the sequel, we assume that gradients have i.i.d. coordinates drawn from some distribution.

Top-k vs Rand-k

We now compare two sparsification operators: Rand-k (5.8) which is unbiased and which we denote as C^k_{rnd}, and Top-k (5.11) which is biased and which we denote as C^k_{top}. We define the variance of the approximation error of x via

$$\omega^k_{rnd}(x) := \mathbb{E}\left\|\frac{k}{d}C^k_{rnd}(x) - x\right\|^2 = \left(1 - \frac{k}{d}\right)\|x\|^2,$$

and

$$\omega^k_{top}(x) := \left\|C^k_{top}(x) - x\right\|^2 = \sum_{i=1}^{d-k} x^2_{(i)},$$

and the energy "saving" via

$$s^k_{rnd}(x) := \|x\|^2 - \omega^k_{rnd}(x) = \mathbb{E}\left\|\frac{k}{d}C^k_{rnd}(x)\right\|^2 = \frac{k}{d}\|x\|^2,$$

and

$$s^k_{top}(x) := \|x\|^2 - \omega^k_{top}(x) = \left\|C^k_{top}(x)\right\|^2 = \sum_{i=d-k+1}^{d} x^2_{(i)}.$$

Expectations in these expressions are taken with respect to the randomization of the compression operator rather than the input vector x. Clearly, there exists x for which these two operators incur identical variance, e.g. $x_1 = \cdots = x_d$. However, in practice, we apply compression to gradients x that evolve over time and may have heterogeneous components. In such situations, $\omega^k_{top}(x)$ could be much smaller than $\omega^k_{rnd}(x)$. This motivates a *quantitative study* of the *average case* behavior in which we make an *assumption* on the distribution of the coordinates of the compressed vector.

Uniform and exponential distribution. We first consider the case of uniform and exponentially distributed entries, and quantify the difference.

Lemma 5.2.19. Assume the coordinates of $x \in \mathbb{R}^d$ are i.i.d.
(a) If they follow a uniform distribution over $[0, 1]$, then

$$\frac{\mathbb{E}\omega^k_{top}}{\mathbb{E}\omega^k_{rnd}} = \left(1 - \frac{k}{d+1}\right)\left(1 - \frac{k}{d+2}\right), \qquad \frac{\mathbb{E}s^1_{top}}{\mathbb{E}s^1_{rnd}} = \frac{3d}{d+2}.$$

(b) If they follow a standard exponential distribution, then

$$\frac{\mathbb{E}s^1_{top}}{\mathbb{E}s^1_{rnd}} = \frac{1}{2}\sum_{i=1}^{d}\frac{1}{i^2} + \frac{1}{2}\left(\sum_{i=1}^{d}\frac{1}{i}\right)^2 = \mathcal{O}(\log^2 d).$$

Empirical comparison. Now we compare these two sparsification methods on an empirical bases and show the significant advantage of the greedy sparsifier over the random sparsifier. We assume that the coordinates of the vector to be compressed are i.i.d. Gaussian random variables.

First, we compare the savings s^k_{top} and s^k_{rnd} of these compressions. For random sparsification,

we have

$$\mathbb{E}s^k_{rnd}(x) = k \cdot (\sigma^2 + \mu^2),$$

where μ and σ^2 are the mean and variance of the Gaussian distribution. For computing $\mathbb{E}s^k_{top}(x)$, we use the probability density function of k-th order statistics (see (2.2.2) of [407]). Table 5.7 shows that Top-3 and Top-5 sparsifiers "save" 3×–40× more information in expectation, and the factor grows with the dimension.

Table 5.7: Information savings of greedy and random sparsifiers for $k = 3$ and $k = 5$.

	Top-3				Top-5			
d	10^2	10^3	10^4	10^5	10^2	10^3	10^4	10^5
$\mathcal{N}(0;1)$	$3 \cdot (\sigma^2 + \mu^2) = 3$				$5 \cdot (\sigma^2 + \mu^2) = 5$			
$\mathbb{E}s^k_{top}(x)$	18.65	31.10	43.98	57.08	27.14	47.70	69.07	90.85
$\mathcal{N}(2;1)$	$3 \cdot (\sigma^2 + \mu^2) = 15$				$5 \cdot (\sigma^2 + \mu^2) = 25$			
$\mathbb{E}s^k_{top}(x)$	53.45	75.27	95.81	115.53	81.60	118.56	153.13	186.22

Next, we compare normalized variances $\frac{\omega^k_{top}(x)}{\|x\|^2}$ and $\frac{\omega^k_{rnd}(x)}{\|x\|^2}$ for randomly generated Gaussian vectors. To provide a dimension-independent comparison, we analyze them against the average number of encoding bits per coordinate, which remains stable with respect to the dimension. Figure 5.6 reveals the superiority of the greedy sparsifier over the random one.

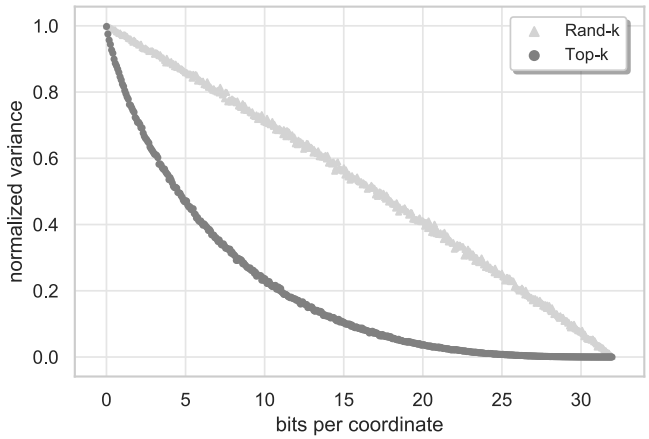


Figure 5.6: The comparison of Top-k and Rand-k sparsifiers with respect to normalized variance and the number of encoding bits used for each coordinate on average. Each point/marker represents a single $d = 10^4$ dimensional vector drawn form Gaussian distribution and then compressed by the specified operator. Each curve was obtained by varying the free parameter $k \in \{1, 2, \ldots, d\}$. Plots for different d look very similar. Notice that, for random sparsification the normalized variance is perfectly linear with respect to the number of bit per coordinate. Letting b be the total number of bits to encode the compressed vector (say in *binary32* system), the normalized variance produced by random sparsifier is almost $1 - \frac{b/d}{32}$. However, greedy sparsifier achieves exponentially lower variance $\approx 0.86^{b/d}$ utilizing the same amount of bits.

Practical distribution. We obtained various gradient distributions via logistic regression (*mushrooms* LIBSVM dataset) and least squares. We utilized the sklearn package and built Gaussian smoothing of the practical gradient density. The second moments, i.e. energy "saving, were already calculated from it using the formula for the density function of k-order statistics, see [407]. We concluded experiments for Top-5 and Rand-5; see Figure 5.7 for details.

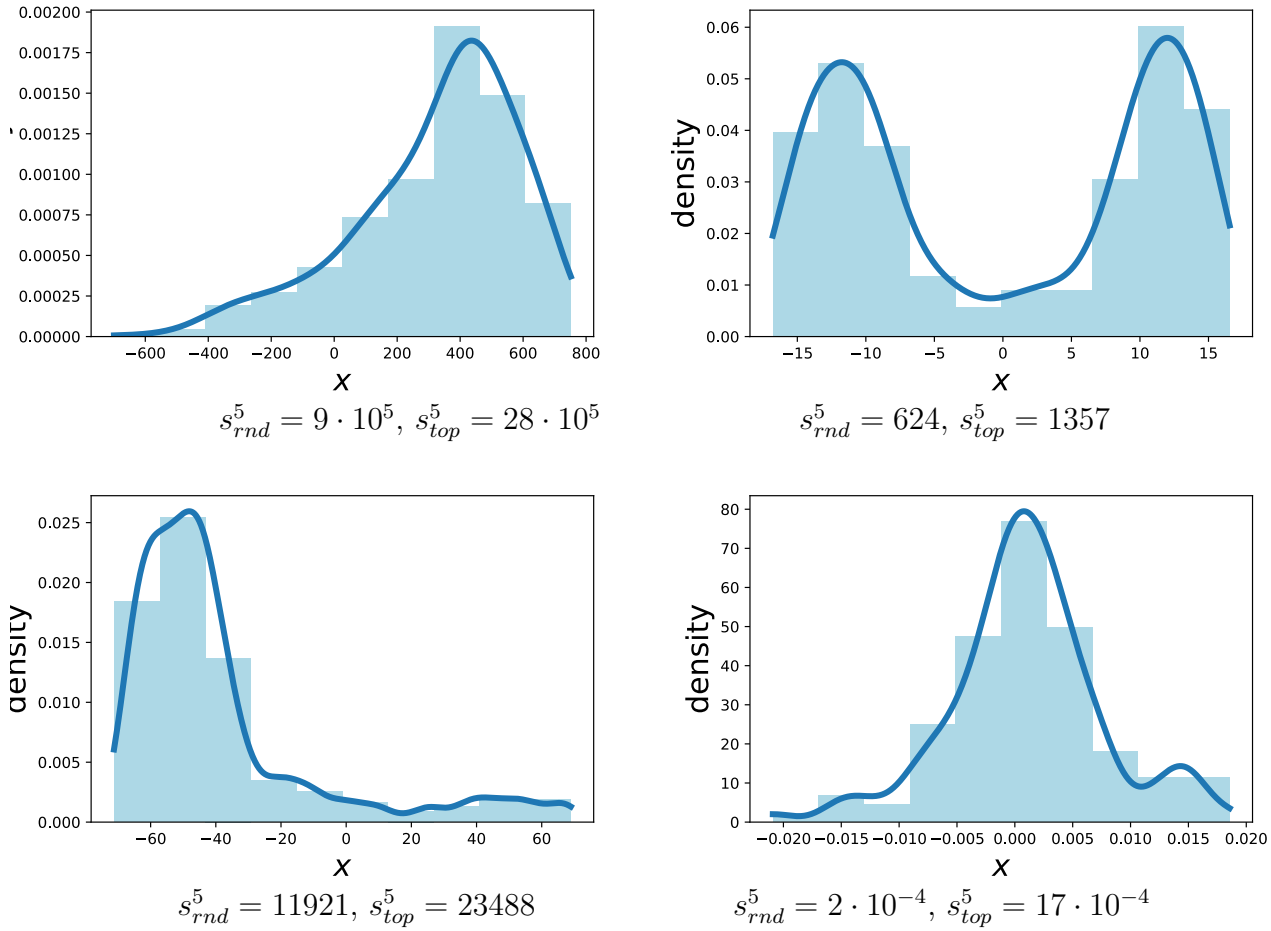


Figure 5.7: Calculations of the Rand-5 and Top-5 energy "saving" for practical gradient distributions ((a),(b),(c): quadratic problem, (d): logistic regression). The results of Top-5 are 3–5× better.

New compressor: Top-k combined with dithering

In Section 5.2.3, we present a new biased compression operator (see (5.16)), which combines the Top-k sparsification operator (see (5.11)) with general exponential dithering (see (5.14)). Consider the composition operator with natural dithering, i.e., with base $b = 2$. We demonstrate that it belongs to $\mathbb{B}^1(\frac{k}{d}, \frac{9}{8})$, $\mathbb{B}^2(\frac{k}{d}, \frac{9}{8})$, and $\mathbb{B}^3(\frac{9d}{8k})$. Figure 5.8 empirically confirms that it attains the lowest compression parameter $\delta \geq 1$ among all other known compressors (see (5.2.7)). Furthermore, the iteration complexity $\mathcal{O}\left(\delta \frac{L}{\mu} \log \frac{1}{\varepsilon}\right)$ of CGD for $C \in \mathbb{B}^3(\delta)$ implies that it enjoys fastest convergence.

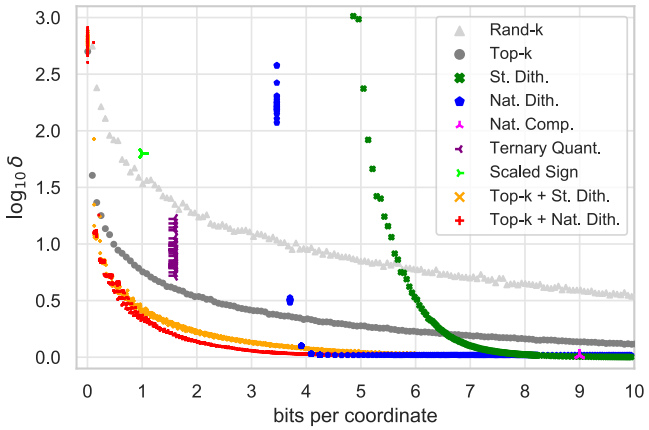


Figure 5.8: Comparison of various compressors (with and without free design parameters) with respect to the parameter $\delta \geq 1$ in $\log_{10}-$scale and the number of encoding bits used for each coordinate on average. Each point/marker represents a single $d = 10^4$ dimensional vector x drawn from Gaussian distribution and then compressed by the specified operator. Each curve was obtained by varying free parameters ($k \in \{1, 2, \ldots, d\}$ for sparsifiers, $s \geq 1$ for ditherings) of the specified compressor. Parameter-free compressors, such as ternary quantization and natural compression, have fixed communication budgets which explains the vertical arrangements of the points.

5.2.4 Basic Algorithms

As we discussed in the previous section, compression operators can have different equivalent parametrizations. Next, we aim to investigate the influence of those parametrizations on the theoretical convergence rate of an algorithm employing compression operators. To achieve a clearer understanding of the interaction between compressor parametrization and convergence rate, we first consider the single-node, unconstrained optimization problem

$$\min_{x \in \mathbb{R}^d} f(x),$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth and μ-strongly convex. We study the method

$$x^{k+1} = x^k - \gamma C^k(\nabla f(x^k)), \tag{5.17}$$

where $C^k : \mathbb{R}^d \to \mathbb{R}^d$ are (potentially biased) compression operators belonging to one of the classes $\mathbb{B}^1$, $\mathbb{B}^2$ and $\mathbb{B}^3$ studied in the previous sections, and $\gamma > 0$ is a stepsize. We refer to this method as CGD: Compressed Gradient Descent.

Complexity theory

We now establish three theorems, performing complexity analysis for each of the three classes $\mathbb{B}^1$, $\mathbb{B}^2$ and $\mathbb{B}^3$ individually. Let $\mathcal{E}_k := \mathbb{E}f(x^k) - f(x^*)$, with $\mathcal{E}_0 = f(x^0) - f(x^*)$.

Theorem 5.2.20. Let $C \in \mathbb{B}^1(\alpha, \beta)$. Then as long as $0 \leq \gamma \leq \frac{2}{\beta L}$, we have

$$\mathcal{E}_k \leq \left(1 - \frac{\alpha}{\beta}\gamma\mu(2 - \gamma\beta L)\right)\mathcal{E}_{k-1}.$$

If we choose $\gamma = \frac{1}{\beta L}$, then

$$\mathcal{E}_k \leq \left(1 - \frac{\alpha}{\beta^2}\frac{\mu}{L}\right)^k \mathcal{E}_0.$$

Theorem 5.2.21. Let $C \in \mathbb{B}^2(\gamma, \beta)$. Then as long as $0 \leq \gamma \leq \frac{2}{\beta L}$, we have

$$\mathcal{E}_k \leq (1 - \gamma\gamma\,(2 - \gamma\beta)L))\,\mathcal{E}_{k-1}.$$

If we choose $\gamma = \frac{1}{\beta L}$, then

$$\mathcal{E}_k \leq \left(1 - \frac{\gamma}{\beta}\frac{\mu}{L}\right)^k \mathcal{E}_0.$$

Theorem 5.2.22. Let $C \in \mathbb{B}^3(\delta)$. Then as long as $0 \leq \gamma \leq \frac{1}{L}$, we have

$$\mathcal{E}_k \leq \left(1 - \frac{1}{\delta}\gamma\mu\right)^k \mathcal{E}_{k-1}.$$

If we choose $\gamma = \frac{1}{L}$, then

$$\mathcal{E}_k \leq \left(1 - \frac{1}{\delta}\frac{\mu}{L}\right)^k \mathcal{E}_0.$$

The iteration complexity for these results can be found in Table 5.4. Note that the identity compressor $C(x) \equiv x$ belongs to $\mathbb{B}^1(1,1), \mathbb{B}^2(1,1)$ and $\mathbb{B}^3(1)$; hence, all these results exactly recover the rate of GD. In the first two theorems, scaling the compressor by a positive scalar $\lambda > 0$ does not influence the rate (see Theorem 5.2.8).

One can note that it is possible to adapt the results of Theorems 5.2.20-5.2.22 to the case with the time-varying parameters of compression operators [242, 408]. In the case of Theorem 5.2.22 and $\mathbb{B}^3$ compressors, it is sufficient to introduce a sequence $\{\delta^i\}$ responsible for the changes of the parameter δ and obtain the following convergence: $\mathcal{E}_k \leq \left[\prod_{i=0}^{k-1}\left(1 - \frac{1}{\delta^i}\frac{\mu}{L}\right)\right]\mathcal{E}_0$. In the case of Theorems 5.2.20 and 5.2.21, the results are similar, but it is necessary to add in the algorithm (5.17) the possibility of using time-varying steps $\gamma_k = \frac{1}{\beta^k L}$.

$\mathbb{B}^3$ and $\mathbb{B}^2$ are better than $\mathbb{B}^1$

In light of the results above, we make the following observation. If $C \in \mathbb{B}^1(\alpha, \beta)$, then by Theorem 5.2.8, $\frac{1}{\beta}C \in \mathbb{B}^3(\frac{\beta^2}{\alpha})$. Applying Theorem 5.2.22, we obtain the bound $\mathcal{O}\left(\frac{\beta^2}{\alpha}\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$. This is the same result as that obtained by Theorem 5.2.20. On the other hand, if $C \in \mathbb{B}^3(\delta)$, then by Theorem 5.2.8, $C \in \mathbb{B}^1(\frac{1}{4\delta^2}, 2)$. Applying Theorem 5.2.20, we obtain the bound $\mathcal{O}\left(16\delta^2\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$.

This is a worse result than what Theorem 5.2.22 offers by a factor of 16δ.

Similarly, if $C \in \mathbb{B}^1(\alpha, \beta)$, then by Theorem 5.2.8, $C \in \mathbb{B}^2(\frac{\alpha}{\beta}, \beta)$. Applying Theorem 5.2.21, we obtain the bound $\mathcal{O}\left(\frac{\beta^2}{\alpha}\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$. This result is the same as that obtained by Theorem 5.2.20. On the other hand, if $C \in \mathbb{B}^2(\gamma, \beta)$, then by Theorem 5.2.8, $C \in \mathbb{B}^1(\gamma^2, \beta)$. Applying Theorem 5.2.20, we derive the bound $\mathcal{O}\left(\frac{\beta^2}{\gamma^2}\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$. This is a worse result than what Theorem 5.2.21 offers by a factor of $\frac{\beta}{\gamma} \geq 1$.

Hence, while $\mathbb{B}^1, \mathbb{B}^2$ and $\mathbb{B}^3$ describe the same classes of compressors, for the purposes of CGD it is better to parameterize them as members of $\mathbb{B}^2$ or $\mathbb{B}^3$.

5.2.5 Distributed Algorithms

We now focus attention on a distributed setup with M machines, each of which owns *non-iid* data defining one loss function f_m. Our goal is to minimize the average loss:

$$\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{M}\sum_{m=1}^{M} f_m(x)\right\}. \tag{5.18}$$

Distributed CGD with unbiased compressors

Perhaps the most straightforward extension of CGD to the distributed setting is to consider the method

$$x^{k+1} = x^k - \gamma\frac{1}{M}\sum_{m=1}^{M} C_i^k(\nabla f_m(x^k)).$$

Indeed, for $M = 1$ this method reduces to CGD. For unbiased compressors belonging to $\mathbb{U}(\zeta)$, this method converges under suitable assumptions on the functions. For instance, if f_m are L-smooth and f is μ-strongly convex, then as long as the stepsize is chosen appropriately, the method converges to a $\mathcal{O}\left(\frac{\gamma D(\zeta-1)}{\mu M}\right)$ neighborhood of the (necessarily unique) solution x^* with the linear rate

$$\mathcal{O}\left(\left(\frac{L}{\mu} + \frac{L(\zeta-1)}{\mu M}\right)\log\frac{1}{\epsilon}\right),$$

where $D := \frac{1}{M}\sum_{m=1}^{M}\|\nabla f_m(x^*)\|^2$ [66]. In particular, in the overparameterized setting when $D = 0$, the method converges to the exact solution, and does so at the same rate as GD as long as $\zeta = \mathcal{O}(M)$. These results hold even if a regularizer is considered, and a proximal step is added to DCGD. Moreover, as shown by [60] and [265], a variance reduction technique can be devised to remove the neighborhood convergence and replace it by convergence to x^*, at the negligible additional cost of $\mathcal{O}((\zeta-1)\log\frac{1}{\epsilon})$.

Failure of DCGD with biased compressors

However, as we now demonstrate by providing counter-examples, DCGD may fail if the compression operators are allowed to be biased. In the first example below, DCGD used with the

Top-1 compressor diverges at an exponential rate.

Example 5.2.23. Consider $n = d = 3$ and define

$$f_1(x) = \langle a, x\rangle^2 + \frac{1}{4}\|x\|^2, \qquad f_2(x) = \langle b, x\rangle^2 + \frac{1}{4}\|x\|^2, \qquad f_3(x) = \langle c, x\rangle^2 + \frac{1}{4}\|x\|^2,$$

where $a = (-3, 2, 2)$, $b = (2, -3, 2)$ and $c = (2, 2, -3)$. Let the starting iterate be $x^0 = (t, t, t)$, where $t > 0$. Then

$$\nabla f_1(x^0) = \frac{t}{2}(-11, 9, 9), \qquad \nabla f_2(x^0) = \frac{t}{2}(9, -11, 9), \qquad \nabla f_3(x^0) = \frac{t}{2}(9, 9, -11).$$

Using the Top-1 compressor, we get $C(\nabla f_1(x^0)) = \frac{t}{2}(-11, 0, 0)$, $C(\nabla f_2(x^0)) = \frac{t}{2}(0, -11, 0)$ and $C(\nabla f_3(x^0)) = \frac{t}{2}(0, 0, -11)$. The next iterate of DCGD is

$$x^1 = x^0 - \gamma\frac{1}{3}\sum_{i=1}^{3} C(\nabla f_m(x^0)) = \left(1 + \frac{11\gamma}{6}\right)x^0.$$

Repeated application gives

$$x^k = \left(1 + \frac{11\gamma}{6}\right)^k x^0.$$

Since $\gamma > 0$, the entries of x^k diverge exponentially fast to $+\infty$.

The above counterexample can be extended to the case of Top-d_1 when $d_1 < \left\lceil \frac{d}{2} \right\rceil$.

Example 5.2.24. Fix the dimension $d \geq 3$ and let $n = \binom{d}{d_1}$ be the number of nodes, where $d_1 < \left\lceil \frac{d}{2} \right\rceil$ and $d_2 = d - d_1 > d_1$. Choose positive numbers $b,\ c > 0$ such that

$$-bd_1 + cd_2 = 1,\ b > c + 1.$$

One possible choice could be $b = d_2 + \frac{d_2}{d_1},\ c = d_1 + \frac{1}{d_2} + 1$. Define vectors $a_j \in \mathbb{R}^d,\ j \in [n]$ via

$$a_j = \sum_{i \in I_j}(-b)e_i + \sum_{i \in [d]\setminus I_j} ce_i,$$

where sets $I_j \subset [d],\ j \in [n]$ are all possible d_1-subsets of $[d]$ enumerated in some way. Define

$$f_j(x) = \langle a_j, x\rangle^2 + \frac{1}{2}\|x\|^2, \quad j \in [n]$$

and let the initial point be $x^0 = te,\ t > 0$, where $e = \sum_{i=1}^{d} e_i$ is the vector of all 1s. Then

$$\nabla f_j(x^0) = 2\langle a_j, x^0\rangle \cdot a_j + x^0 = 2t(-bd_1 + cd_2) \cdot a_j + te = t(2a_j + e).$$

Since $|2(-b)+1| > |2c+1|$, then using the Top-d_1 compressor, we obtain

$$C(\nabla f_j(x^0)) = -t(2b-1)\sum_{i\in I_j} e_i.$$

Therefore, the next iterate of DCGD is

$$\begin{aligned} x^1 &= x^0 - \gamma\frac{1}{M}\sum_{j=1}^{M} C(\nabla f_j(x^0)) = x^0 + \frac{\gamma t(2b-1)}{M}\sum_{j=1}^{M}\sum_{i\in I_j} e_i \\ &= x^0 + \frac{\gamma(2b-1)}{M}\binom{d}{d_1-1}x^0 = \left(1+\frac{\gamma(2b-1)d_1}{d_2+1}\right)x^0, \end{aligned}$$

which implies

$$x^k = \left(1+\frac{\gamma(2b-1)d_1}{d_2+1}\right)^k x^0.$$

Since $\gamma > 0$ and $b > 1$, the entries of x^k diverge exponentially fast to $+\infty$.

The above examples suggest that one needs to devise a different approach to solving the distributed problem (5.18) with biased compressors. We resolve this problem by employing a memory feedback mechanism.

Error Feedback

We show that the distributed version of Distributed SGD with Error-Feedback [215], displayed in Algorithm 50, resolves the issue. Moreover, this algorithm allows for the computation of stochastic gradients. Each step starts with all machines m in parallel computing a stochastic gradient g_m^k of the form

$$g_m^k = \nabla f_m(x^k) + \xi_m^k, \tag{5.19}$$

where $\nabla f_m(x^k)$ is the true gradient, and ξ_m^k is a stochastic error. Then, this is multiplied by a stepsize γ_k and added to the memory/error-feedback term e_m^k, and subsequently compressed. The compressed messages are communicated and aggregated. The difference between the message we intend to send and its compressed version is stored as e_m^{k+1} for further correction in the next communication round. The output $\overline{x}^K$ is an ergodic average of the form

$$\overline{x}^K := \frac{1}{W_K}\sum_{k=0}^{K} w_k x^k, \qquad W_K := \sum_{k=0}^{K} w_k.$$

Complexity theory

We assume the stochastic error ξ_m^k in (5.19) satisfies the following condition.

Assumption 5.2.25. Stochastic error ξ_m^k is unbiased, i.e. $\mathbb{E}\xi_m^k = 0$, and for some constants

Algorithm 50 Distributed SGD with Biased Compression and Error Feedback

Parameters: compressors $C_m^k \in \mathbb{B}^3(\delta)$; stepsizes $\{\gamma_k\}_{k\geq 0}$; iteration count K
Initialization: choose $x^0 \in \mathbb{R}^d$ and $e_m^0 = 0$ for all m

1: **for** $k = 0, 1, 2, \ldots, K$ **do**
2: Server sends x^k to all M machines
3: All machines in parallel perform these updates:

$$\tilde{g}_m^k = C_m^k(e_m^k + \gamma_k g_m^k) \tag{5.20}$$
$$e_m^{k+1} = e_m^k + \gamma_k g_m^k - \tilde{g}_m^k \tag{5.21}$$

4: Each machine m sends $\tilde{g}_m^k$ to the server
5: Server performs aggregation:

$$x^{k+1} = x^k - \frac{1}{M}\sum_{m=1}^{M} \tilde{g}_m^k$$

6: **end for**

$B, C \geq 0$

$$\mathbb{E}\left\|\xi_m^k\right\|^2 \leq B\left\|\nabla f_m(x^k)\right\|^2 + C, \qquad \forall m \in [M], k \geq 0. \tag{5.22}$$

Note that this assumption is much weaker than the bounded variance assumption (i.e., $\mathbb{E}\left\|\xi_m^k\right\|^2 \leq C$) and the bounded gradient assumption (i.e., $\mathbb{E}\left\|g_m^k\right\|^2 \leq C$). We can now state the main result of this section. To the best of our knowledge, *this was an open problem*: we are not aware of any convergence results for distributed optimization that tolerates general classes of *biased* compression operators and has reasonable assumptions on the stochastic gradient.

Theorem 5.2.26 (Convergence guarantees for Algorithm 50)**.** Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 50 for solving problem (5.6), where each f_m is L-smooth and μ-strongly convex. Let x^* be the minimizer of f and let $f^* := f(x^*)$ and

$$D := \frac{1}{M}\sum_{m=1}^{M} \left\|\nabla f_m(x^*)\right\|^2.$$

Assume the compression operator used by all nodes is in $\mathbb{B}^3(\delta)$. Then we have the following convergence rates under three different stepsize and iterate weighting regimes:

(i) $\mathcal{O}(\frac{1}{k})$ **stepsizes & $\mathcal{O}(k)$ weights.** Let, for all $k \geq 0$, the stepsizes and weights be set as $\gamma_k = \frac{4}{\mu(\kappa+k)}$ and $w_k = \kappa + k$, respectively, where $\kappa = \frac{56(2\delta+B)L}{\mu}$. Then

$$\mathbb{E}f(\bar{x}^K) - f^* = \mathcal{O}\left(\frac{A_1}{K^2} + \frac{A_2}{K}\right),$$

where $A_1 := \frac{L^2(2\delta+B)^2}{\mu}\left\|x^0 - x^*\right\|^2$ and $A_2 := \frac{C(1+1/M)+D(2B/M+3\delta)}{\mu}$.

(ii) $\mathcal{O}(1)$ **stepsizes & $\mathcal{O}(e^{-k})$ weights.** Let, for all $k \geq 0$, the stepsizes and weights be set as

$\gamma_k = \gamma$ and $w_k = (1-\mu\gamma/2)^{-(k+1)}$, respectively, where $\gamma \leq \frac{1}{14(2\delta+B)L}$. Then

$$\mathbb{E}f(\bar{x}^K) - f^* = \tilde{\mathcal{O}}\left(A_3 \exp\left[-\frac{K}{A_4}\right] + \frac{A_2}{K}\right),$$

where $A_3 := L(2\delta+B)\left\|x^0 - x^*\right\|^2$ and $A_4 := \frac{28L(2\delta+B)}{\mu}$.

(iii) $\mathcal{O}(1)$ **stepsizes & equal weights.** Let, for all $k \geq 0$, the stepsizes and weights be set as $\gamma_k = \gamma$ and $w_k = 1$, respectively, where $\gamma \leq \frac{1}{14(2\delta+B)L}$. Then

$$\mathbb{E}f(\bar{x}^K) - f^* = \mathcal{O}\left(\frac{A_3}{K} + \frac{A_5}{\sqrt{K}}\right),$$

where $A_5 := \sqrt{C\left(1+{}^1\!/_M\right) + D\left({}^{2B}\!/_M + 3\delta\right)}\left\|x^0 - x^*\right\|$.

Let us make a few observations on these results. First, Algorithm 50 employing general biased compressors and an error feedback mechanism indeed *resolves* convergence issues of the DCGD method by *converging to the optimal solution* x^*. Second, note that the choice of stepsizes γ_k and weights w_k leading to convergence is not unique, and *several schedules are feasible*. Third, all the rates are sublinear and based on the second rate *(ii)* above, *linear convergence* is guaranteed if $C = D = 0$. Based on (5.22), one setup when the condition $C = 0$ holds is when all devices compute full local gradients (i.e., $g_m^k = \nabla f_m(x^k)$). Furthermore, the condition $D = 0$ is equivalent to $\nabla f_m(x^*) = 0$ for all $m \in [M]$, which is typically satisfied for over-parameterized models. Lastly, under these two assumptions (i.e., devices can compute full local gradients and the model is over-parameterized), we show that distributed SGD method with error feedback converges with the same $\mathcal{O}\left(\delta\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$ linear rate as single node CGD algorithm. *To the best of our knowledge, this was the first regime where distributed first order method with biased compression is guaranteed to converge linearly.*

5.2.6 Experiments

In this subsection, we present our experiments, which primarily focus on supporting our theoretical findings. Therefore, we simulate these experiments on one machine which enable us to do rapid direct comparisons against the prior methods.

Lower empirical variance induced by biased compressors during deep network training

Motivated by our theoretical results in Section 5.2.3, we show that similar behavior appears in the empirical variance of gradients. We conduct two sets of experiments with ResNet18 on the CIFAR10 dataset. In Figure 5.9, we display empirical variance, which is obtained by running a training procedure with specific compression. We compare unbiased and biased compressions with the same communication complexities–deterministic with classic/unbiased C_{nat} and Top-k with Rand-k with k to be ${}^1\!/_5$ of coordinates. One can clearly see that there is a gap in empirical variance between biased and unbiased methods, similar to what we have shown in theory; see Section 5.2.3.

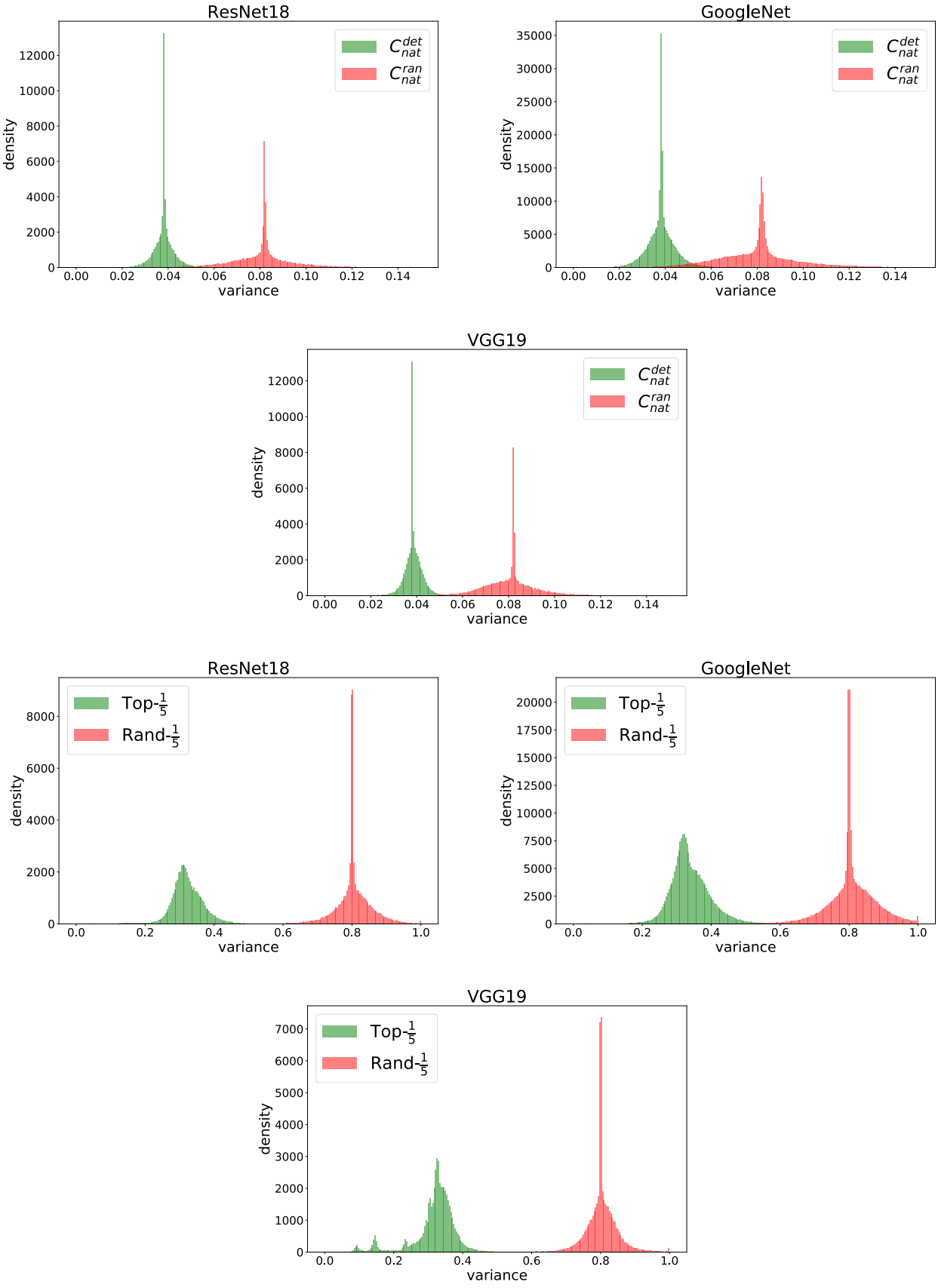


Figure 5.9: Comparison of empirical variance $\|C(x) - x\|^2 / \|x\|^2$ during training procedure for two pairs of method– deterministic with classic/unbiased C_{nat} and Top-k with Rand-k with Top-$1/5$ of coordinates. Both of the plots were produced using ResNet18, GoogleNet, and VGG19 on CIFAR10 dataset.

Error-feedback is needed in distributed training with biased compression

The next experiment demonstrates the necessity of error feedback for methods utilizing biased compression operators. Based on Example 1, error feedback is essential to avoid divergence from the optimal solution. Figure 5.10 illustrates the training/test loss and accuracy for VGG19 on CIFAR10 with data equally distributed among 4 nodes. We employ plain SGD with a default step size equal to 0.01 for all methods, including Top-5 with and without error feedback, Rand-5, and no compression. As suggested by the counterexample, the absence of error feedback can significantly impair performance when biased compression methods are used. Additionally, it is noteworthy that the performance of Rand-5 is considerably worse than that of Top-5.

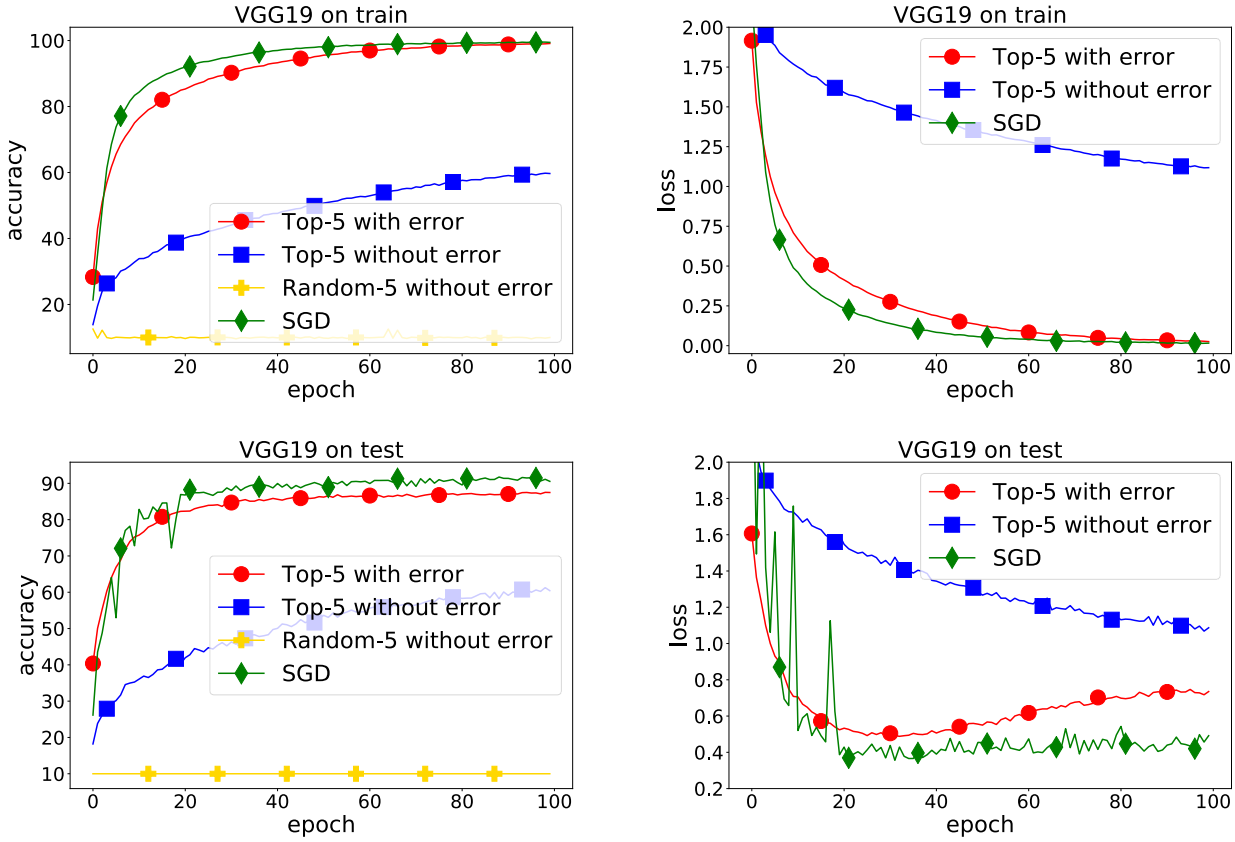


Figure 5.10: Training/Test loss and accuracy for VGG19 on CIFAR10 distributed among 4 nodes for 4 different compression operators.

Top-k mixed with natural dithering saves in communication significantly

Next, we experimentally demonstrate the superiority of our newly proposed compressor–Top-k combined with natural dithering. We compare this against the current state-of-the-art low bandwidth approach Top-k for some small k. In Figure 5.11, we plot a comparison of 5 methods–Top-k, Rand-k, natural dithering, Top-k combined with natural dithering, and plain SGD. We use 2 levels with infinity norm for natural dithering and $k = 5$ for sparsification methods. For all the compression operators, we train VGG11 on CIFAR10 with plain SGD as an optimizer and a default step size equal to 0.01. We observe that adding natural dithering after Top-k has the same effect as the natural dithering comparing to no compression, which is a significant reduction in

communications without almost no effect on convergence or generalization. Using this intuition, one can come to the conclusion that Top-k with natural dithering is the best compression operator for any bandwidth, where we adjust to the given bandwidth by adjusting k. This aligns precisely with our previous theoretical variance estimates displayed in Figure 5.8.

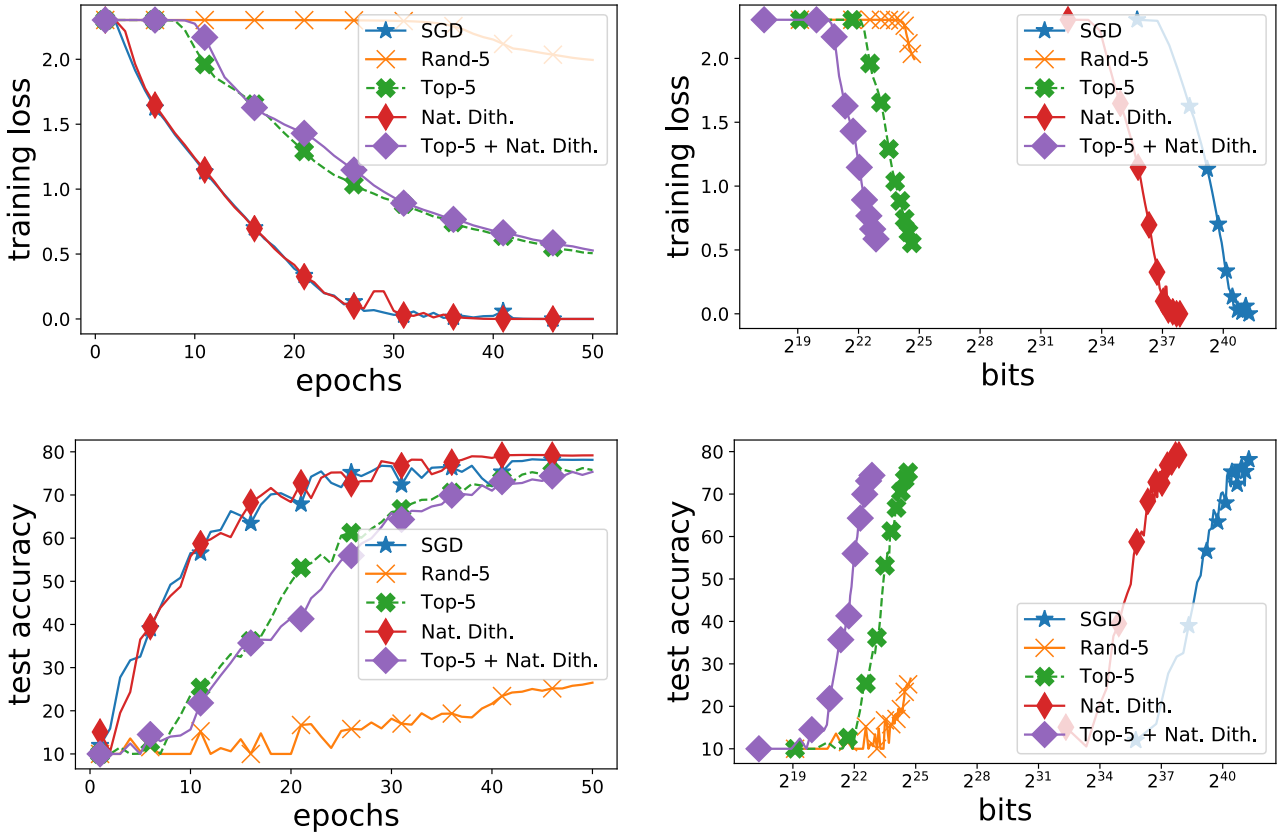


Figure 5.11: Training loss and test accuracy for VGG11 on CIFAR10 distributed among 4 nodes for 5 different compression operators.

Theoretical behavior predicts the actual performance in practice

In the next experiment, we provide numerical results to further demonstrate that our predicted theoretical behavior matches the actual performance observed in practice. We conduct two regression experiments optimized by gradient descent with step-size $\gamma = \frac{1}{L}$. We utilize a slightly adjusted version of Theorem 5.2.22 with adaptive step-sizes, specifically

$$\frac{f(x^k) - f(x^*)}{f(x^0) - f(x^*)} \leq \prod_{i=1}^{k} \left(1 - \frac{\mu}{L\delta_i}\right),$$

where

$$1 - \frac{1}{\delta_i} = \frac{\|C(\nabla f(x^i)) - \nabla f(x^i)\|^2}{\|\nabla f(x^i)\|^2}.$$

Note that this is the direct consequence of our analysis. We apply this property to illustrate the theoretical convergence. In the first experiment depicted in Figure 5.12, we randomly generate a square matrix A of dimension 100, constructed as follows: we sample a random diagonal matrix

D, whose elements are independently sampled from the uniform distribution $(1, 10)$, $(1, 100)$, and $(1, 1000)$, respectively. A is then constructed using $Q^\top DQ$, where $P = QR$ is a random matrix and QR is obtained via QR-decomposition. The label y is generated in the same manner from the uniform distribution $(0, 1)$. The optimization objective is then

$$\min_{x \in \mathbb{R}^d} x^\top Ax - y^\top x.$$

For the second experiment shown in Figure 5.13, we run standard linear regression on two scikit-learn datasets—*Boston* and *Diabetes*—and apply data normalization as the preprocessing step.

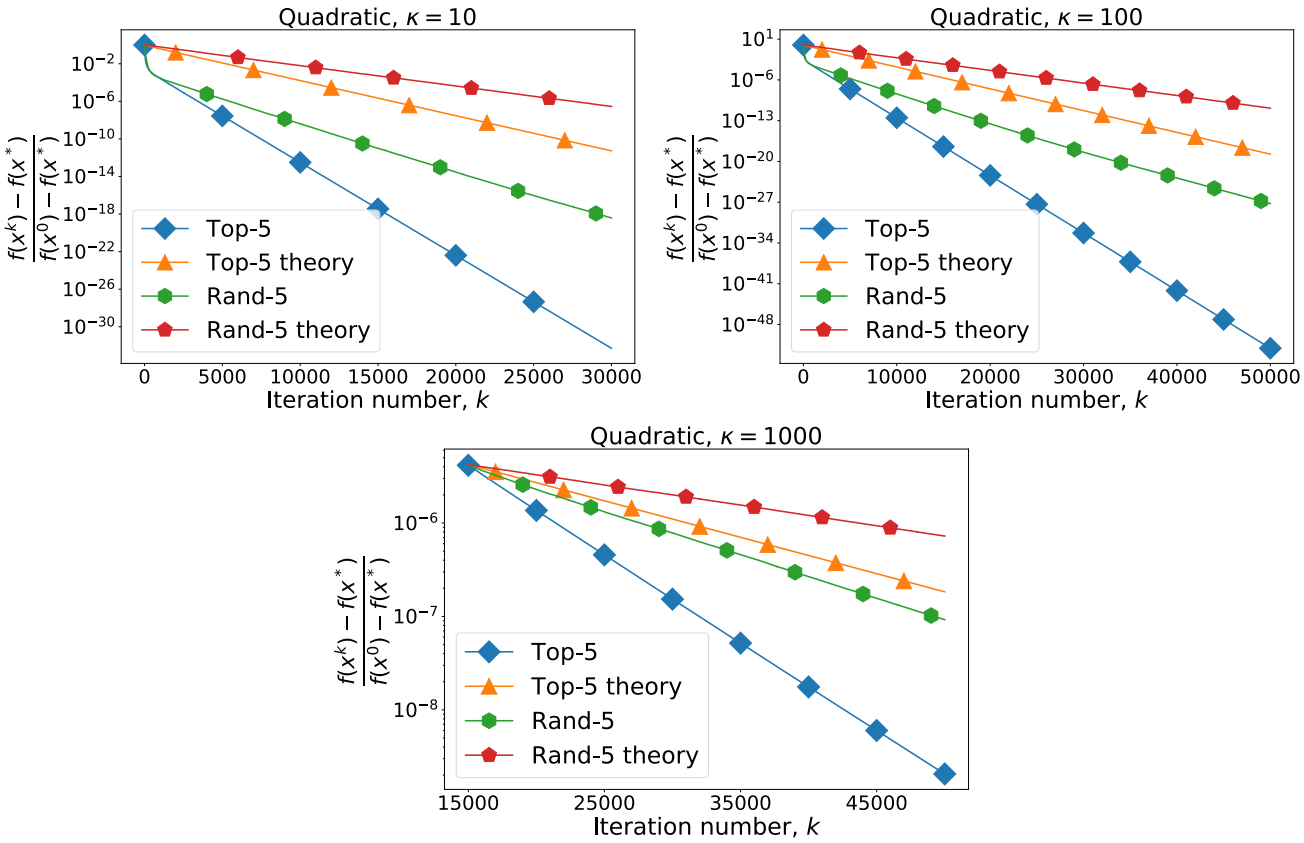


Figure 5.12: Theoretical vs. Practical Convergence of Compressed Gradient Descent on Quadratics problem with different condition number κ for Top-5 and Rand-5 compression operators.

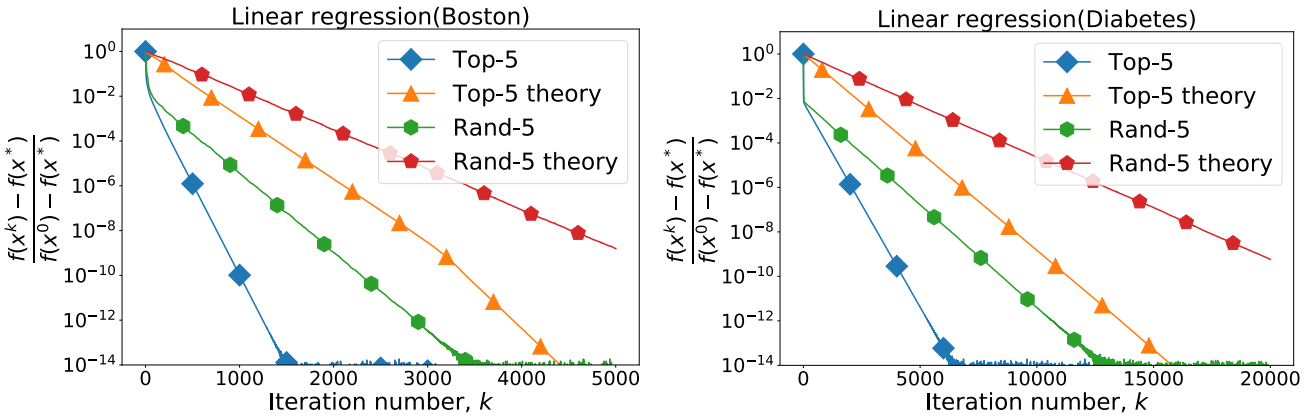


Figure 5.13: Theoretical vs. Practical Convergence of Compressed Gradient Descent on Linear Regression problem for *Boston* and *Diabetes* datasets with Top-5 and Rand-5 compression operators.

Looking into Figures 5.12 and 5.13, one can clearly see that, as predicted by our theory, biased compression with less empirical variance leads to better convergence in practice, and the gap almost matches the improvement.

Transformer training

In the last experiment, we work with a real big model. In particular, we train ALBERT-large [238] (18M parameters) with layer sharing on a combination of Bookcorpus [409] and Wikipedia [234] datasets. We use the same optimizer (LAMB) and the same tuning for it as in the original paper [238]. Our goal is to find an unbiased and biased operators such that we maximize the improvement in terms of communication cost without losing much in terms of training quality. In this case, we include in the communication cost both the time to perform the communication round and the time to perform the compression and decompression operations. It is important to note that we do not compress packages with gradients corresponding to LayerNorm scales, but this is less than 1 percent of the whole package. Among unbiased compressors, we try natural compression (Section 2.2 (g)) and random sparsification (Section 2.2 (a)). The best result is shown by natural compression, which compresses the packages by a factor of 4. The Rand-25 operator (which also compresses the information by a factor of 4) performs much worse even with the use of the error feedback technique. For communications with natural compression, we use the classical allreduce procedure. Among unbiased compressors, we try Top-k sparsification (Section 2.2 (d)) and Power compression [42]. The best result is shown by Power compression with the rank parameter r=8 and the error feedback. The organization of communications (allreduce procedures) occurs as in the original paper [42]. We measure how the training loss changes (Figure 5.14) as well as at the end of training we evaluate the final performance for each model on several popular tasks from [245] (Table 5.8). The results show that the use of biased compression can significantly reduce the communication time cost compared to uncompressed and even unbiased compression setups. At the same time, the quality of the training does not drop much.

Table 5.8: Comparison of average time per one communication round and downstream evaluation scores on GLUE benchmark tasks.

Setup	Avg time	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST2	STS-B
Without compression	9.62 ± 0.03	46.2	**81.1**	82.5	87.9	**88.0**	**66.3**	85.1	**88.0**
Natural compression	4.05 ± 0.05	**48.3**	81.0	**87.5**	87.8	84.4	63.2	87.8	86.9
Power compression	**1.04 ± 0.04**	42.4	80.3	85.1	**88.2**	85.3	46.3	**88.0**	87.4

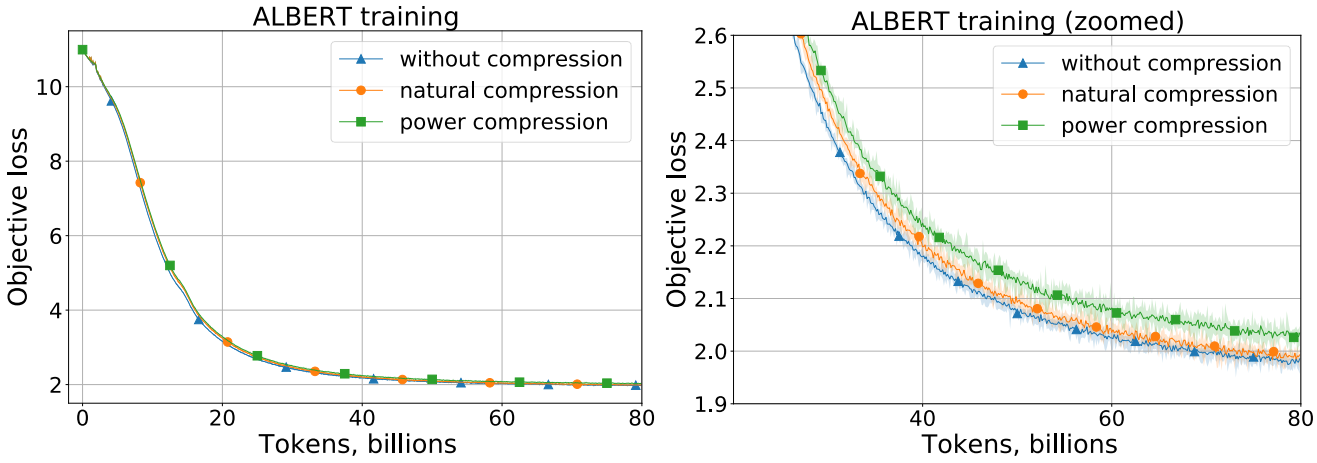


Figure 5.14: Training objective value for ALBERT-large with different compression operators.

5.3 Markovian Stochasticity

In this section, we study the minimization problem

$$\min_{x\in\mathbb{R}^d} f(x) := \mathbb{E}_{Z\sim\pi}[F(x,Z)]\,, \tag{5.23}$$

where the access to the function f and its gradient is available only through the (unbiased) noisy oracle $F(x,Z)$ and $\nabla F(x,Z)$, respectively. It is a standard stochastic setting that has been encountered a few times before in this thesis. And we already know that stochastic gradient methods are an essential ingredient for solving various optimization problems, with a wide range of applications in various fields such as machine learning [97, 410], empirical risk minimization problems [411], and reinforcement learning [412, 413, 414]. Various stochastic gradient descent methods (SGD) and their accelerated versions [34, 415] have been extensively studied under different statistical frameworks [416, 65]. The standard assumption for stochastic optimization algorithms is to consider independent and identically distributed noise variables. However, the growing usage of stochastic optimization methods in reinforcement learning [417, 418, 419] and distributed optimization [420, 421, 422] has led to increased interest in problems with Markovian noise. Despite this, existing theoretical works that consider Markov noise have significant limitations, and their analysis often results in suboptimal finite-time error bounds.

Our research aims to fill the gap in the existing literature on the first-order Markovian setting. By focusing on uniformly geometrically ergodic Markov chains, we obtain finite-time complexity bounds for achieving ε-accurate solutions that scale linearly with the mixing time of the underlying Markov chain. Our approach is based on careful applications of randomized batch size schemes and provides a unified view on both non-convex and strongly convex minimization problems, as well as variational inequalities.

5.3.1 Related Works and Contributions

• **Stochastic gradient methods.** Numerous research papers have reported significant improvements achieved by accelerated methods for stochastic optimization with stochastic gradient oracles involving independent and identically distributed (i.i.d.) noise. These methods have been extensively studied in theory [423, 424, 425, 48, 426, 427, 428, 65, 49, 429, 430, 431] and have demonstrated practical success [202, 432]. The finite-time analysis of first-order methods in i.i.d. noise settings has been extensively studied by many authors, as discussed in [433] and references therein. In Table 5.9 we include only some important results, as the i.i.d. setting is not the focus of this section.

While the literature on i.i.d. noise is extensive, existing research on the first-order Markovian setting is relatively sparse. In this study, we focus on Markov chains that are uniformly geometrically ergodic. We note that the complexity bounds which scale linearly with the mixing time of the underlying general Markov chain are currently available only for general convex and non-convex minimization problems. Namely, [434] has investigated a version of the ergodic mirror descent

algorithm that yields optimal convergence rates for Lipschitz, general convex, and non-convex problems. Recently, [435] proposed a random batch size algorithm that adapts to the mixing time of the underlying chain for non-convex optimization with a compact domain. In particular, [435, Theorem 4.3] yields optimal complexity rates in terms of the number of oracle calls required for non-convex problems, which is consistent with the results obtained in [434]. Unlike previous studies, this method is insensitive to the mixing time of the noise sequence.

For the general case of Markovian noise, the finite-time analysis of non-accelerated SGD-type algorithms was carried out in [436] and [437]. However, [436] heavily relies on the bounded domain assumption and uniformly bounded stochastic gradient oracles, while its bound in [436, Theorem 5] exhibits a suboptimal dependence on the mixing time of the underlying chain, see Table 5.9. Additionally, [436] does not cover the strongly convex setting. On the other hand, [437] covers both non-convex and strongly convex settings, but the bounds of [437, Theorem 1] has terms that are *exponential* in the mixing time, and a careful examination reveals suboptimal dependence on the initial condition for strongly convex problems when SGD is applied.

Table 5.9: This table summarizes our results on first-order method with Markovian noise. The columns of the table indicate whether the authors consider optimization over bounded domain, potentially unbounded gradients, and whether or not they assume additional restrictions on the Markovian noise (finite state space or reversibility). For ease of comparison we provide the respective results on SGD and ASGD (accelerated SGD) in the i.i.d. setting.

		Unbounded					
	Method	**Domain**	**Gradient noise**	**General MC**	**Acceleration**	**Oracle complexity (Smooth and non-convex)**	**Oracle complexity (Smooth and strongly convex)**
i.i.d.	SGD [46, 47, 398]	✓	✗	N/A	✗	$\tilde{\mathcal{O}}\left(L(f(x^0)-f(x^*))\left[\frac{1}{\varepsilon^2}+\frac{\sigma^2}{\varepsilon^4}\right]\right)$	$\tilde{\mathcal{O}}\left(\frac{L}{\mu}\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{\sigma^2}{\mu^2\varepsilon}\right)$
	ASGD [65, 438] [(1)]	✓	✓	N/A	✓	$\tilde{\mathcal{O}}\left(L(f(x^0)-f(x^*))\left[\frac{1+\delta^2}{\varepsilon^2}+\frac{\sigma^2}{\varepsilon^4}\right]\right)$	$\tilde{\mathcal{O}}\left((1+\delta^2)\sqrt{\frac{L}{\mu}}\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{\sigma^2}{\mu^2\varepsilon}\right)$
Markovian	EMD [434][(2)]	✗	✗	✓	✗	$\tilde{\mathcal{O}}\left(\frac{\tau G^2 D^2}{\varepsilon^4}\right)$	✗
	MC SGD [436][(3)]	✓	✗	✗	✗	$\tilde{\mathcal{O}}\left(h(G,L)\left(\frac{\tau}{\varepsilon^2}\right)^{1/(1-q)}\right)$	✗
	MC SGD [437][(4)]	✓	✓	✓	✗	$\tilde{\mathcal{O}}\left(\frac{\tau L^2(1+\|x^*\|^2+\|x^0-x^*\|^2)}{\varepsilon^4}\right)$	$\tilde{\mathcal{O}}\left(e^{\tau(L/\mu)^2}\left[h(\frac{L}{\mu})\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{\tau^2L^2(1+\|x^*\|^2)}{\mu^2\varepsilon}\right]\right)$
	ASGD[439][(5)]	✗	✗	✗	✓	$\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^4}\left[B^2+G^6(L^2\tau^2+1)\right]\right)$	$\tilde{\mathcal{O}}\left(\sqrt{\frac{L}{\mu}}\frac{\|x^0-x^*\|^2}{\varepsilon^{1/2}}+\frac{\tau^2(G^2+\mu GD+\mu LD^2)}{\mu^2\varepsilon}\right)$
	MAG [435][(6)]	✓	✗	✓	✗	$\tilde{\mathcal{O}}\left(\frac{\tau(G+L+B)^2G^2}{\varepsilon^4}\right)$	✗
	MC SGD [440] (Sec. 5.1) [(7)]	✓	✗	✗	✗	$\mathcal{O}\left(\frac{\tau(L(f(x^0)-f(x^*))+\sigma^2)}{\varepsilon^2}+\frac{\tau(L(f(x^0)-f(x^*))+\sigma^2)\sigma^2}{\varepsilon^4}\right)$	$\mathcal{O}\left(\frac{\tau L}{\mu}\log\frac{(f(x^0)-f(x^*))/\mu+\sigma^2/(\mu L)}{\varepsilon}+\frac{\tau\sigma^2}{\mu^2\varepsilon}\right)$
	MC SGD [440] (Sec. 5.2) [(8)]	✓	✗	✗	✗	✗	$O\left(\frac{L}{\mu}\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{L\tau\sigma_*^2}{\mu^3\varepsilon}\right)$
	RASGD (ours)	✓	✓	✓	✓	$\tilde{\mathcal{O}}\left(\tau L(f(x^0)-f(x^*))\left[\frac{1+\delta^2}{\varepsilon^2}+\frac{\sigma^2}{\varepsilon^4}\right]\right)$	$\tilde{\mathcal{O}}\left(\tau\left[(1+\delta^2)\sqrt{\frac{L}{\mu}}\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{\sigma^2}{\mu^2\varepsilon}\right]\right)$

Notation: μ and L are as in Assumption 5.3.1 and Assumption 5.3.2,$G=\sup_{x,z}\|\nabla F(x,z)\|$. Note that $G\geq L$ and $G^2\geq\sigma^2$ under Assumption 5.3.4. We also set $B=\sup_x|f(x)|$; x^0 - starting point, x^* - solution, $\mathcal{D}$ - optimisation domain; $D=\sup_{x\in\mathcal{D}}\|x-x^*\|$, σ and δ - stochasticity parameters (see Assumption 5.3.4); σ_* - stochasticity parameter in x^* ; τ - mixing time of the chain (see Assumption 5.3.3), ε - accuracy of the solution, measured as $\mathbb{E}[\|\nabla f(x)\|^2]\lesssim\varepsilon^2$ for non-convex problems and $\mathbb{E}[\|x-x^*\|^2]\lesssim\varepsilon$ for the strongly convex ones. Functions $h(L/\mu)$ and $h(G,L)$ stands for an implicit dependence of the respective parameters.

[(1)] gives results with stepsize as a parameter, we choose it the close way as in our Corollary 5.3.8. [(2)] covers more general noise setting, then just Markovian. [(3)] for general state-space Markov noise the analysis of [436] requires reversibility. Parameter $q\in(1/2;1)$ refers to the step size $\sim 1/k^q$. [(4)] The fluctuation terms in [437, Theorem 1,3] contain hidden dependence on the initial error and $\|x^*\|$ in the fluctuation terms, making the result comparison complicated. They also contain hidden factors, which are exponential in $C=\tau/\log 4$ in the notations of this section. Moreover, the analysis of [437] requires that $F(x,z)$ is Lipschitz w.r.t. x for any $z\in\mathsf{Z}$. [(5)] considers Markovian noise with finite state space and a specifically decreasing step size. Moreover, in the proof of [439, Theorem 3] (equations $(64)-(66)$) the authors lost the factor C^2, with $C=\tau/\log 4$. The result in the table accounts for this lost factor. [(6)] considers the adaptive tuning of batch size, which is oblivious to τ. [(7)] considers Markov noise with finite state space and additionally assumes that all stochastic realization $F(\cdot,Z)$ are L-smooth. [(8)] considers Markovian noise with finite state space, σ_* bounds noise only in x^*, but additionally assumes that all stochastic realization $F(\cdot,Z)$ are L-smooth and μ-strongly convex.

In the study of Nesterov-accelerated SGD with Markovian noise, the authors of [439] considered the use of a batch size of 1 and achieved a rate of forgetting the initial condition that matches that of the i.i.d. noise setting. However, their result is suboptimal regarding the variance terms in both non-convex and strongly convex settings, as detailed in Table 5.9. We emphasize that the case of unbounded gradient oracles with Markov noise is not treated, in contrast to the i.i.d. setup [65, 441].

The above papers deal with general Markovian noise optimization. Additionally, there are results pertaining to Markovian stochasticity within a finite state space. Here we can highlight the

work [440], where the author gives quite extensive results and achieves linear scaling by mixing time in the non-convex as well as strongly convex cases. Recently, numerous papers have emerged that focus on the special scenario of distributed optimization [442]. [443] investigates the generalization and stability of Markov SGD with special attention to the excess variance guarantees. We note that first, these algorithms only need to deal with a very special case of Markov gradients, and second, the corresponding dependence on the mixing time of the Markov chain is again quadratic. At the same time, there exist particular results, e.g. [444], which provide a lower bound for the particular finite sum problems in the Markovian setting.

• **Variational inequalities.** As we know from Chapters 2 and 3, variational inequalities and saddle point problems have well-established theory and methods. Unlike minimization problems, solving variational inequalities does not rely on (accelerated) gradient descent. Instead, the Extragradient method [78], various modified versions [80, 229], or similar techniques [86] are recommended as the basic and theoretically optimal methods. While deterministic methods have long been used for solving variational inequalities, stochastic methods have gained importance only in the last 15 years, following pioneering works by [445, 85]. We summarise the results on methods for stochastic variational inequalities with the Lipschitz operator and smooth stochastic saddle point problems in Table 5.10. The number of papers dealing with stochastic variational inequalities and saddle point problems is small compared to those dealing with stochastic optimization, we include in Table 5.10 papers with the i.i.d. noise (which we do not do for stochastic optimization). The only competing work dealing with Markovian noise in saddle point problems consider the finite sum problem and thus the finite Markov chain [443], therefore we do not include it in Table 5.10. Moreover, the results from [443] has much worse oracle complexity guarantees $\mathcal{O}(\tau^2/\varepsilon^2)$ in terms of τ. There are more papers dealing with stochastic finite-sum variational inequalities or saddle point problems, but in the i.i.d. setting [100, 227, 218, 214, 19]. We also do not consider in Table 5.10 because of the difference in the stochastic oracle structure. It is important to note that, unlike most previous works, we consider the most general formulation of variational inequality itself for an arbitrary optimization set and composite term.

Our contributions. Our main contributions are as follows:

• **Accelerated SGD.** We present the first analysis of SGD, including the Nesterov accelerated SGD method, with Markov noise, without the assumption of a bounded domain and uniformly bounded stochastic gradient estimates. Our results are summarized in Table 5.9 and cover both strongly convex and non-convex scenarios. Our findings for non-convex minimization problems complement the results obtained in [435].

• **Lower bounds.** In Section 5.3.3, we provide lower bounds demonstrating that the presence of mixing time in the upper complexity bounds is not an artifact of the proof. This is consistent with the results reported in [444].

• **Extensions.** We provide, as far as we know, the first analysis for variational inequalities with a general stochastic Markov oracle, an arbitrary optimization set, and an arbitrary composite

term. Our finite-time performance analysis provides complexity bounds in terms of oracle calls that scale linearly with the mixing time of the underlying chain, which is an improvement over the bounds obtained in [443] for the Markov setting.

Table 5.10: This table summarizes the findings on methods for solving stochastic (strongly) monotone variational inequalities with a Lipschitz operator and (un)bounded stochasticity. The columns of the table indicate whether the authors consider variational inequalities or only certain saddle point problems, the arbitrariness of the sets, and the use of additional composite terms. The columns on stochasticity provide information on the assumptions made with respect to the stochastic operator, such as bounded variance and the Markovian noise setting. Note that with the exception of our work, all other studies assume the independent noise.

		Statement			Stochasticity		
	Method	**VI?**	**Any set?**	**Composite?**	**Unbounded?**	**Markovian?**	**Oracle complexity**
Strongly monotone	SPEG [81, 229]	✓	✓	✗	✗	✗	$\tilde{\mathcal{O}}\left(\frac{L}{\mu}\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{\sigma^2}{\mu^2\varepsilon}\right)$ (1)
	SEG [113]	✓	✓	✗	✗	✗	$\tilde{\mathcal{O}}\left(\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{B^2+\sigma^2+(B+\sigma)(1+LD)}{\sigma^2\varepsilon}\right)$
	SS-SEG [196, 267]	✓	✓	✓	✓	✗	$\tilde{\mathcal{O}}\left(\frac{L}{\mu}\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{\sigma_*^2}{\mu^2\varepsilon}\right)$
	SEG (Ch. 2)	✗	✓	✗	✗	✗	$\tilde{\mathcal{O}}\left(\frac{L}{\mu}\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{\sigma^2}{\mu^2\varepsilon}\right)$
	DSEG [114]	✓	✗	✗	✓	✗	$\mathcal{O}\left(\left[\frac{L^2\sigma^2}{\mu^4\varepsilon}\right]^3\right)$ (2)
	UEG [267]	✓	✗	✗	✓	✗	$\mathcal{O}\left(\left(\frac{L+\Delta}{\mu}+\frac{\Delta^2}{\mu^2}\right)\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{\sigma^2}{\mu^2\varepsilon}\right)$
	SGDA (Ch. 3)	✓	✓	✓	✗	✗	$\mathcal{O}\left(\frac{L^2}{\mu^2}\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{\sigma^2}{\mu^2\varepsilon}\right)$ (3)
	REG (this section)	✓	✓	✓	✓	✓	$\tilde{\mathcal{O}}\left(\tau\cdot\left[\left(\frac{L+\Delta}{\mu}+\frac{\Delta^2}{\mu^2}\right)\log\frac{\|x^0-x^*\|^2}{\varepsilon}+\frac{\sigma^2}{\mu^2\varepsilon}\right]\right)$
Monotone	SMP [85]	✓	✓	✗	✗	✗	$\mathcal{O}\left(\frac{LD^2}{\varepsilon}+\frac{\sigma^2\Delta^2}{\varepsilon^2}\right)$
	VR-SEG [111]	✓	✓	✗	✓	✗	$\mathcal{O}\left(\frac{(\sigma+\Delta)^8D^4}{\varepsilon^2}+\frac{D^4}{\varepsilon^2}\right)$
	IPM [446]	✓	✓	✓	✓	✗	$\mathcal{O}\left(\tilde{\mathcal{O}}\left(\frac{L^4D^4}{\varepsilon^2}+\frac{\sigma_*^2D^4}{\varepsilon^2}\right)\right)$
	SS-SEG [196]	✓	✓	✓	✓	✗	$\tilde{\mathcal{O}}\left(\frac{L^2D^4}{\varepsilon}+\frac{\sigma_*^4}{L^2\varepsilon^2}\right)$
	SEG (Ch. 2)	✗	✓	✗	✗	✗	$\mathcal{O}\left(\frac{LD^2}{\varepsilon}+\frac{\sigma^2\Delta^2}{\varepsilon^2}\right)$
	REG (this section)	✓	✓	✓	✓	✓	$\tilde{\mathcal{O}}\left(\tau\cdot\left[\frac{LD^2}{\varepsilon}+\frac{\sigma^2D^2}{\varepsilon^2}+\frac{\Delta^2D^4}{\varepsilon^2}\right]\right)$

Notation: μ = constant of strong monotonicity of operator F, L = Lipschitz constant of F, B = uniform bound of F, D = uniform bound of iterations x^k, x^0 = starting point, x^* = solution, Δ and σ = stochasticity parameters (see Assumption 5.3.16, [85, 81, 229, 113] and Chapters 2 and 3 take $\Delta = 0$), σ_* = stochasticity parameter in x^* (see [196]), τ = mixing time of the chain (see Assumption 5.3.3), ε = accuracy of the solution.
(1) give results with stepsize as a parameter, we choose it according to Section 3 from [194]. (2) consider Assumption 5.3.16, but do not provide explicit rates if $\Delta \neq 0$ (see also [267, Table 1]). (3) consider the cocoercive case, for which in general $\ell = L^2/\mu$.

5.3.2 Assumptions

In the following presentation, we impose at least one of the following regularity constraints on the underlying function f itself:

Assumption 5.3.1. The function f is L-smooth on $\mathbb{R}^d$ with $L > 0$, i.e., it is differentiable, and there exists a constant $L > 0$ such that the following inequality holds for all $x, y \in \mathbb{R}^d$:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

Assumption 5.3.2. The function f is μ-strongly convex on $\mathbb{R}^d$, i.e., it is continuously differentiable and there is a constant $\mu > 0$ such that the following inequality holds for all $x, y \in \mathbb{R}^d$:

$$\frac{\mu}{2}\|x - y\|^2 \leq f(x) - f(y) - \langle \nabla f(y), x - y \rangle .$$

Next, we specify our assumptions on the sequence of noise variables $\{Z_i\}_{i=0}^{\infty}$. We consider the general setting of $\{Z_i\}_{i=0}^{\infty}$ being a time-homogeneous Markov chain. Such problems naturally arise in stochastic optimization. In empirical risk minimization problems, this context arises with non-random minibatch choices. Indeed, a random choice of a batch number may lose to a non-random one, see [386, 388]. A wide range of problems dealing with Markovian noise is spawned by the reinforcement learning methods. The usual MDP setting falls naturally within this paradigm; moreover, the analysis of non-tabular RL problems requires addressing general state-space Markov noise. Here, the potential range of applications includes policy evaluation methods, such as temporal difference methods [447], and policy optimization algorithms, such as the policy gradient family, e.g. the celebrated REINFORCE algorithm [448].

We denote by Q the Markov kernel corresponding to the sequence $\{Z_i\}_{i=0}^{\infty}$ and impose the following assumption on the mixing properties of Q:

Assumption 5.3.3. $\{Z_i\}_{i=0}^{\infty}$ is a stationary Markov chain on $(\mathsf{Z}, \mathcal{Z})$ with Markov kernel Q and unique invariant distribution π. Moreover, Q is uniformly geometrically ergodic with mixing time $\tau \in \mathbb{N}$, i.e., for every $k \in \mathbb{N}$,

$$\Delta(\mathrm{Q}^k) = \sup_{z, z' \in \mathsf{Z}} \frac{1}{2} \left\| \mathrm{Q}^k(z, \cdot) - \mathrm{Q}^k(z', \cdot) \right\|_{[\mathsf{TV}]} \leq \left(\frac{1}{4}\right)^{\lfloor k/\tau \rfloor} .$$

Assumption 5.3.3 is classical in the literature on optimization methods with Markovian noise and has been considered, in particular, in recent works [436, 435, 439]. In particular, this assumption covers finite state-space Markov chains with irreducible and aperiodic transition matrix considered in [440]. However, our definition of the mixing time τ aligns more closely with classical probability literature [449] and slightly differs from the one presented, for example, in [440, 422]. Next, we specify our assumptions on stochastic gradient:

Assumption 5.3.4. For all $x \in \mathbb{R}^d$ it holds that $\mathbb{E}_{\pi}[\nabla F(x, Z)] = \nabla f(x)$. Moreover, for all $z \in \mathsf{Z}$ and $x \in \mathbb{R}^d$ it holds that

$$\|\nabla F(x, z) - \nabla f(x)\|^2 \leq \sigma^2 + \delta^2 \|\nabla f(x)\|^2 . \tag{5.24}$$

Assumption 5.3.4 resembles the strong growth condition [65], which is classical for the over-parametrized learning setup [65, 450]. The main difference is that Assumption 5.3.4 concerns the almost sure bound in (5.24), which is unavoidable when dealing with uniformly geometrically ergodic Markovian noise (Assumption 5.3.3). Note that it is possible that the quantity δ^2 in (5.24) is not instance-independent and scales with the ratio L/μ from Assumption 5.3.1-Assumption 5.3.2

in the particular problems. With Assumption 5.3.3 and Assumption 5.3.4 we can prove the result on the mean squared error of the stochastic gradient estimate computed over batch size n under arbitrary initial distribution. This result is summarized below in Lemma 5.3.5:

Lemma 5.3.5. Let Assumption 5.3.3 and Assumption 5.3.4 hold. Then, for any $n \geq 1$ and $x \in \mathbb{R}^d$, it holds that

$$\mathbb{E}_{\pi}\left[\left\|n^{-1}\sum\nolimits_{i=1}^{n}\nabla F(x,Z_i)-\nabla f(x)\right\|^2\right] \leq \frac{8\tau}{n}\left(\sigma^2+\delta^2\|\nabla f(x)\|^2\right) .$$

Moreover, for any initial distribution ξ on $(\mathsf{Z},\mathcal{Z})$, that

$$\mathbb{E}_{\xi}\left[\left\|n^{-1}\sum\nolimits_{i=1}^{n}\nabla F(x,Z_i)-\nabla f(x)\right\|^2\right] \leq \frac{C_1\tau}{n}\left(\sigma^2+\delta^2\|\nabla f(x)\|^2\right) ,$$

where $C_1 = 16\left(1+\frac{1}{\ln^2 4}\right)$.

The proof of Lemma 5.3.5 simplifies the arguments in [435, Lemma 4] and allows us to obtain tighter values for the constants when dealing with the randomized batch size. Note that it is especially important to have the result for MSE under arbitrary initial distribution ξ, since in the proofs of our main results, we inevitably deal with the conditional expectations with respect to the previous iterate. We provide more details on the bias and variance of the Markov SGD gradients in the next section.

5.3.3 Algorithms and Lower Bounds

Accelerated method

We begin with a version of Nesterov accelerated SGD with randomized batch size, described in Algorithm 51. Due to the unboundedness of the stochastic gradient variance (see Assumption 5.3.4), using of the classical Nesterov accelerated method [397, Section 2.2.] does not give the desired result, it is necessary to introduce an additional momentum [67, 65]. We use our own version, but partially similar to [67, 65]. The main feature of Algorithm 51 is that the number of samples used during the k-th gradient computation scales as 2^{J_k}, where J_k comes from a truncated geometric distribution. The truncation parameter needs to be adopted (see Theorem 5.3.7) in order to control the computational complexity of the algorithm.

Randomized batch size allows for efficient *bias* reduction in the stochastic gradient estimates and can be seen as a particular case of the so-called multilevel MCMC [451, 452]. In the optimization context, this approach was successfully used by [435] for non-convex problems. Indeed, this bias naturally appears under the Markovian stochastic gradient oracles. It is easy to see that, with the counter T^k defined in Algorithm 51, we have

$$\mathbb{E}_k[\nabla F(x^k, Z_{T^k+i})] \neq \nabla f(x^k) .$$

Algorithm 51 `Randomized Accelerated GD`

Parameters: stepsize $\gamma > 0$, momentums θ, η, β, p, batchsize limit M
Initialization: choose $x^0 = x_f^0$

for $k = 0, 1, 2, \ldots$ **do**
  $x_g^k = \theta x_f^k + (1-\theta)x^k$
  Sample $J_k \sim \text{Geom}\,(1/2)$
  $g^k = g_0^k + \begin{cases} 2^{J_k}\left(g_{J_k}^k - g_{J_k-1}^k\right), & \text{if } 2^{J_k} \le M \\ 0, & \text{otherwise} \end{cases}$ with $g_j^k = 2^{-j}B^{-1}\sum\limits_{i=1}^{2^j B} \nabla f(x_g^k, Z_{T^k+i})$
  $x_f^{k+1} = x_g^k - p\gamma g^k$
  $x^{k+1} = \eta x_f^{k+1} + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k$
  $T^{k+1} = T^k + 2^{J_k}B$
end for

Below we show how the bias of the gradient estimate scales with the truncation parameter M. The statement of Lemma 5.3.6 indicates that the gradient estimates g_k introduced above possess a bias that decreases *quadratically* with M.

Lemma 5.3.6. Let Assumption 5.3.3 and Assumption 5.3.4 hold. Then for the gradient estimates g^k from Algorithm 51 it holds that $\mathbb{E}_k[g^k] = \mathbb{E}_k[g^k_{\lfloor \log_2 M \rfloor}]$. Moreover,

$$\mathbb{E}_k[\|\nabla f(x^k) - g^k\|^2] \lesssim \left(\tau B^{-1}\log_2 M + \tau^2 B^{-2}\right)(\sigma^2 + \delta^2\|\nabla f(x^k)\|^2)\,,$$
$$\|\nabla f(x^k) - \mathbb{E}_k[g^k]\|^2 \lesssim \tau^2 M^{-2} B^{-2}(\sigma^2 + \delta^2\|\nabla f(x^k)\|^2)\,.$$

Note that Lemma 5.3.6 is a natural counterpart of the deterministic bound (Lemma 5.3.5). Moreover, it gives the idea of the trade-off between the parameters B and M. Namely, the expected number of oracle calls to compute g_k is $\mathcal{O}(B\log_2(M))$ with the bias scaling as M^{-2}. Thus, the increase of M drastically reduces the bias with only a logarithmic payment in variance. At the same time, gradient variance scales as $(\tau/B)^2$, but the increase of B is much more expensive for the computational cost of the whole procedure. Taking into account the considerations above, we can prove the following result:

Theorem 5.3.7. Let Assumption 5.3.1 – Assumption 5.3.4 hold. Let the problem (5.23) be solved by Algorithm 51. Then for any $b \in \mathbb{N}$, $\gamma \in (0; \frac{3}{4L}]$, and $\beta, \theta, \eta, p, M, B$ satisfying

$$p \simeq (1 + (1+\gamma L)[\delta^2\tau b^{-1} + \delta^2\tau^2 b^{-2}])^{-1}, \quad \beta \simeq \sqrt{p^2\mu\gamma}, \quad \eta \simeq \sqrt{\frac{1}{\mu\gamma}},$$
$$\theta \simeq \frac{p\eta^{-1} - 1}{\beta p \eta^{-1} - 1}, \quad M \simeq \max\{2; \sqrt{p^{-1}(1 + p/\beta)}\}, \quad B = \lceil b\log_2 M\rceil\,,$$

it holds that

$$\mathbb{E}\left[\|x^K - x^*\|^2 + \frac{6}{\mu}(f(x_f^K) - f(x^*))\right] = \mathcal{O}\left(\exp\left(-K\sqrt{\frac{p^2\mu\gamma}{3}}\right)\left[\|x^0 - x^*\|^2 + \frac{6}{\mu}(f(x^0) - f(x^*))\right]\right.$$

$$+ \frac{p\sqrt{\gamma}}{\mu^{3/2}} \left(\sigma^2 \tau b^{-1} + \sigma^2 \tau^2 b^{-2}\right) \Bigg). \quad (5.25)$$

The result of Theorem 5.3.7 can be rewritten as an upper complexity bound under an appropriate choice of the remaining free parameter b:

Corollary 5.3.8. Under the conditions of Theorem 5.3.7, choosing $b = \tau$ and γ as

$$\gamma \simeq \min\left\{\frac{1}{L}; \frac{1}{p^2 \mu K^2} \ln\left(\max\left\{2; \frac{\mu^2 K[\|x^0 - x^*\|^2 + 6\mu^{-1}(f(x_f^0) - f(x^*))]}{\sigma^2}\right\}\right)\right\},$$

in order to achieve ε-approximate solution (in terms of $\mathbb{E}[\|x - x^*\|^2] \lesssim \varepsilon$) it takes

$$\tilde{\mathcal{O}}\left(\tau\left[(1+\delta^2)\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon} + \frac{\sigma^2}{\mu^2\varepsilon}\right]\right) \quad \text{oracle calls}. \quad (5.26)$$

The results of Corollary 5.3.8 are obtained with fixed parameters of the method. In Corollary 5.3.8 these parameters are selected somewhat artificially; for instance, the stepsize γ depends on the iteration horizon K.

Comparison. Running the procedure above requires knowledge of the mixing time τ. Estimating the mixing time from a single trajectory of the running Markov chain is known to be a computationally hard problem; see, e.g. [453] and references therein. At the same time, methods that share the same (optimal) linear scaling of the sample complexity with respect to the mixing time also share the same drawback as our method. In particular, this holds true for the EMD algorithm [434], SGD-DD algorithm [444], and usual SGD with Markovian data [440]. In the non-convex scenario, the paper [435] is truly oblivious to mixing time, allowing obtaining sample complexity rates for non-convex problems that are homogeneous with respect to τ with AdaGrad-type learning rate. An interesting direction for the future work is to suggest a procedure that would allow to generalize the results of [435] to the accelerated SGD setting.

It is possible that the sample complexity bound (5.26) is worse than the respective bounds for non-accelerated SGD with Markov data, provided that δ^2 grows quickly with L/μ. At the same time, this drawback is shared by the classical results on learning under the strong growth condition, see e.g. [65]. As shown in [441], the respective rates can be worse than the ones obtained by usual SGD even under the i.i.d. noise setting, see Appendix F.3 in [441]. Making the analysis of accelerated SGD 'backward compatible' regarding the rates of usual SGD requires performing analysis in terms of additional problem-specific quantities, see [454, 441].

The closest equivalent of the result from Corollary 5.3.8 is given by [439, Theorem 3]. However, the corresponding bound of [439, Theorem 3] is incomplete, since the factor τ^2 is lost in the proof (see equations $(64 - 66)$). With this completion, the bound of [439, Theorem 3] yields a variance term of order $\tilde{\mathcal{O}}\left(\frac{\sigma^2\tau^2}{\mu^2\varepsilon}\right)$, which is suboptimal with respect to τ. Moreover, the corresponding analysis relies heavily on the assumption of a bounded domain. In [440], the author considers Markovian noise with a finite number of states and manages to obtain a rather interesting result of

the form $O\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}+\frac{L\tau\sigma_*^2}{\mu^3\varepsilon}\right)$. Here the first term does not depend on τ, and the second consists only σ^* (stochasticity in x^*), but the price for this is an additional factor L/μ in the second term and a stricter assumption that all realizations $F(\cdot,z)$ are smooth and strongly convex. In the context of overparameterized learning, our results are almost consistent with the bound of [65, Theorem 1] under i.i.d. sampling. The difference is that the term δ^2 in Assumption 5.3.4 can be more pessimistic than the expectation bound in [65].

Lower bounds

We start with a lower bound for the complexity of Markovian stochastic optimization under Assumption 5.3.1 –Assumption 5.3.4. Below we provide a result that highlights that the bound of Theorem 5.3.7 is tight provided that δ does not scale with the instance-dependent quantities, e.g., condition number L/μ.

Theorem 5.3.9. There exists an instance of the optimization problem satisfying Assumption 5.3.1 –Assumption 5.3.4 with $\delta = 1$ and arbitrary $\sigma \geq 0, L, \mu > 0, \tau \in \mathbb{N}$, such that for any first-order gradient method, it takes at least

$$\Omega\left(\tau\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}+\frac{\tau\sigma^2}{\mu^2\varepsilon}\right)$$

oracle calls in order to achieve $\mathbb{E}[\|x-x^*\|^2]\leq\varepsilon$.

The idea of the constructed lower bound for the deterministic part $\Omega\big(\tau\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\big)$ goes back to [397, Theorem 2.1.13]. The stochastic part lower bound is rooted in classical statistical reasoning and is well explained for i.i.d. noise in [433, Chapter 4.1]. Our adaptation for the Markovian setting is based on Le Cam's theory, as detailed in [455, Theorem 8] and [456]. For the case of Markov noise, this lower bound is, to the best of our knowledge, original. The closest result to ours is the stochastic term lower bound in [440, Proposition 1], but it is valid only for the vanilla stochastic gradient methods. Below, we provide another lower bound showing that the dependence of the sample complexity from Corollary 5.3.8 on δ is not an artefact of the proof.

Proposition 5.3.10. There exists an instance of the optimization problem satisfying Assumption 5.3.1 –Assumption 5.3.4 with arbitrary $L, \mu > 0, \tau \in \mathbb{N}$, $\delta = \frac{L}{\mu}$, and $\sigma = 0$, such that for any first-order gradient method, it takes at least

$$\Omega\left(\tau\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$$

gradient calls in order to achieve $\mathbb{E}[\|x-x^*\|^2]\leq\varepsilon$.

This lower bound is adapted from the information-theoretic lower bound [444]. Recent studies [457, 444, 438] have revealed the impossibility of accelerating stochastic gradient descent (SGD) for online linear regression problems with specific noise structures. To address this issue, researchers

have proposed various solutions, such as the MaSS algorithm [441] and the approach presented in [458]. However, these methods rely heavily on the particular structure of the online regression setup. Another question that naturally arises is whether one can get rid of the dependence on τ in the deterministic part of (5.26) if $\delta = 0$. The following counterexample demonstrates that this is not the case in general.

Proposition 5.3.11. There exists an instance of the optimisation problem satisfying Assumption 5.3.1 –Assumption 5.3.4 with arbitrary $L, \mu > 0, \tau \in \mathbb{N}$, $\sigma = 1, \delta = 0$, such that for any first-order gradient method, it takes at least

$$\Omega\left(\left(\tau + \sqrt{\frac{L}{\mu}}\right)\log\frac{1}{\varepsilon}\right)$$

oracle calls in order to achieve $\mathbb{E}[\|x - x^*\|^2] \leq \varepsilon$.

Non-convex problems

Now we proceed with a randomized batch size version of the simple SGD algorithm. It is summarized in Algorithm 52 and can be shown to achieve optimal rates of convergence for smooth non-convex problems. For the case of non-convex problems with Markov noise, similar analyses appeared in [435, Theorem 4].

Algorithm 52 `Randomized GD`

Parameters: stepsize $\gamma > 0$, number of iterations K, bound on batchsize B, mixing time τ
Initialization: choose $x^0 \in \mathcal{X}$

1: **for** $k = 0, 1, 2, \ldots, K-1$ **do**
2: Sample $J_k \sim \text{Geom}\left(\frac{1}{2}\right)$
3: $g^k = g_0^k + \begin{cases} 2^{J_k}\left(g^k_{J_k} - g^k_{J_k - 1}\right), & \text{if } 2^{J_k} \leq M \\ 0, & \text{otherwise} \end{cases}$ with $g_j^k = 2^{-j}B^{-1}\sum\limits_{i=1}^{2^j B}\nabla f(x^k, Z_{T^k+i})$
4: $x^{k+1} = x^k - \gamma g^k$
5: $T^{k+1} = T^k + 2^{J_k}B$
6: **end for**

By balancing the values of B and M with Lemma 5.3.6, we establish the following result:

Theorem 5.3.12. Let Assumption 5.3.1, Assumption 5.3.3, Assumption 5.3.4 hold. Let the problem (5.23) be solved by Algorithm 52. Let f^* be a global (possibly non-unique) minimum of f. Then for any $b \in \mathbb{N}$, and γ, M satisfying

$$\gamma \lesssim (L[1 + \delta^2\tau b^{-1} + \delta^2\tau^2 b^{-2}])^{-1}, \quad M \simeq \max\{2; \sqrt{\gamma^{-1}L^{-1}}\}, \quad B = \lceil b\log_2 M\rceil,$$

it holds that

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla f(x^k)\|^2\right] \lesssim \frac{f(x^0)-f^*}{\gamma K} + L\gamma\cdot\left[\sigma^2\tau b^{-1} + \sigma^2\tau^2 b^{-2}\right].$$

The following corollary immediately follows from the theorem.

Corollary 5.3.13. Under the conditions of Theorem 5.3.12, if we choose $b = \tau$ and γ given by

$$\gamma \simeq \min\left\{\frac{1}{L(1+\delta^2)}; \sqrt{\frac{f(x^0)-f^*}{LK\sigma^2}}\right\},$$

then to achieve the ε-solution (in terms of $\mathbb{E}[\|\nabla f(x)\|^2] \lesssim \varepsilon^2$) we need

$$\tilde{\mathcal{O}}\left(\tau\cdot\left[\frac{(1+\delta^2)L(f(x^0)-f^*)}{\varepsilon^2} + \frac{L(f(x^0)-f^*)\sigma^2}{\varepsilon^4}\right]\right) \quad \text{oracle calls.}$$

Comparison. The respective bound for the non-convex setting provided in [439, Theorem 1] yields the sample complexity of order $\tilde{\mathcal{O}}\left(\frac{\tau^2 L(f(x^0)-f(x^*))\sigma^2}{\varepsilon^4}\right)$. Additionally, we can note the results of [440, Theorem 2] with the following estimate $O\left(\frac{\tau(L(f(x^0)-f(x^*))+\sigma^2)}{\varepsilon^2} + \frac{\tau(L(f(x^0)-f(x^*))+\sigma^2)\sigma^2}{\varepsilon^4}\right)$.

Variational inequalities

In this subsection, we address a class of problems well known from the previous chapters. For the sake of completeness, we will provide a formal introduction. We are interested in the following problem:

$$\text{Find } z^* \in \mathcal{Z} \text{ such that } \langle F(z^*), z - z^*\rangle + g(z) - g(z^*) \geq 0 \text{ for all } z \in \mathcal{Z}. \tag{5.27}$$

Assumption 5.3.14. The operator F is L-Lipschitz continuous on $\mathcal{Z}$ with $L > 0$, i.e., the following inequality holds for all $z, y \in \mathcal{Z}$:

$$\|F(z) - F(y)\| \leq L\|z - y\|, \qquad \forall z, y \in \mathcal{Z}.$$

Assumption 5.3.15. The operator F is μ_F-strongly monotone on $\mathcal{Z}$, i.e., the following inequality holds for all $z, y \in \mathcal{Z}$:

$$\langle F(z) - F(y), z - y\rangle \geq \mu_F\|z - y\|^2.$$

The function g is μ_g-strongly convex on $\mathcal{Z}$, i.e. for all $z, y \in \mathcal{Z}$ and any $g'(z) \in \partial g(z)$ we have

$$g(y) \geq g(z) + \langle g'(z), y - z\rangle + (\mu_g/2)\|z - y\|^2.$$

We consider two cases: strongly monotone/convex with $\mu_F + \mu_g > 0$ and monotone/convex with $\mu_F + \mu_g = 0$.

Assumption 5.3.16. For all $z \in \mathbb{R}^d$ it holds that $\mathbb{E}_\pi[F(z, Z)] = F(z)$. Moreover, for all $z \in \mathsf{Z}$ and $x \in \mathcal{Z}$ it holds that

$$\|F(z, Z) - F(z)\|^2 \leq \sigma^2 + \Delta^2 \|z - z^*\|^2,$$

where z^* is a point from the solution set.

Assumption 5.3.16 is found in the literature on variational inequalities [114, 111, 267] and is considered analog to Assumption 5.3.4 on overparametrized learning.

Just as the Nesterov accelerated method is optimal for smooth convex minimization problems, the Extragradient method [78, 80, 85] is optimal for monotone variational inequalities. Therefore, we take it as a base. In the extrapolation step of Algorithm 53, we simply collect a batch of size B. However, in the main step, we utilize randomization as in Algorithm 51. The next theorem gives the convergence of our method.

Theorem 5.3.17. Let Assumption 5.3.14, Assumption 5.3.15 with $\mu_F + \mu_r > 0$, Assumption 5.3.3, Assumption 5.3.16 hold. Let the problem (5.27) be solved by Algorithm 53. Then for any $b \in \mathbb{N}$, and γ, M satisfying

$$\gamma \lesssim \min\left\{(\mu_F + \mu_r)^{-1}; L^{-1}; (\mu_F + \mu_r)(\Delta^2 \tau b^{-1} + \Delta^2 \tau^2 b^{-2})^{-1}; \sqrt{\Delta^{-2}\tau^{-1}b}\right\},$$
$$M \simeq \max\{2; \sqrt{\gamma^{-1}(\mu_F + \mu_r)^{-1}}\}, \quad B = \lceil b \log_2 M \rceil,$$

it holds that

$$\mathbb{E}[\|z^K - z^*\|^2] \lesssim \exp\left(-\frac{K(\mu_F + \mu_r)\gamma}{2}\right)\|z^0 - z^*\|^2 + \frac{\gamma}{\mu}(\sigma^2 \tau b^{-1} + \sigma^2 \tau^2 b^{-2}).$$

One can obtain an estimate of the oracle complexity.

Algorithm 53 `Randomized ExtraGradient`

Parameters: stepsize $\gamma > 0$
Initialization: choose $z^0 \in \mathcal{Z}$

for $k = 0, 1, 2, \ldots, K-1$ **do**
  $z^{k+1/2} = \text{prox}_{\gamma r}\left(z^k - \gamma B^{-1} \sum_{i=1}^B F(z^k, Z_{T^k+i})\right)$
  $T^{k+1/2} = T^k + B$
  Sample $J_k \sim \text{Geom}\left(\frac{1}{2}\right)$
  $g^k = g_0^k + \begin{cases} 2^{J_k}\left(g^k_{J_k} - g^k_{J_k - 1}\right), & \text{if } 2^{J_k} \leq M \\ 0, & \text{otherwise} \end{cases}$ with $g_j^k = 2^{-j} B^{-1} \sum_{i=1}^{2^j \cdot B} F(z^{k+1/2}, Z_{T^{k+1/2}+i})$
  $z^{k+1} = \text{prox}_{\gamma r}\left(z^k - \gamma g^k\right)$
  $T^{k+1} = T^{k+1/2} + 2^{J_k} B$
end for

Corollary 5.3.18. Under the conditions of Theorem 5.3.17, if we choose $b = \tau$ and γ as follows

$$\gamma \simeq \min\left\{\frac{1}{\mu_F+\mu_r}; \frac{1}{L}; \frac{\mu_F+\mu_r}{\Delta^2}; \frac{1}{\Delta}; \frac{1}{K(\mu_F+\mu_r)}\ln\left(\max\left\{2; \frac{\mu K\|x^0-x^*\|^2}{\sigma^2}\right\}\right)\right\},$$

then to achieve the ε-solution (in terms of $\mathbb{E}[\|x-x^*\|^2] \lesssim \varepsilon$) we need

$$\tilde{\mathcal{O}}\left(\tau\cdot\left[\left(1+\frac{L}{\mu_F+\mu_r}+\frac{\Delta}{\mu_F+\mu_r}+\frac{\Delta^2}{(\mu_F+\mu_r)^2}\right)\log\frac{1}{\varepsilon}+\frac{\sigma^2}{\mu^2\varepsilon}\right]\right) \quad \text{oracle calls.}$$

Note that one provides an (almost) matching lower complexity bounds for variational inequalities via lower bounds for saddle point problems, which are a special case of variational inequalities. The method for obtaining lower bounds for saddle point problems is reduced to obtaining estimates for the strongly convex minimization problem (see [159, 170] for respective deterministic lower bounds), which we provide in Section 5.3.3. Similarly, the question of constructing a lower bound which is tight w.r.t. Δ remains open.

For the monotone case, we use the gap function as a convergence criterion:

$$\text{Gap}(z) = \sup_{y\in\mathcal{Z}}\left[\langle F(y), z-y\rangle + g(z) - g(y)\right].$$

An important assumption for the gap function is the boundedness of the set $\mathcal{Z}$.

Assumption 5.3.19. The set $\mathcal{Z}$ is bounded and has a diameter D, i.e., for all $z, y \in \mathcal{Z}$: $\|x-y\|^2 \le D^2$.

As we know, Assumption 5.3.19 can be slightly relaxed. We need to use a simple trick from [120]. In particular, we need to consider $\mathcal{C}$ – a compact subset of $\mathcal{X}$ and change $\mathcal{X}$ to $\mathcal{C}$. However, such a technique is rather technical and does not change the essence. Finally, the following result holds.

Theorem 5.3.20. Let Assumption 5.3.14, Assumption 5.3.15 with $\mu_F + \mu_r = 0$, Assumption 5.3.19, Assumption 5.3.3, Assumption 5.3.16 hold. Let the problem (5.27) be solved by Algorithm 53. Then for any $B \in \mathbb{N}$, and γ, M satisfying $\gamma \lesssim L^{-1}$, $M = \sqrt{K}$, it holds that

$$\mathbb{E}[\text{Gap}(\bar{x}^K)] \lesssim \frac{D^2}{\gamma K} + \gamma(\tau B^{-1}\log_2 K + \tau^2 B^{-2})(\sigma^2 + \Delta^2 D^2) \quad \text{where} \quad \bar{x}^K = \frac{1}{K}\sum_{k=0}^{K-1} x^{k+1/2}.$$

The following corollary holds.

Corollary 5.3.21. Under the conditions of Theorem 5.3.20, if we choose $B = \tau$ and γ as follows

$$\gamma \simeq \min\left\{\frac{1}{L}; \sqrt{\frac{D^2}{(\sigma^2+\Delta^2D^2)K}}\right\},$$

then to achieve ε-solution (in terms of $\mathbb{E}[\text{Gap}(x)] \lesssim \varepsilon$) we need

$$\tilde{\mathcal{O}}\left(\tau\left[\frac{LD^2}{\varepsilon} + \frac{\sigma^2 D^2 + \Delta^2 D^4}{\varepsilon^2}\right]\right) \quad \text{oracle calls.}$$

Comparison. These results represent the first findings for variational inequalities with Markovian stochasticity, applicable to both strongly monotone and monotone cases. The only closely related work is [443]. The authors address convex-concave saddle point problems and provide the following estimate on the oracle complexity $\mathcal{O}\left(\tau^2 \cdot \frac{G^4}{\varepsilon^2} + \frac{D^2}{\varepsilon^2}\right)$ (with G – the uniform bound of the operator), which is inferior to ours at least in terms of τ. Moreover, the authors examine the case of a finite Markov chain, which is a special case of our setup.

Chapter 6

Sliding for Distributed and Federated Optimization

In this chapter, we consider the known from the previous chapters distributed optimization problem of the form:

$$\min_{x\in\mathbb{R}^d} \frac{1}{M}\sum_{m=1}^{M} f_m(x), \tag{6.1}$$

where each agent/device $m \in [M]$ has an associated local loss function $f_m : \mathbb{R}^d \to \mathbb{R}$, as well as its own locally stored data.

Federated Learning (FL) [105, 253] is a subset of distributed machine learning, where it is assumed that computing agents are general user devices (for example, smartphones, tablets, laptops, and personal computers), and where different devices may have varying memory capacity and computing power. This leads to many new and important problems and questions that did not arise previously in the classical distributed setting [459]. For example, data may be spread unequally between devices, privacy considerations may prohibit the sharing of data between certain devices on the network, poor or unreliable connectivity may inhibit the flow of data, and data on certain devices may be of poorer quality compared to others.

Given these issues, this work focuses on *personalization* for federated learning. Notice that in (6.1), the model parameter x is found using global data (from all agents). However, the inclusion of a particular individual agent might negatively impact the global training process if their local data differs markedly from the global data, or if they possess low-quality local data; in this case, training the global model may lead to a poor solution. On the other hand, each user may have very little local data, and the process of training a model solely on local data may also lead to a poor-quality solution. The question of how to balance the two extremes, *global versus local*, gave rise to *Personalized Federated Learning (PFL)* [460, 461, 462].

For PFL, each agent m has its own parameter x_m, but the discrepancy between the parameters held on different devices is penalized. Correspondingly, PFL can be formulated as the following regularized optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^{Md}} \frac{1}{M}\sum_{m=1}^{M} f_m(x_m) + \frac{\lambda}{2} g(\mathbf{x}), \tag{6.2}$$

where the vector $\mathbf{x} = [x_1^T, \ldots, x_M^T]^T \in \mathbb{R}^{Md}$ is the concatenation of the M local vectors $\{x_m \in \mathbb{R}^d\}_{m=1}^M$, $g(\mathbf{x})$ is a convex penalty function, and the weight parameter λ balances the degree of personalization. There are many possible choices for the penalty function $g(\mathbf{x})$. A simple option is to let $g(\mathbf{x})$ be the deviation between the local models and their average [461, 460, 462]:

$$g(\mathbf{x}) = \frac{1}{2M} \sum_{m=1}^{M} \|x_m - \bar{x}\|^2, \quad \text{where } \bar{x} = \frac{1}{M}(x_1 + \cdots + x_M). \tag{6.3}$$

This is a reasonable choice in the centralized distributed setting, where devices communicate with a central server, sending and receiving information without failures. In this case, calculating the average $\bar{x}$ is straightforward: all agents m simply send their local x_M to the central server, which then calculates the average $\bar{x}$ and communicates it back to every agent. In this section, we consider a more general setup, where different penalty functions may be more appropriate.

Throughout this work, we consider *decentralized distributed learning*, where there is no main server (node); instead, all devices are connected via a large network. Moreover, each agent in the network can only communicate with its neighbors. Mathematically, the network is represented by a fixed, undirected, connected graph, where each node corresponds to an agent, and connections between agents are represented by edges. Although a decentralized setting is assumed here, our problem formulation is general enough to include a centralized setup as a special case (simply take a complete graph). However, a decentralized setup perhaps better captures the federated learning setting, where each device only communicates with a limited number of other agents, corresponding to an incomplete graph. As previously mentioned, communication links between certain agents may be inaccessible, for example, due to the poor quality of the connection between agents or due to remoteness of location, leading to missing edges in the graph.

In a decentralized setting, employing the penalty in (6.3) is not sensible due to the impracticality of calculating the average $\bar{x}$. (Note that to calculate $\bar{x}$, all local x_m's must be sent to one device (node) and then the average $\bar{x}$ broadcast back to every node, which is a long and expensive operation, especially for large networks.) With this in mind, here we propose the use of a different penalty $g(\mathbf{x})$, which is more suitable for a decentralized setup. Hence, the problem formulation considered in this work is:

$$\min_{\mathbf{x} \in \mathbb{R}^{Md}} F(\mathbf{x}) := \underbrace{\frac{1}{M} \sum_{m=1}^{M} f_m(x_m)}_{f(\mathbf{x})} + \underbrace{\frac{\lambda}{2} \langle \mathbf{x}, \mathbf{W}\mathbf{x} \rangle}_{g(\mathbf{x})}, \tag{6.4}$$

where $\mathbf{W} = W \otimes I$ with a communication gossip matrix W which reflects the properties of the network. We have already met with this matrix in Chapter 2. For the sake of completeness, we introduce it here as well.

For a fixed network, communication can be modeled via an undirected connected graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = [M]$ are vertices (workers) and $\mathcal{E} = \{(i,j) \,|\, i,j \in \mathcal{V}\}$ are edges. Note that $(i,j) \in \mathcal{E}$ if and only if there exists a communication link between agents i and j. For such a graph,

a gossip matrix $\hat{W}$ is defined as follows.

Definition 6.0.1 (Gossip matrix)**.** A matrix $W \in \mathbb{R}^{M\times M}$, associated with a graph $\mathcal{G}$, is called a gossip matrix, if it satisfies the following conditions:

1. W is symmetric positive semi-definite;
2. The kernel of W consists of the vector $\mathbf{1} = (1, \dots, 1)^\top$;
3. W is defined on the edges of the communication network: $w_{i,j} \neq 0$ if and only if $i = j$ or $(i, j) \in \mathcal{E}$.

The function $g(\mathbf{x}) = \langle \mathbf{x}, \mathbf{W}\mathbf{x}\rangle$ penalizes the difference between neighboring local models in the network, and is computationally friendlier than (6.3) in a decentralized setting. The matrix W determines how much an agent depends on each of the other nodes in the learning process. This is achieved due to the fact that W represents the structure of the communication graph, gives information about the remoteness of the nodes, the speed of transfer between them, and carries weights of how much to rely on one or another neighbour in the network. Note that $\mathbf{W}\mathbf{x} = 0$ (and consequently $r(\mathbf{x}) = 0$) if and only if $x_1 = \cdots = x_M$. This penalty function is not new and has been used in the literature in several contexts, for example, for classical decentralized minimization with large λ [463, 464], and for multitask PFL with small λ [465, 466].

The parameter λ balances the 'global vs local' trade-off. For example, consider the following extremes:

* If $\lambda = 0$, then (6.4) becomes $\min_{\mathbf{x}\in\mathbb{R}^{nd}} \sum_{i=1}^n f_i(x_i)$, where the local function f_i held by agent i is minimized by x_i^*, and x_i^* is likely to be different than that obtained for agent j. This is equivalent to independent local training of the models.
* As $\lambda \to +\infty$, (6.4) tends to the distributed problem where the local arguments are constrained to be equal: i.e., $\min_{x_1=\cdots=x_n\in\mathbb{R}^d} \sum_{i=1}^n f_i(x_i)$. This is equivalent to problem (6.1) and the training of one global model.

Formally, we can look at the problem (6.4) as a general composite problem of the form:

$$\min_{x\in\mathcal{X}} \Psi_0(x) := f(x) + g(x), \tag{6.5}$$

where we minimize the two-part function Ψ_0 on some set $\mathcal{X}$. The problem (6.5) is the focus of this chapter. Having obtained the results for (6.5) we transfer them automatically to (6.4).

In this chapter, we examine the problem (6.4) through the lens of the problem (6.5) from different perspectives. First, we assume that the function $f(\mathbf{x})$ is non-smooth and that only a zero-order oracle is available for it. In the second section, we consider the smooth function $f(\mathbf{x})$.

6.1 Sliding for Black Box Non-Smooth Case

One can note that the function g form (6.5) can be a regularizer; usually, it is a simple function for which the gradient ∇g can be computed. At the same time, the objective function f can be quite complex. In this section, we focus on the case when, for the function f, only a zero-order oracle (i.e., only the values of the function f, but not its gradient) is available. In the literature, this concept is sometimes referred to as a black box. It arises when the calculation of the gradient is expensive (in adversarial training [327], optimization [69], structured-prediction learning [326]) or impossible (in reinforcement learning [467, 468, 469], bandit problem [470, 341], black-box ensemble learning [328]). To make the problem statement even more practical, we assume that we have access to inexact values of the function $f(x, \xi)$ with some random noise ξ. With the help of this oracle, it is possible to make some approximation of the gradient in terms of finite differences. Next, we highlight two main approaches for such gradient estimation. The first approach is called a two-point feedback:

$$\tilde{f}'_r(x, \xi) := \frac{n}{2r}(f(x + re, \xi) - f(x - re, \xi))e, \tag{6.6}$$

where $e \in \mathbb{R}^n$ is uniformly distributed on the unit Euclidean sphere, $r > 0$ is the smoothing parameter responsible for the margin length of the approximation. For the two-point feedback there are many papers with theoretical analysis [355, 69, 358, 341, 363, 342]. An important aspect of this approach is the assumption that we are able to obtain the values of the function at points $x + re$ and $x - re$ with the same realization of the noise ξ. However, from a practical point of view, this is a very strong and idealistic assumption. Therefore, it is proposed to consider the concept of one-point feedback [358, 359, 471]:

$$\tilde{f}'_r(x, \xi^{\pm}) := \frac{n}{2r}(f(x + re, \xi^{+}) - f(x - re, \xi^{-}))e. \tag{6.7}$$

In general $\xi^{+} \neq \xi^{-}$. In this section, we work with the one-point concept.

The function f is "bad", while the function g is "good". The question arises how to minimize Ψ_0 from (6.5). The easiest option is to add the gradient of g and the "gradient" of f (from (6.6) or (6.7)) and make a step along it. However, when examining the problems (6.4), computing the gradient of g necessitates communication, while calculating the "gradient" of f does not. Communications serve as the bottleneck of distributed algorithms, requiring significantly more time than local computations. Therefore, there is a desire to reduce the number of communications and to calculate the gradient g as rarely as possible.

This brings us to the goal of this section: to formulate an algorithm that addresses the composite optimization problem, where one component has either a two-point or one-point zero-order oracle, while the other component relies on a gradient. At the same time, we want to make calls to the gradient as rarely as possible.

6.1.1 Related Works and Contributions

• **Sliding.** The naive approach to (6.5) looks at it as a whole problem and does not take into account its composite structure. This oversight can significantly degrade the oracle complexity (number of the oracle calls) for one of the functions. The sliding technique facilitates the avoidance of these losses and enables the separation of oracle complexities. In particular, if we can solve a separate problem $\min f(x)$ by T_f oracle calls (these can be calls of gradient or any other oracle, for example, zero-order), and a problem $\min g(x)$ by T_g oracle calls, then the sliding technique gives that we can solve the composite problem by $\mathcal{O}(T_f)$ oracle calls corresponding to f and $\mathcal{O}(T_g)$ oracle calls corresponding to g. As mentioned above, if we employ the naive approach, we maintain the same complexity $\mathcal{O}(\max(T_g; T_f))$ for both f and g.

There are various types of sliding in the literature, depending on the assumptions made for (6.5).

- The sliding was justified for smooth f and g with gradient oracles for convex optimization problems in [85, 472, 280, 473, 474].
- In [475, 476], they considered case with smooth f and g with zero-order oracle for f and gradient oracle for g.
- In [211, 477], the sliding technique was used for non-smooth f and smooth g with stochastic subgradient oracle for f and gradient oracle for g.
- The authors of [211] adapted the sliding approach for non-smooth f and smooth g with gradient oracle for g.

The development of the sliding technique is a quite popular issue in the literature, but on the other hand that there are still many open problems especially for the mixture oracle.

• **Gradient-free distributed setup.** For one-point stochastic feedback we know only one result [478], they assume that the target function f is highly-smooth and strongly convex. However, the method is very expensive in terms of decentralized communications. The reason for this issue is that, in [478], the authors fight only for the oracle calls criteria and do not use the sliding technique that allows to split communication complexity from the oracle one. In this section, using the sliding technique, we split these complexities in the problem (6.4) and obtain much better guarantees on the number of communications. For the two-point case, we know of no such work.

The contributions of this section can be summarized as follows:

We present a new method based on the sliding technique for the convex problem (6.5) with the mixture oracle: first-order for the smooth part g and zero-order for the non-smooth part f. Our method solves the problems mentioned above in the introduction. It reduces the number of calls to the gradient ∇g of the smooth part of the composite problem, while using one-point feedback for the non-smooth part f.

Note that all the results were obtained in the general (non-Euclidean) proximal setting to take

into account the geometry of the problem. This was achieved by using the Bregman divergence instead of the classical Euclidean distance. It allows sometimes to reduce the f-oracle calls complexity $\sim d$-times in comparison with algorithms that use the Euclidean setup, where d is a dimension of the problem – see Table 6.1.

Table 6.1: Summary of complexity results on the number of ∇g and f calls for finding an ε-solution of (6.5) in the different geometric setups: Euclidean and non-Euclidean. Convergence is measured by the function distance to the solution.

	$\|\cdot\| = \|\cdot\|_2$	$\|\cdot\| = \|\cdot\|_1$
∇g	$O\left(\sqrt{\frac{LD^2}{\varepsilon}}\right)$	$O\left(\sqrt{\frac{LD^2}{\varepsilon}}\right)$
f	$\tilde{O}\left(\sqrt{\frac{LD^2}{\varepsilon}} + \frac{dG^2D^2}{\varepsilon^2} + \frac{d^2G^2\sigma^2D^2}{\varepsilon^4}\right)$	$O\left(\sqrt{\frac{LD^2}{\varepsilon}} + \frac{\ln d G^2D^2}{\varepsilon^2} + \frac{d\ln dG^2\sigma^2D^2}{\varepsilon^4}\right)$

Notation: L = constant of L-smoothness of g in the corresponding norm, G = upper bound of f' in the ℓ_2-norm ($\|f'(x)\|_2 \leq G$ for all $f'(x) \in \partial f(x)$ and $x \in \mathcal{X}$), d = dimension of vector x ($\mathcal{X} \subset \mathbb{R}^d$), σ^2 = variance of unbiased noise ξ (for two-point feedback $\sigma = 0$), D = diameter of $\mathcal{X}$(in the corresponding geometry).

We also present the applicability and relevance of our new method for distributed and federated learning problems in both centralized (6.2)+(6.3) and decentralized (6.4) setups – see Table 6.2. It turns out that this method can be useful in terms of reducing the number of communications.

Table 6.2: Summary of complexity results on communications (comm) and local computations (local) for finding an ε-solution of centralized (6.2)+(6.3) and decentralized (6.4) distributed problems. Convergence is measured by the function distance to the solution.

	Centralized	**Decentralized**
`comm`	$O\left(\sqrt{\frac{\lambda D^2}{\varepsilon}}\right)$	$O\left(\sqrt{\frac{\lambda\lambda_{\max}(W)D^2}{\varepsilon}}\right)$
`local`	$O\left(\sqrt{\frac{\lambda D^2}{\varepsilon}} + \frac{MdG^2D^2}{\varepsilon^2} + \frac{Md^2G^2\sigma^2D^2}{\varepsilon^4}\right)$	$O\left(\sqrt{\frac{\lambda\lambda_{\max}(W)D^2}{\varepsilon}} + \frac{MdG^2D^2}{\varepsilon^2} + \frac{Md^2G^2\sigma^2D^2}{\varepsilon^4}\right)$

Notation: $\lambda_{\max}(W)$ = maximum eigenvalue of W, G = upper bound of f'_m in the ℓ_2-norm ($\|f'_m(x)\|_2 \leq G$ for all $f'(x) \in \partial f(x)$, $x \in \mathcal{X}$ and m), d = dimension of vector x ($\mathcal{X} \subset \mathbb{R}^d$), σ^2 = variance of unbiased noise ξ (for two-point feedback $\sigma = 0$), D = diameter of $\mathcal{X}$ (in the Euclidean norm).

6.1.2 Assumptions

As in Section 3.2, here, we address non-Euclidean geometry for the first time in this thesis. Prior to this, we have consistently utilized short notation $\|\cdot\| = \|\cdot\|_2$. In this section, we assume that $\mathcal{X} \subseteq \mathbb{R}^d$ is a normed vector space with a dual space $\mathcal{X}^*$ and a primal-dual norm pair $\|\cdot\|$ and $\|\cdot\|_*$. As $\|\cdot\|$, we employ the norm $\|\cdot\|_p$ for $p \in [1, 2]$. Moreover, $\|\cdot\|_* = \|\cdot\|_q$ with $1/p + 1/q = 1$.

To work in arbitrary geometry, it is suggested to introduce the following object.

Definition 6.1.1 (Bregman divergence)**.** Suppose some continuously differentiable function $\phi(x)$ is 1-strongly convex w.r.t. norm $\|\cdot\|$ on $\mathcal{X}$. Then for any two points $x, y \in \mathcal{X}$ we define the Bregman divergence $V(x, y)$ associated with $\phi(x)$ as follows:

$$V(x, y) := \phi(y) - \phi(x) - \langle \nabla\phi(x), y - x\rangle.$$

We denote the Bregman diameter of the set $\mathcal{X}$ w.r.t. $V(x, y)$ as $D_{\mathcal{X},V} := \max\{\sqrt{2V(x, y)} \mid x, y \in \mathcal{X}\}$.

Next, we introduce some assumptions valid for (6.5). Furthermore, we will show how they are related to the assumptions for (6.4).

Assumption 6.1.2. The function g is convex and L-smooth w.r.t. norm $\|\cdot\|$ on $\mathcal{X}$ with $L > 0$, i.e., it is differentiable and there is a constant $L > 0$ such that the following inequalities hold for all $x, y \in \mathcal{X}$:

$$\|\nabla g(x) - \nabla g(y)\|_* \leq L\|x - y\|,$$

and

$$g(y) \geq g(x) + \langle \nabla g(x), y - x\rangle.$$

Assumption 6.1.3. The function f is convex and G-Lipschitz w.r.t. norm $\|\cdot\|_2$ on $\mathcal{X}$ with $G > 0$, i.e., there is a constant $G > 0$ such that the following inequalities hold for all $x, y \in \mathcal{X}$:

$$|f(x) - f(y)| \leq G\|x - y\|_2 \quad \text{for all} \quad x, y \in \mathcal{X}.$$

and for all $\alpha \in [0; 1]$

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y).$$

Assumption 6.1.4. The set $\mathcal{X} \subset \mathbb{R}^d$ is a compact and convex set with the Bregman diameter $D_{\mathcal{X},V}$.

6.1.3 Algorithm

From first to zero-order

Before presenting the main algorithm, let us understand the properties of the approximation (6.7) that we use. Most of these properties have already been encountered in the literature [341], we modify only a few of them for our case. These properties are associated with the following object

$$F(x) := \mathbb{E}[f(x + r\tilde{e})], \tag{6.8}$$

where $\tilde{e} \in \mathbb{R}^n$ is a vector randomly uniformly generated from the unit Euclidean ball and $r > 0$ is the smoothing parameter. The function F is called the "smoothed" version of the function f. Our algorithm does not use it in any way, but it will be used in the theoretical analysis.

It is worth noting that the use of F as well as the approximation (6.7) implies that the function f has to be defined not only on the set $\mathcal{X}$ itself, but also on some neighborhood of it, in order to make the calculation of $f(x + r\tilde{e})$ valid for any $x \in \mathcal{X}$ and $\tilde{e}$ from the unit Euclidean ball. We will not focus much on this fact, which is standard in the literature on gradient-free optimization. Let us only note that there are two ways to fulfill this assumption if the function f is defined strictly on the set $\mathcal{X}$. The first option is to reduce the initial set in the problem (6.5) and consider $\mathcal{X}_{new}$ such that $\mathcal{X}_{new} + B_2^d(r) \subseteq \mathcal{X}$, where $B_2^d(r)$ is the Euclidean ball with a radius of r (see Section 3.2 from [356] for more details). In addition, it is possible to predefine the function f over the entire space $\mathbb{R}^n$. In particular, it can be done as follows: $f_{new}(x) := \min_{z \in \mathcal{X}} [f(z) + G\|x - z\|]$. One can prove that f_{new} is equivalent to f on $\mathcal{X}$, G-Lipschitz w.r.t. norm $\|\cdot\|_2$ on $\mathbb{R}^n$ and convex on $\mathbb{R}^n$.

Now let us move to the important technical results.

Lemma 6.1.5. $F(x)$ from (6.8) is convex, differentiable and it holds that

$$\sup_{x \in \mathcal{X}} |F(x) - f(x)| \leq rG, \tag{6.9}$$

$$\mathbb{E}[\tilde{f}_r'(x, \xi^{\pm})] - \nabla F(x) = 0, \tag{6.10}$$

$$\text{for two-point feedback: } \mathbb{E}[\|\tilde{f}_r'(x, \xi^{\pm})\|_*^2] \leq 8dp^2(d)G^2, \tag{6.11}$$

$$\text{for one-point feedback: } \mathbb{E}[\|\tilde{f}_r'(x, \xi^{\pm})\|_*^2] \leq p^2(d)\left(8dG^2 + \frac{2d^2\sigma^2}{r^2}\right), \tag{6.12}$$

where $p^2(d) := \min\{2q - 1, 32 \log d - 8\} d^{\frac{2}{q}-1}$.

Let us discuss these facts. The property (6.10) means that the approximation (6.7) is an unbiased estimate of the gradient, not of the original function f, but of the smoothed function F. This means that we can replace the original problem (6.5) with $\min[F(x) + g(x)]$ and now consider the oracle (6.7) for F as an unbiased stochastic gradient with a second moment equal to (6.11) or (6.12). The question arises, how much the new problem is different from the original one? (6.9) states that for a small parameter r, the original problem (6.5) and the new one are very close. The proof of convergence of the algorithm will be built is based on this idea. As mentioned above, all the difference between the two-point case and the one-point case is the presence of σ.

Algorithm and convergence analysis

As mentioned above, our algorithm is based on the sliding algorithm [211, 1]. Our method represents a modification of the first-order sliding with a zero-order oracle. The sliding (complexities splitting) effect is achieved by the fact that the method consists of outer and inner loops. At the outer iterations, we compute the gradient of the function g, while in the inner loop (prox-sliding procedure), only the function f is utilized, with fixed information about the gradient of g.

The following theorem gives an estimate for the convergence rate of this method:

Algorithm 54 Zero-Order Sliding Algorithm (ZOSA)

Parameters: $\{\beta_k\} > 0$, $\{\gamma_k\} > 0$, and $\{T_k\} \in \mathbb{N}$, iteration limit K
Initialization: choose $\overline{x}^0 = x_0 \in \mathcal{X}$

1: **for** $k = 0, \ldots, K-1$ **do**
2: Set $\underline{x}^k = (1-\gamma_k)\overline{x}^{k-1} + \gamma_k x^{k-1}$
3: Let $h_k(y) = g(\underline{x}^k) + \left\langle \nabla g(\underline{x}^k), y - \underline{x}^k \right\rangle$
4: Set $(x^k, \tilde{x}^k) = \texttt{PS}(h_k, x^{k-1}, \beta_k, T_k)$
5: Set $\overline{x}^k = (1-\gamma_k)\overline{x}^{k-1} + \gamma_k \tilde{x}^k$
6: **end for**

PS procedure: $(x^+, \tilde{x}^+) = \texttt{PS}(h, x, \beta, T)$

Parameters: $\{p_t\} > 0$, $\{\theta_t\} > 0$, iteration limit T
Initialization: function $h : \mathbb{R}^d \to \mathbb{R}$, initial point $u_0 = \tilde{u}_0 = x \in \mathcal{X}$, $\beta > 0$

1: **for** $t = 0, \ldots, T-1$ **do**
2: $u_t = \underset{u \in \mathcal{X}}{\operatorname{argmin}} \left\{ h(u) + \left\langle \tilde{f}'_r(u_{t-1}, \xi^{\pm}_{t-1}), u \right\rangle + \beta V(x,u) + \beta p_t V(u_{t-1}, u) \right\}$
3: $\tilde{u}_t = (1-\theta_t)\tilde{u}_{t-1} + \theta_t u_t$
4: **end for**
5: **Output:** $x^+ = u_{T-1}$ and $\tilde{x}^+ = \tilde{u}_{T-1}$

Theorem 6.1.6. Suppose that $p_t = \frac{t}{2}$, $\theta_t = \frac{2(t+1)}{t(t+3)}$, $\beta_k = \frac{2L}{k}$, $\gamma_k = \frac{2}{k+1}$,

$$T_k = \max\left\{1; \frac{16Kk^2}{3D^2_{\mathcal{X},V}L^2} \cdot \left(14p^2(d)dG + \frac{p^2(d)d^2\sigma^2}{r^2}\right)\right\}$$

for $t \geq 1, k \geq 1$. Then for any number of iterations N of Algorithm 54 it holds that

$$\mathbb{E}[\Psi_0(\overline{x}^K) - \Psi_0(x^*)] \leq 2rG + \frac{20LD^2_{\mathcal{X},V}}{K(K+1)}. \tag{6.13}$$

Additionally, the total number of PS procedure iterations is

$$T^{\text{total}} = \frac{(K+1)^4}{D^2_{\mathcal{X},V}L^2} \cdot \left(10p^2(d)dG + \frac{2p^2(d)d^2\sigma^2}{r^2}\right) + N, \tag{6.14}$$

where $p^2(d) := \min\{2q-1, 32\log d - 8\}d^{\frac{2}{q}-1}$.

This theorem demonstrates the significance of the choice of r. In particular, it follows from (6.13) that r should be taken as small as possible. On the other hand, it follows from (6.14) that as r decreases, the total number of internal iterations increases. This situation creates a challenge: the parameter r must be controlled and adjusted carefully.

Corollary 6.1.7. Under the assumptions of Theorem 6.1.6 and if we put $r = \Theta\left(\frac{\varepsilon}{G}\right)$, then the number of calls of ∇g and f required by Algorithm 1 to find an ε-solution $\overline{x}^K$ of (6.5) (i.e.

$\mathbb{E}[\Psi_0(\overline{x}^K)] - \Psi_0(x^*) \leq \varepsilon)$ is respectively bounded by

$$O\left(\sqrt{\frac{LD^2_{\mathcal{X},V}}{\varepsilon}}\right) \quad \text{and}$$

$$O\left(\sqrt{\frac{LD^2_{\mathcal{X},V}}{\varepsilon}} + \frac{d \cdot p^2(d) G^2 D^2_{\mathcal{X},V}}{\varepsilon^2} + \frac{d^2 \cdot p^2(d) G^2 \sigma^2 D^2_{\mathcal{X},V}}{\varepsilon^4}\right),$$

where $p^2(d) := \min\{2q - 1, 32 \log d - 8\} d^{\frac{2}{q}-1}$.

This is the result that we wanted to achieve by the use of the sliding. The number of ∇g calculations is not affected in any way by the fact that we use a "very bad" oracle for f. Our results also match the obtained bounds for one-point feedback in the non-distributed composite-free case [358].

Remark. Note that the second estimate depends on the "geometry" of the problem. In particular, in the Euclidean case $\|\cdot\| = \|\cdot\|_2$ with $q = 2$, we have the following oracle complexity (number of calls) for f

$$O\left(\sqrt{\frac{LD^2_{\mathcal{X}}}{\varepsilon}} + \frac{dG^2 D^2_{\mathcal{X}}}{\varepsilon^2} + \frac{d^2 G^2 \sigma^2 D^2_{\mathcal{X}}}{\varepsilon^4}\right), \tag{6.15}$$

where we use $D_{\mathcal{X}}$ as the Euclidean diameter of the set $\mathcal{X}$. A more interesting case is the case when we work in the non-Euclidean setting. In particular, let us consider a probability simplex with $\|\cdot\| = \|\cdot\|_1$, $q = \infty$ and $D^2_{\mathcal{X},V} = 2 \ln d$. In this case, the estimate is transformed into

$$O\left(\sqrt{\frac{\ln d \cdot L}{\varepsilon}} + \frac{\ln^2 d \cdot G^2}{\varepsilon^2} + \frac{d \ln^2 d \cdot G^2 \sigma^2}{\varepsilon^4}\right). \tag{6.16}$$

It can be observed that (6.16) enhances the estimate (6.15) d times by utilizing a different geometric setup. Moreover, if the noise $\sigma = 0$, our estimates are consistent (up to $\ln d$) with the estimates for the full-gradient method [211].

Applications to distributed optimization

Let us now look at some examples, including those for which sliding gives the estimates necessary in practice. We consider the problems (6.2)+(6.3) and (6.4). In the introduction, we briefly mentioned that in these problems, we need to reduce the number of ∇g calls, and thus the number of communications. Indeed, in order to calculate the gradient g in the problem (6.2)+(6.3), we need to know the value of $\bar{x}$. However, it is impossible to calculate this using local computations only, it means that one needs help from the central server: we send all current x_m and get the average $\bar{x}$. At the same time, all calculations of f do not require any communication. To compute $f(\mathbf{x})$ we need to compute $f_m(x_m)$, and these are just the values of local functions on local variables. For the problem (6.4), the same reasoning is valid, but communication takes place with neighbors using the gossip protocol with W. In more details, the computation of ∇g requires the calculation of the product $\mathbf{Wx}$. Since the matrix W has only non-zero weights, when

two workers are connected, the multiplication of $\mathbf{Wx}$ corresponds to the exchange of information between neighbours, where each of the workers, having received parcels from neighbours, averages them with weights according to the matrix W. Therefore, the number of $\mathbf{Wx}$ multiplications is the total number of required communications in the case of a decentralized protocol. Computing ∇g requires such a multiplication, then the number of communications is equivalent to the calls of the first-order oracle for g.

Now we are ready to obtain estimates for the problems (6.2)+(6.3) and (6.4) from the general results of the previous section. We consider the Euclidean case. It remains only to describe the properties of the problems (6.2)+(6.3) and (6.4). For this purpose we give the following lemma.

Lemma 6.1.8. Let the functions f_m from (6.2)+(6.3) and (6.4) be G-Lipschitz w.r.t. $\|\cdot\|_2$, the local noises ξ_m for all $f_m(x_m, \xi_m) = f_m(x_m) + \xi_m$ be independent, unbiased and bounded: $\mathbb{E}\xi_m = 0$, $\mathbb{E}[\xi_m^2] \leq \sigma^2$, the Euclidean diameter of the set $\mathcal{X}$ be equal to $D_{\mathcal{X}}$. Then the following facts for (6.2)+(6.3) and (6.4) are valid:

- $f(\mathbf{x})$ is $(G/\sqrt{M})$-Lipschitz w.r.t. $\|\cdot\|_2$;
- $g(\mathbf{x})$ from (6.2)+(6.3) is (λ/M)-smooth w.r.t. $\|\cdot\|_2$, and g from (6.4) is $(\lambda\lambda_{\max}(W)/M)$-smooth w.r.t. $\|\cdot\|_2$;
- the Euclidean diameter $D_{\mathcal{X}^M}$ of the set $\mathcal{X}^M$ is equal to $\sqrt{M}D_{\mathcal{X}}$;
- the noise of the function $f(\mathbf{x}, \xi_1, \ldots, \xi_M)$ is unbiased and bounded by σ^2/M.

One can note that if $\mathcal{X} \in \mathbb{R}^n$, then the dimension of vector $\mathbf{x}$ is $d \cdot M$. Recall that the number of computations of $\nabla g(\mathbf{x})$ corresponds to the number of communication rounds, and the calls $f(\mathbf{x})$ – to the local gradient-free calculations. Then the following estimates are valid for the number of communications and local iterations to find ε-solution in terms of the target functions:

- in the centralized case

$$O\left(\sqrt{\frac{\lambda D_{\mathcal{X}}^2}{\varepsilon}}\right) \quad \text{communication rounds} \quad \text{and}$$

$$O\left(\sqrt{\frac{\lambda D_{\mathcal{X}}^2}{\varepsilon}} + \frac{MdG^2D_{\mathcal{X}}^2}{\varepsilon^2} + \frac{Md^2G^2\sigma^2D_{\mathcal{X}}^2}{\varepsilon^4}\right) \quad \text{local computations;}$$

- and in the decentralized case

$$O\left(\sqrt{\frac{\lambda\lambda_{\max}(W)D_{\mathcal{X}}^2}{\varepsilon}}\right) \quad \text{communication rounds} \quad \text{and}$$

$$O\left(\sqrt{\frac{\lambda\lambda_{\max}(W)D_{\mathcal{X}}^2}{\varepsilon}} + \frac{MdG^2D_{\mathcal{X}}^2}{\varepsilon^2} + \frac{Md^2G^2\sigma^2D_{\mathcal{X}}^2}{\varepsilon^4}\right) \quad \text{local computations.}$$

This is a rather remarkable result. We have "very bad" local functions (they are non-smooth and only with zero-order information), but this fact does not dramatically affect the communication complexity.

It is important to note that using (6.7) for $f(x_1, \ldots, x_M)$ implies that we use a common vector e for the full vector $\mathbf{x}$. It turns out that each device for its x_m have to take the corresponding part from the common e. This can be accomplished by using the same random generator with the same seed on all devices. In fact, similar results could be obtained using individual and independent directions on all devices.

6.1.4 Experiments

The purpose of our experiments is to compare how our method works in practice in comparison with classical methods. In particular, we compare our method with approaches that do not consider the composite structure of the problem (6.5). For this comparison, we analyze Mirror Descent in two settings: in the first case, we examine full-gradient Mirror Descent [33], which utilizes $f' + \nabla g$ as a subgradient; we also evaluate a gradient-free version, which employs $\tilde{f}'_r(x, \xi^{\pm}) + \nabla g$ as a "subgradient".

Comparison is made on the problem of distributed computation of the geometric median [479]. We have N vectors $\{b_i\}_{i=1}^N \in \mathbb{R}^d$:

$$\min_{x \in \mathbb{R}^d} f(x) = \sum_{i=1}^{N} \|x - b_i\|_2. \tag{6.17}$$

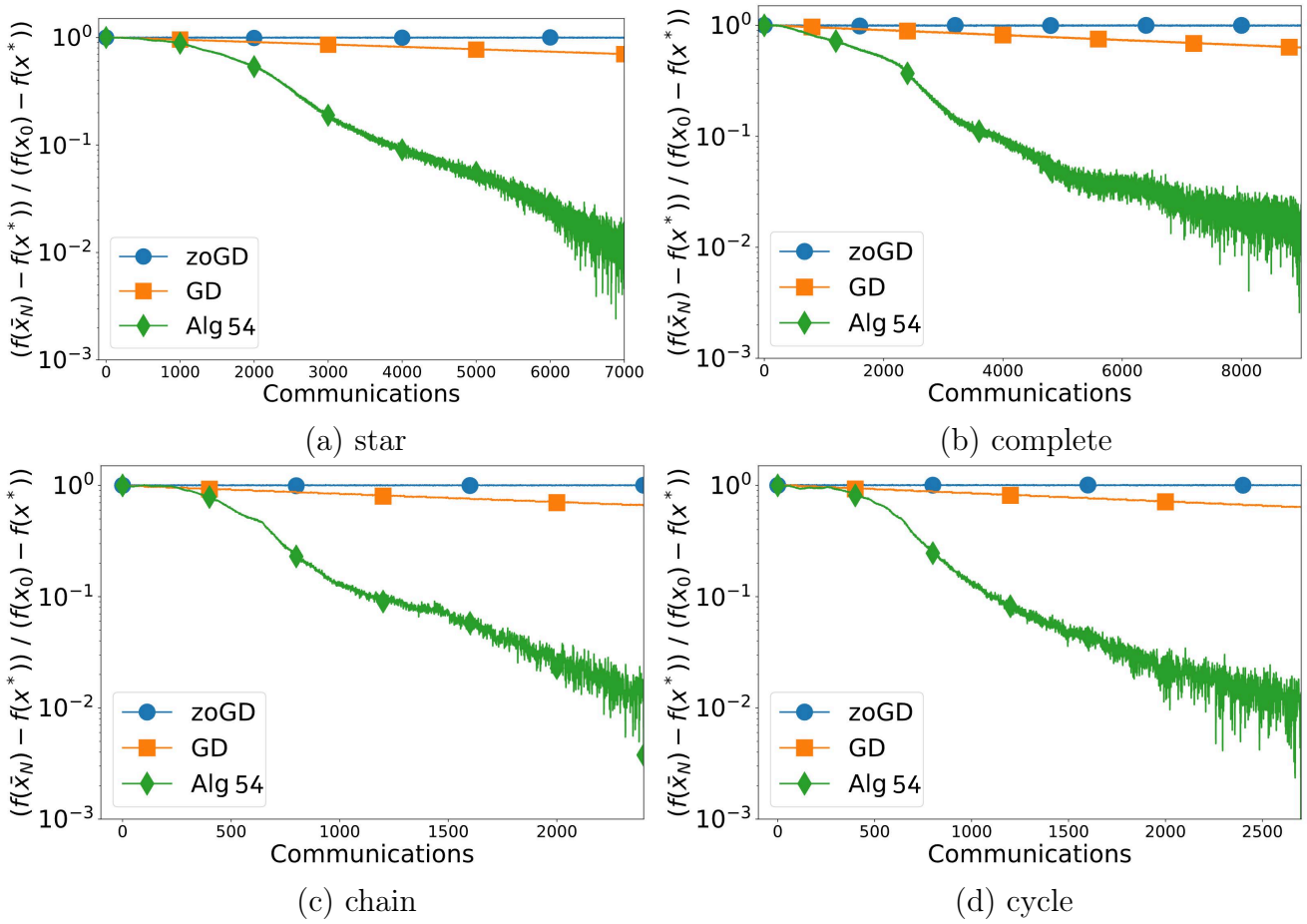
We distributed the vectors $\{b_i\}$ among $M = 10$ computing devices. Then the problem can be written in the form (6.4) with $f(\mathbf{x})$:

$$f(\mathbf{x}) = \sum_{m=1}^{M} \underbrace{\sum_{i=1}^{\lfloor N/M \rfloor} \|x_m - \hat{b}_m^i\|_2}_{f_m(x_m)}, \tag{6.18}$$

where $\hat{b}_m^i = b_{(m-1)\cdot\lfloor N/M \rfloor + i}$. It is easy to verify that each f_m is non-smooth, but 1-Lipschitz and convex, also, as noted before, g is $\lambda\lambda_{\max}(W)$-smooth and convex. To make our setting stochastic, each time when we call f' or value f, we independently generate normal noise vectors $\{\xi_i\}$ for each vector $\{b_i\}$ and compute f' or f using $\{b_i + \xi_i\}$ instead of true $\{b_i\}$.

We run Algorithm 54, the first-order Mirror Descent [33] and the zero-order Mirror Descent [355] on the problem (6.18) with $d = 100$, $N = 50$ and $\lambda = 10^2$. The vectors $b_1, \ldots, b_N$ are generated as i.i.d. samples from the normal distribution $\mathcal{N}(\mathbf{1}, 2I)$, the noise ξ is also generated from the normal distribution $\mathcal{N}(0, \sigma^2 I)$, where $\sigma = 0.01$. We consider different decentralized topologies: star, complete graph, chain, and cycle. Both competitor-methods are tuned for better convergence: in the first-order Mirror Descent we tune the step size, in the zero-order Mirror Descent we put

$r = 10^{-2}$ in (6.7) and also tune the step size. Algorithm 54 is not tuned and is used with a theoretical set of parameters from Theorem 6.1.6 and $r = 10^{-2}$ in (6.7). The main comparison criterion is the number of communications, i.e. calls to the oracle $\nabla g = \lambda W X$. We measure the quality of the solution with respect to the target function f of the original problem (6.17), the problem (6.18) is auxiliary and arises due to the distributional nature of the formulation. See Figure 6.1 for the results. We notice that in these experiments Algorithm 54 outperforms even Mirror Descent which is the first-order method. This shows the importance of taking into account the composite structure of the problem.



(a) star (b) complete

(c) chain (d) cycle

Figure 6.1: Algorithm 54 (green), first-order Mirror Descent (orange) and zero-order Mirror Descent (blue) applied to solve (6.17) in distributed form (6.18) with different network topologies.

6.2 Sliding for First-Order Smooth Case

6.2.1 Contributions

In this section, we study the personalized federated learning formulation (6.4). Lower complexity bounds for communication and local computation are proposed, and several algorithms capable of achieving these lower bounds are developed. Our results extend the work in [461], which used the penalty (6.3), to problem (6.4), which involves a penalty more amenable to the decentralized setting. Our contributions are summarized below.

• **Lower bounds.** We present lower bounds for the decentralized personalized federated learning problem (6.4) in the deterministic case (i.e., when we have access to full gradients for each f_i); see Section 6.2.2. The lower bounds are valid for all values of the parameter λ. In particular, in the smooth strongly convex case with small λ, the lower bounds are of the order $\sqrt{\lambda\lambda_{\max}(W)/\mu}$, which can be a significant improvement over the bound $\sqrt{\chi L/\mu}$ in the general, non-personalized case, [123]. This reflects a key advantage of the formulation (6.4), as it allows for both personalizing the models and significantly reducing the total number of communications. This is an important factor not only in federated learning but also in general distributed learning. Note that the lower bounds obtained in the work [461] are a special case of our lower bounds when the communication network is represented by a fully connected graph. A summary of these lower bounds is presented in Table 6.3.

Table 6.3: Summary of complexity results (upper and lower bounds) on communications (comm) and local computations (local) for finding an ε-optimal solution of (6.4) in the deterministic (gradient) case.

	Lower bounds	**Upper bounds**
`comm`	$\widetilde{\Omega}\left(\min\left\{\sqrt{\frac{\lambda\lambda_{\max}(W)}{\mu}}, \sqrt{\frac{L}{\mu}\chi}\right\}\right)$	$\widetilde{\mathcal{O}}\left(\min\left\{\sqrt{\frac{\lambda\lambda_{\max}(W)}{\mu}}, \sqrt{\frac{L}{\mu}\chi}\right\}\right)$
`local`	$\widetilde{\Omega}\left(\sqrt{\frac{L}{\mu}}\right)$	$\widetilde{\mathcal{O}}\left(\sqrt{\frac{L}{\mu}}\right)$

• **Optimal algorithm.** Another contribution is the development of optimal algorithms that match the theoretical lower bounds. The Accelerated Meta-Algorithm of [480] (for general composite problems) serves as the base algorithm. The application of this algorithm to our problem formulation (6.4) is discussed, and specific implementation modes are suggested depending on small and large values of the regularization parameter λ. The analysis of convergence in these modes shows that using this approach we achieve the lower optimal bounds up to logarithmic factors (Section 6.2.3). Hence, our algorithm is 'near-optimal' in the deterministic case; see Section 6.2.3.

• **Experiments.** We present numerical experiments to demonstrate the benefits of our approach. In particular, we used several datasets from the benchmark LIBSVM library and considered various graph structures. We also conducted experiments for multiple values of the penalty parameter λ to better understand the impact of personalization; see Section 6.2.4.

6.2.2 Lower Bounds

In this section, optimal algorithms for problems of the form (6.4) are described, and lower bounds on the local computation and communication costs for such optimal algorithms are presented. We begin with the following assumption, which describes the properties of algorithms relevant for this work, specifically the properties of the algorithms for which the lower bounds developed later in this section are valid. Such an assumption is common in the literature; see, for example, [461, 168, 123].

Assumption 6.2.1. Consider an Algorithm $\mathcal{A}$, for problem (6.4). Then, the iterates $\{\mathbf{x}^k\}_{k=1}^K$ of Algorithm $\mathcal{A}$ are generated using only components available in local memory, where, for each node of graph $\mathcal{G}$ the sequence of local memory $\{\mathcal{M}_{i,k}\}_{k=1}^K$ for $1 \leq i \leq n$ is:

$$\begin{aligned} \mathcal{M}_{i,0} &= \left\{x_i^0\right\}, \\ \mathcal{M}_{i,k+1} &= \begin{cases} \operatorname{span}\left\{\mathcal{M}_{i,k}, \nabla f_i(y_i)\right\}, \forall y_i \in \mathcal{M}_{i,k} & \text{if local comp. at iteration } k \\ \operatorname{span}\left\{\bigcup\limits_{j:(i,j)\in\mathcal{E}} \mathcal{M}_{j,k}\right\} & \text{if communication at iteration } k. \end{cases} \end{aligned}$$

Assumption 6.2.1 can be interpreted as follows. Initially, each agent i (corresponding to a node on graph $\mathcal{G}$) has local memory $\mathcal{M}_{i,0}$, which comprises the initial point x_i^0. At any iteration $k \geq 1$, the algorithm can either perform a computation using the locally available memory or carry out a communication step. If the algorithm performs a local computation, then each device can calculate the gradient at any point from its current memory $\mathcal{M}_{i,k}$ and take a linear combination of this gradient with the previously generated points stored in $\mathcal{M}_{i,k}$. If the algorithm performs a communication step, then information is exchanged with neighbors, and the current local memory $\mathcal{M}_{i,k}$ is combined with that held by its neighbors. Such an algorithm is first order because it generates its iterates using linear combinations of local points and gradients.

We are now ready to present our first theorem, which provides a lower bound on the number of communications needed by an algorithm whose iterates are generated according to Assumption 6.2.1.

Theorem 6.2.2. Let $\chi \geq 3$, $L \geq 2\mu$, and $\lambda\lambda_{\min}^+(W) \geq \mu$. Then there exist functions $f_1, f_2, \ldots, f_n : \mathbb{R}^d \to \mathbb{R}$ satisfying Assumption 6.2.6, a graph $\mathcal{G}$ with associated matrix $\hat{W}$ satisfying Definition 6.0.1, and an initial point $\mathbf{x}^0 = [(x_1^0)^T, \ldots (x_n^0)^T]^T \in \mathbb{R}^{Md}$, such that any algorithm $\mathcal{A}$ (satisfying Assumption 6.2.1) among K iterations need to make at least

$$\Omega\left(\min\left\{\sqrt{\frac{\lambda\lambda_{\max}(W)}{\mu}}, \sqrt{\frac{(L-\mu)\chi}{\mu}}\right\}\log\frac{1}{\varepsilon}\right) \quad \text{communications}$$

to achieve ε-optimal solution in the outputs ($\|x_j - x_j^*\|^2 \leq \varepsilon$ for all j).

It remains to develop lower bounds for the local computation costs for any algorithm $\mathcal{A}$

satisfying Assumption 6.2.1. Hence, consider a special instance of problem (6.4), where $\mathbf{x}^0 \in \mathbb{R}^{Md}$, $f_1 = f_2 = \cdots = f_M$, and W is the Laplace matrix for a fully connected graph. Then (6.4) reduces to the minimization of the single local function f_1 (communication is unnecessary, irrespective of λ, because the functions are all identical). Now, if f_1 is chosen to be the worst-case quadratic from [397], then the lower bound of at least

$$N^{\text{loc}} = \Omega\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right) \tag{6.19}$$

gradient calls are needed to find an ε-optimal solution.

6.2.3 Optimal Algorithms

The goal of this section is to develop an optimal algorithm for problem (6.4), i.e., to develop an algorithm whose iterates satisfy the lower bounds in Section 6.2.2. In Section 6.2.3, we discuss an algorithm that can be applied to general composite optimization problems. In Section 6.2.3, this algorithm is specialized to the application considered in this work, that of decentralized personalized federated learning (6.4).

Accelerated Meta-Algorithm

As in the previous section, we construct an algorithm for the problem (6.5), and later transfer it to the general formulation.

Assumption 6.2.3. The following assumptions are valid:

• The function g is convex and L_g-smooth with $L_g > 0$, i.e., it is differentiable and there is a constant $L_g > 0$ such that the following inequalities hold for all $x, y \in \mathbb{R}^d$:

$$\|\nabla g(x) - \nabla g(y)\| \leq L_g\|x - y\|,$$

and

$$g(y) \geq g(x) + \langle \nabla g(x), y - x\rangle.$$

• The function f is convex and L_f-smooth with $L_f > 0$, i.e., it is differentiable and there is a constant $L_f > 0$ such that the following inequalities hold for all $x, y \in \mathbb{R}^d$:

$$\|\nabla f(x) - \nabla f(y)\| \leq L_f\|x - y\|$$

and for all $\alpha \in [0; 1]$

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle.$$

• The function Ψ_0 is μ-strongly convex with $\mu > 0$, i.e., it is differentiable and there is a constant

$\mu > 0$ such that the following inequality holds for all $x, y \in \mathbb{R}^d$:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|x - y\|^2.$$

There are many efficient algorithms that can be applied to the problem (6.5) under Assumption 6.2.3, including the Accelerated-Meta-Algorithm (see Algorithm 55) proposed in [480].

Algorithm 55 Accelerated Meta-Algorithm (`AMA`) [480]

Parameters: $N = \max\left\{\left\lceil 4 \cdot \sqrt{\frac{2\gamma}{\mu}} \right\rceil, 1\right\}$, $\gamma > 0$, $\delta > 0$
Initialization: choose $x^0 \in \mathbb{R}^d$

for $k = 0, \ldots, K-1$ **do**
  $x^{k+1} = \texttt{MA}(x^k, N, \gamma, \delta)$
end for

`MA` procedure: $x^+ = \texttt{MA}(x, T, \gamma, \delta)$
Parameters: number of iterations T, $\gamma > 0$, $\delta > 0$
Initialization: $x_0 \in \mathbb{R}^d$, $A_0 = 0$, $y_0 = x_0$, $\tau = \frac{1}{2\gamma}$

for $t = 0, \ldots, T-1$ **do**
  $a_{t+1} = \frac{\tau + \sqrt{\tau^2 + 4\tau A_t}}{2}$
  $A_{t+1} = A_t + a_{t+1}$
  $w_t = \frac{A_t}{A_{t+1}} y_t + \frac{a_{t+1}}{A_{t+1}} x_t$
  Find $y_{t+1} \in \mathbb{R}^d$, such that $\|\hat{y}_{t+1} - y_{t+1}\|^2 \leq \delta$, where

$$\hat{y}_{t+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ \langle \nabla f(w_t), y - w_t \rangle + g(y) + \frac{\gamma}{2}\|y - w_k\|^2 \right\} \tag{6.20}$$

  $x_{t+1} = x_t - a_{t+1} \nabla \Psi_0(y_{t+1})$
end for
Output: y_T

Remark 6.2.4. Note that (6.20) can be solved, for example, by Accelerated Gradient Descent [397].

The following convergence results hold for `AMA` (Algorithm 55) applied to problem (6.5).

Theorem 6.2.5 (Theorem 3 in [280]). Let Assumption 6.2.3 hold, let $\gamma \geq 2L_f$, let $\varepsilon > 0$, and let

$$\delta \leq \frac{\varepsilon \mu}{864^2 (L_f + L_g + \gamma)^2}.$$

If Algorithm 55 runs for

$$S = \mathcal{O}\left(\sqrt{\frac{L_f}{\mu}} \log \frac{1}{\varepsilon}\right) \tag{6.21}$$

iterations, generating output x^S, then $\Psi_0(x^S) - \Psi_0(x^*) \leq \varepsilon$, where x^* denotes the optimal solution to (6.5).

Convergence Analysis

We introduced Algorithm 55 for the general problem (6.5), with associated convergence results. The purpose of this section is to connect the results for Algorithm 55 to their applicability in the context of personalized federated learning (i.e., problem (6.4)). Moreover, the lower bounds established in Section 6.2.2 relate to local computation and communication costs for an optimal algorithm for problem (6.4). Thus, another goal is to demonstrate that Algorithm 55 is an optimal algorithm for (6.4), by showing that it achieves the lower bounds on communication and local computation costs presented in Section 6.2.2.

By comparing problems (6.4) and (6.5), it can be seen that they are both convex and composite. The key here is that we do not make a one-to-one correspondence between $(f(\mathbf{x}), g(\mathbf{x}))$ from (6.4) and $(f(x), g(x))$ from (6.5). That is, depending on the parameter λ, two different cases — one in which $f(x) \equiv f(\mathbf{x})$ and $g(x) \equiv g(\mathbf{x})$, while the other in which $f(x) \equiv g(\mathbf{x})$ and $g(x) \equiv f(\mathbf{x})$ — are considered.

Regardless, to apply Algorithm 55, it is necessary to compute the gradients for both f and g (recall the subproblem (6.20)), and therefore for both f and g when extending to the original problem (6.4). As discussed in the previous section, the computation of $\nabla f(\mathbf{x})$ from (6.4) does not require communication. Indeed, each block m has a corresponding gradient ∇f_m (taken with respect to the local variables x_m), and the 'long' gradient ∇f is simply the concatenation of the block gradients. On the other hand, $\nabla g(\mathbf{x}) = \lambda \mathbf{W}\mathbf{x}$, and to compute the matrix-vector product $\mathbf{W}\mathbf{x}$ requires communication with neighbors (recall the gossip protocol described in Definition 6.0.1). It can be shown that computing $\nabla g(\mathbf{x})$ is equivalent to one communication. Therefore, if we know how many times $\nabla f(\mathbf{x})$ and $\nabla g(\mathbf{x})$ are called by Algorithm 55, then complexities for the number of local computations and communications, respectively, can be obtained.

Let us make the following assumption regarding the functions in (6.4).

Assumption 6.2.6. It is assumed that each $f_m : \mathbb{R}^d \to \mathbb{R}$ in the problem (6.4) is

- L-smooth, i.e. for all $u, v \in \mathbb{R}^d$, $\|\nabla f_m(u) - \nabla f_m(v)\| \leq L\|u - v\|$;
- μ-strongly convex, i.e. for all $u, v \in \mathbb{R}^d$, $f_m(u) - f_m(v) \geq \langle \nabla f_m(v), u - v\rangle + \frac{\mu}{2}\|u - v\|^2$.

By Assumption 6.2.6, f in (6.4) is L-smooth and μ-strongly convex.

We are now ready to present the main convergence theorem of this section, which provides complexity results for the local computation and communication costs for an optimal algorithm for problem (6.4).

Theorem 6.2.7. Let Assumption 6.2.6 hold, and let the graph $\mathcal{G}$ have an associated matrix W that satisfies Definition 6.0.1. Then, to obtain an ε-optimal solution to problem (6.4), solving by Algorithm 55 with

$$\delta = \frac{\varepsilon\mu}{3000^2(L + \lambda\lambda_{\max}(W))^2}, \tag{6.22}$$

requires the number of communications and local computations, respectively, to be of the order

$$N^{\text{comm}} = \mathcal{O}\left(\min\left\{\sqrt{\frac{\lambda\lambda_{\max}(W)}{\mu}}, \sqrt{\frac{L}{\mu}\chi}\right\}\log\tfrac{1}{\varepsilon}\log\tfrac{1}{\delta}\right), \tag{6.23}$$

and

$$N^{\text{loc}} = \mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\tfrac{1}{\varepsilon}\log\tfrac{1}{\delta}\right). \tag{6.24}$$

Proof. First, note that $g(\mathbf{x})$ from (6.4) is a quadratic function with a positive semi-definite Hessian, then it is $\lambda_{\max}(W)$-smooth and convex. Moreover, it is $\lambda\lambda^{+}_{\min}(W)$-strongly convex on the subspace $(\mathbf{Ker}\ W)^{\perp}$. By Assumption 6.2.6, $f(\mathbf{x})$ is L-smooth and μ-strongly convex. Hence, $F(\mathbf{x})$ is strongly convex. Thus, the conditions of Theorem 6.2.5 hold, and the application of its analysis is valid. The remainder of the analysis is split into two cases.

Case 1: $\lambda\lambda_{\max}(W) \geq L$. Here, let $f(x) = f(\mathbf{x})$ and $g(x) = g(\mathbf{x})$. Theorem 6.2.5 gives the complexity for the function $f(x) = f(\mathbf{x})$, i.e. the number of local computations N^{loc} is given in (6.24). Also, it can be shown that δ in (6.22) satisfies the condition in Theorem 6.2.5.

Next, consider the auxiliary problem (6.20). By Definition 6.0.1, $\mathbf{Ker}\ W$ is not empty, and the function $g(\mathbf{x})$ takes a zero on this subspace. Then we can divide our problem into two subproblems: minimization of a quadratic form with matrix $\gamma \cdot I$ on $\mathbf{Ker}\ W$ and minimization of a quadratic form with matrix $\lambda W + \gamma \cdot I$ on $(\mathbf{Ker}\ W)^{\perp}$. The complexity of the first problem is $\mathcal{O}(1)$. The second problem is $\lambda\lambda^{+}_{\min}(W)$-strongly convex, and if the Accelerated Gradient Method [397] is used to solve this subproblem, then the complexity is

$$\mathcal{O}\left(\sqrt{\frac{\gamma+\lambda\lambda_{\max}(W)}{\max\{\gamma,\lambda\lambda^{+}_{\min}(W)\}}}\log\tfrac{1}{\delta}\right). \tag{6.25}$$

This is the complexity for a single subproblem (6.20) solve, but (6.20) is solved (6.19) times. Overall, this means that the total number of calls of ∇g is:

$$N^{\text{comm}} = \mathcal{O}\left(\sqrt{\frac{L}{\mu}}\sqrt{\frac{\gamma+\lambda\lambda_{\max}(W)}{\max\{\gamma,\lambda\lambda^{+}_{\min}(W)\}}}\log\tfrac{1}{\varepsilon}\log\tfrac{1}{\delta}\right).$$

Noting that

$$\sqrt{\frac{\gamma+\lambda\lambda_{\max}(W)}{\max\{\gamma,\lambda\lambda^{+}_{\min}(W)\}}} = \min\left\{\sqrt{\frac{\gamma+\lambda\lambda_{\max}(W)}{\gamma}}, \sqrt{\frac{\gamma+\lambda\lambda_{\max}(W)}{\lambda\lambda^{+}_{\min}(W)}}\right\},$$

and taking $\gamma = 2L$, gives (6.23).

Case 2: $\lambda\lambda_{\max}(W) < L$. Here, let $f(x) = g(\mathbf{x})$ and $g(x) = f(\mathbf{x})$. Theorem 6.2.5 gives the complexity for the function $f(x) = g(\mathbf{x})$, i.e. the number of communications is

$$N^{\text{comm}} = \mathcal{O}\left(\sqrt{\frac{\lambda\lambda_{\max}(W)}{\mu}}\log\tfrac{1}{\varepsilon}\right) = \mathcal{O}\left(\min\left\{\sqrt{\frac{\lambda\lambda_{\max}(W)}{\mu}}, \sqrt{\frac{L}{\mu}\chi}\right\}\log\tfrac{1}{\varepsilon}\log\tfrac{1}{\delta}\right). \tag{6.26}$$

In last step we additionally use that $\chi \geq 1$. Also, it can be shown that δ in (6.22) satisfies the

condition in Theorem 6.2.5. If the Accelerated Gradient Method [397] is used to solve subproblem (6.20), the complexity for a single subproblem solve is again given by (6.25), and this subproblem is solved (6.26) times. Then we can find the number of calls for $\nabla f(\mathbf{x})$:

$$N^{\text{loc}} = \mathcal{O}\left(\sqrt{\frac{\lambda\lambda_{\max}(W)}{\mu}}\sqrt{\frac{L+\gamma}{\mu+\gamma}}\log\tfrac{1}{\varepsilon}\log\tfrac{1}{\delta}\right).$$

Taking $\gamma = 2\lambda\lambda_{\max}(W)$ gives (6.24).

Finally, combining the two cases establishes the theorem statement.

□

Remark 6.2.8. Note that in the centralized case (with a completely connected communication network) we have that $\chi = 1,\ \lambda_{\max}(W) = 1$ and our method converges with the following rates:

$$N^{comm} = \widetilde{\mathcal{O}}\left(\min\left\{\sqrt{\tfrac{\lambda}{\mu}}, \sqrt{\tfrac{L}{\mu}}\right\}\right), \quad N^{loc} = \widetilde{\mathcal{O}}\left(\sqrt{\tfrac{L}{\mu}}\right).$$

These bounds coincide with lower bounds for centralized PFL [461].

6.2.4 Experiments

In this subsection, we present several numerical experiments to demonstrate the practical advantages of the proposed approach for problem (6.4). We study logistic loss functions,

$$f_m(x_m) = \left(\frac{1}{n_m}\sum_{j=1}^{n_m}\log(1 + e^{-y_j^m a_j^m x_m})\right),$$

where $\{(a_j^m, y_j^m)\}_{j=1}^{n_m}$ is local dataset stored on each machine $m \in \{1, 2, \ldots, M\}$, $a_j^m \in \mathbb{R}^d$ represents the feature vector and $b_j^m \in \{-1, 1\}$ is the label. In the experiments, the power method was used to estimated the smoothness parameter of the objective function, as well as $\lambda_{\max}(W)$.

The experiments were performed on datasets from the LIBSVM [171] database. Table 6.4 shows the basic characteristics of the datasets used.

Table 6.4: The number of features and number of samples for each dataset used in the numerical experiments.

dataset	# features (d)	# samples
mushrooms	112	8,124
a9a	123	32,561
covtype.scale	54	581,012
rcv1.binary	47,236	20,242

In the experiments, three different network topologies were considered:

1. *Cyclic:* In this topology, devices are connected in a cycle, where each device is connected to it's two closest neighbours only. In this communication network, it takes $\sim \frac{n}{2}$ iterations to transmit information between two devices on opposite sides of the cycle.
2. *Grid:* Here devices are organized in a $\sqrt{n} \times \sqrt{n}$ grid, and are connected to their nearest neighbours.
3. *Erdos:* A random communication graph, also known as an Erdős-Rényi graph.

We used the **networkx** Python package. to generate random bi-directional graphs with the structures described above. As highlighted in the theory, the algorithm depends on the parameters λ, $\lambda_{\max}(W)$ and L. We ran several experiments with varying values of λ, where

$$\lambda = r\frac{L}{\lambda_{\max}(W)}, \qquad r > 0. \tag{6.27}$$

In Figure 6.2 we show the evolution of $\|\nabla f(x_t)\|$ for various selections of parameter T. Observe that the behaviour for $T \in \{2, 4, \ldots, 32\}$ is almost identical (in terms of the iterations of the algorithm), however, larger T requires additional rounds of communications. Therefore, in the following experiments we selected $T = 2$.

Effect of the regularization parameter

The main benefit of personalized federated learning is the ability to have slightly different local models, x_i, for each device i. The regularization term $\lambda x^T W x$ penalizes local models (i.e., the x_i's) that are different from their mean, where the parameter λ controls the emphasis placed on this penalty term. When λ is large, the problem (6.4) tends to a consensus/classical federated learning problem, because there is a large penalty for models that are different at distinct devices. The current work focuses on *personalized* federated learning, so here we consider the small λ regime.

Recall that for each problem instance considered, two iterations of accelerated gradient descent ($T = 2$) were used to provide an approximate solution to subproblem (6.20). The parameter λ is defined in (6.27), and several values of $r \in \{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\}$ were utilized. Let us stress that, as the number of local functions n increases, the matrix W changes; hence, $\lambda_{\max}(W)$ also varies. One can observe that, as expected, larger values of λ (that corresponds to larger values of r) lead to solutions $x \in \mathbb{R}^{nd}$ that exhibit a smaller value of the penalty term $x^T W x$. Figures 6.3 (cyclic network), 6.4 (grid network) and 6.5 (Erdős-Rényi network) illustrate the results of several numerical experiments.

Local Training Accuracy

In Figure 6.6 we demonstrate the main benefit of using PFL—namely, the ability for each device to have a slightly different local model, thereby capturing small differences in the local data. This is accomplished by selecting various values of λ and observing the effect that has on the training

accuracy over various local functions f_i. We plot the average accuracy over local accuracies (each using their own set of parameters). For the *mushrooms* and *a9a* datasets, the algorithm quickly achieved very good local accuracy for all local models. However, the *covtype* (more samples) and *rcv1* (more features) datasets were more challenging. For the *rcv1* dataset, we can see that initially (mainly due to the over-parametrization of the data) the local models achieve better accuracy (for smaller values of λ (r)), demonstrating the advantages of PFL.

Partial worker participation

One of the challenges of the FL setting is the fact that not all devices can always participate in all the communications [219, 481]. To simulate such a scenario, we conducted the following two experiments:

1. **Randomly dropping communication edge(s).** For each iteration, and each communication edge e, the edge is kept with probability p_e, or dropped with probability $1 - p_e$. The result is that the gossip matrix W is randomly modified at each iteration. In Figures 6.7 and 6.8 we demonstrate empirically that keeping some communication edges with probability $p_e \in \{1.0, 0.9, 0.5, 0.1\}$ only mildly affects the convergence.
2. **Randomly dropping the device(s) from communication.** In this case, a subset of devices is randomly selected. In particular, at each iteration, a device is kept with probability p_d, and excluded/dropped with probability $1 - p_d$. As before, the effect is that the gossip matrix W is randomly modified at each iteration. In Figures 6.9 and 6.10 we demonstrate empirically that keeping only some devices with probability $p_d \in \{1.0, 0.9, 0.5, 0.1\}$ only mildly affects the convergence.

The benefit of personalized training

In Section 6.2.4 we discussed the case when $\lambda \leq \frac{L}{\lambda_{\max}(W)}$ that allows for more personalization of local models. Note that we use Algorithm 55 with different settings for $f(x)$ and $g(x)$ of (6.5) depending on the value of λ. In Figure 6.11 we investigate the behavior of Algorithm 55 for $\lambda = r\frac{L}{\lambda_{\max}(W)}$ with $r \in \{0.125, 16\}$. Note that a larger value of λ (r) corresponds to larger penalization if the model deviates from the mean $(x^T W x)$; consequently, this allows less personalization.

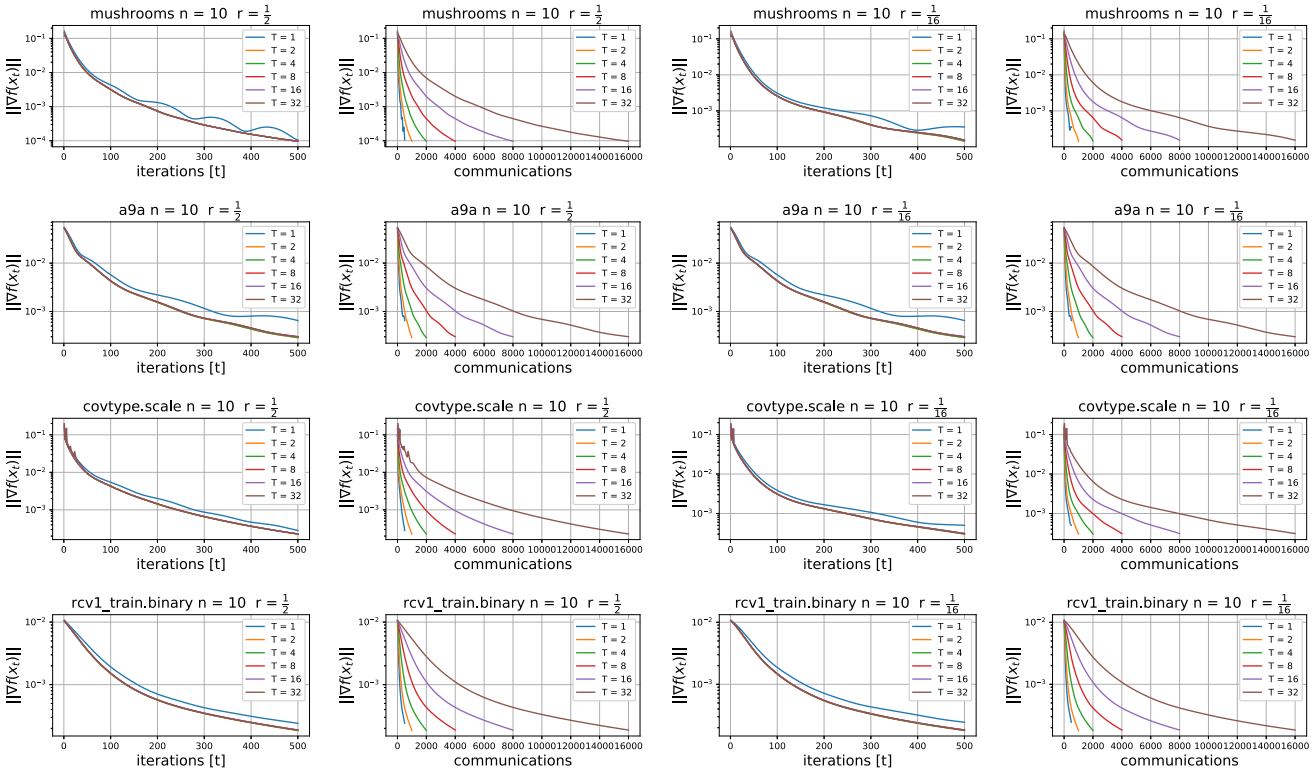


Figure 6.2: Evolution of $\|\nabla f(x_t)\|$ for different datasets, regularization parameter and different level of solving the subproblem (6.20) (larger T means we optimize the subproblem better).

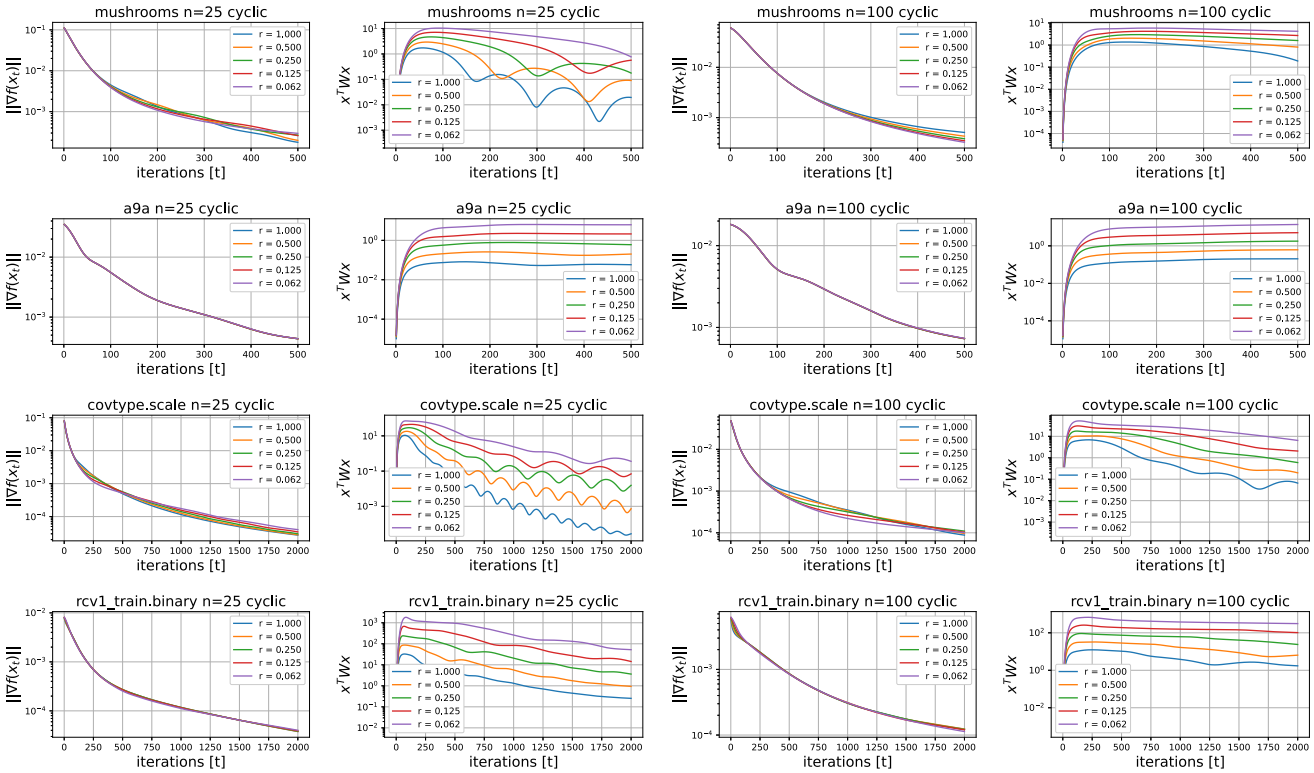


Figure 6.3: Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, levels of λ and number of devices $n \in \{25, 100\}$ with cyclic network.

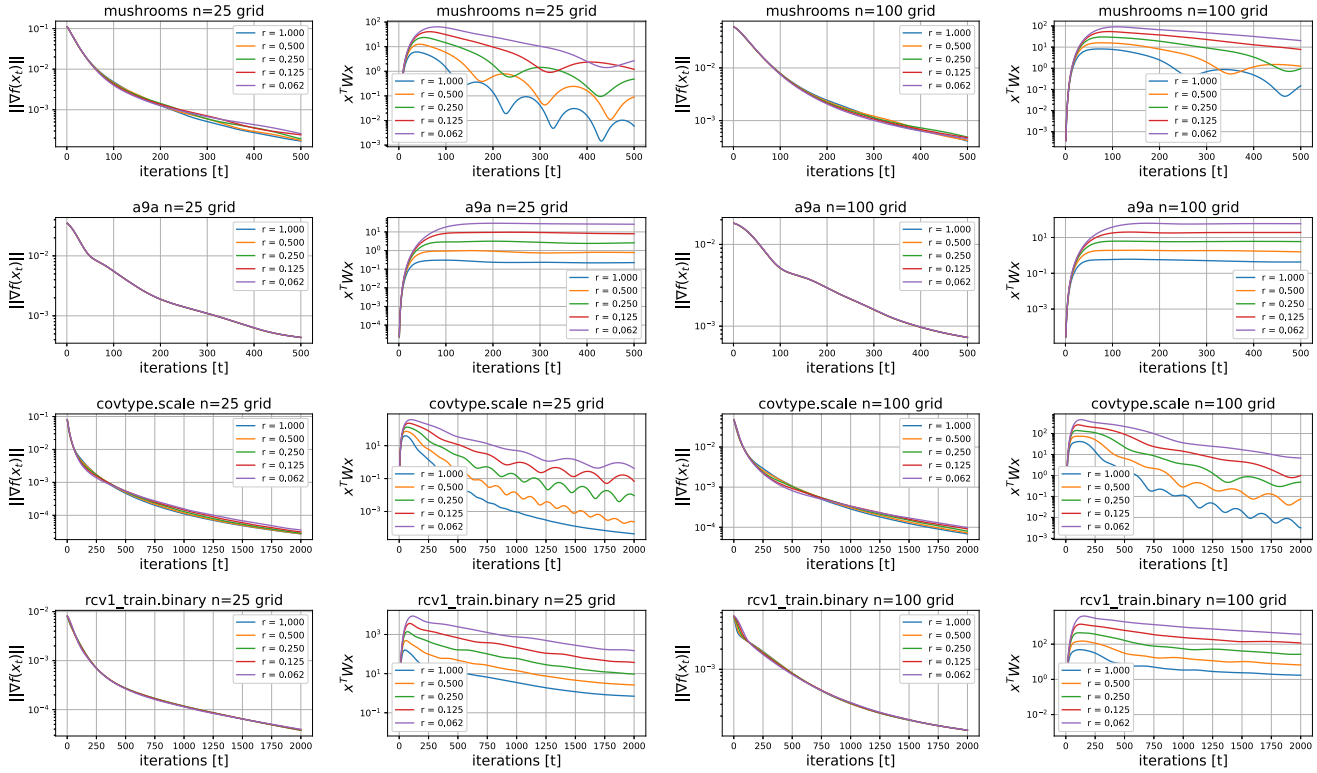


Figure 6.4: Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, levels of λ and number of devices $n \in \{25, 100\}$ with grid network.

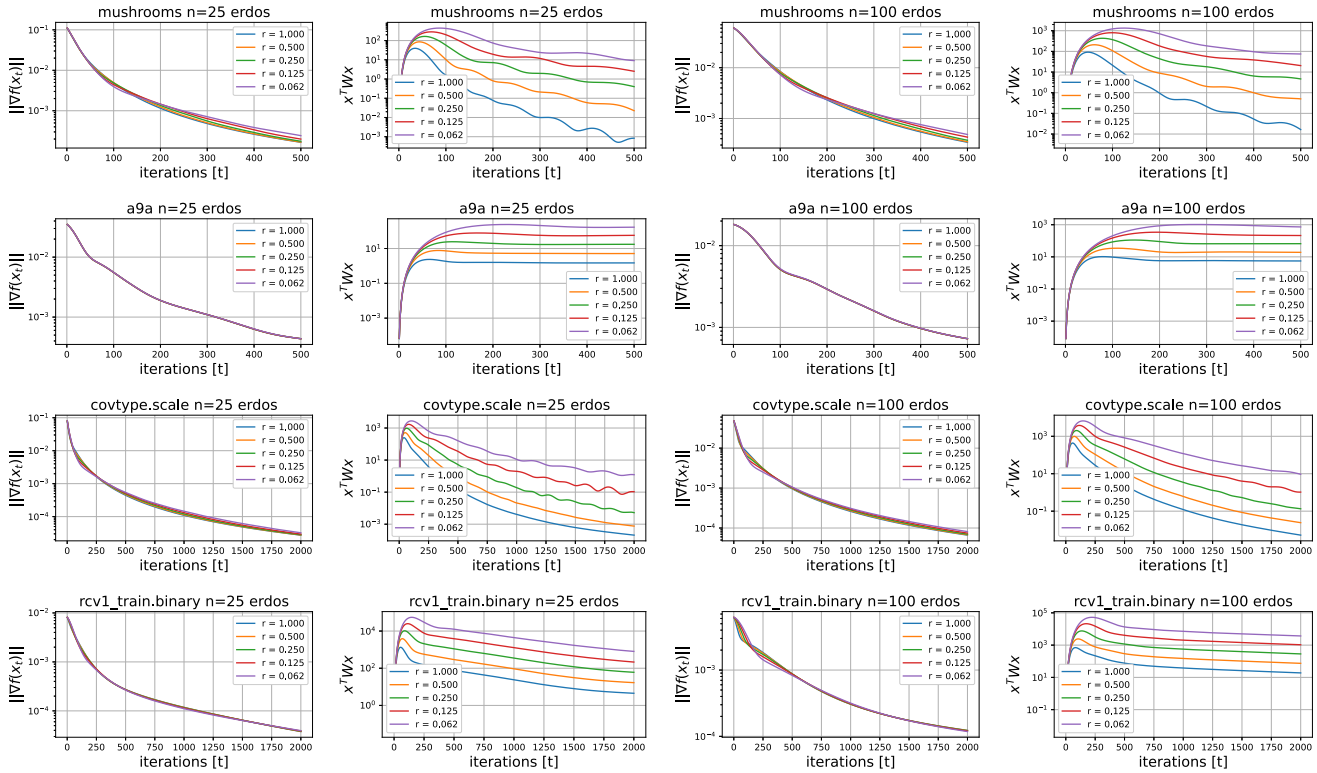


Figure 6.5: Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, levels of λ and number of devices $n \in \{25, 100\}$ with erdos network.

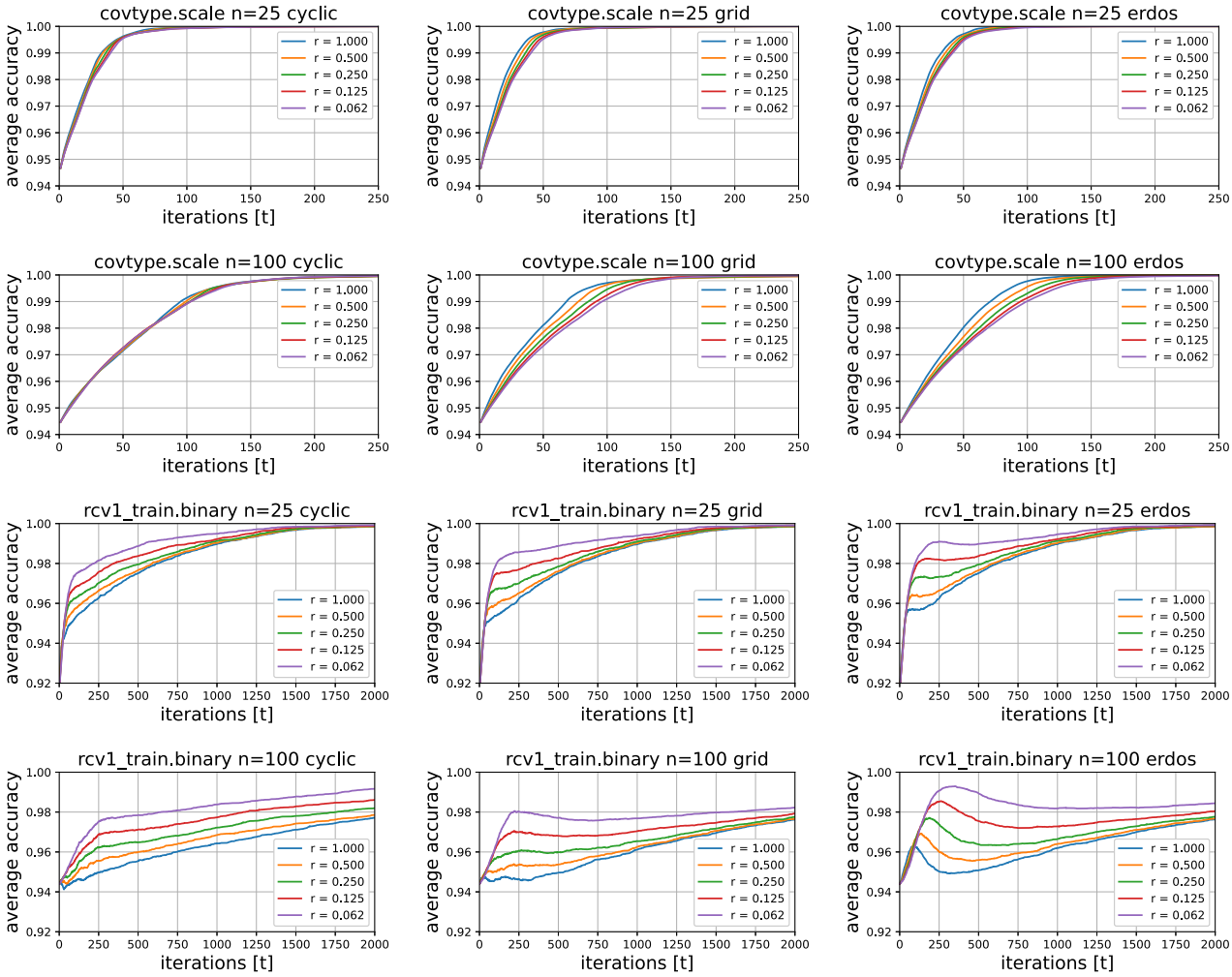


Figure 6.6: Comparison of average accuracy of local models on local data for various communication networks and levels of λ (r).

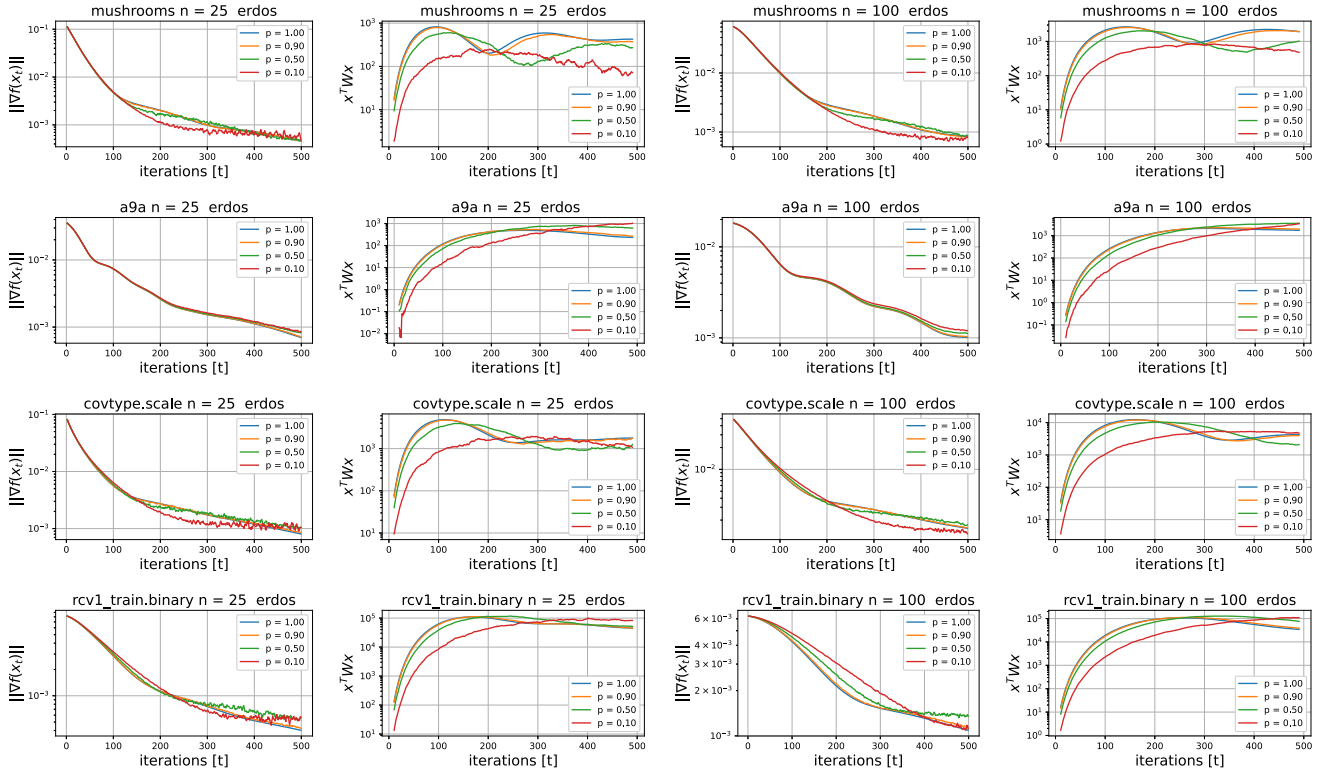


Figure 6.7: Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, number of devices $n \in \{25, 100\}$ and different probability of keeping the communication edge p with the erdos network.

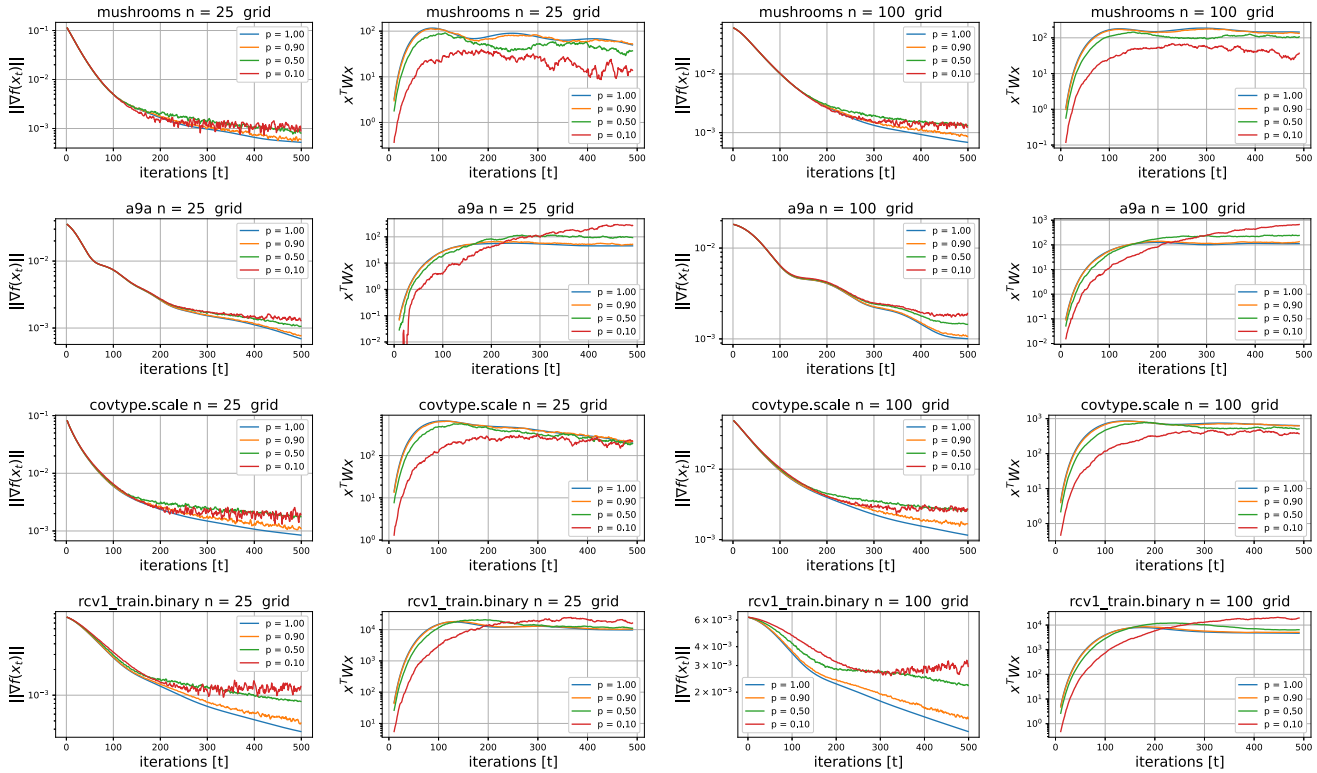


Figure 6.8: Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, number of devices $n \in \{25, 100\}$ and different probability of keeping the communication edge p with grid network.

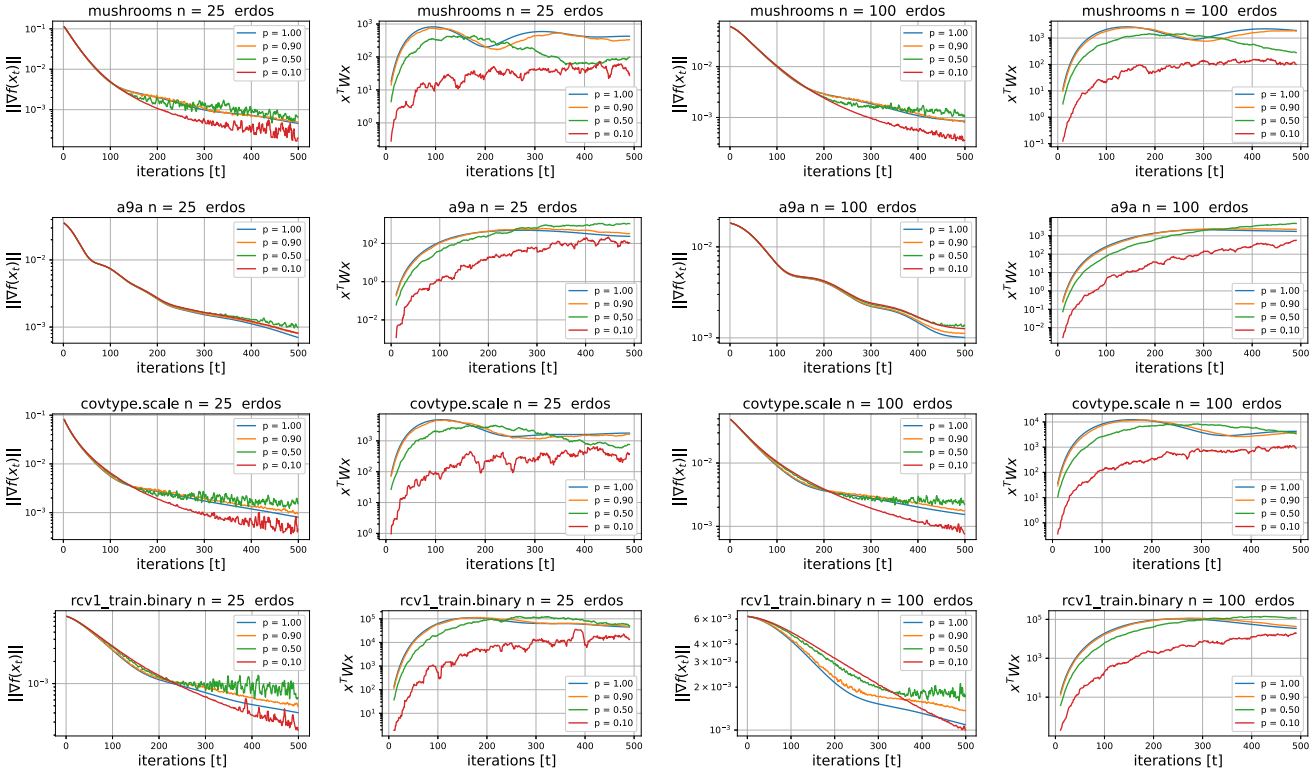


Figure 6.9: Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, number of devices $n \in \{25, 100\}$ and different probability p of keeping the device in the communication with erdos network.

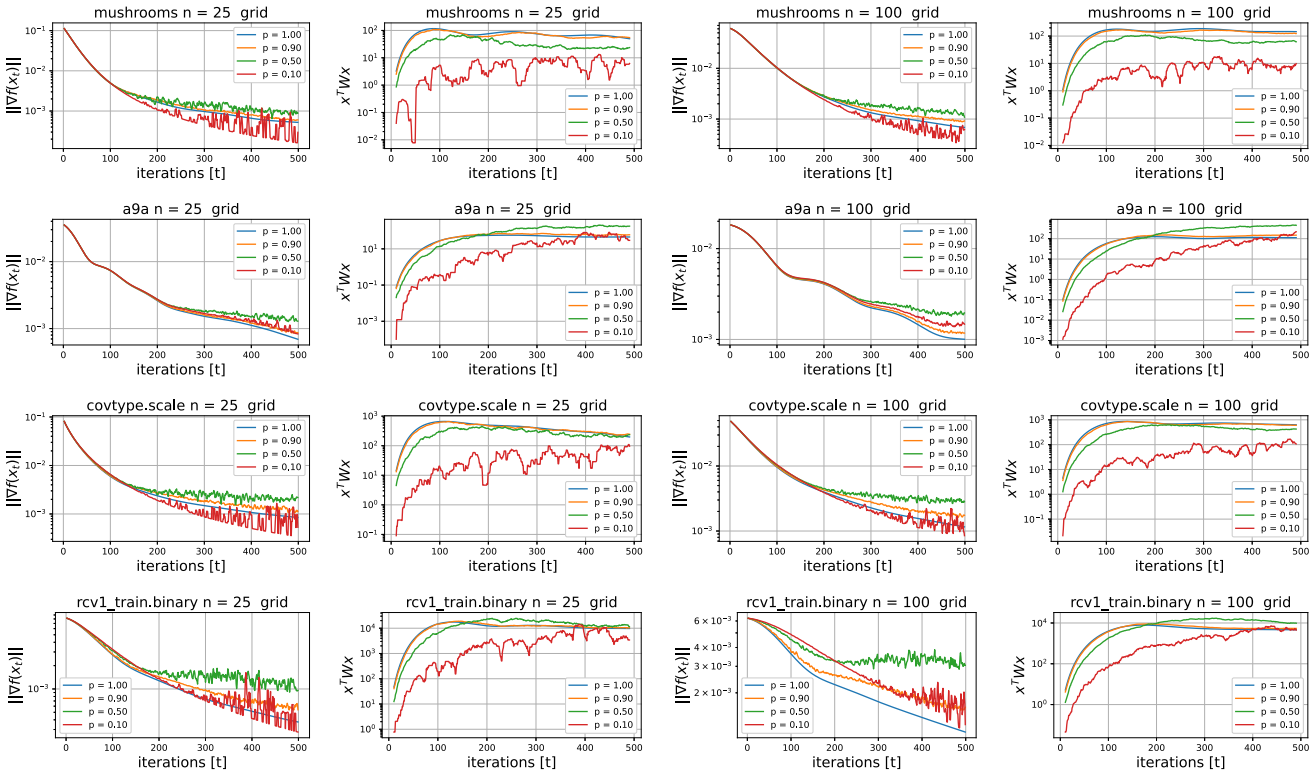


Figure 6.10: Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, number of devices $n \in \{25, 100\}$ and different probability p of keeping the device in the communication with grid network.

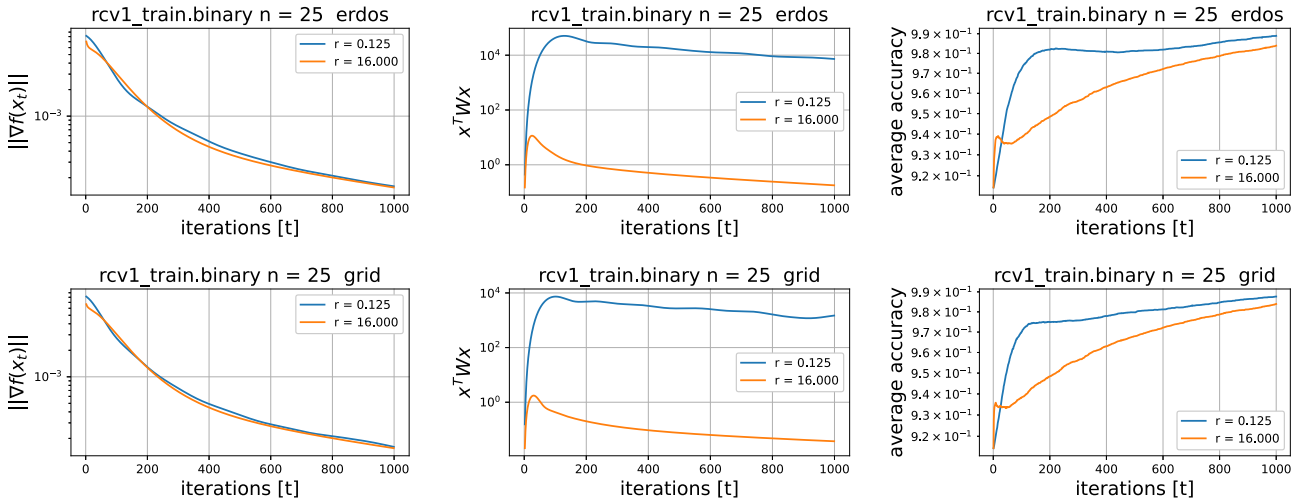


Figure 6.11: Evolution of $\|\nabla f(x_t)\|$, $x_t^T W x_t$ and average accuracy of local models for rcv1 datasets and erdos and grid network. We compare with two levels of regularization: low with $\lambda = 0.125 \cdot \frac{L}{\lambda_{\max}(W)}$ and high with $\lambda = 16 \cdot \frac{L}{\lambda_{\max}(W)}$. Note that, although both regularization values give comparable $\|\nabla f(x_t)\|$, the average accuracy for the case with smaller penalization is better.

Chapter 7

Conclusion

The thesis provided a comprehensive overview of various plots from numerical optimization methods, examined multiple classes of optimization problems from the perspectives of lower bounds and optimal algorithms, offered a unified view of entire families of methods, and also closely analyzed specific plots. The main results of the dissertation can be summarized as follows:

- Lower and upper bounds for stochastic distributed saddle point problems and variational inequalities were obtained. The thesis considered various formulations of these problems: from the perspective of the assumption on the stochasticity of the target function/operator and from the point of view of the communication organization.
- Local methods for distributed variational inequalities were developed. The analysis of local methods under the similarity among local operators on the devices were provided.
- Methods with compression for distributed variational inequalities were developed. The thesis provided algorithms that utilize unbiased and more practical biased compression operators. The thesis also presented methods that, in addition to compression, incorporate the similarity of local operators.
- A unified view on the stochastic methods for cocoercive variational inequalities was obtained. The thesis considered the SGDA method and proposed a universal theorem that allows for the analysis of a whole family of methods surrounding SGDA.
- Algorithms with optimal batching for solving stochastic variational inequalities of finite sum form were developed.
- A unified view on the stochastic Frank-Wolfe method was obtained. The thesis proposed a universal theorem that allows for the analysis of a whole family of methods based on the Frank-Wolfe concept.
- The first variance-reduced method that requires neither significant additional memory overhead nor regular computation of the full gradients of the target function were developed.
- Stochastic methods with Markovian stochasticity in the gradient were developed. The thesis proposed accelerated methods that achieve advanced convergence estimates for smooth minimization problems. In addition, lower bounds guaranteeing the optimality of the proposed methods were presented.

- A theoretical analysis of distributed optimization methods using biased compression operators was obtained.
- Several sliding algorithms for solving distributed optimization problems with various features in a communication-efficient manner were developed. The thesis proposed methods for dealing with personalized formulations and with non-smooth problems, with access only to zero-order information.

Directions for future research

If we discuss the expansion of research around the results of the dissertation, the following directions can be highlighted:

- Firstly, one can identify a significant area for applying the theoretical findings from the thesis to contemporary machine learning problems. In the past year, there was an increase in interest in optimization techniques through the use of the LMO oracle [482, 483], which is employed in the Frank–Wolfe method discussed in Chapter 4. Additionally, game-theoretic formulations of learning problems are of particular interest [484]. The machinery of variational inequalities developed in Chapters 2 and 3 provides a solid foundation.
- An important theoretical direction is the work in the field of special unusual classes of problems. For example, these include non-independent stochasticity. The results of Chapter 5 can be a good starting point, but even they contain a number of open questions and opportunities for improved understanding. It is clear that the mentioned Markovian methods can be investigated from various angles: the non-Euclidean case, the application of the Markovian property in compression operators, and other ideas that have already been used in the independent case.
- If we talk about the various sliding techniques explored in both Chapter 2 and Chapter 6, then the research horizon also seems wide. This is due to the fact that the popularity sum-type tasks in machine learning grows, and many specialized tasks have this type. As we know, slidings help to properly look at the features of these tasks and make methods of their solution more effective.

Author's publications

[1] Beznosikov Aleksandr, Gorbunov Eduard, Gasnikov Alexander. Derivative-free method for composite optimization with applications to decentralized distributed optimization // IFAC-PapersOnLine. — 2020. — Vol. 53, no. 2. — P. 4038–4043.

[2] Beznosikov Aleksandr, Rogozin Alexander, Kovalev Dmitry, Gasnikov Alexander. Near-Optimal Decentralized Algorithms for Saddle Point Problems over Time-Varying Networks // Optimization and Applications. — 2021. — Vol. 13078. — P. 246–257.

[3] Beznosikov Aleksandr, Gasnikov Alexander. Compression and data similarity: Combination of two techniques for communication-efficient solving of distributed variational inequalities // Lecture Notes in Computer Science. — 2022. — Vol. 13781. — P. 151–162.

[4] Beznosikov Aleksandr, Gasnikov Alexander. Sarah-based variance-reduced algorithm for stochastic finite-sum cocoercive variational inequalities // Springer Optimization and Its Applications. — 2023. — Vol. 202. — P. 47–57.

[5] Beznosikov Aleksandr Nikolaevich, Gasnikov Alexander Vladimirovich, Zainullina Karina E et al. A unified analysis of variational inequality methods: variance reduction, sampling, quantization, and coordinate descent // Computational Mathematics and Mathematical Physics. — 2023. — Vol. 63, no. 2. — P. 147–174.

[6] Beznosikov Aleksandr, Horváth Samuel, Richtárik Peter, Safaryan Mher. On biased compression for distributed learning // Journal of Machine Learning Research. — 2023. — Vol. 24, no. 276. — P. 1–50.

[7] Beznosikov Aleksandr, Takáč Martin. Random-reshuffled SARAH does not need full gradient computations // Optimization Letters. — 2024. — Vol. 18, no. 3. — P. 727–749.

[8] Beznosikov Aleksandr, Samokhin Valentin, Gasnikov Alexander. Distributed saddle point problems: lower bounds, near-optimal and robust algorithms // Optimization Methods and Software. — 2025. — P. 1–18.

[9] Beznosikov Aleksandr, Stepanov Ivan, Voronov Artyom, Gasnikov Alexander. One-point feedback for composite optimization with applications to distributed and federated learning // Optimization Methods and Software. — 2025. — P. 1–16.

[10] Rogozin Alexander, Beznosikov Aleksandr, Dvinskikh Darina et al. Decentralized saddle point problems via non-Euclidean mirror prox // Optimization Methods and Software. — 2024. — P. 1–26.

[11] Gorbunov Eduard, Rogozin Alexander, Beznosikov Aleksandr et al. Recent theoretical advances

in decentralized distributed convex optimization // High-Dimensional Optimization and Probability: With a View Towards Data Science. — 2022. — P. 253–325.

[12] Sadiev Abdurakhmon, Borodich Ekaterina, Beznosikov Aleksandr et al. Decentralized personalized federated learning: Lower bounds and optimal algorithm for all personalization modes // EURO Journal on Computational Optimization. — 2022. — Vol. 10. — P. 100041.

[13] Pichugin Alexander, Pechin Maksim, Beznosikov Aleksandr et al. Optimal analysis of method with batching for monotone stochastic finite-sum variational inequalities // Doklady Mathematics. — 2023. — Vol. 108, no. 2. — P. S348–S359.

[14] Pichugin Alexander, Pechin Maksim, Beznosikov Aleksandr et al. Method with batching for stochastic finite-sum variational inequalities in non-Euclidean setting // Chaos, Solitons & Fractals. — 2024. — Vol. 187. — P. 115396.

[15] Veprikov Andrey, Bogdanov Alexander, Minashkin Vladislav, Beznosikov Aleksandr. New aspects of black box conditional gradient: Variance reduction and one point feedback // Chaos, Solitons & Fractals. — 2024. — Vol. 189. — P. 115654.

[16] Beznosikov Aleksandr, Scutari Gesualdo, Rogozin Alexander, Gasnikov Alexander. Distributed saddle-point problems under data similarity // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 8172–8184.

[17] Beznosikov Aleksandr, Dvurechenskii Pavel, Koloskova Anastasiia et al. Decentralized local stochastic extra-gradient for variational inequalities // Advances in Neural Information Processing Systems. — 2022. — Vol. 35. — P. 38116–38133.

[18] Beznosikov Aleksandr, Richtárik Peter, Diskin Michael et al. Distributed methods with compressed communication for solving variational inequalities, with theoretical guarantees // Advances in Neural Information Processing Systems. — 2022. — Vol. 35. — P. 14013–14029.

[19] Beznosikov Aleksandr, Gorbunov Eduard, Berard Hugo, Loizou Nicolas. Stochastic gradient descent-ascent: Unified theory and new efficient methods // Proceedings of the 26th International Conference on Artificial Intelligence and Statistics, PMLR. — 2023. — P. 172–235.

[20] Beznosikov Aleksandr, Takac Martin, Gasnikov Alexander. Similarity, Compression and Local Steps: Three Pillars of Efficient Communications for Distributed Variational Inequalities // Advances in Neural Information Processing Systems. — 2023. — Vol. 36. — P. 28663–28677.

[21] Beznosikov Aleksandr, Samsonov Sergey, Sheshukova Marina et al. First order methods with markovian noise: from acceleration to variational inequalities // Advances in Neural Information Processing Systems. — 2023. — Vol. 36. — P. 44820–44835.

[22] Beznosikov Aleksandr, Dobre David, Gidel Gauthier. Sarah Frank-Wolfe: Methods for Constrained Optimization with Best Rates and Practical Features // Proceedings of the 41st International Conference on Machine Learning, PMLR. — 2024. — P. 3725–3750.

[23] Kovalev Dmitry, Beznosikov Aleksandr, Sadiev Abdurakhmon et al. Optimal algorithms for decentralized stochastic variational inequalities // Advances in Neural Information Processing Systems. — 2022. — Vol. 35. — P. 31073–31088.

[24] Kovalev Dmitry, Beznosikov Aleksandr, Borodich Ekaterina et al. Optimal gradient sliding and its application to optimal distributed optimization under similarity // Advances in Neural Information Processing Systems. — 2022. — Vol. 35. — P. 33494–33507.

[25] Nazykov Ruslan, Shestakov Aleksandr, Solodkin Vladimir et al. Stochastic frank-wolfe: Unified analysis and zoo of special cases // Proceedings of The 27th International Conference on Artificial Intelligence and Statistics, PMLR. — 2024. — P. 4870–4878.

References

[26] Cauchy Augustin. Methode generale pour la resolution des systemes d'equations simultanees // Comptes rendus hebdomadaires des séances de l'Académie des sciences. — 1847. — Vol. 55. — P. 536–538.

[27] Newton Isaac. De analysi per aequationes numero terminorum infinitas. — 1711.

[28] L. V. Kantorovich. Mathematical methods of production organization and planning. — LSU Publishing House, 1939.

[29] Dantzig George B. Origins of the simplex method // A history of scientific computing. — 1990. — P. 141–151.

[30] Curry Haskell B. The method of steepest descent for non-linear minimization problems // Quarterly of Applied Mathematics. — 1944. — Vol. 2, no. 3. — P. 258–261.

[31] Polyak Boris T. Gradient methods for the minimisation of functionals // USSR Computational Mathematics and Mathematical Physics. — 1963. — Vol. 3, no. 4. — P. 864–878.

[32] Polyak B.T. Some methods of speeding up the convergence of iteration methods // USSR Computational Mathematics and Mathematical Physics. — 1964. — Vol. 4, no. 5. — P. 1–17.

[33] Nemirovski Arkadi, Yudin David B. Problem complexity and method efficiency in optimization. — Wiley Interscience, 1983.

[34] Nesterov Yurii E. A method for solving the convex programming problem with convergence rate O (1/kˆ 2) // Dokl. akad. nauk Sssr. — 1983. — Vol. 269, no. 3. — P. 543–547.

[35] Nesterov Yurii, Nemirovskii Arkadii. Interior-point polynomial algorithms in convex programming. — SIAM, 1994.

[36] Verbraeken Joost, Wolting Matthijs, Katzy Jonathan et al. A Survey on Distributed Machine Learning // ACM Computing Surveys (CSUR). — 2019. — Vol. 53, no. 2.

[37] McMahan Brendan, Moore Eider, Ramage Daniel, Agüera y Arcas Blaise. Federated Learning of Deep Networks using Model Averaging // arXiv preprint arXiv:1602.05629. — 2016. — URL: `https://arxiv.org/pdf/1602.05629v1/1000` (visited on June 9, 2025).

[38] Li Tian, Sahu Anit Kumar, Zaheer Manzil et al. Federated optimization in heterogeneous networks // Proceedings of Machine learning and systems. — 2020. — Vol. 2. — P. 429–450.

[39] Mishchenko Konstantin, Malinovsky Grigory, Stich Sebastian, Richtárik Peter. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! // International Conference on Machine Learning. — 2022. — Vol. 162. — P. 15750–15769.

[40] Karimireddy Sai Praneeth, Kale Satyen, Mohri Mehryar et al. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning // Proceedings of the 37th International Conference on Machine Learning. — 2020. — Vol. 119. — P. 5132–5143.

[41] Hendrikx Hadrien, Xiao Lin, Bubeck Sebastien et al. Statistically preconditioned accelerated gradient method for distributed optimization // International Conference on Machine Learning. — 2020. — Vol. 119. — P. 4151–4150.

[42] Vogels Thijs, Karinireddy Sai Praneeth, Jaggi Martin. PowerSGD: Practical low-rank gradient compression for distributed optimization // Advances In Neural Information Processing Systems 32 (Nips 2019). — 2019. — Vol. 18. — P. 14192–14201.

[43] Alistarh Dan, Grubic Demjan, Li Jerry et al. QSGD: Communication-efficient SGD via gradient quantization and encoding // Advances in Neural Information Processing Systems. — 2017. — P. 1709–1720.

[44] Seide Frank, Fu Hao, Droppo Jasha et al. 1-Bit Stochastic Gradient Descent and Application to Data-Parallel Distributed Training of Speech DNNs // Interspeech 2014. — 2014. — Vol. 1. — P. 1058–1062.

[45] Richtárik Peter, Sokolov Igor, Fatkhullin Ilyas. EF21: A New, Simpler, Theoretically Better, and Practically Faster Error Feedback // 35nd Conference on Neural Information Processing Systems. — 2021. — Vol. 6. — P. 4384–4396.

[46] Robbins Herbert, Monro Sutton. A Stochastic Approximation Method // The Annals of Mathematical Statistics. — 1951. — Vol. 22, no. 3. — P. 400 – 407.

[47] Moulines Eric, Bach Francis. Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning // Advances in Neural Information Processing Systems. — 2011. — Vol. 24. — P. 451–460.

[48] Lan Guanghui. An optimal method for stochastic composite optimization // Mathematical Programming. — 2012. — Vol. 133, no. 1-2. — P. 365–397.

[49] Taylor Adrien, Bach Francis. Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions // Conference on Learning Theory. — 2019. — Vol. 99. — P. 2934–2992.

[50] Ustimenko Aleksei, Beznosikov Aleksandr. Ito Diffusion Approximation of Universal Ito Chains for Sampling, Optimization and Boosting // The Twelfth International Conference on Learning Representations. — 2024. — URL=https://openreview.net/pdf?id=fjpfCOV4ru (visited on June 9, 2025).

[51] Shalev-Shwartz Shai, Ben-David Shai. Understanding machine learning: From theory to algorithms. — Cambridge university press, 2014.

[52] Johnson Rie, Zhang Tong. Accelerating stochastic gradient descent using predictive variance

reduction // Advances in neural information processing systems. — 2013. — Vol. 26. — P. 315–323.

[53] Nguyen Lam M, Liu Jie, Scheinberg Katya, Takáč Martin. SARAH: A novel method for machine learning problems using stochastic recursive gradient // International Conference on Machine Learning. — 2017. — Vol. 70. — P. 2613–2621.

[54] Li Zhize, Bao Hongyan, Zhang Xiangliang, Richtárik Peter. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization // International Conference on Machine Learning. — 2021. — Vol. 139. — P. 6286–6295.

[55] Defazio Aaron, Bach Francis, Lacoste-Julien Simon. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives // Advances in neural information processing systems. — 2014. — Vol. 27. — P. 1646–1654.

[56] Allen-Zhu Zeyuan. Katyusha: The first direct acceleration of stochastic gradient methods // Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing. — 2017. — P. 1200–1205.

[57] Medyakov Daniil, Molodtsov Gleb, Chezhegov Savelii et al. Variance Reduction Methods Do Not Need to Compute Full Gradients: Improved Efficiency through Shuffling // arXiv preprint arXiv:2502.14648. — 2025. — URL: https://arxiv.org/pdf/2502.14648 (visited on June 9, 2025).

[58] Defazio Aaron, Bottou Léon. On the ineffectiveness of variance reduced optimization for deep learning // Advances in Neural Information Processing Systems. — 2019. — Vol. 32. — P. 1744–1754.

[59] Hanzely Filip, Mishchenko Konstantin, Richtárik Peter. SEGA: Variance reduction via gradient sketching // Advances in Neural Information Processing Systems. — 2018. — Vol. 31. — P. 1646—-1654.

[60] Mishchenko Konstantin, Gorbunov Eduard, Takáč Martin, Richtárik Peter. Distributed learning with compressed gradient differences // arXiv preprint arXiv:1901.09269. — 2019. — URL: https://arxiv.org/pdf/1901.09269 (visited on June 9, 2025).

[61] Gorbunov Eduard, Danilova Marina, Gasnikov Alexander. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 15042–15053.

[62] Kornilov Nikita, Zmushko Philip, Semenov Andrei et al. Sign Operator for Coping with Heavy-Tailed Noise: High Probability Convergence Bounds with Extensions to Distributed Optimization and Comparison Oracle // arXiv preprint arXiv:2502.07923. — 2025. — URL: https://arxiv.org/pdf/2502.07923 (visited on June 9, 2025).

[63] Simsekli Umut, Sagun Levent, Gurbuzbalaban Mert. A tail-index analysis of stochastic

gradient noise in deep neural networks // International Conference on Machine Learning. — 2019. — Vol. 97. — P. 5827–5837.

[64] Davis Damek, Drusvyatskiy Dmitriy, Xiao Lin, Zhang Junyu. From low probability to high confidence in stochastic convex optimization // Journal of machine learning research. — 2021. — Vol. 22, no. 49. — P. 1–38.

[65] Vaswani Sharan, Bach Francis, Schmidt Mark. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron // The 22nd international conference on artificial intelligence and statistics. — 2019. — Vol. 89. — P. 1195–1204.

[66] Gorbunov Eduard, Hanzely Filip, Richtárik Peter. A Unified Theory of SGD: Variance Reduction, Sampling, Quantization and Coordinate Descent // The 23rd International Conference on Artificial Intelligence and Statistics. — 2020. — Vol. 108. — P. 680–690.

[67] Nesterov Yu. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems // SIAM Journal on Optimization. — 2012. — Vol. 22, no. 2. — P. 341–362.

[68] Richtárik Peter, Takáč Martin. Parallel coordinate descent methods for big data optimization // Mathematical Programming. — 2016. — Vol. 156, no. 1-2. — P. 433–484.

[69] Nesterov Yurii, Spokoiny Vladimir. Random gradient-free minimization of convex functions // Foundations of Computational Mathematics. — 2017. — Vol. 17, no. 2. — P. 527–566.

[70] Gasnikov Alexander, Dvinskikh Darina, Dvurechensky Pavel et al. Randomized gradient-free methods in convex optimization // Encyclopedia of Optimization. — 2023. — P. 1–15.

[71] Scutari Gesualdo, Palomar Daniel P, Facchinei Francisco, Pang Jong-Shi. Convex optimization, game theory, and variational inequality theory // IEEE Signal Processing Magazine. — 2010. — Vol. 27, no. 3. — P. 35–49.

[72] Morgenstern Oskar, Von Neumann John. Theory of games and economic behavior. — Princeton university press, 1953.

[73] Stampacchia G. Formes bilineaires coercitives sur les ensembles convexes // Académie des Sciences de Paris. — 1964. — Vol. 258. — P. 4413–4416.

[74] Browder F. E. Existence and approximation of solutions of nonlinear variational inequalities // Proceedings of the National Academy of Sciences. — 1966. — Vol. 56, no. 4. — P. 1080–1086.

[75] Beznosikov Aleksandr, Polyak Boris, Gorbunov Eduard et al. Smooth monotone stochastic variational inequalities and saddle point problems: A survey // European Mathematical Society Magazine. — 2023. — no. 127. — P. 15–28.

[76] Bakushinskii A.B., Polyak B.T. On the solution of variational inequalities // Soviet Math. Doklady. — 1974. — Vol. 15, no. 6. — P. 1705–1710.

[77] Martinet Bernard. Regularisation, d'inéquations variationelles par approximations succe-

sives // Revue Francaise d'informatique et de Recherche operationelle. — 1970. — Vol. 4, no. 3. — P. 154–158.

[78] Korpelevich G. M. The extragradient method for finding saddle points and other problems // Matecon. — 1976. — Vol. 12. — P. 747–756.

[79] Popov Leonid Denisovich. A modification of the Arrow-Hurwicz method for search of saddle points // Mathematical notes of the Academy of Sciences of the USSR. — 1980. — Vol. 28, no. 5. — P. 845–848.

[80] Nemirovski Arkadi. Prox-method with rate of convergence O (1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems // SIAM Journal on Optimization. — 2004. — Vol. 15, no. 1. — P. 229–251.

[81] Gidel Gauthier, Berard Hugo, Vincent Pascal, Lacoste-Julien Simon. A Variational Inequality Perspective on Generative Adversarial Nets // arXiv preprint arXiv:1802.10551. — 2019. — URL: `https://arxiv.org/pdf/1802.10551` (visited on June 9, 2025).

[82] Neumann J., Morgenstern O. Theory of games and economic behavior. — Princeton University Press, 1944.

[83] Harker P. T., Pang J.-S. Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications // Mathematical programming. — 1990. — Vol. 48, no. 2. — P. 161–220.

[84] Facchinei Francisco, Pang Jong-Shi. Finite-Dimensional Variational Inequalities and Complementarity Problems. — 2003.

[85] Juditsky Anatoli, Nemirovski Arkadi, Tauvel Claire. Solving variational inequalities with stochastic mirror-prox algorithm // Stochastic Systems. — 2011. — Vol. 1, no. 1. — P. 17–58.

[86] Tseng Paul. A modified forward-backward splitting method for maximal monotone mappings // SIAM Journal on Control and Optimization. — 2000. — Vol. 38, no. 2. — P. 431–446.

[87] Joachims Thorsten. A Support Vector Method for multivariate performance measures // ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning. — 2005. — Vol. 119. — P. 377–384.

[88] Bach Francis, Jenatton Rodolphe, Mairal Julien, Obozinski Guillaume. Optimization with sparsity-inducing penalties // arXiv preprint arXiv:1108.0775. — 2011. — URL: `https://arxiv.org/pdf/1108.0775` (visited on June 9, 2025).

[89] Xu Linli, Neufeld James, Larson Bryce, Schuurmans Dale. Maximum Margin Clustering // Advances in Neural Information Processing Systems. — 2004. — Vol. 17.

[90] Bach Francis, Mairal Julien, Ponce Jean. Convex sparse matrix factorizations // arXiv preprint arXiv:0812.1869. — 2008. — URL=`https://arxiv.org/pdf/0812.1869` (visited on June 9, 2025).

[91] Esser Ernie, Zhang Xiaoqun, Chan Tony F. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science // SIAM Journal on Imaging Sciences. — 2010. — Vol. 3, no. 4. — P. 1015–1046.

[92] Chambolle Antonin, Pock Thomas. A first-order primal-dual algorithm for convex problems with applications to imaging // Journal of mathematical imaging and vision. — 2011. — Vol. 40, no. 1. — P. 120–145.

[93] Ben-Tal Aharon, Ghaoui Laurent El, Nemirovski Arkadi. Robust Optimization. — Princeton University Press, 2009.

[94] Omidshafiei Shayegan, Pazis Jason, Amato Christopher et al. Deep Decentralized Multi-task Multi-Agent Reinforcement Learning under Partial Observability // Proceedings of the 34th International Conference on Machine Learning (ICML). — 2017. — Vol. 70. — P. 2681–2690.

[95] Jin Yujia, Sidford Aaron. Efficiently solving MDPs with stochastic mirror descent // International Conference on Machine Learning. — 2020. — P. 4890–4900.

[96] Madry Aleksander, Makelov Aleksandar, Schmidt Ludwig et al. Towards Deep Learning Models Resistant to Adversarial Attacks // International Conference on Learning Representations. — 2018. — Vol. 6. — P. 4138–4160.

[97] Goodfellow Ian J., Pouget-Abadie Jean, Mirza Mehdi et al. Generative Adversarial Nets // Neural Information Processing Systems. — 2014. — Vol. 1. — P. 2672–2680.

[98] Daskalakis Constantinos, Ilyas Andrew, Syrgkanis Vasilis, Zeng Haoyang. Training gans with optimism // arXiv preprint arXiv:1711.00141. — 2017. — URL: `https://arxiv.org/pdf/1711.00141` (visited on June 9, 2025).

[99] Mertikopoulos Panayotis, Lecouat Bruno, Zenati Houssam et al. Optimistic mirror descent in saddle-point problems: Going the extra(-gradient) mile // International Conference on Learning Representations. — 2019. — Vol. 8. — P. 5925–5947.

[100] Chavdarova Tatjana, Gidel Gauthier, Fleuret Francois, Lacoste-Julien Simon. Reducing Noise in GAN Training with Variance Reduced Extragradient // Advances in Neural Information Processing Systems 32 (NeurIPS). — 2019. — Vol. 1. — P. 393–403.

[101] Liang Tengyuan, Stokes James. Interaction Matters: A Note on Non-asymptotic Local Convergence of Generative Adversarial Networks // Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics. — 2019. — Vol. 89. — P. 907–915.

[102] Peng Wei, Dai Yu-Hong, Zhang Hui, Cheng Lizhi. Training GANs with centripetal acceleration // Optimization Methods and Software. — 2020. — Vol. 35, no. 5. — P. 955–973.

[103] Arora Sanjeev, Cohen Nadav, Hazan Elad. On the Optimization of Deep Networks: Implicit Acceleration by Overparameterization // Proceedings of the 35th International Conference on Machine Learning (ICML). — 2018. — Vol. 80. — P. 244–253.

[104] Gallager Robert G, Humblet Pierre A, Spira Philip M. A distributed algorithm for minimum-weight spanning trees // ACM Transactions on Programming Languages and systems (TOPLAS). — 1983. — Vol. 5, no. 1. — P. 66–77.

[105] Konečnỳ Jakub, McMahan H Brendan, Yu Felix X et al. Federated learning: Strategies for improving communication efficiency // arXiv preprint arXiv:1610.05492. — 2016. — URL: https://arxiv.org/abs/1610.05492 (visited on June 9, 2025).

[106] Li Tian, Sahu Anit Kumar, Talwalkar Ameet, Smith Virginia. Federated learning: challenges, methods, and future directions // IEEE Signal Processing Magazine. — 2020. — Vol. 37, no. 3. — P. 50–60.

[107] Mangasarian L. O. Parallel Gradient Distribution in Unconstrained Optimization // SIAM Journal on Control and Optimization. — 1995. — Vol. 33, no. 6. — P. 1916–1925.

[108] Shamir Ohad, Srebro Nati, Zhang Tong. Communication-Efficient Distributed Optimization using an Approximate Newton-type Method // Proceedings of the 31st International Conference on Machine Learning. — 2014. — Vol. 32, no. 2. — P. 1000–1008.

[109] Goodfellow Ian J, Shlens Jonathon, Szegedy Christian. Explaining and harnessing adversarial examples // arXiv preprint arXiv:1412.6572. — 2014. — URL: https://arxiv.org/pdf/1412.6572 (visited on June 9, 2025).

[110] Dang Cong D, Lan Guanghui. On the convergence properties of non-Euclidean extragradient methods for variational inequalities with generalized monotone operators // Computational Optimization and Applications. — 2015. — Vol. 60, no. 2. — P. 277–310.

[111] Iusem Alfredo N, Jofré Alejandro, Oliveira Roberto Imbuzeiro, Thompson Philip. Extragradient method with variance reduction for stochastic variational inequalities // SIAM Journal on Optimization. — 2017. — Vol. 27, no. 2. — P. 686–724.

[112] Liu Weijie, Mokhtari Aryan, Ozdaglar Asuman et al. A decentralized proximal point-type method for saddle point problems // arXiv preprint arXiv:1910.14380. — 2019. — URL: https://arxiv.org/pdf/1910.14380 (visited on June 9, 2025).

[113] Kannan Aswin, Shanbhag Uday V. Optimal stochastic extragradient schemes for pseudomonotone stochastic variational inequality problems and their variants // Computational Optimization and Applications. — 2019. — Vol. 74, no. 3. — P. 779–820.

[114] Hsieh Yu-Guan, Iutzeler Franck, Malick Jérôme, Mertikopoulos Panayotis. Explore aggressively, update conservatively: Stochastic extragradient methods with variable stepsize scaling // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 16223–16234.

[115] Diakonikolas Jelena, Daskalakis Constantinos, Jordan Michael. Efficient methods for structured nonconvex-nonconcave min-max optimization // International Conference on Artificial Intelligence and Statistics (AISTATS). — 2021. — Vol. 130. — P. 2746–2754.

[116] Liu Mingrui, Mroueh Youssef, Ross Jerret et al. Towards better understanding of adaptive gradient algorithms in generative adversarial nets // arXiv preprint arXiv:1912.11940. — 2019. — URL: https://arxiv.org/pdf/1912.11940 (visited on June 9, 2025).

[117] Liu Mingrui, Zhang Wei, Mroueh Youssef et al. A Decentralized Parallel Algorithm for Training Generative Adversarial Nets // Advances in Neural Information Processing Systems (NeurIPS). — 2020. — Vol. 14. — P. 11056–11071.

[118] Dou Zehao, Li Yuanzhi. On the One-sided Convergence of Adam-type Algorithms in Non-convex Non-concave Min-max Optimization // arXiv preprint arXiv:2109.14213. — 2021. — URL: https://arxiv.org/abs/2109.14213 (visited on June 9, 2025).

[119] Barazandeh Babak, Huang Tianjian, Michailidis George. A decentralized adaptive momentum method for solving a class of min-max optimization problems // Signal Processing. — 2021. — Vol. 189. — P. 108245.

[120] Nesterov Yurii. Dual extrapolation and its applications to solving variational inequalities and related problems // Mathematical Programming. — 2007. — Vol. 109, no. 2. — P. 319–344.

[121] Nedic Angelia, Ozdaglar Asuman. Distributed subgradient methods for multi-agent optimization // IEEE Transactions on Automatic Control. — 2009. — Vol. 54, no. 1. — P. 48–61.

[122] Shi Wei, Ling Qing, Wu Gang, Yin Wotao. EXTRA: An exact first-order algorithm for decentralized consensus optimization // SIAM Journal on Optimization. — 2015. — Vol. 25, no. 2. — P. 944–966.

[123] Scaman Kevin, Bach Francis, Bubeck Sébastien et al. Optimal algorithms for smooth and strongly convex distributed optimization in networks // arXiv preprint arXiv:1702.08704. — 2017. — URL: https://arxiv.org/pdf/1702.08704 (visited on June 9, 2025).

[124] Li Huan, Lin Zhouchen. Accelerated gradient tracking over time-varying graphs for decentralized optimization // arXiv preprint arXiv:2104.02596. — 2021. — URL: https://arxiv.org/pdf/2104.02596 (visited on June 9, 2025).

[125] Qu Guannan, Li Na. Accelerated distributed Nesterov gradient descent // IEEE Transactions on Automatic Control. — 2019. — Vol. 65, no. 6. — P. 2566–2581.

[126] Nedić Angelia, Olshevsky Alex, Shi Wei. Achieving geometric convergence for distributed optimization over time-varying graphs // SIAM Journal on Optimization. — 2017. — Vol. 27, no. 4. — P. 2597–2633.

[127] Ye Haishan, Luo Luo, Zhou Ziang, Zhang Tong. Multi-consensus decentralized accelerated gradient descent // arXiv preprint arXiv:2005.00797. — 2020. — URL: https://arxiv.org/pdf/2005.00797 (visited on June 9, 2025).

[128] Koloskova Anastasiia, Lin Tao, Stich Sebastian U. An Improved Analysis of Gradient Tracking for Decentralized Machine Learning // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 11422–11435.

[129] Alghunaim Sulaiman A, Ryu Ernest K, Yuan Kun, Sayed Ali H. Decentralized proximal gradient algorithms with linear convergence rates // IEEE Transactions on Automatic Control. — 2020. — Vol. 66, no. 6. — P. 2787–2794.

[130] Qu Guannan, Li Na. Harnessing smoothness to accelerate distributed optimization // IEEE Transactions on Control of Network Systems. — 2017. — Vol. 5, no. 3. — P. 1245–1260.

[131] Pu S., Nedić A. Distributed stochastic gradient tracking methods // Mathematical Programming. — 2021. — Vol. 187. — P. 409–457.

[132] Li Huan, Lin Zhouchen. Revisiting EXTRA for Smooth Distributed Optimization // arXiv preprint arXiv:2002.10110. — 2020. — URL=https://arxiv.org/pdf/2002.10110 (visited on June 9, 2025).

[133] Ye Haishan, Zhou Ziang, Luo Luo, Zhang Tong. Decentralized Accelerated Proximal Gradient Descent // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 18308–18317.

[134] Kovalev Dmitry, Salim Adil, Richtárik Peter. Optimal and practical algorithms for smooth and strongly convex decentralized optimization // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 18342–18352.

[135] Koloskova Anastasia, Loizou Nicolas, Boreiri Sadra et al. A unified theory of decentralized sgd with changing topology and local updates // arXiv preprint arXiv:2003.10422. — 2020. — URL: https://arxiv.org/pdf/2003.10422 (visited on June 9, 2025).

[136] Maros Marie, Jaldén Joakim. PANDA: A Dual Linearly Converging Method for Distributed Optimization over Time-Varying Undirected Graphs // IEEE Conference on Decision and Control (CDC). — 2018. — P. 6520–6525.

[137] Rogozin Alexander, Uribe César A, Gasnikov Alexander V et al. Optimal distributed convex optimization on slowly time-varying graphs // IEEE Transactions on Control of Network Systems. — 2019. — Vol. 7, no. 2. — P. 829–841.

[138] Kovalev Dmitry, Shulgin Egor, Richtárik Peter et al. ADOM: Accelerated decentralized optimization method for time-varying networks // arXiv preprint arXiv:2102.09234. — 2021. — URL: https://arxiv.org/pdf/2102.09234 (visited on June 9, 2025).

[139] Kovalev Dmitry, Gasanov Elnur, Richtárik Peter, Gasnikov Alexander. Lower Bounds and Optimal Algorithms for Smooth and Strongly Convex Decentralized Optimization Over Time-Varying Networks // arXiv preprint arXiv:2106.04469. — 2021. — URL: https://arxiv.org/pdf/2106.04469 (visited on June 9, 2025).

[140] Rogozin Alexander, Gasnikov Alexander. Projected gradient method for decentralized optimization over time-varying networks // arXiv preprint arXiv:1911.08527. — 2019. — URL: https://arxiv.org/pdf/1911.08527 (visited on June 9, 2025).

[141] Rabbat M., Nowak R. Distributed optimization in sensor networks // Third International Symposium on Information Processing in Sensor Networks. — 2004. — P. 20–27.

[142] Beck Amir, Nedić Angelia, Ozdaglar Asuman, Teboulle Marc. An O (1/k) gradient method for network resource allocation problems // IEEE Transactions on Control of Network Systems. — 2014. — Vol. 1, no. 1. — P. 64–73.

[143] Giselsson Pontus, Doan Minh Dang, Keviczky Tamás et al. Accelerated gradient methods and dual decomposition in distributed model predictive control // Automatica. — 2013. — Vol. 49, no. 3. — P. 829–833.

[144] Bazerque Juan Andrés, Giannakis Georgios B. Distributed spectrum sensing for cognitive radio networks by exploiting sparsity // IEEE Transactions on Signal Processing. — 2009. — Vol. 58, no. 3. — P. 1847–1862.

[145] Granichin Oleg, Amelina Natalia. Simultaneous Perturbation Stochastic Approximation for Tracking Under Unknown but Bounded Disturbances // IEEE Transactions on Automatic Control. — 2015. — Vol. 60, no. 6. — P. 1653–1658.

[146] Ram S. S., Veeravalli V. V., Nedic A. Distributed Non-Autonomous Power Control through Distributed Convex Optimization. — 2009. — P. 3001–3005.

[147] Gan Lingwen, Topcu Ufuk, Low Steven H. Optimal decentralized protocol for electric vehicle charging // IEEE Transactions on Power Systems. — 2012. — Vol. 28, no. 2. — P. 940–951.

[148] Jadbabaie A., Lin Jie, Morse A.S. Coordination of groups of mobile autonomous agents using nearest neighbor rules // IEEE Transactions on Automatic Control. — 2003. — Vol. 48, no. 6. — P. 988–1001.

[149] Ren Wei. Consensus based formation control strategies for multi-vehicle systems // American Control Conference. — 2006. — P. 6.

[150] Antal Cathrine, Granichin Oleg, Levi Sergey. Adaptive autonomous soaring of multiple UAVs using Simultaneous Perturbation Stochastic Approximation // 49th IEEE Conference on Decision and Control (CDC). — 2010. — P. 3656–3661.

[151] Amelina Natalia, Granichin Oleg, Kornivetc Aleksandra. Local voting protocol in decentralized load balancing problem with switched topology, noise, and delays. — 2013. — P. 4613–4618.

[152] Amelina Natalia, Fradkov Alexander, Jiang Yuming, Vergados Dimitrios J. Approximate Consensus in Stochastic Networks With Application to Load Balancing // IEEE Transactions on Information Theory. — 2015. — Vol. 61, no. 4. — P. 1739–1752.

[153] Forero Pedro A, Cano Alfonso, Giannakis Georgios B. Consensus-based distributed linear support vector machines // Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks. — 2010. — P. 35–46.

[154] Nedić Angelia, Olshevsky Alex, Uribe César A. Fast convergence rates for distributed non-

Bayesian learning // IEEE Transactions on Automatic Control. — 2017. — Vol. 62, no. 11. — P. 5538–5553.

[155] Nedic Angelia. Distributed Gradient Methods for Convex Machine Learning Problems in Networks: Distributed Optimization // IEEE Signal Processing Magazine. — 2020. — Vol. 37, no. 3. — P. 92–101.

[156] Mukherjee Soham, Chakraborty Mrityunjoy. A decentralized algorithm for large scale min-max problems // 59th IEEE Conference on Decision and Control (CDC). — 2020. — P. 2967–2972.

[157] Deng Yuyang, Mahdavi Mehrdad. Local Stochastic Gradient Descent Ascent: Convergence Analysis and Communication Efficiency // International Conference on Artificial Intelligence and Statistics (AISTATS). — 2021. — Vol. 130. — P. 1387–1395.

[158] Hou Charlie, Thekumparampil Kiran K, Fanti Giulia, Oh Sewoong. Efficient Algorithms for Federated Saddle Point Optimization // arXiv preprint arXiv:2102.06333. — 2021. — URL: `https://arxiv.org/pdf/2102.06333` (visited on June 9, 2025).

[159] Zhang Junyu, Hong Mingyi, Zhang Shuzhong. On lower iteration complexity bounds for the saddle point problems // arXiv preprint arXiv:1912.07481. — 2019. — URL: `https://arxiv.org/pdf/1912.07481` (visited on June 9, 2025).

[160] Ouyang Yuyuan, Xu Yangyang. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems // Mathematical Programming. — 2021. — Vol. 135. — P. 1–35.

[161] Foster Dylan J, Sekhari Ayush, Shamir Ohad et al. The complexity of making the gradient small in stochastic convex optimization // Conference on Learning Theory. — 2019. — Vol. 99. — P. 1319–1345.

[162] Arjevani Yossi, Shamir Ohad. Communication complexity of distributed convex learning and optimization // Advances in neural information processing systems. — 2015. — Vol. 28. — P. 1756–1764.

[163] Liu Ji, Morse A Stephen. Accelerated linear iterations for distributed averaging // Annual Reviews in Control. — 2011. — Vol. 35, no. 2. — P. 160–165.

[164] Minty George J. Monotone (nonlinear) operators in Hilbert space // Duke Mathematical Journal. — 1962. — Vol. 29, no. 3. — P. 341 – 346.

[165] Kempe David, Dobra Alin, Gehrke Johannes. Gossip-based computation of aggregate information // 44th Annual IEEE Symposium on Foundations of Computer Science. Proceedings. — 2003. — P. 482–491.

[166] Boyd Stephen, Ghosh Arpita, Prabhakar Balaji, Shah Devavrat. Randomized gossip algorithms // IEEE transactions on information theory. — 2006. — Vol. 52, no. 6. — P. 2508–2530.

[167] Nedich Angelia, Olshevsky Alex, Shi Wei. A geometrically convergent method for distributed

optimization over time-varying graphs // IEEE 55th Conference on Decision and Control (CDC). — 2016. — P. 1023–1029.

[168] Hendrikx Hadrien, Bach Francis, Massoulie Laurent. An optimal algorithm for decentralized finite-sum optimization // SIAM Journal on Optimization. — 2021. — Vol. 31, no. 4. — P. 2753–2783.

[169] Barazandeh Babak, Tarzanagh Davoud Ataee, Michailidis George. Solving a class of non-convex min-max games using adaptive momentum methods // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). — 2021. — P. 3625–3629.

[170] Han Yuze, Xie Guangzeng, Zhang Zhihua. Lower complexity bounds of finite-sum optimization problems: The results and construction // Journal of Machine Learning Research. — 2024. — Vol. 25, no. 2. — P. 1–86.

[171] Chang Chih-Chung, Lin Chih-Jen. LIBSVM: a library for support vector machines // ACM transactions on intelligent systems and technology (TIST). — 2011. — Vol. 2, no. 3. — P. 1–27.

[172] Mcdonald Ryan, Mohri Mehryar, Silberman Nathan et al. Efficient Large-Scale Distributed Training of Conditional Maximum Entropy Models // Advances in Neural Information Processing Systems 22. — 2009. — Vol. 1. — P. 1231–1239.

[173] Zinkevich Martin, Weimer Markus, Li Lihong, Smola Alex J. Parallelized Stochastic Gradient Descent // Advances in Neural Information Processing Systems 23. — 2010. — Vol. 1. — P. 2595–2603.

[174] Lin Tao, Stich Sebastian U, Patel Kumar Kshitij, Jaggi Martin. Don't Use Large Mini-Batches, Use Local SGD // arXiv preprint arXiv:1808.07217. — 2018. — URL: https://arxiv.org/pdf/1808.07217 (visited on June 9, 2025).

[175] Stich Sebastian U. Local SGD converges fast and communicates little // arXiv preprint arXiv:1805.09767. — 2018. — URL: https://arxiv.org/pdf/1805.09767 (visited on June 9, 2025).

[176] Khaled Ahmed, Mishchenko Konstantin, Richtárik Peter. Tighter Theory for Local SGD on Identical and Heterogeneous Data // International Conference on Artificial Intelligence and Statistics. — 2020. — Vol. 108. — P. 4519–4529.

[177] Woodworth Blake, Patel Kumar Kshitij, Stich Sebastian U et al. Is Local SGD Better than Minibatch SGD? // arXiv preprint arXiv:2002.07839. — 2020. — URL: https://arxiv.org/pdf/2002.07839 (visited on June 9, 2025).

[178] Woodworth Blake, Patel Kumar Kshitij, Srebro Nathan. Minibatch vs Local SGD for Heterogeneous Distributed Learning // arXiv preprint arXiv:2006.04735. — 2020. — URL: https://arxiv.org/pdf/2006.04735 (visited on June 9, 2025).

[179] Tsaknakis Ioannis, Hong Mingyi, Liu Sijia. Decentralized Min-Max Optimization: Formula-

tions, Algorithms and Applications in Network Poisoning Attack // 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). — 2020. — P. 5755–5759.

[180] Rogozin Alexander, Dvurechensky Pavel, Dvinkikh Darina et al. Decentralized Distributed Optimization for Saddle Point Problems // arXiv preprint arXiv:2102.07758. — 2021. — URL: https://arxiv.org/pdf/2102.07758 (visited on June 9, 2025).

[181] Xian Wenhan, Huang Feihu, Zhang Yanfu, Huang Heng. A Faster Decentralized Algorithm for Nonconvex Minimax Problems // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 25865–25877.

[182] Radford Alec, Metz Luke, Chintala Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks // arXiv preprint arXiv:1511.06434. — 2015. — URL=https://arxiv.org/pdf/1511.06434 (visited on June 9, 2025).

[183] Krizhevsky Alex, Nair Vinod, Hinton Geoffrey. CIFAR-10 (Canadian Institute for Advanced Research). — URL=https://www.cs.toronto.edu/~kriz/cifar.html (visited on June 9, 2025).

[184] Sayed Ali H. Diffusion Adaptation Over Networks // Academic Press Library in Signal Processing. — 2014. — Vol. 3. — P. 323–453.

[185] Sayed Ali H, Tu Sheng-Yuan, Chen Jianshu et al. Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior // IEEE Signal Processing Magazine. — 2013. — Vol. 30, no. 3. — P. 155–171.

[186] Xie Siyu, Guo Lei. Analysis of Distributed Adaptive Filters Based on Diffusion Strategies Over Sensor Networks // IEEE Transactions on Automatic Control. — 2018. — Vol. 63, no. 11. — P. 3643–3658.

[187] Alghunaim Sulaiman A, Yuan Kun. A unified and refined convergence analysis for non-convex decentralized learning // IEEE Transactions on Signal Processing. — 2022. — Vol. 70. — P. 3264–3279.

[188] Matveev Alexey S., Almodarresi Mostafa, Ortega Romeo et al. Diffusion-Based Distributed Parameter Estimation Through Directed Graphs With Switching Topology: Application of Dynamic Regressor Extension and Mixing // IEEE Transactions on Automatic Control. — 2022. — Vol. 67, no. 8. — P. 4256–4263.

[189] Gorbunov Eduard, Hanzely Filip, Richtarik Peter. Local SGD: Unified Theory and New Efficient Methods // Proceedings of The 24th International Conference on Artificial Intelligence and Statistics. — 2021. — Vol. 130. — P. 3556–3564.

[190] Nedic Angelia, Olshevsky Alex, Ozdaglar Asuman, Tsitsiklis John N. On Distributed Averaging Algorithms and Quantization Effects // IEEE Transactions on Automatic Control. — 2009. — Vol. 54, no. 11. — P. 2506–2517.

[191] Agafonov Artem, Dvurechensky Pavel, Scutari Gesualdo et al. An accelerated second-order method for distributed stochastic optimization // arXiv preprint arXiv:2103.14392. — 2021. — URL: https://arxiv.org/pdf/2103.14392 (visited on June 9, 2025).

[192] Daneshmand Amir, Scutari Gesualdo, Dvurechensky Pavel, Gasnikov Alexander. Newton method over networks is fast up to the statistical precision // International Conference on Machine Learning. — 2021. — Vol. 139. — P. 2398–2409.

[193] Dvurechensky Pavel, Kamzolov Dmitry, Lukashevich Aleksandr et al. Hyperfast Second-Order Local Solvers for Efficient Statistically Preconditioned Distributed Optimization // EURO Journal on Computational Optimization. — 2022. — P. 100045.

[194] Stich Sebastian U. Unified optimal analysis of the (stochastic) gradient method // arXiv preprint arXiv:1907.04232. — 2019. — URL: https://arxiv.org/pdf/1907.04232 (visited on June 9, 2025).

[195] Koloskova Anastasia, Stich Sebastian, Jaggi Martin. Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication // Proceedings of the 36th International Conference on Machine Learning (ICML). — 2019. — Vol. 97. — P. 3478–3487.

[196] Mishchenko Konstantin, Kovalev Dmitry, Shulgin Egor et al. Revisiting stochastic extragradient // International Conference on Artificial Intelligence and Statistics. — 2020. — Vol. 108. — P. 4573–4582.

[197] Moritz Philipp, Nishihara Robert, Wang Stephanie et al. Ray: A Distributed Framework for Emerging AI Applications // 13th USENIX Symposium on Operating Systems Design and Implementation. — 2018. — P. 561–577.

[198] Mirza Mehdi, Osindero Simon. Conditional generative adversarial nets // arXiv preprint arXiv:1411.1784. — 2014. — URL: https://arxiv.org/pdf/1411.1784 (visited on June 9, 2025).

[199] Li Yuanzhi, Yuan Yang. Convergence analysis of two-layer neural networks with relu activation // arXiv preprint arXiv:1705.09886. — 2017. — URL: https://arxiv.org/pdf/1705.09886 (visited on June 9, 2025).

[200] Kleinberg Bobby, Li Yuanzhi, Yuan Yang. An alternative view: When does SGD escape local minima? // International Conference on Machine Learning. — 2018. — Vol. 80. — P. 2698–2707.

[201] Zhou Yi, Yang Junjie, Zhang Huishuai et al. SGD converges to global minimum in deep learning via star-convex path // arXiv preprint arXiv:1901.00451. — 2019. — URL: https://arxiv.org/pdf/1901.00451 (visited on June 9, 2025).

[202] Kingma D. P., Ba J. Adam: A method for stochastic optimization // International Conference on Learning Representations. — 2015. — P. 2305–2313.

[203] Tropp Joel A. User-friendly tail bounds for sums of random matrices // Foundations of computational mathematics. — 2012. — Vol. 12. — P. 389–434.

[204] Sun Ying, Daneshmand Amir, Scutari Gesualdo. Distributed optimization based on gradient-tracking revisited: Enhancing convergence rate via surrogation // arXiv preprint arXiv:1905.02637. — 2019. — URL: https://arxiv.org/pdf/1905.02637 (visited on June 9, 2025).

[205] Zhang Yuchen, Lin Xiao. DiSCO: Distributed Optimization for Self-Concordant Empirical Loss // Proceedings of the 32nd International Conference on Machine Learning. — 2015. — Vol. 37. — P. 362–370.

[206] Reddi Sashank J, Konečnỳ Jakub, Richtárik Peter et al. Aide: Fast and communication efficient distributed optimization // arXiv preprint arXiv:1608.06879. — 2016. — URL: https://arxiv.org/pdf/1608.06879 (visited on June 9, 2025).

[207] Lin Hongzhou, Mairal Julien, Harchaoui Zaid. Catalyst acceleration for first-order convex optimization: from theory to practice // Journal of Machine Learning Research. — 2018. — Vol. 18, no. 1. — P. 7854–7907.

[208] Yuan Xiao-Tong, Li Ping. On convergence of distributed approximate newton methods: Globalization, sharper bounds and beyond // arXiv preprint arXiv:1908.02246. — 2019. — URL: https://arxiv.org/pdf/1908.02246 (visited on June 9, 2025).

[209] Tian Ye, Scutari Gesualdo, Cao Tianyu, Gasnikov Alexander. Acceleration in Distributed Optimization Under Similarity // arXiv preprint arXiv:2110.12347. — 2021. — URL: https://arxiv.org/pdf/2110.12347 (visited on June 9, 2025).

[210] Zhang Yuchen, Xiao Lin. Communication-efficient distributed optimization of self-concordant empirical loss // Large-Scale and Distributed Optimization. — 2018. — P. 289–341.

[211] Lan Guanghui. Gradient sliding for composite optimization // Mathematical Programming. — 2016. — Vol. 159, no. 1. — P. 201–235.

[212] Gorbunov Eduard, Burlachenko Konstantin, Li Zhize, Richtárik Peter. MARINA: Faster nonconvex distributed learning with compression // 38th International Conference on Machine Learning. — 2021. — Vol. 139. — P. 3788–3798.

[213] Yuan Deming, Ma Qian, Wang Zhen. Dual averaging method for solving multi-agent saddle-point problems with quantized information // Transactions of the Institute of Measurement and Control. — 2014. — Vol. 36, no. 1. — P. 38–46.

[214] Alacaoglu Ahmet, Malitsky Yura. Stochastic variance reduction for variational inequality methods // Conference on Learning Theory. — 2022. — Vol. 178. — P. 778–816.

[215] Karimireddy Sai Praneeth, Rebjock Quentin, Stich Sebastian, Jaggi Martin. Error feedback fixes signsgd and other gradient compression schemes // International Conference on Machine Learning. — 2019. — Vol. 97. — P. 3252–3261.

[216] Li Zhize, Kovalev Dmitry, Qian Xun, Richtarik Peter. Acceleration for Compressed Gradient Descent in Distributed and Federated Optimization // Proceedings of the 37th International Conference on Machine Learning. — 2020. — Vol. 119. — P. 5895–5904.

[217] Qian Xun, Richtarik Peter, Zhang Tong. Error Compensated Distributed SGD Can Be Accelerated // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 30401–30413.

[218] Yang Junchi, Kiyavash Negar, He Niao. Global Convergence and Variance Reduction for a Class of Nonconvex-Nonconcave Minimax Problems // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 1153–1165.

[219] Kairouz Peter, McMahan H Brendan, Avent Brendan et al. Advances and open problems in federated learning // Foundations and Trends® in Machine Learning. — 2021. — Vol. 14, no. 1–2. — P. 1–210.

[220] Horváth Samuel, Richtárik Peter. A Better Alternative to Error Feedback for Communication-Efficient Distributed Learning // arXiv preprint arXiv:2006.11077. — 2020. — URL: https://arxiv.org/pdf/2006.11077 (visited on June 9, 2025).

[221] Philippenko Constantin, Dieuleveut Aymeric. Bidirectional compression in heterogeneous settings for distributed or federated learning with partial participation: tight convergence guarantees // arXiv preprint arXiv:2006.14591. — 2020. — URL: https://arxiv.org/pdf/2006.14591 (visited on June 9, 2025).

[222] Horvath Samuel, Ho Chen-Yu, Horvath Ludovit et al. Natural compression for distributed deep learning // arXiv preprint arXiv:1905.10988. — 2019. — URL: https://arxiv.org/pdf/1905.10988 (visited on June 9, 2025).

[223] Tang Hanlin, Yu Chen, Lian Xiangru et al. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression // International Conference on Machine Learning. — 2019. — Vol. 16. — P. 10747–10760.

[224] Zheng Shuai, Huang Ziyue, Kwok James. Communication-efficient distributed blockwise momentum SGD with error-feedback // Advances in Neural Information Processing Systems. — 2019. — Vol. 32. — P. 11450–11460.

[225] Liu Xiaorui, Li Yao, Tang Jiliang, Yan Ming. A double residual compression algorithm for efficient distributed learning // International Conference on Artificial Intelligence and Statistics. — 2020. — Vol. 108. — P. 133–143.

[226] Fatkhullin Ilyas, Sokolov Igor, Gorbunov Eduard et al. EF21 with bells & whistles: Practical algorithmic extensions of modern error feedback // arXiv preprint arXiv:2110.03294. — 2021. — URL: https://arxiv.org/pdf/2110.03294 (visited on June 9, 2025).

[227] Palaniappan Balamurugan, Bach Francis. Stochastic variance reduction methods for saddle-

point problems // Advances in Neural Information Processing Systems. — 2016. — Vol. 29. — P. 1416–1424.

[228] Goodfellow Ian. Nips 2016 tutorial: Generative adversarial networks // arXiv preprint arXiv:1701.00160. — 2016. — URL: https://arxiv.org/pdf/1701.00160 (visited on June 9, 2025).

[229] Hsieh Yu-Guan, Iutzeler Franck, Malick Jérôme, Mertikopoulos Panayotis. On the convergence of single-call stochastic extra-gradient methods // arXiv preprint arXiv:1908.08465. — 2019. — URL=https://arxiv.org/pdf/1908.08465 (visited on June 9, 2025).

[230] Mokhtari Aryan, Ozdaglar Asuman E, Pattathil Sarath. Convergence Rate of O(1/k) for Optimistic Gradient and Extragradient Methods in Smooth Convex-Concave Saddle Point Problems // SIAM Journal on Optimization. — 2020. — Vol. 30, no. 4. — P. 3230–3251.

[231] Stich Sebastian U, Cordonnier Jean-Baptiste, Jaggi Martin. Sparsified SGD with Memory // Advances in Neural Information Processing Systems 31. — 2018. — Vol. 1. — P. 4447–4458.

[232] Stich Sebastian U, Karimireddy Sai Praneeth. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication // arXiv preprint arXiv:1909.05350. — 2019. — URL: https://arxiv.org/pdf/1909.05350 (visited on June 9, 2025).

[233] Vaswani Ashish, Shazeer Noam, Parmar Niki et al. Attention is All you Need // Advances in Neural Information Processing Systems 30. — 2017. — Vol. 1. — P. 5998–6008.

[234] Devlin Jacob, Chang Ming-Wei, Lee Kenton, Toutanova Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // NAACL-HLT. — 2019. — Vol. 1. — P. 4171–4187.

[235] Liu Yinhan, Ott Myle, Goyal Naman et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach // arXiv preprint arXiv:1907.11692. — 2019. — URL: https://arxiv.org/pdf/1907.11692 (visited on June 9, 2025).

[236] Zhu Chen, Cheng Yu, Gan Zhe et al. Freelb: Enhanced adversarial training for natural language understanding // arXiv preprint arXiv:1909.11764. — 2019. — URL: https://arxiv.org/pdf/1909.11764 (visited on June 9, 2025).

[237] Liu Xiaodong, Cheng Hao, He Pengcheng et al. Adversarial training for large neural language models // arXiv preprint arXiv:2004.08994. — 2020. — URL: https://arxiv.org/pdf/2004.08994 (visited on June 9, 2025).

[238] Lan Zhen-Zhong, Chen Mingda, Goodman Sebastian et al. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations // International Conference on Learning Representations. — 2020. — Vol. 15. — P. 11574–11590.

[239] Ryabinin Max, Gorbunov Eduard, Plokhotnyuk Vsevolod, Pekhimenko Gennady. Moshpit SGD: Communication-Efficient Decentralized Training on Heterogeneous Unreliable Devices //

Advances in Neural Information Processing Systems 34 (NeurIPS 2021). — 2021. — Vol. 22. — P. 18195–18211.

[240] Diskin Michael, Bukhtiyarov Alexey, Ryabinin Max et al. Distributed Deep Learning in Open Collaborations // arXiv preprint arXiv:2106.10207. — 2021. — URL: https://arxiv.org/pdf/2106.10207 (visited on June 9, 2025).

[241] Dettmers Tim. 8-bit approximations for parallelism in deep learning // arXiv preprint arXiv:1511.04561. — 2015. — URL=https://arxiv.org/pdf/1511.04561 (visited on June 9, 2025).

[242] Lin Yujun, Han Song, Mao Huizi et al. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training // International Conference on Learning Representations. — 2018. — Vol. 3. — P. 2366–2379.

[243] Lei Ba Jimmy, Ryan Kiros Jamie, Hinton Geoffrey E. Layer Normalization // arXiv preprint arXiv:1607.06450. — 2016. — URL: https://arxiv.org/pdf/1607.06450 (visited on June 9, 2025).

[244] Fedus William, Zoph Barret, Shazeer Noam. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity // arXiv preprint arXiv:2101.03961. — 2021. — URL: https://arxiv.org/pdf/2101.03961 (visited on June 9, 2025).

[245] Wang Alex, Singh Amanpreet, Michael Julian et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding // arXiv preprint arXiv:1804.07461. — 2018. — URL: https://arxiv.org/pdf/1804.07461 (visited on June 9, 2025).

[246] Basu Debraj, Data Deepesh, Karakus Can, Diggavi Suhas. Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations // Advances in Neural Information Processing Systems. — 2019. — Vol. 32. — P. 14628–14639.

[247] Nadiradze Giorgi, Sabour Amirmojtaba, Davies Peter et al. Decentralized SGD with asynchronous, local and quantized updates // arXiv preprint arXiv:1910.12308. — 2019. — URL: https://arxiv.org/pdf/1910.12308 (visited on June 9, 2025).

[248] Malinovsky Grigory, Yi Kai, Richtárik Peter. Variance reduced proxskip: Algorithm, theory and application to federated learning // arXiv preprint arXiv:2207.04338. — 2022. — URL: https://arxiv.org/pdf/2207.04338 (visited on June 9, 2025).

[249] Condat Laurent, Agarsky Ivan, Richtárik Peter. Provably Doubly Accelerated Federated Learning: The First Theoretically Successful Combination of Local Training and Compressed Communication // arXiv preprint arXiv:2210.13277. — 2022. — URL: https://arxiv.org/pdf/2210.13277 (visited on June 9, 2025).

[250] Szlendak Rafał, Tyurin Alexander, Richtárik Peter. Permutation Compressors for Provably Faster Distributed Nonconvex Optimization // arXiv preprint arXiv:2110.03300. — 2021. — URL: https://arxiv.org/pdf/2110.03300 (visited on June 9, 2025).

[251] LeCun Yann, Cortes Corinna. MNIST handwritten digit database. — 2010. — URL: http://yann.lecun.com/exdb/mnist/ (visited on June 9, 2025).

[252] Deng Jia, Dong Wei, Socher Richard et al. ImageNet: A large-scale hierarchical image database // 2009 IEEE Conference on Computer Vision and Pattern Recognition. — 2009. — P. 248–255.

[253] McMahan Brendan, Moore Eider, Ramage Daniel et al. Communication-efficient learning of deep networks from decentralized data // Artificial Intelligence and Statistics. — 2017. — P. 1273–1282.

[254] Woodworth Blake, Bullins Brian, Shamir Ohad, Srebro Nathan. The Min-Max Complexity of Distributed Stochastic Convex Optimization with Intermittent Communication // arXiv preprint arXiv:2102.01583. — 2021. — URL: https://arxiv.org/pdf/2102.01583 (visited on June 9, 2025).

[255] Chan Ernie, Heimlich Marcel, Purkayastha Avi, van de Geijn Robert. Collective communication: theory, practice, and experience // Concurrency and Computation: Practice and Experience. — 2007. — Vol. 19, no. 13. — P. 1749–1783.

[256] Li Shigang, Hoefler Torsten. Near-optimal sparse allreduce for distributed deep learning // Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. — 2022. — P. 135–149.

[257] Gorbunov Eduard, Taylor Adrien, Horváth Samuel, Gidel Gauthier. Convergence of Proximal Point and Extragradient-Based Methods Beyond Monotonicity: the Case of Negative Comonotonicity // arXiv preprint arXiv:2210.13831. — 2022. — URL: https://arxiv.org/pdf/2210.13831 (visited on June 9, 2025).

[258] Loizou Nicolas, Berard Hugo, Gidel Gauthier et al. Stochastic gradient descent-ascent and consensus optimization for smooth games: Convergence analysis under expected co-coercivity // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 19095–19108.

[259] Dem'yanov Vladimir Fedorovich, Pevnyi Aleksandr Borisovich. Numerical methods for finding saddle points // USSR Computational Mathematics and Mathematical Physics. — 1972. — Vol. 12, no. 5. — P. 11–52.

[260] Nemirovski Arkadi, Juditsky Anatoli, Lan Guanghui, Shapiro Alexander. Robust stochastic approximation approach to stochastic programming // SIAM Journal on Optimization. — 2009. — Vol. 19, no. 4. — P. 1574–1609.

[261] Sadiev Abdurakhmon, Beznosikov Aleksandr, Dvurechensky Pavel, Gasnikov Alexander. Zeroth-order algorithms for smooth saddle-point problems // International Conference on Mathematical Optimization Theory and Operations Research. — 2021. — P. 71–85.

[262] Beck Amir. First-order methods in optimization. — Society for Industrial and Applied Mathematics (SIAM), 2017.

[263] Hofmann Thomas, Lucchi Aurelien, Lacoste-Julien Simon, McWilliams Brian. Variance reduced stochastic gradient descent with neighbors // Advances in Neural Information Processing Systems. — 2015. — Vol. 28. — P. 2305–2313.

[264] Kovalev Dmitry, Horváth Samuel, Richtárik Peter. Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop // Algorithmic Learning Theory. — 2020. — Vol. 117. — P. 451–467.

[265] Horváth Samuel, Kovalev Dmitry, Mishchenko Konstantin et al. Stochastic distributed learning with gradient quantization and variance reduction // arXiv preprint arXiv:1904.05115. — 2019. — URL: `https://arxiv.org/pdf/1904.05115` (visited on June 9, 2025).

[266] Gorbunov Eduard, Hanzely Filip, Richtarik Peter. A Unified Theory of SGD: Variance Reduction, Sampling, Quantization and Coordinate Descent // Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. — 2020. — Vol. 108. — P. 680–690.

[267] Gorbunov Eduard, Berard Hugo, Gidel Gauthier, Loizou Nicolas. Stochastic extragradient: General analysis and improved rates // International Conference on Artificial Intelligence and Statistics. — 2022. — Vol. 151. — P. 7865–7901.

[268] Richtárik Peter, Takác Martin. Stochastic reformulations of linear systems: algorithms and convergence theory // SIAM Journal on Matrix Analysis and Applications. — 2020. — Vol. 41, no. 2. — P. 487–524.

[269] Loizou Nicolas, Richtárik Peter. Convergence analysis of inexact randomized iterative methods // SIAM Journal on Scientific Computing. — 2020. — Vol. 42, no. 6. — P. A3979–A4016.

[270] Loizou Nicolas, Richtárik Peter. Momentum and stochastic momentum for stochastic gradient, newton, proximal point and subspace descent methods // Computational Optimization and Applications. — 2020. — Vol. 77, no. 3. — P. 653–710.

[271] Gower Robert Mansel, Loizou Nicolas, Qian Xun et al. SGD: General analysis and improved rates // International conference on machine learning. — 2019. — Vol. 97. — P. 5200–5209.

[272] Gower Robert, Sebbouh Othmane, Loizou Nicolas. Sgd for structured nonconvex functions: Learning rates, minibatching and interpolation // International Conference on Artificial Intelligence and Statistics. — 2021. — Vol. 130. — P. 1315–1323.

[273] Hanzely Filip, Richtárik Peter. Accelerated coordinate descent with arbitrary sampling and best rates for minibatches // The 22nd International Conference on Artificial Intelligence and Statistics. — 2019. — P. 304–312.

[274] Qian Xun, Qu Zheng, Richtárik Peter. SAGA with arbitrary sampling // International Conference on Machine Learning. — 2019. — Vol. 97. — P. 5190–5199.

[275] Qian Xun, Qu Zheng, Richtárik Peter. L-SVRG and L-Katyusha with arbitrary sampling // Journal of Machine Learning Research. — 2021. — Vol. 22, no. 112. — P. 1–47.

[276] Nguyen Lam M, Scheinberg Katya, Takáč Martin. Inexact SARAH algorithm for stochastic optimization // Optimization Methods and Software. — 2021. — Vol. 36, no. 1. — P. 237–258.

[277] Hu Wenqing, Li Chris Junchi, Lian Xiangru et al. Efficient smooth non-convex stochastic compositional optimization via stochastic recursive gradient descent // Advances in Neural Information Processing Systems. — 2019. — Vol. 32. — P. 6897–6905.

[278] Cutkosky Ashok, Orabona Francesco. Momentum-based variance reduction in non-convex sgd // arXiv preprint arXiv:1905.10018. — 2019. — URL=https://arxiv.org/pdf/1905.10018 (visited on June 9, 2025).

[279] Chavdarova Tatjana, Hsieh Ya-Ping, Jordan Michael I. Continuous-time analysis for variational inequalities: An overview and desiderata // arXiv preprint arXiv:2207.07105. — 2022. — URL: https://arxiv.org/pdf/2207.07105 (visited on June 9, 2025).

[280] Tominin Vladislav, Tominin Yaroslav, Borodich Ekaterina et al. On accelerated methods for saddle-point problems with composite structure // arXiv preprint arXiv:2103.09344. — 2021. — URL: https://arxiv.org/pdf/2103.09344 (visited on June 9, 2025).

[281] Carmon Yair, Jin Yujia, Sidford Aaron, Tian Kevin. Variance reduction for matrix games // arXiv preprint arXiv:1907.02056. — 2019. — URL: https://arxiv.org/pdf/1907.02056 (visited on June 9, 2025).

[282] Allen-Zhu Zeyuan, Yuan Yang. Improved SVRG for non-strongly-convex or sum-of-non-convex objectives // International conference on machine learning. — 2016. — P. 1080–1089.

[283] Mokhtari Aryan, Ozdaglar Asuman, Pattathil Sarath. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach // International Conference on Artificial Intelligence and Statistics. — 2020. — Vol. 108. — P. 1497–1507.

[284] Malitsky Yura, Tam Matthew K. A forward-backward splitting method for monotone inclusions without cocoercivity // SIAM Journal on Optimization. — 2020. — Vol. 30, no. 2. — P. 1451–1472.

[285] Polyak Boris T. Introduction to optimization. — New York, Optimization Software, 1987.

[286] Gidel Gauthier, Hemmat Reyhane Askari, Pezeshki Mohammad et al. Negative momentum for improved game dynamics // The 22nd International Conference on Artificial Intelligence and Statistics. — 2019. — Vol. 89. — P. 1802–1811.

[287] Yoon TaeHo, Ryu Ernest K. Accelerated Algorithms for Smooth Convex-Concave Minimax Problems with O (1/kˆ 2) Rate on Squared Gradient Norm // International Conference on Machine Learning. — 2021. — Vol. 139. — P. 12098–12109.

[288] Alacaoglu Ahmet, Malitsky Yura, Cevher Volkan. Forward-reflected-backward method with variance reduction // Computational optimization and applications. — 2021. — Vol. 80, no. 2. — P. 321–346.

[289] Nemirovski Arkadi. Mini-Course on Convex Programming Algorithms. — Citeseer, 2013.

[290] Condat Laurent. Fast projection onto the simplex and the l 1 ball // Mathematical Programming. — 2016. — Vol. 158, no. 1-2. — P. 575–585.

[291] Frank Marguerite, Wolfe Philip. An algorithm for quadratic programming // Naval research logistics quarterly. — 1956. — Vol. 3, no. 1-2. — P. 95–110.

[292] Levitin Evgeny S, Polyak Boris T. Constrained minimization methods // USSR Computational mathematics and mathematical physics. — 1966. — Vol. 6, no. 5. — P. 1–50.

[293] Demianov Vladimir Fedorovich, Rubinov Aleksandr Moiseevich. Approximate methods in optimization problems. No. 32. — Elsevier Publishing Company, 1970.

[294] Dunn Joseph C, Harshbarger S. Conditional gradient algorithms with open loop step size rules // Journal of Mathematical Analysis and Applications. — 1978. — Vol. 62, no. 2. — P. 432–444.

[295] Patriksson Michael. Partial linearization methods in nonlinear programming // Journal of Optimization Theory and Applications. — 1993. — Vol. 78, no. 2. — P. 227–246.

[296] Jaggi Martin. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization // Proceedings of the 30th International Conference on Machine Learning. — 2013. — Vol. 28, no. 1. — P. 427–435.

[297] Lacoste-Julien Simon, Jaggi Martin. On the global linear convergence of Frank-Wolfe optimization variants // Advances in neural information processing systems. — 2015. — Vol. 28. — P. 496–504.

[298] Bach Francis. Learning with Submodular Functions: A Convex Optimization Perspective // arXiv preprint arXiv:1111.6453. — 2011. — URL: `https://arxiv.org/abs/1111.6453` (visited on June 9, 2025).

[299] Miech Antoine, Alayrac Jean-Baptiste, Bojanowski Piotr et al. Learning from video and text via large-scale discriminative clustering // Proceedings of the IEEE international conference on computer vision. — 2017. — P. 5257–5266.

[300] Bojanowski Piotr, Lajugie Rémi, Bach Francis et al. Weakly supervised action labeling in videos under ordering constraints // European Conference on Computer Vision. — 2014. — P. 628–643.

[301] Krishnan Rahul G, Lacoste-Julien Simon, Sontag David. Barrier Frank-Wolfe for marginal inference // Advances in Neural Information Processing Systems. — 2015. — Vol. 28.

[302] Braun Gábor, Carderera Alejandro, Combettes Cyrille W et al. Conditional gradient methods // arXiv preprint arXiv:2211.14103. — 2022. — URL: https://arxiv.org/pdf/2211.14103 (visited on June 9, 2025).

[303] Lacoste-Julien Simon. Convergence rate of frank-wolfe for non-convex objectives // arXiv preprint arXiv:1607.00345. — 2016.

[304] Reddi Sashank J, Sra Suvrit, Póczos Barnabás, Smola Alex. Stochastic frank-wolfe methods for nonconvex optimization // 2016 54th annual Allerton conference on communication, control, and computing (Allerton). — 2016. — P. 1244–1251.

[305] Schmidt Mark, Le Roux Nicolas, Bach Francis. Minimizing finite sums with the stochastic average gradient // Mathematical Programming. — 2017. — Vol. 162, no. 1-2. — P. 83–112.

[306] Yang Zhuang, Chen Zengping, Wang Cheng. Accelerating mini-batch SARAH by step size rules // Information Sciences. — 2021. — Vol. 558. — P. 157–173.

[307] Mairal Julien. Incremental majorization-minimization optimization with application to large-scale machine learning // SIAM Journal on Optimization. — 2015. — Vol. 25, no. 2. — P. 829–855.

[308] Li Zhize, Richtárik Peter. ZeroSARAH: Efficient Nonconvex Finite-Sum Optimization with Zero Full Gradient Computation // arXiv preprint arXiv:2103.01447. — 2021. — URL: https://arxiv.org/pdf/2103.01447 (visited on June 9, 2025).

[309] Fang Cong, Li Chris Junchi, Lin Zhouchen, Zhang Tong. Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator // arXiv preprint arXiv:1807.01695. — 2018. — URL: https://arxiv.org/pdf/1807.01695 (visited on June 9, 2025).

[310] Hazan Elad, Kale Satyen. Projection-free online learning // arXiv preprint arXiv:1206.4657. — 2012. — URL: https://arxiv.org/pdf/1206.4657 (visited on June 9, 2025).

[311] Lan Guanghui, Zhou Yi. Conditional gradient sliding for convex optimization // SIAM Journal on Optimization. — 2016. — Vol. 26, no. 2. — P. 1379–1409.

[312] Hazan Elad, Luo Haipeng. Variance-reduced and projection-free stochastic optimization // International Conference on Machine Learning. — 2016. — Vol. 48. — P. 1263–1271.

[313] Qu Chao, Li Yan, Xu Huan. Non-convex Conditional Gradient Sliding // Proceedings of the 35th International Conference on Machine Learning. — 2018. — Vol. 80. — P. 4208–4217.

[314] Yurtsever Alp, Sra Suvrit, Cevher Volkan. Conditional Gradient Methods via Stochastic Path-Integrated Differential Estimator // Proceedings of the 36th International Conference on Machine Learning. — 2019. — Vol. 97. — P. 7282–7291.

[315] Gao Hongchang, Huang Heng. Can Stochastic Zeroth-Order Frank-Wolfe Method Converge

Faster for Non-Convex Problems? // Proceedings of the 37th International Conference on Machine Learning. — 2020. — Vol. 119. — P. 3377–3386.

[316] Shen Zebang, Fang Cong, Zhao Peilin et al. Complexities in Projection-Free Stochastic Non-convex Minimization // Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics. — 2019. — Vol. 89. — P. 2868–2876.

[317] Mokhtari Aryan, Hassani Hamed, Karbasi Amin. Stochastic conditional gradient methods: From convex minimization to submodular maximization // Journal of machine learning research. — 2020. — Vol. 21, no. 105. — P. 1–49.

[318] Akhtar Zeeshan, Rajawat Ketan. Momentum based Projection Free Stochastic Optimization Under Affine Constraints // American Control Conference (ACC). — 2021. — P. 2619–2624.

[319] Hou Jie, Zeng Xianlin, Wang Gang et al. Distributed momentum-based Frank-Wolfe algorithm for stochastic optimization // IEEE/CAA Journal of Automatica Sinica. — 2023. — Vol. 10, no. 3. — P. 685–699.

[320] Négiar Geoffrey, Dresdner Gideon, Tsai Alicia et al. Stochastic Frank-Wolfe for constrained finite-sum minimization // International Conference on Machine Learning. — 2020. — P. 7253–7262.

[321] Lu Haihao, Freund Robert M. Generalized stochastic Frank–Wolfe algorithm with stochastic "substitute" gradient for structured convex optimization // Mathematical Programming. — 2021. — Vol. 187, no. 1. — P. 317–349.

[322] Weber Melanie, Sra Suvrit. Projection-free nonconvex stochastic optimization on Riemannian manifolds // IMA Journal of Numerical Analysis. — 2022. — Vol. 42, no. 4. — P. 3241–3271.

[323] Zhang Mingrui, Shen Zebang, Mokhtari Aryan et al. One sample stochastic frank-wolfe // International Conference on Artificial Intelligence and Statistics. — 2020. — Vol. 108. — P. 4012–4023.

[324] Allen-Zhu Zeyuan. Katyusha x: Practical momentum method for stochastic sum-of-nonconvex optimization // arXiv preprint arXiv:1802.03866. — 2018. — URL: `https://arxiv.org/pdf/1802.03866` (visited on June 9, 2025).

[325] Bellet Aurélien, Liang Yingyu, Garakani Alireza Bagheri et al. A distributed frank-wolfe algorithm for communication-efficient sparse learning // Proceedings of the 2015 SIAM international conference on data mining. — 2015. — P. 478–486.

[326] Taskar Ben, Chatalbashev Vassil, Koller Daphne, Guestrin Carlos. Learning structured prediction models: A large margin approach // Proceedings of the 22nd international conference on Machine learning. — 2005. — P. 896–903.

[327] Chen Pin-Yu, Zhang Huan, Sharma Yash et al. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models // Proceedings of the 10th ACM workshop on artificial intelligence and security. — 2017. — P. 15–26.

[328] Lian Xiangru, Huang Yijun, Li Yuncheng, Liu Ji. Asynchronous parallel stochastic gradient for nonconvex optimization // Advances in neural information processing systems. — 2015. — Vol. 28. — P. 2737–2745.

[329] Duchi John C, Bartlett Peter L, Wainwright Martin J. Randomized smoothing for stochastic optimization // SIAM Journal on Optimization. — 2012. — Vol. 22, no. 2. — P. 674–701.

[330] Tu Chun-Chen, Ting Paishun, Chen Pin-Yu et al. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks // Proceedings of the AAAI conference on artificial intelligence. — 2019. — Vol. 33, no. 1. — P. 742–749.

[331] Bai Yang, Wang Yisen, Zeng Yuyuan et al. Query efficient black-box adversarial attack on deep neural networks // Pattern Recognition. — 2023. — Vol. 133. — P. 109037.

[332] Lei Yuheng, Chen Jianyu, Li Shengbo Eben, Zheng Sifa. Zeroth-order actor-critic // arXiv preprint arXiv:2201.12518. — 2022. — URL: `https://arxiv.org/pdf/2201.12518` (visited on June 9, 2025).

[333] Nakashima So, Kobayashi Tetsuya J. Ancestral Reinforcement Learning: Unifying Zeroth-Order Optimization and Genetic Algorithms for Reinforcement Learning // arXiv preprint arXiv:2408.09493. — 2024. — URL: `https://arxiv.org/pdf/2408.09493` (visited on June 9, 2025).

[334] Richtárik Peter, Takáč Martin. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function // Mathematical Programming. — 2014. — Vol. 144, no. 1. — P. 1–38.

[335] Wright Stephen J. Coordinate descent algorithms // Mathematical programming. — 2015. — Vol. 151, no. 1. — P. 3–34.

[336] Nesterov Yurii, Stich Sebastian U. Efficiency of the Accelerated Coordinate Descent Method on Structured Optimization Problems // SIAM Journal on Optimization. — 2017. — Vol. 27, no. 1. — P. 110–123.

[337] Gasnikov Alexander, Novitskii Anton, Novitskii Vasilii et al. The power of first-order smooth optimization for black-box non-smooth problems // International Conference on Machine Learning. — 2022. — Vol. 162. — P. 7241–7265.

[338] Statkevich Ekaterina, Bondar Sofiya, Dvinskikh Darina et al. Gradient-free algorithm for saddle point problems under overparametrization // Chaos, Solitons & Fractals. — 2024. — Vol. 185. — P. 115048.

[339] Gasnikov Alexander, Lagunovskaya Anastasia, Usmanova Ilnura, Fedorenko Fedor. Gradient-free proximal methods with inexact oracle for convex stochastic nonsmooth optimization problems on the simplex // Automation and Remote Control. — 2016. — Vol. 77, no. 11. — P. 2018–2034.

[340] Akhavan Arya, Chzhen Evgenii, Pontil Massimiliano, Tsybakov Alexandre. A gradient estimator via l1-randomization for online zero-order optimization with two point feedback // Advances in Neural Information Processing Systems. — 2022. — Vol. 35. — P. 7685–7696.

[341] Shamir Ohad. An Optimal Algorithm for Bandit and Zero-Order Convex Optimization with Two-Point Feedback // Journal of Machine Learning Research. — 2017. — Vol. 18, no. 52. — P. 1–11.

[342] Gorbunov Eduard, Dvurechensky Pavel, Gasnikov Alexander. An Accelerated Method for Derivative-Free Smooth Stochastic Convex Optimization // SIAM Journal on Optimization. — 2022. — Vol. 32, no. 2. — P. 1210–1238.

[343] Lacoste-Julien Simon, Jaggi Martin, Schmidt Mark, Pletscher Patrick. Block-coordinate Frank-Wolfe optimization for structural SVMs // International Conference on Machine Learning. — 2013. — Vol. 28, no. 1. — P. 53–61.

[344] Dvinskikh Darina, Tominin Vladislav, Tominin Iaroslav, Gasnikov Alexander. Noisy zeroth-order optimization for non-smooth saddle point problems // International Conference on Mathematical Optimization Theory and Operations Research. — 2022. — P. 18–33.

[345] Lobanov Aleksandr, Anikin Anton, Gasnikov Alexander et al. Zero-Order Stochastic Conditional Gradient Sliding Method for Non-smooth Convex Optimization // arXiv preprint arXiv:2303.02778. — 2023. — URL: `https://arxiv.org/pdf/2303.02778` (visited on June 9, 2025).

[346] Sahu Anit Kumar, Zaheer Manzil, Kar Soummya. Towards gradient free and projection free stochastic optimization // The 22nd International Conference on Artificial Intelligence and Statistics. — 2019. — Vol. 89. — P. 3468–3477.

[347] Akhtar Zeeshan, Rajawat Ketan. Zeroth and first order stochastic Frank-Wolfe algorithms for constrained optimization // IEEE Transactions on Signal Processing. — 2022. — Vol. 70. — P. 2119–2135.

[348] Lobanov Aleksandr, Veprikov Andrew, Konin Georgiy et al. Non-smooth setting of stochastic decentralized convex optimization problem over time-varying graphs // Computational Management Science. — 2023. — Vol. 20, no. 48.

[349] Ji Kaiyi, Wang Zhe, Zhou Yi, Liang Yingbin. Improved zeroth-order variance reduced algorithms and analysis for nonconvex optimization // International conference on machine learning. — 2019. — Vol. 97. — P. 3100–3109.

[350] Risteski Andrej, Li Yuanzhi. Algorithms and matching lower bounds for approximately-convex optimization // Advances in Neural Information Processing Systems. — 2016. — Vol. 29. — P. 4752–4760.

[351] Sahu Anit Kumar, Jakovetic Dusan, Bajovic Dragana, Kar Soummya. Distributed zeroth

order optimization over random networks: A Kiefer-Wolfowitz stochastic approximation approach // IEEE Conference on Decision and Control (CDC). — 2018. — P. 4951–4958.

[352] Liu Sijia, Kailkhura Bhavya, Chen Pin-Yu et al. Zeroth-order stochastic variance reduction for nonconvex optimization // Advances in Neural Information Processing Systems. — 2018. — Vol. 31. — P. 3727–3737.

[353] Bogolubsky Lev, Dvurechenskii Pavel, Gasnikov Alexander et al. Learning supervised pagerank with gradient-based and gradient-free optimization methods // Advances in neural information processing systems. — 2016. — Vol. 29. — P. 4914–4922.

[354] Bayandina Anastasia Sergeevna, Gasnikov Alexander V, Lagunovskaya Anastasia A. Gradient-free two-point methods for solving stochastic nonsmooth convex optimization problems with small non-random noises // Automation and Remote Control. — 2018. — Vol. 79. — P. 1399–1408.

[355] Duchi John C, Jordan Michael I, Wainwright Martin J, Wibisono Andre. Optimal rates for zero-order convex optimization: The power of two function evaluations // IEEE Transactions on Information Theory. — 2015. — Vol. 61, no. 5. — P. 2788–2806.

[356] Beznosikov Aleksandr, Sadiev Abdurakhmon, Gasnikov Alexander. Gradient-free methods with inexact oracle for convex-concave stochastic saddle-point problem // Communications in Computer and Information Science. — 2020. — Vol. 1275. — P. 105–119.

[357] Flaxman Abraham D, Kalai Adam Tauman, McMahan H Brendan. Online convex optimization in the bandit setting: gradient descent without a gradient // arXiv preprint arXiv:cs/0408007. — 2004. — URL: `https://arxiv.org/pdf/cs/0408007` (visited on June 9, 2025).

[358] Gasnikov Alexander V, Krymova Ekaterina A, Lagunovskaya Anastasia A et al. Stochastic online optimization. Single-point and multi-point non-linear multi-armed bandits. Convex and strongly-convex case // Automation and remote control. — 2017. — Vol. 78. — P. 224–234.

[359] Akhavan Arya, Pontil Massimiliano, Tsybakov Alexandre. Exploiting higher order smoothness in derivative-free optimization and continuous bandits // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 9017–9027.

[360] Beznosikov Aleksandr, Novitskii Vasilii, Gasnikov Alexander. One-point gradient-free methods for smooth and non-smooth saddle-point problems // Mathematical Optimization Theory and Operations Research: 20th International Conference. — 2021. — P. 144–158.

[361] Agarwal Alekh, Foster Dean P, Hsu Daniel J et al. Stochastic convex optimization with bandit feedback // Advances in Neural Information Processing Systems. — 2011. — Vol. 24. — P. 1035–1043.

[362] Bach Francis, Perchet Vianney. Highly-smooth zero-th order online optimization // Conference on Learning Theory. — 2016. — Vol. 49. — P. 257–283.

[363] Dvurechensky Pavel, Gorbunov Eduard, Gasnikov Alexander. An accelerated directional derivative method for smooth stochastic convex optimization // European Journal of Operational Research. — 2021. — Vol. 290, no. 2. — P. 601–621.

[364] Deng Li. The mnist database of handwritten digit images for machine learning research // IEEE Signal Processing Magazine. — 2012. — Vol. 29, no. 6. — P. 141–142.

[365] Wang Yu-Xiang, Sadhanala Veeranjaneyulu, Dai Wei et al. Parallel and distributed block-coordinate frank-wolfe algorithms // International Conference on Machine Learning. — 2016. — Vol. 48. — P. 1548–1557.

[366] Li Zhize, Richtárik Peter. A unified analysis of stochastic gradient methods for nonconvex federated optimization // arXiv preprint arXiv:2006.07013. — 2020. — URL: `https://arxiv.org/pdf/2006.07013` (visited on June 9, 2025).

[367] Khaled Ahmed, Sebbouh Othmane, Loizou Nicolas et al. Unified analysis of stochastic gradient methods for composite convex and smooth optimization // arXiv preprint arXiv:2006.11573. — 2020. — URL=`https://arxiv.org/pdf/2006.11573` (visited on June 9, 2025).

[368] Richtárik Peter, Takáč Martin. On optimal probabilities in stochastic coordinate descent methods // Optimization Letters. — 2016. — Vol. 10, no. 6. — P. 1233–1243.

[369] Qu Zheng, Richtárik Peter. Coordinate descent with arbitrary sampling I: Algorithms and complexity // Optimization Methods and Software. — 2016. — Vol. 31, no. 5. — P. 829–857.

[370] Alistarh Dan, Hoefler Torsten, Johansson Mikael et al. The convergence of sparsified gradient methods // Advances in Neural Information Processing Systems. — 2018. — Vol. 31. — P. 5977–5987.

[371] Duchi John, Shalev-Shwartz Shai, Singer Yoram, Chandra Tushar. Efficient projections onto the l 1-ball for learning in high dimensions // Proceedings of the 25th international conference on Machine learning. — 2008. — P. 272–279.

[372] Shalev-Shwartz Shai, Singer Yoram, Srebro Nathan, Cotter Andrew. Pegasos: Primal estimated sub-gradient solver for svm // Mathematical programming. — 2011. — Vol. 127, no. 1. — P. 3–30.

[373] Takác Martin, Bijral Avleen, Richtárik Peter, Srebro Nati. Mini-batch primal and dual methods for SVMs // International Conference on Machine Learning. — 2013. — Vol. 28, no. 3. — P. 1022–1030.

[374] Bottou Léon, Curtis Frank E, Nocedal Jorge. Optimization methods for large-scale machine learning // Siam Review. — 2018. — Vol. 60, no. 2. — P. 223–311.

[375] Nguyen Lam M, Nguyen Phuong Ha, Richtárik Peter et al. New Convergence Aspects of Stochastic Gradient Algorithms // Journal of Machine Learning Research. — 2019. — Vol. 20. — P. 1–49.

[376] Khaled Ahmed, Richtárik Peter. Better theory for SGD in the nonconvex world // arXiv preprint arXiv:2002.03329. — 2020. — URL: https://arxiv.org/pdf/2002.03329 (visited on June 9, 2025).

[377] Li Bingcong, Ma Meng, Giannakis Georgios B. On the convergence of SARAH and beyond // International Conference on Artificial Intelligence and Statistics. — 2020. — Vol. 108. — P. 223–233.

[378] Liu Deyi, Nguyen Lam M, Tran-Dinh Quoc. An optimal hybrid variance-reduced algorithm for stochastic composite nonconvex optimization // arXiv preprint arXiv:2008.09055. — 2020. — URL: https://arxiv.org/pdf/2008.09055 (visited on June 9, 2025).

[379] Bottou Léon. Curiously fast convergence of some stochastic gradient descent algorithms // Proceedings of the symposium on learning and data science, Paris. — 2009. — Vol. 8. — P. 2624–2633.

[380] Recht Benjamin, Ré Christopher. Parallel stochastic gradient algorithms for large-scale matrix completion // Mathematical Programming Computation. — 2013. — Vol. 5, no. 2. — P. 201–226.

[381] Bengio Yoshua. Practical recommendations for gradient-based training of deep architectures // Neural Networks: Tricks of the Trade: Second Edition. — 2012. — Vol. 19. — P. 437–478.

[382] Sun Ruo-Yu. Optimization for deep learning: An overview // Journal of the Operations Research Society of China. — 2020. — Vol. 8, no. 2. — P. 249–294.

[383] Ying Bicheng, Yuan Kun, Vlaski Stefan, Sayed Ali H. Stochastic learning under random reshuffling with constant step-sizes // IEEE Transactions on Signal Processing. — 2018. — Vol. 67, no. 2. — P. 474–489.

[384] Jain Prateek, Nagaraj Dheeraj, Netrapalli Praneeth. Sgd without replacement: Sharper rates for general smooth convex functions // arXiv preprint arXiv:1903.01463. — 2019. — URL: https://arxiv.org/pdf/1903.01463 (visited on June 9, 2025).

[385] Ahn Kwangjun, Yun Chulhee, Sra Suvrit. Sgd with shuffling: optimal rates without component convexity and large epoch requirements // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 17526–17535.

[386] Mishchenko Konstantin, Khaled Ragab Bayoumi Ahmed, Richtárik Peter. Random reshuffling: Simple analysis with vast improvements // Advances in Neural Information Processing Systems. — 2020. — Vol. 33.

[387] Nguyen Lam M, Tran-Dinh Quoc, Phan Dzung T et al. A unified convergence analysis for shuffling-type gradient methods // The Journal of Machine Learning Research. — 2021. — Vol. 22, no. 1. — P. 9397–9440.

[388] Koloskova Anastasia, Doikov Nikita, Stich Sebastian U, Jaggi Martin. Shuffle SGD is Always Better than SGD: Improved Analysis of SGD with Arbitrary Data Orders // arXiv preprint

arXiv:2305.19259. — 2023. — URL: https://arxiv.org/pdf/2305.19259 (visited on June 9, 2025).

[389] Vanli Nuri Denizcan, Gurbuzbalaban Mert, Ozdaglar Asu. A stronger convergence result on the proximal incremental aggregated gradient method // arXiv preprint arXiv:1611.08022. — 2016. — URL: https://arxiv.org/pdf/1611.08022 (visited on June 9, 2025).

[390] Gurbuzbalaban Mert, Ozdaglar Asuman, Parrilo Pablo A. On the convergence rate of incremental aggregated gradient algorithms // SIAM Journal on Optimization. — 2017. — Vol. 27, no. 2. — P. 1035–1048.

[391] Mokhtari Aryan, Gurbuzbalaban Mert, Ribeiro Alejandro. Surpassing gradient descent provably: A cyclic incremental method with linear convergence rate // SIAM Journal on Optimization. — 2018. — Vol. 28, no. 2. — P. 1420–1447.

[392] Sun Tao, Sun Yuejiao, Li Dongsheng, Liao Qing. General proximal incremental aggregated gradient algorithms: Better and novel results under general scheme // Advances in Neural Information Processing Systems. — 2019. — Vol. 32.

[393] Park Youngsuk, Ryu Ernest K. Linear convergence of cyclic SAGA // Optimization Letters. — 2020. — Vol. 14, no. 6. — P. 1583–1598.

[394] Ying Bicheng, Yuan Kun, Sayed Ali H. Variance-reduced stochastic learning under random reshuffling // IEEE Transactions on Signal Processing. — 2020. — Vol. 68. — P. 1390–1408.

[395] Malinovsky Grigory, Sailanbayev Alibek, Richtárik Peter. Random Reshuffling with Variance Reduction: New Analysis and Better Rates // arXiv preprint arXiv:2104.09342. — 2021. — URL: https://arxiv.org/pdf/2104.09342 (visited on June 9, 2025).

[396] Huang Xinmeng, Yuan Kun, Mao Xianghui, Yin Wotao. An Improved Analysis and Rates for Variance Reduction under Without-replacement Sampling Orders // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 3232–3243.

[397] Nesterov Yurii. Introductory lectures on convex optimization: A basic course. — Springer Science & Business Media, 2013. — Vol. 87.

[398] Ghadimi Saeed, Lan Guanghui, Zhang Hongchao. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization // Mathematical Programming. — 2016. — Vol. 155, no. 1-2. — P. 267–305.

[399] Cohen Michael, Diakonikolas Jelena, Orecchia Lorenzo. On acceleration with noise-corrupted gradients // International Conference on Machine Learning. — 2018. — Vol. 80. — P. 1019–1028.

[400] Sapio Amedeo, Canini Marco, Ho Chen-Yu et al. Scaling Distributed Machine Learning with In-Network Aggregation // arXiv preprint arXiv:1903.06701. — 2019. — URL: https://arxiv.org/pdf/1903.06701 (visited on June 9, 2025).

[401] Sun Haobo, Shao Yingxia, Jiang Jiawei et al. Sparse Gradient Compression for Distributed SGD // Database Systems for Advanced Applications. — 2019. — P. 139–155.

[402] Wu Jiaxiang, Huang Weidong, Huang Junzhou, Zhang Tong. Error Compensated Quantized SGD and its Applications to Large-scale Distributed Optimization // Proceedings of the 35th International Conference on Machine Learning. — 2018. — Vol. 80. — P. 5325–5333.

[403] Zhao Shen-Yi, Xie Yinpeng, Gao Hao, Li Wu-Jun. Global Momentum Compression for Sparse Communication in Distributed SGD // arXiv preprint arXiv:1905.12948. — 2019. — URL: https://arxiv.org/pdf/1905.12948 (visited on June 9, 2025).

[404] Ajalloeian Ahmad, Stich Sebastian U. On the Convergence of SGD with Biased Gradients // arXiv preprint arXiv:2008.00051. — 2021. — URL: https://arxiv.org/pdf/2008.00051 (visited on June 9, 2025).

[405] Gorbunov Eduard, Kovalev Dmitry, Makarenko Dmitry, Richtárik Peter. Linearly converging error compensated SGD // 34th Conference on Neural Information Processing Systems. — 2020. — P. 20889–20900.

[406] Wen Wei, Xu Cong, Yan Feng et al. TernGrad: ternary gradients to reduce communication in distributed deep learning // Proceedings of the 31st International Conference on Neural Information Processing Systems. — 2017. — Vol. 1. — P. 1508–1518.

[407] Arnold Barry C., Balakrishnan N., Nagaraja H. N. A First Course in order Statistics. — John Wiley and Sons Inc., 1992.

[408] Agarwal Saurabh, Wang Hongyi, Lee Kangwook et al. Accordion: Adaptive gradient communication via critical learning regime identification // arXiv preprint arXiv:2010.16248. — 2020. — URL: https://arxiv.org/pdf/2010.16248 (visited on June 9, 2025).

[409] Zhu Yukun, Kiros Ryan, Zemel Rich et al. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books // Proceedings of the IEEE international conference on computer vision. — 2015. — P. 19–27.

[410] Goodfellow Ian J., Bengio Yoshua, Courville Aaron. Deep Learning. — Cambridge, MA, USA : MIT Press, 2016. — http://www.deeplearningbook.org.

[411] Van der Vaart Aad W. Asymptotic statistics. — Cambridge university press, 2000. — Vol. 3.

[412] Sutton Richard S., Barto Andrew G. Reinforcement Learning: An Introduction. — Second edition. — The MIT Press, 2018.

[413] Schulman John, Levine Sergey, Abbeel Pieter et al. Trust region policy optimization // International conference on machine learning. — 2015. — Vol. 37. — P. 1889–1897.

[414] Mnih Volodymyr, Kavukcuoglu Koray, Silver David et al. Human-level control through deep reinforcement learning // Nature. — 2015. — Vol. 518, no. 7540. — P. 529–533.

[415] Ghadimi Saeed, Lan Guanghui. Stochastic first-and zeroth-order methods for nonconvex stochastic programming // SIAM Journal on Optimization. — 2013. — Vol. 23, no. 4. — P. 2341–2368.

[416] Dieuleveut Aymeric, Flammarion Nicolas, Bach Francis. Harder, better, faster, stronger convergence rates for least-squares regression // The Journal of Machine Learning Research. — 2017. — Vol. 18, no. 1. — P. 3520–3570.

[417] Bhandari Jalaj, Russo Daniel, Singal Raghav. A finite time analysis of temporal difference learning with linear function approximation // Conference on learning theory. — 2018. — Vol. 75. — P. 1691–1692.

[418] Srikant Rayadurgam, Ying Lei. Finite-time error bounds for linear stochastic approximation andtd learning // Conference on Learning Theory. — 2019. — Vol. 99. — P. 2803–2830.

[419] Durmus Alain, Moulines Eric, Naumov Alexey et al. On the stability of random matrix product with markovian noise: Application to linear stochastic approximation and td learning // Conference on Learning Theory. — 2021. — Vol. 134. — P. 1711–1752.

[420] Lopes Cassio G., Sayed Ali H. Incremental Adaptive Strategies Over Distributed Networks // IEEE Transactions on Signal Processing. — 2007. — Vol. 55, no. 8. — P. 4064–4077.

[421] Dimakis Alexandros G., Kar Soummya, Moura José M. F. et al. Gossip Algorithms for Distributed Signal Processing // Proceedings of the IEEE. — 2010. — Vol. 98, no. 11. — P. 1847–1864.

[422] Mao Xianghui, Yuan Kun, Hu Yubin et al. Walkman: A Communication-Efficient Random-Walk Algorithm for Decentralized Optimization // IEEE Transactions on Signal Processing. — 2020. — Vol. 68. — P. 2513–2528.

[423] Hu Chonghai, Pan Weike, Kwok James. Accelerated gradient methods for stochastic optimization and online learning // Advances in Neural Information Processing Systems. — 2009. — Vol. 22. — P. 781–789.

[424] Cotter Andrew, Shamir Ohad, Srebro Nati, Sridharan Karthik. Better mini-batch algorithms via accelerated gradient methods // Advances in neural information processing systems. — 2011. — Vol. 24. — P. 1647–1655.

[425] Stochastic first order methods in smooth convex optimization : Rep. / CORE ; Executor: Olivier Devolder et al. : 2011.

[426] Lin Qihang, Chen Xi, Pena Javier. A smoothing stochastic gradient method for composite optimization // Optimization Methods and Software. — 2014. — Vol. 29, no. 6. — P. 1281–1301.

[427] Dvurechensky Pavel, Gasnikov Alexander. Stochastic intermediate gradient method for convex problems with stochastic inexact oracle // Journal of Optimization Theory and Applications. — 2016. — Vol. 171. — P. 121–145.

[428] Gasnikov Alexander Vladimirovich, Nesterov Yu E. Universal method for stochastic composite optimization problems // Computational Mathematics and Mathematical Physics. — 2018. — Vol. 58. — P. 48–64.

[429] Aybat Necdet Serhat, Fallah Alireza, Gurbuzbalaban Mert, Ozdaglar Asuman. A universally optimal multistage accelerated stochastic gradient method // Advances in neural information processing systems. — 2019. — Vol. 32. — P. 8493–8504.

[430] Gorbunov Eduard, Danilova Marina, Shibaev Innokentiy et al. Near-optimal high probability complexity bounds for non-smooth stochastic optimization with heavy-tailed noise // arXiv preprint arXiv:2106.05958. — 2021. — URL: https://arxiv.org/pdf/2106.05958 (visited on June 9, 2025).

[431] Woodworth Blake E, Srebro Nathan. An even more optimal stochastic optimization algorithm: minibatching and interpolation learning // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 7333–7345.

[432] Sutskever Ilya, Martens James, Dahl George, Hinton Geoffrey. On the importance of initialization and momentum in deep learning // Proceedings of the 30th International Conference on Machine Learning. — 2013. — Vol. 28, no. 3. — P. 1139–1147.

[433] Lan Guanghui. First-order and Stochastic Optimization Methods for Machine Learning. — 2020. — 01. — ISBN: 978-3-030-39567-4.

[434] Duchi John C, Agarwal Alekh, Johansson Mikael, Jordan Michael I. Ergodic mirror descent // SIAM Journal on Optimization. — 2012. — Vol. 22, no. 4. — P. 1549–1578.

[435] Dorfman Ron, Levy Kfir Yehuda. Adapting to mixing time in stochastic optimization with markovian data // International Conference on Machine Learning. — 2022. — Vol. 162. — P. 5429–5446.

[436] Sun Tao, Sun Yuejiao, Yin Wotao. On Markov chain gradient descent // Advances in neural information processing systems. — 2018. — Vol. 31.

[437] Doan Thinh T. Finite-Time Analysis of Markov Gradient Descent // IEEE Transactions on Automatic Control. — 2023. — Vol. 68, no. 4. — P. 2140–2153.

[438] Chen You-Lin, Na Sen, Kolar Mladen. Convergence Analysis of Accelerated Stochastic Gradient Descent under the Growth Condition // arXiv preprint arXiv:2006.06782. — 2020. — URL: https://arxiv.org/pdf/2006.06782 (visited on June 9, 2025).

[439] Doan Thinh T, Nguyen Lam M, Pham Nhan H, Romberg Justin. Convergence rates of accelerated markov gradient descent with applications in reinforcement learning // arXiv preprint arXiv:2002.02873. — 2020. — URL: https://arxiv.org/pdf/2002.02873 (visited on June 9, 2025).

[440] Even Mathieu. Stochastic Gradient Descent under Markovian Sampling Schemes // arXiv

preprint arXiv:2302.14428. — 2023. — URL: https://arxiv.org/pdf/2302.14428 (visited on June 9, 2025).

[441] Liu Chaoyue, Belkin Mikhail. Accelerating sgd with momentum for over-parameterized learning // arXiv preprint arXiv:1810.13395. — 2018. — URL: https://arxiv.org/pdf/1810.13395 (visited on June 9, 2025).

[442] Sun Tao, Li Dongsheng, Wang Bao. Adaptive Random Walk Gradient Descent for Decentralized Optimization // Proceedings of the 39th International Conference on Machine Learning. — 2022. — Vol. 162. — P. 20790–20809.

[443] Wang Puyu, Lei Yunwen, Ying Yiming, Zhou Ding-Xuan. Stability and Generalization for Markov Chain Stochastic Gradient Methods // arXiv preprint arXiv:2209.08005. — 2022. — URL: https://arxiv.org/pdf/2209.08005 (visited on June 9, 2025).

[444] Nagaraj Dheeraj, Wu Xian, Bresler Guy et al. Least squares regression with markovian data: Fundamental limits and algorithms // Advances in neural information processing systems. — 2020. — Vol. 33. — P. 16666–16676.

[445] Jiang Houyuan, Xu Huifu. Stochastic Approximation Approaches to the Stochastic Variational Inequality Problem // IEEE Transactions on Automatic Control. — 2008. — Vol. 53, no. 6. — P. 1462–1475.

[446] Iusem Alfredo N, Jofré Alejandro, Thompson Philip. Incremental constraint projection methods for monotone stochastic variational inequalities // Mathematics of Operations Research. — 2019. — Vol. 44, no. 1. — P. 236–263.

[447] Sutton Richard S. Learning to predict by the methods of temporal differences // Machine learning. — 1988. — Vol. 3. — P. 9–44.

[448] Williams Ronald J. Simple statistical gradient-following algorithms for connectionist reinforcement learning // Machine learning. — 1992. — Vol. 8. — P. 229–256.

[449] Paulin Daniel. Concentration inequalities for Markov chains by Marton couplings and spectral methods // Electronic Journal of Probability. — 2015. — Vol. 20. — P. 1 – 32.

[450] Vaswani Sharan, Mishkin Aaron, Laradji Issam et al. Painless Stochastic Gradient: Interpolation, Line-Search, and Convergence Rates // CoRR. — 2019.

[451] Glynn Peter W, Rhee Chang-han. Exact estimation for Markov chain equilibrium expectations // Journal of Applied Probability. — 2014. — Vol. 51, no. A. — P. 377–389.

[452] Giles Michael B. Multilevel Monte Carlo Path Simulation // Operations Research. — 2008. — Vol. 56, no. 3. — P. 607–617.

[453] Wolfer Geoffrey, Kontorovich Aryeh. Estimating the mixing time of ergodic markov chains // Conference on Learning Theory. — 2019. — Vol. 99. — P. 3120–3159.

[454] Jain Prateek, Kakade Sham M, Kidambi Rahul et al. Accelerating stochastic gradient descent for least squares regression // Conference On Learning Theory. — 2018. — Vol. 75. — P. 545–604.

[455] Aamari Eddie, Levrard Clément. Nonasymptotic rates for manifold, tangent space and curvature estimation // The Annals of Statistics. — 2019. — Vol. 47, no. 1. — P. 177 – 204.

[456] Yu Bin. Assouad, Fano, and Le Cam // Festschrift for Lucien Le Cam: Research Papers in Probability and Statistics. — 1997. — P. 423–435.

[457] Kidambi Rahul, Netrapalli Praneeth, Jain Prateek, Kakade Sham. On the insufficiency of existing momentum schemes for stochastic optimization // 2018 Information Theory and Applications Workshop (ITA). — 2018. — P. 1–9.

[458] Jain Prateek, Kakade Sham M., Kidambi Rahul et al. Parallelizing Stochastic Gradient Descent for Least Squares Regression: Mini-batching, Averaging, and Model Misspecification // Journal of Machine Learning Research. — 2018. — Vol. 18, no. 223. — P. 1–42.

[459] Kulkarni Viraj, Kulkarni Milind, Pant Aniruddha. Survey of personalization techniques for federated learning // 2020 fourth world conference on smart trends in systems, security and sustainability (WorldS4). — 2020. — Vol. 1. — P. 794–797.

[460] Hanzely Filip, Richtárik Peter. Federated learning of a mixture of global and local models // arXiv preprint arXiv:2002.05516. — 2020. — URL: https://arxiv.org/pdf/2002.05516 (visited on June 9, 2025).

[461] Hanzely Filip, Hanzely Slavomír, Horváth Samuel, Richtárik Peter. Lower bounds and optimal algorithms for personalized federated learning // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 2304–2315.

[462] Hanzely Filip, Zhao Boxin, Kolar Mladen. Personalized federated learning: A unified framework and universal optimization techniques // arXiv preprint arXiv:2102.09743. — 2021. — URL: https://arxiv.org/pdf/2102.09743 (visited on June 9, 2025).

[463] Li Huan, Fang Cong, Yin Wotao, Lin Zhouchen. Decentralized Accelerated Gradient Methods With Increasing Penalty Parameters // IEEE Transactions on Signal Processing. — 2020. — Vol. 68. — P. 4855–4870.

[464] Gorbunov Eduard, Dvinskikh Darina, Gasnikov Alexander. Optimal decentralized distributed algorithms for stochastic convex optimization // arXiv preprint arXiv:1911.07363. — 2019. — URL: https://arxiv.org/pdf/1911.07363 (visited on June 9, 2025).

[465] Smith Virginia, Chiang Chao-Kai, Sanjabi Maziar, Talwalkar Ameet S. Federated multi-task learning // Advances in neural information processing systems. — 2017. — Vol. 30. — P. 4425–4435.

[466] Wang Weiran, Wang Jialei, Kolar Mladen, Srebro Nathan. Distributed stochastic multi-task

learning with graph regularization // arXiv preprint arXiv:1802.03830. — 2018. — URL: https://arxiv.org/pdf/1802.03830 (visited on June 9, 2025).

[467] Fazel Maryam, Ge Rong, Kakade Sham, Mesbahi Mehran. Global convergence of policy gradient methods for the linear quadratic regulator // International conference on machine learning. — 2018. — Vol. 80. — P. 1467–1476.

[468] Choromanski Krzysztof, Rowland Mark, Sindhwani Vikas et al. Structured evolution with compact architectures for scalable policy optimization // International Conference on Machine Learning. — 2018. — Vol. 80. — P. 970–978.

[469] Salimans Tim, Ho Jonathan, Chen Xi et al. Evolution strategies as a scalable alternative to reinforcement learning // arXiv preprint arXiv:1703.03864. — 2017. — URL: https://arxiv.org/pdf/1703.03864 (visited on June 9, 2025).

[470] Bubeck Sébastien, Cesa-Bianchi Nicolo et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems // Foundations and Trends® in Machine Learning. — 2012. — Vol. 5, no. 1. — P. 1–122.

[471] Novitskii Vasilii, Gasnikov Alexander. Improved exploitation of higher order smoothness in derivative-free optimization // Optimization Letters. — 2022. — Vol. 16, no. 7. — P. 2059–2071.

[472] Alkousa Mohammad S, Gasnikov Alexander Vladimirovich, Dvinskikh Darina Mikhailovna et al. Accelerated methods for saddle-point problem // Computational Mathematics and Mathematical Physics. — 2020. — Vol. 60, no. 11. — P. 1787–1809.

[473] Lan Guanghui, Ouyang Yuyuan. Mirror-prox sliding methods for solving a class of monotone variational inequalities // arXiv preprint arXiv:2111.00996. — 2021. — URL: https://arxiv.org/pdf/2111.00996 (visited on June 9, 2025).

[474] Lin Qihang, Xu Yangyang. Inexact accelerated proximal gradient method with line search and reduced complexity for affine-constrained and bilinear saddle-point structured convex problems // arXiv preprint arXiv:2201.01169. — 2022. — URL: https://arxiv.org/pdf/2201.01169 (visited on June 9, 2025).

[475] Dvinskikh Darina, Omelchenko Sergey, Gasnikov Alexander, Tyurin Alexander. Accelerated gradient sliding for minimizing a sum of functions // Doklady Mathematics. — 2020. — Vol. 101, no. 3. — P. 244–246.

[476] Ivanova Anastasiya, Vorontsova Evgeniya, Pasechnyuk Dmitry et al. Oracle complexity separation in convex optimization // arXiv preprint arXiv:2002.02706. — 2020. — URL: https://arxiv.org/pdf/2002.02706 (visited on June 9, 2025).

[477] Lan Guanghui. Lectures on Optimization Methods for Machine Learning. — H. Milton Stewart School of Industrial and Systems Engineering Georgia Institute of Technology, Atlanta, GA, 2019.

[478] Akhavan Arya, Pontil Massimiliano, Tsybakov Alexandre. Distributed zero-order optimization under adversarial noise // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 10209–10220.

[479] Minsker Stanislav et al. Geometric median and robust estimation in Banach spaces // Bernoulli. — 2015. — Vol. 21, no. 4. — P. 2308–2335.

[480] Gasnikov Alexander Vladimirovich, Dvinskikh Darina Mikhailovna, Dvurechensky Pavel Evgenievich et al. Accelerated meta-algorithm for convex optimization problems // Computational Mathematics and Mathematical Physics. — 2021. — Vol. 61, no. 1. — P. 17–28.

[481] Wang Jianyu, Charles Zachary, Xu Zheng et al. A field guide to federated optimization // arXiv preprint arXiv:2107.06917. — 2021. — URL: https://arxiv.org/pdf/2107.06917 (visited on June 9, 2025).

[482] Pethick Thomas, Xie Wanyun, Antonakopoulos Kimon et al. Training deep learning models with norm-constrained lmos // arXiv preprint arXiv:2502.07529. — 2025. — URL: https://arxiv.org/pdf/2502.07529 (visited on June 9, 2025).

[483] Sfyraki Maria-Eleni, Wang Jun-Kun. Lions and muons: Optimization via stochastic frank-wolfe // arXiv preprint arXiv:2506.04192. — 2025. — URL: https://arxiv.org/pdf/2506.04192 (visited on June 9, 2025).

[484] Munos Remi, Valko Michal, Calandriello Daniele et al. Nash Learning from Human Feedback // International Conference on Machine Learning / PMLR. — 2024. — P. 36743–36768.

Appendix A

Missing Details

A.1 Notation and Basic Facts

Norms. We use $\langle x, y\rangle := \sum_{i=1}^{d} x_i y_i$ to denote the standard inner product of $x, y \in \mathbb{R}^d$, where x_i corresponds to the i-th component of x in the standard basis in $\mathbb{R}^d$. It induces ℓ_2-norm in $\mathbb{R}^n$ in the following way $\|x\|_2 := \sqrt{\langle x, x\rangle}$. We denote ℓ_p-norms as $\|x\|_p := \left(\sum_{i=1}^{d} |x_i|^p\right)^{1/p}$ for $p \in (1, \infty)$ and for $p = \infty$ we use $\|x\|_\infty := \max_{1\le i\le d} |x_i|$. The dual norm $\|\cdot\|_*$ for the norm $\|\cdot\|$ is defined in the following way: $\|y\|_* := \max\{\langle x, y\rangle \mid \|x\| \le 1\}$.

In most of the paper, we use the short notation $\|\cdot\| = \|\cdot\|_2$. For the Euclidean norm the dual is also equal to $\|\cdot\|_2$. In some sections we depart from this notation, where it is specified separately.

Matrices. To denote positive eigenvalues of positive semidefinite matrix $A \in \mathbb{R}^{d\times d}$ we use $\lambda_{\max}(A) = \lambda_1(A) \ge \ldots \ge \lambda_m(A) = \lambda_{\min}(A)$, we use $\lambda^+_{\min}(A)$ for a minimal positive eigenvalue, $\chi(A) := {}^{\lambda_{\max}(A)}/_{\lambda^+_{\min}(A)}$ to denote condition number of A. To define the Kronecker product of two matrices $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{n\times n}$ we use $A \otimes B \in \mathbb{R}^{nm\times nm}$. The identity matrix is denoted in our paper by I.

Others. Operator $\mathbb{E}[\cdot]$ denotes full mathematical expectation and operator $\mathbb{E}_\xi[\cdot]$ express conditional mathematical expectation w.r.t. all randomness coming from random variable ξ. Vectors $e_1, \ldots, e_d \in \mathbb{R}^d$ form a standard basis in $\mathbb{R}^d$. We use $[n]$ to define $\{1, \ldots, n\}$. We introduce $\mathrm{dom}(f)$ to describe the domain of f.

Lemma A.1.1 (Cauchy Schwartz inequality)**.** For any $a, b, x_1, \ldots, x_n \in \mathbb{R}^d$ and $c > 0$ the following inequalities hold:

$$2\langle a, b\rangle \le \frac{\|a\|^2}{c} + c\|b\|^2, \tag{A.1}$$

$$\|a + b\|^2 \le \left(1 + \frac{1}{c}\right)\|a\|^2 + (1 + c)\|b\|^2, \tag{A.2}$$

$$\left\|\sum_{i=1}^{n} x_i\right\|^2 \le n \cdot \sum_{i=1}^{n} \|x_i\|^2.$$

Lemma A.1.2 (Jensen's inequality)**.** If f is a convex function, then for any $n \in \mathbb{N}^*$ and $x_1, \ldots, x_n \in \mathbb{R}^d$ the following inequality holds:

$$f\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right) \leq \frac{1}{n}\sum_{i=1}^{n} f(x_i).$$

Lemma A.1.3 (Theorem 6.39 (iii) from [262])**.** Let g be a proper lower semicontinuous convex function and $x^+ = \text{prox}_{\gamma g}(x)$. Then for all $z \in \mathbb{R}^d$ the following inequality holds:

$$\langle x^+ - x, z - x^+ \rangle \geq \gamma \left(R(x^+) - R(z)\right).$$

Lemma A.1.4. Let $K > 0$ be a positive integer and $\eta_1, \eta_2, \ldots, \eta_K$ be random vectors such that $\mathbb{E}_k[\eta_k] := \mathbb{E}[\eta_k \mid \eta_1, \ldots, \eta_{k-1}] = 0$ for $k = 2, \ldots, K$. Then

$$\mathbb{E}\left[\left\|\sum_{k=1}^{K} \eta_k\right\|^2\right] = \sum_{k=1}^{K} \mathbb{E}[\|\eta_k\|^2].$$

Proof. We start with the following derivation:

$$\begin{aligned}
\mathbb{E}\left[\left\|\sum_{k=1}^{K} \eta_k\right\|^2\right] &= \mathbb{E}[\|\eta_K\|^2] + 2\mathbb{E}\left[\left\langle \eta_K, \sum_{k=1}^{K-1} \eta_k \right\rangle\right] + \mathbb{E}\left[\left\|\sum_{k=1}^{K-1} \eta_k\right\|^2\right] \\
&= \mathbb{E}[\|\eta_K\|^2] + 2\mathbb{E}\left[\mathbb{E}_K\left[\left\langle \eta_K, \sum_{k=1}^{K-1} \eta_k \right\rangle\right]\right] + \mathbb{E}\left[\left\|\sum_{k=1}^{K-1} \eta_k\right\|^2\right] \\
&= \mathbb{E}[\|\eta_K\|^2] + 2\mathbb{E}\left[\left\langle \mathbb{E}_K[\eta_K], \sum_{k=1}^{K-1} \eta_k \right\rangle\right] + \mathbb{E}\left[\left\|\sum_{k=1}^{K-1} \eta_k\right\|^2\right] \\
&= \mathbb{E}[\|\eta_K\|^2] + \mathbb{E}\left[\left\|\sum_{k=1}^{K-1} \eta_k\right\|^2\right].
\end{aligned}$$

Applying similar steps to $\mathbb{E}\left[\left\|\sum_{k=1}^{K-1} \eta_k\right\|^2\right], \mathbb{E}\left[\left\|\sum_{k=1}^{K-2} \eta_k\right\|^2\right], \ldots, \mathbb{E}\left[\left\|\sum_{k=1}^{2} \eta_k\right\|^2\right]$, we get the result. □

A.2 Selected Proofs

A.2.1 Proof of Theorems 2.2.4, 2.2.6

The idea of obtaining lower bounds is to find some "bad" example in the class of problems. In particular, we consider saddle point problems (as a special case of variational inequality) and look for the "bad" function among them. The paper [159] gives such an example for non-distributed deterministic problems. To get a distributed bounds, this problem needs to be divided between computing devices, which are connected in some kind of "bad" computing network [123, 139]. Then on each of the devices the function must be further divided in order to obtain a stochastic sum type problem [168, 170].

As stated above, to obtain lower bounds, we consider a particular case of the variational inequality (2.1)+(2.7)+(2.14), the saddle point problem:

$$\min_{x\in\mathbb{R}^d}\max_{y\in\mathbb{R}^d} f(x,y) := \sum_{m=1}^{M} f_m(x,y) := \sum_{m=1}^{M}\frac{1}{n}\sum_{i=1}^{n} f_{m,i}(x,y). \tag{A.3}$$

In this case $F(z) = F(x,y) = [\nabla_x f(x,y), -\nabla_y f(x,y)]$ and $g \equiv 0$. Next we rewrite Assumptions 2.0.4 and 2.0.5 (SM) for (A.3).

Assumption A.2.1. Suppose that

• each f_m is $\overline{L}$-average smooth, on $\mathbb{R}^d \times \mathbb{R}^d$, i.e. for all $x_1, x_2 \in \mathbb{R}^d$, $y_1, y_2 \in \mathbb{R}^d$ it holds that

$$\frac{1}{n}\sum_{i=1}^{n}(\|\nabla_x f_{m,i}(x_1,y_1) - \nabla_x f_{m,i}(x_2,y_2)\|^2 + \|\nabla_y f_{m,i}(x_1,y_1) - \nabla_y f_{mi}(x_2,y_2)\|^2) \\ \leq \overline{L}^2\left(\|x_1-x_2\|^2 + \|y_1-y_2\|^2\right);$$

• each f_m is L-smooth, on $\mathbb{R}^d \times \mathbb{R}^d$, i.e. for all $x_1, x_2 \in \mathbb{R}^d$, $y_1, y_2 \in \mathbb{R}^d$ it holds that

$$\|\nabla_x f_m(x_1,y_1) - \nabla_x f_m(x_2,y_2)\|^2 + \|\nabla_y f_m(x_1,y_1) - \nabla_y f_m(x_2,y_2)\|^2 \\ \leq L^2\left(\|x_1-x_2\|^2 + \|y_1-y_2\|^2\right);$$

• f is μ - strongly-convex-strongly-concave on $\mathbb{R}^d \times \mathbb{R}^d$, i.e. for all $x_1, x_2 \in \mathbb{R}^d$, $y_1, y_2 \in \mathbb{R}^d$ it holds that

$$\langle \nabla_x f(x_1,y_1) - \nabla_x f(x_2,y_2); x_1 - x_2\rangle - \langle \nabla_y f(x_1,y_1) - \nabla_y f(x_2,y_2); y_1 - y_2\rangle \\ \geq \mu\left(\|x_1-x_2\|^2 + \|y_1-y_2\|^2\right).$$

Next we rewrite Definition 2.2.3 for (A.3). Since $g \equiv 0$, then $\text{prox}_{\rho g}(z) = z$.

Definition A.2.2 (Oracle)**.** Each agent m has its own local memories $\mathcal{M}_m^x$ and $\mathcal{M}_m^y$ for the x- and y-variables, respectively–with initialization $\mathcal{M}_m^x = \mathcal{M}_m^y = \{0\}$. $\mathcal{M}_m^x$ and $\mathcal{M}_m^x$ are updated as follows.

• **Local computation:** At each local iteration device m can sample uniformly and independently batch S_m of any size b from $\{f_{m,i}\}$ and adds to its $\mathcal{M}_m^x$ and $\mathcal{M}_m^y$ a finite number of points x, y, satisfying

$$x \in \text{span}\{x' \ , \ \sum_{i_m \in S_m} \nabla_x f_{m,i_m}(x'', y'')\},$$
$$y \in \text{span}\{y' \ , \ \sum_{i_m \in S_m} \nabla_y f_{m,i_m}(x'', y'')\},$$

for given $x', x'' \in \mathcal{M}_m^x$ and $y', y'' \in \mathcal{M}_m^y$. Such call needs b local computations to collect the batch. Batch of the size n is equal to the full f_m;

• **Communication:** Based upon communication rounds among neighbouring nodes, at the communication with the number t, $\mathcal{M}_m^x$ and $\mathcal{M}_m^y$ are updated according to

$$\mathcal{M}_m^x := \text{span}\left\{\bigcup_{(i,m)\in\mathcal{E}(t)} \mathcal{M}_i^x\right\}, \quad \mathcal{M}_m^y := \text{span}\left\{\bigcup_{(i,m)\in\mathcal{E}(t)} \mathcal{M}_i^y\right\}.$$

• **Output:** The final global output is calculated as:

$$\hat{x} \in \text{span}\left\{\bigcup_{m=1}^{M} \mathcal{M}_m^x\right\}, \quad \hat{y} \in \text{span}\left\{\bigcup_{m=1}^{M} \mathcal{M}_m^y\right\}.$$

We construct the following bilinearly functions with $\overline{L}, \mu$. Let us consider a communication graph G with vertexes $\{1, \dots M\}$. Define $B = \{1\}$ and $\bar{B} = \{M\}$, with $|B| = |\bar{B}| = 1$. We then construct the following bilinear functions on the graph:

$$f_m(x,y) = \begin{cases} f_1(x,y) = \frac{\overline{L}}{4\sqrt{n}} x^T A_1 y + \frac{\mu}{6}\|x\|^2 - \frac{\mu}{6}\|y\|^2 + \frac{\overline{L}^2}{2n\mu} e_1^T y, & m \in B; \\ f_2(x,y) = \frac{\overline{L}}{4\sqrt{n}} x^T A_2 y + \frac{\mu}{6}\|x\|^2 - \frac{\mu}{6}\|y\|^2, & m \in \bar{B}; \\ f_3(x,y) = \frac{1}{M-2}\left(\frac{\mu}{6}\|x\|^2 - \frac{\mu}{6}\|y\|^2\right), & \text{otherwise;} \end{cases} \tag{A.4}$$

where $e_1 = (1, 0 \ldots, 0)$ and

$$A_1 = \begin{pmatrix} 1 & 0 & & & & & & \\ & 1 & -2 & & & & & \\ & & 1 & 0 & & & & \\ & & & 1 & -2 & & & \\ & & & & \cdots & \cdots & & \\ & & & & & 1 & -2 & \\ & & & & & & 1 & 0 \\ & & & & & & & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & -2 & & & & & & \\ & 1 & 0 & & & & & \\ & & 1 & -2 & & & & \\ & & & 1 & 0 & & & \\ & & & & \cdots & \cdots & & \\ & & & & & 1 & 0 & \\ & & & & & & 1 & -2 \\ & & & & & & & 1 \end{pmatrix}.$$

Then we construct functions $f_{m,i}$. Let define $a_{1,q}$ qth row of the matrix A_1. Then we split function f_1 to finite sum $\frac{1}{n}\sum_{i=1}^n f_{1,i}$ in following way:

$$\begin{aligned} f_{1,i} =& \sqrt{n} \cdot \frac{\overline{L}}{4} x^T \left[\sum_{j \equiv (i-1) \mod n} \left(e_{2j+1} a_{1,2j+1}^T + e_{2j+2} a_{1,2j+2}^T \right) \right] y \\ &+ \frac{\mu}{6}\|x\|^2 - \frac{\mu}{6}\|y\|^2 + \frac{\overline{L}^2}{2n\mu} e_1^T y \end{aligned} \tag{A.5}$$

Let define $A_{1,i} = \sum_{j \equiv (i-1) \mod n} \left(e_{2j+1} a_{1,2j+1}^T + e_{2j+2} a_{1,2j+2}^T \right)$. The same way we can construct $f_{2,i}$ with $A_{2,i}$ and $f_{3,i}$ with $A_{3,i} = 0$. Consider the global objective function:

$$\begin{aligned} f(x,y) &= \sum_{m=1}^M f_m(x,y) = |B| \cdot f_1(x,y) + |\bar{B}| \cdot f_2(x,y) + (M - |B| - |\bar{B}|) \cdot f_3(x,y) \\ &= \frac{\overline{L}}{2\sqrt{n}} x^T A y + \frac{\mu}{2}\|x\|^2 - \frac{\mu}{2}\|y\|^2 + \frac{\overline{L}^2}{2n\mu} e_1^T y, \end{aligned} \tag{A.6}$$

with $A = \frac{1}{2}(A_1 + A_2)$.

Lemma A.2.3. Problem (A.4) with f_m from (A.5) satisfies Assumption A.2.1 with $\overline{L}, L = \frac{\overline{L}}{\sqrt{n}}, \mu$.

Proof. It is easy to verify that f is μ - strongly-convex-strongly-concave. Also with $\|A_1\|, \|A_2\| \leq 3$, we get that $f_m(x,y)$ is $L = \frac{\overline{L}}{\sqrt{n}}$ - smooth. Then we need to check that f_m is $\overline{L}$ - average smooth:

$$\frac{1}{n}\sum_{i=1}^n \left[\|\nabla_x f_{1,i}(x_1,y_1) - \nabla_x f_{1,i}(x_2,y_2)\|^2 + \|\nabla_y f_{1,i}(x_1,y_1) - \nabla_y f_{1,i}(x_2,y_2)\|^2 \right]$$

$$
\begin{aligned}
&= \frac{1}{n}\sum_{i=1}^{n}\left[\left\|\sqrt{n}\cdot\frac{\overline{L}}{4}A_{1,i}(y_1-y_2)+\frac{\mu}{3}(x_1-x_2)\right\|^2+\left\|\sqrt{n}\cdot\frac{\overline{L}}{4}A_{1,i}^T(x_2-x_1)+\frac{\mu}{3}(y_1-y_2)\right\|^2\right]\\
&\le \frac{1}{n}\sum_{i=1}^{n}\left[\frac{nL^2}{8}\left\|A_{1,i}(y_1-y_2)\right\|^2+\frac{2\mu^2}{9}\left\|x_1-x_2\right\|^2+\frac{nL^2}{8}\left\|A_{1,i}^T(x_1-x_2)\right\|^2+\frac{2\mu^2}{9}\left\|y_1-y_2\right\|^2\right]\\
&= \frac{\overline{L}^2}{8}\sum_{i=1}^{n}\left[(y_1-y_2)^TA_{1,i}^TA_{1,i}(y_1-y_2)+(x_1-x_2)^TA_{1,i}A_{1,i}^T(x_1-x_2)\right]\\
&\quad+\frac{2\mu^2}{9}\left\|x_1-x_2\right\|^2+\frac{2\mu^2}{9}\left\|y_1-y_2\right\|^2\\
&= \frac{\overline{L}^2}{8}\sum_{i=1}^{n}(y_1-y_2)^T\left[\sum_{j\equiv(i-1)\bmod n}\left(a_{1,2j+1}e_{2j+1}^T+a_{1,2j+2}e_{2j+2}^T\right)\right]\\
&\qquad\qquad\left[\sum_{j\equiv(i-1)\bmod n}\left(e_{2j+1}a_{1,2j+1}^T+e_{2j+2}a_{1,2j+2}^T\right)\right](y_1-y_2)\\
&\quad+\frac{\overline{L}^2}{8}\sum_{i=1}^{n}(x_1-x_2)^T\left[\sum_{j\equiv(i-1)\bmod n}\left(e_{2j+1}a_{1,2j+1}^T+e_{2j+2}a_{1,2j+2}^T\right)\right]\\
&\qquad\qquad\left[\sum_{j\equiv(i-1)\bmod n}\left(a_{1,2j+1}e_{2j+1}^T+a_{1,2j+2}e_{2j+2}^T\right)\right](x_1-x_2)\\
&\quad+\frac{2\mu^2}{9}\left\|x_1-x_2\right\|^2+\frac{2\mu^2}{9}\left\|y_1-y_2\right\|^2\\
&= \frac{\overline{L}^2}{8}\sum_{i=1}^{n}(y_1-y_2)^T\left[\sum_{j\equiv(i-1)\bmod n}\left(a_{1,2j+1}a_{1,2j+1}^T+a_{1,2j+2}a_{1,2j+2}^T\right)\right](y_1-y_2)\\
&\quad+\frac{\overline{L}^2}{8}\sum_{i=1}^{n}(x_1-x_2)^T\left[\sum_{j\equiv(i-1)\bmod n}\left(e_{2j+1}e_{2j+1}^T+5e_{2j+2}e_{2j+2}^T\right)\right]^T(x_1-x_2)\\
&\quad+\frac{2\mu^2}{9}\left\|x_1-x_2\right\|^2+\frac{2\mu^2}{9}\left\|y_1-y_2\right\|^2\\
&\le \frac{\overline{L}^2}{8}\sum_{j=1}^{d}(y_1-y_2)^T\left(a_{1,j}a_{1,j}^T\right)(y_1-y_2)+\frac{5\overline{L}^2}{8}\sum_{j=1}^{d}(x_1-x_2)^T\left(e_je_j^T\right)(x_1-x_2)\\
&\quad+\frac{2\mu^2}{9}\left\|x_1-x_2\right\|^2+\frac{2\mu^2}{9}\left\|y_1-y_2\right\|^2\\
&\le \frac{\overline{L}^2}{8}\sum_{j=1}^{d}(y_1-y_2)^T\left(a_{1,j}a_{1,j}^T\right)(y_1-y_2)+\frac{5\overline{L}^2}{8}\left\|x_1-x_2\right\|^2\\
&\quad+\frac{2\mu^2}{9}\left\|x_1-x_2\right\|^2+\frac{2\mu^2}{9}\left\|y_1-y_2\right\|^2\\
&\le \frac{3\overline{L}^2}{4}\left\|y_1-y_2\right\|^2+\frac{5\overline{L}^2}{8}\left\|x_1-x_2\right\|^2+\frac{2\mu^2}{9}\left\|x_1-x_2\right\|^2+\frac{2\mu^2}{9}\left\|y_1-y_2\right\|^2.
\end{aligned}
$$

The last inequality follows from $\lambda_{\max}\left(\sum\limits_{j=1}^{d}\left(a_{1,j}a_{1,j}^T\right)\right) \leq 6$. Finally, with $\mu \leq \overline{L}$ we get

$$\frac{1}{n}\sum_{i=1}^{n}\left[\|\nabla_x f_{1,i}(x_1,y_1) - \nabla_x f_{1,i}(x_2,y_2)\|^2 + \|\nabla_y f_{1,i}(x_1,y_1) - \nabla_y f_{1,i}(x_2,y_2)\|^2\right] \leq \overline{L}^2\left(\|x_1-x_2\|^2 + \|y_1-y_2\|^2\right).$$

□

The next two lemmas give an idea of how quickly we approximate the solution of (A.6) depending on the number of communications and local iterations. For simplicity, we divide a situation in two parts: one part is devoted to communications (taking into account the fact that we are not limited in the number of local iterations), the second - on the contrary (we concentrate on local computations and assume that communications cost nothing).

Lemma A.2.4. Let the problem (A.4) be solved by any method that satisfies Definition A.2.2. Then after K communication rounds, only the first $\left\lfloor \frac{K}{l} \right\rfloor$ coordinates of the global output can be non-zero while the rest of the $d - \left\lfloor \frac{K}{l} \right\rfloor$ coordinates are strictly equal to zero. Here l is "distance" between B and $\bar{B}$ (how quickly can we transfer information from B to $\bar{B}$).

Proof. We begin introducing some notation for our proof. Let

$$E_0 := \{0\}, \quad E_K := \text{span}\{e_1, \ldots, e_K\}.$$

Note that, the initialization gives $\mathcal{M}_m^x = E_0$, $\mathcal{M}_m^y = E_0$.

Suppose that, for some m, $\mathcal{M}_m^x = E_K$ and $\mathcal{M}_m^y = E_K$, at some given time. Let us analyze how $\mathcal{M}_m^x, \mathcal{M}_m^y$ can change by performing only local computations.

Firstly, we consider the case when K odd. After one local update, we have the following:

- For machines m which own f_1, it holds

$$\begin{aligned} x &\in \text{span}\left\{e_1\ ,\ x'\ ,\ A_1 y'\right\} = E_K, \\ y &\in \text{span}\left\{e_1\ ,\ y'\ ,\ A_1^T x'\right\} = E_K, \end{aligned} \tag{A.7}$$

for given $x' \in \mathcal{M}_m^x$ and $y' \in \mathcal{M}_m^y$. In details, each local iteration uses matrices $A_{1,i}$ (in the stochastic) or A_1 (in the deterministic). But here we talks only about communications and do not pay attention to the number of local iterations. Therefore, without loss of generality, we can immediately assume that all local calculations change our output according to (A.7). Since A_1 has a block diagonal structure, after local computations, we have $\mathcal{M}_m^x = E_K$ and $\mathcal{M}_m^y = E_K$. The situation does not change, no matter how many local computations one does.

- For machines m which own f_2, it holds

$$x \in \text{span}\{x' \ , \ A_2 y'\} = E_{K+1},$$
$$y \in \text{span}\{y' \ , \ A_2^T x'\} = E_{K+1},$$

for given $x' \in \mathcal{M}_m^x$ and $y' \in \mathcal{M}_m^y$. It means that, after local computations, one has $\mathcal{M}_m^x = E_{K+1}$ and $\mathcal{M}_m^y = E_{K+1}$. Therefore, machines with function f_2 can progress by one new non-zero coordinate.

This means that we constantly have to transfer progress from the group of machines with f_1 to the group of machines with f_2 and back. Initially, all devices have zero coordinates. Further, machines with f_1 can receive the first nonzero coordinate (but only the first, the second is not), and the rest of the devices are left with all zeros. Next, we pass the first non-zero coordinate to machines with f_2. To do this, l communication rounds are needed. By doing so, they can make the second coordinate non-zero, and then transfer this progress to the machines with f_1. Then the process continues in the same way. This completes the proof. □

In the next lemma, we will give an understanding of how local progress towards a solution occurs. For this, we will assume that communications cost nothing.

Lemma A.2.5. Let Problem (A.4) be solved by any method that satisfies Definition A.2.2. Then after N local calls (for each node), in expectation only the first $\left\lfloor \frac{N}{n} \right\rfloor$ coordinates of the global output can be non-zero while the rest of the $d - \left\lfloor \frac{N}{n} \right\rfloor$ coordinates are strictly equal to zero.

Proof. As is clear from the previous lemma, communications make sense if an update ($E_K \to E_{K+1}$) is reached on one of the nodes. Depending on K, this happens on the nodes with f_1 or f_2 (but not simultaneously). The question is how many local calls should be made to get this update. One can understand it by looking at the structure of matrices $A_{1,i}$ and $A_{2,i}$ from (A.5). Only one of n matrices is suitable for us. For example, in case of $K = 2k$, we need $f_{1,j}$ with $j \equiv k \bmod n$.

Suppose that $s_1, s_2 \ldots$ times call stochastic oracle with batchsize $1, 2 \ldots$. Then $\sum_{j=1}^{n} j s_j + N$. Due to the fact that the choice of batches is random and uniform, the random variable responsible for the total number of updates during the operation of the algorithm has the sum of binomial distribution with pairs of parameters $\left(s_j; \frac{j}{n}\right)$. It remains only to take the mathematical expectation and the lemma is proved. □

The next lemma is devoted to provide an approximate solution of problem (A.6), and shows that this approximation is close to a real solution. The proof of the lemma follows closely that of Lemma 3.3 from [159], and is reported for the sake of completeness.

Lemma A.2.6 (Lemma 3.3 from [159])**.** Let $\alpha = \frac{2n\mu^2}{L^2}$ and $q = \frac{1}{2}\left(2 + \alpha - \sqrt{\alpha^2 + 4\alpha}\right) \in (0; 1)$–the

smallest root of $q^2 - (2+\alpha)q + 1 = 0$; and let define

$$\bar{y}_i^* = \frac{q^i}{1-q}, \quad i \in [d].$$

The following bound holds when $\bar{y}^* := [y_1^*, \dots y_d^*]^\top$ is used to approximate the solution y^*:

$$\|\bar{y}^* - y^*\| \leq \frac{q^{d+1}}{\alpha(1-q)}.$$

Proof. Let us write the dual function for (A.6):

$$h(y) = -\frac{1}{2} y^T \left(\frac{\overline{L}^2}{2n\mu} A^T A + \mu I \right) y + \frac{\overline{L}^2}{2n\mu} e_1^T y,$$

where it is not difficult to check that

$$AA^T = \begin{pmatrix} 1 & -1 & & & & & & \\ -1 & 2 & -1 & & & & & \\ & -1 & 2 & -1 & & & & \\ & & -1 & 2 & -1 & & & \\ & & & -1 & 2 & -1 & & \\ & & & & & \dots & & \\ & & & & & -1 & 2 & -1 \\ & & & & & & -1 & 2 \end{pmatrix}.$$

The optimality of dual problem $\nabla h(y^*) = 0$ gives

$$\left(\frac{\overline{L}^2}{2n\mu} A^T A + \mu I \right) y^* = \frac{\overline{L}^2}{2n\mu} e_1,$$

or

$$\left(A^T A + \alpha I \right) y^* = e_1.$$

Equivalently, we can write

$$\begin{cases} (1+\alpha)y_1^* - y_2^* = 1, \\ -y_1^* + (2+\alpha)y_2^* - y_3^* = 0, \\ \dots \\ -y_{d-2}^* + (2+\alpha)y_{d-1}^* - y_d^* = 0, \\ -y_{d-1}^* + (2+\alpha)y_d^* = 0. \end{cases}$$

On the other hand, the approximation $\bar{y}^*$ satisfies the following set of equations:

$$\begin{cases} (1+\alpha)\bar{y}_1^* - \bar{y}_2^* = 1, \\ -\bar{y}_1^* + (2+\alpha)\bar{y}_2^* - \bar{y}_3^* = 0, \\ \dots \\ -\bar{y}_{d-2}^* + (2+\alpha)\bar{y}_{d-1}^* - \bar{y}_d^* = 0, \\ -\bar{y}_{d-1}^* + (2+\alpha)\bar{y}_d^* = \frac{q^{d+1}}{1-q}, \end{cases}$$

or equivalently

$$\left(A^T A + \alpha I\right)\bar{y}^* = e_1 + \frac{q^{d+1}}{1-q}e_d.$$

Therefore, the difference between $\bar{y}^*$ and y^* reads

$$\bar{y}^* - y^* = \left(A^T A + \alpha I\right)^{-1}\frac{q^{d+1}}{1-q}e_d.$$

The statement of the lemma follow from the above equality and $\alpha^{-1}I \succeq \left(A^T A + \alpha I\right)^{-1} \succ 0$. □

The next lemma provides a lower bound for the solution of (A.6) in the distributed case (A.4). The proof follows closely that of Lemma 3.4 from [159] and is reported for the sake of completeness.

Lemma A.2.7. Consider a distributed saddle-point problem with objective function given by (A.6). For any K, N, choose any problem size $d \geq \max\left\{2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right), 2K, 2N\right\}$, where $\alpha = \frac{2n\mu^2}{L^2}$ and $q = \frac{1}{2}\left(2+\alpha-\sqrt{\alpha^2+4\alpha}\right) \in (0;1)$. Then, any output $\hat{x}, \hat{y}$ produced by any method satisfying Definition A.2.2 after K communications and N local calls, is such that

$$\mathbb{E}\left[\|\hat{x}-x^*\|^2 + \|\hat{y}-y^*\|^2\right] \geq \left(q^{\frac{2K}{l}} + q^{\frac{2N}{n}}\right)\frac{\|y_0-y^*\|^2}{16}.$$

Proof. Let us assume that in output we have k non-zero coordinates. By definition of $\bar{y}^*$, with $q < 1$ and $k \leq \frac{d}{2}$, we have

$$\begin{aligned}\|\hat{y} - \bar{y}^*\|^2 &\geq \sqrt{\sum_{j=k+1}^{d} (\bar{y}_j^*)^2} = \frac{q^k}{1-q}\sqrt{q^2 + q^4 + \ldots + q^{2(d-k)}} \\ &\geq \frac{q^k}{\sqrt{2}(1-q)}\sqrt{q^2 + q^4 + \ldots + q^{2d}} = \frac{q^k}{\sqrt{2}}\|\bar{y}^*\|^2 = \frac{q^k}{\sqrt{2}}\|y_0 - \bar{y}^*\|^2.\end{aligned}$$

Using Lemma A.2.6 for $d \geq 2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right)$ we can guarantee that $\bar{y}^* \approx y^*$ (for more detailed proof see [159]) and

$$\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2 \geq \|\hat{y} - y^*\|^2 \geq \frac{q^{2k}}{16}\|y_0 - y^*\|^2.$$

It remains only to note that k depends on the number of nonzero coordinates from communications k_c and local computations k_l. For this we use Lemmas A.2.4 and A.2.5:

$$\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2 \geq \frac{q^{2k}}{16}\|y_0 - y^*\|^2 \geq \frac{q^{2\min(k_c,k_l)}}{16}\|y_0 - y^*\|^2 \geq \frac{q^{k_c+k_l}}{16}\|y_0 - y^*\|^2.$$

By Lemma A.2.4 we have $k_c \leq \left\lfloor \frac{K}{l} \right\rfloor$, where l is "distance" between B and $\bar{B}$. By Lemma A.2.5 we get that k_l has binomial distribution with parameters N and $\frac{1}{n}$.

$$\begin{aligned}\mathbb{E}\left[\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2\right] &\geq \frac{q^{2\left\lfloor \frac{K}{l} \right\rfloor}}{32}\|y_0 - y^*\|^2 + \mathbb{E}\left[\frac{q^{2k_l}}{32}\|y_0 - y^*\|^2\right] \\ &\geq \frac{q^{\frac{2K}{l}}}{32}\|y_0 - y^*\|^2 + \frac{\mathbb{E}\left[q^{2k_l}\right]}{32}\|y_0 - y^*\|^2 \\ &\geq \left(q^{\frac{2K}{l}} + q^{2\mathbb{E}[k_l]}\right)\cdot\frac{1}{32}\|y_0 - y^*\|^2 \\ &\geq \left(q^{\frac{2K}{l}} + q^{\frac{2N}{n}}\right)\cdot\frac{1}{32}\|y_0 - y^*\|^2.\end{aligned}$$

Here we use Jensen's inequality. □

It remains to get an estimate on l ("distance" between B and $\bar{B}$). For the fixed network it is real distance, for time-varying – rate how fast information transmits in the network from B to $\bar{B}$.

Fixed network

Theorem A.2.8 (Theorem 2.2.4)**.** Let $\overline{L} > \mu > 0$, $n \in \mathbb{N}$ (with $\overline{L}/\mu \geq \sqrt{n}$), $\chi \geq 1$ and $K, N \in \mathbb{N}$. There exists a distributed saddle-point problem over fixed network (Assumption 2.1.3). For which the following statements are true:

- a gossip matrix W has $\chi(W) = \chi$,
- $f = \sum\limits_{m=1}^{M} \frac{1}{n}\sum\limits_{i=1}^{n} f_{m.i} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is μ – strongly-convex-strongly-concave,

- f_m are $\overline{L}$-average smooth and $L = \frac{\overline{L}}{\sqrt{n}}$ - smooth,
- size $d \geq \max\left\{2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right), 2K, 2N\right\}$, where $\alpha = \frac{2n\mu^2}{\overline{L}^2}$ and $q = \frac{1}{2}\left(2 + \alpha - \sqrt{\alpha^2 + 4\alpha}\right) \in (0; 1)$,
- the solution of the problem is non-zero: $x^* \neq 0$, $y^* \neq 0$.

Then for any output $\hat{z}$ of any procedure (Definition A.2.2) with K communication rounds and N local computations, one can obtain the following estimate:

$$\|\hat{z} - z^*\|^2 = \Omega\left(\exp\left(-\frac{80}{1 + \sqrt{\frac{2L^2}{\mu^2} + 1}} \cdot \frac{K}{\sqrt{\chi}}\right)\|y_0 - y^*\|^2\right);$$

$$\|\hat{z} - z^*\|^2 = \Omega\left(\exp\left(-\frac{16}{n + \sqrt{\frac{2n\overline{L}^2}{\mu^2} + n^2}} \cdot N\right)\|y_0 - y^*\|^2\right).$$

Proof. Applying Lemma A.2.7, we have

$$\left(\frac{1}{q}\right)^{\frac{2K}{l}} \geq \frac{\|y_0 - y^*\|^2}{32\mathbb{E}\left[\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2\right]} \quad \text{and} \quad \left(\frac{1}{q}\right)^{\frac{2N}{n}} \geq \frac{\|y_0 - y^*\|^2}{32\mathbb{E}\left[\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2\right]}.$$

Taking the logarithm on both sides, we get

$$\frac{2K}{l} \geq \ln\left(\frac{\|y_0 - y^*\|^2}{32\mathbb{E}\left[\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2\right]}\right)\frac{1}{\ln(q^{-1})}.$$

Next, we work with

$$\begin{aligned}\frac{1}{\ln(q^{-1})} &= \frac{1}{\ln(1 + (1-q)/q))} \geq \frac{q}{1-q} = \frac{1 + \frac{n\mu^2}{\overline{L}^2} - \sqrt{\frac{2n\mu^2}{\overline{L}^2} + \left(\frac{n\mu^2}{\overline{L}^2}\right)^2}}{\sqrt{\frac{2n\mu^2}{\overline{L}^2} + \left(\frac{n\mu^2}{\overline{L}^2}\right)^2} - \frac{n\mu^2}{\overline{L}^2}} \\ &\geq \frac{\sqrt{\frac{2n\mu^2}{\overline{L}^2} + \left(\frac{n\mu^2}{\overline{L}^2}\right)^2} + \frac{n\mu^2}{\overline{L}^2}}{\frac{8n\mu^2}{\overline{L}^2}} = \frac{1}{8}\left(1 + \sqrt{\frac{2\overline{L}^2}{n\mu^2} + 1}\right).\end{aligned}$$

One can then obtain

$$\frac{2K}{l} \geq \ln\left(\frac{\|y_0 - y^*\|^2}{32\mathbb{E}\left[\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2\right]}\right) \cdot \frac{1}{8}\left(1 + \sqrt{\frac{2\overline{L}^2}{n\mu^2} + 1}\right),$$

and

$$\exp\left(\frac{1}{1 + \sqrt{\frac{2\overline{L}^2}{n\mu^2} + 1}}\frac{16K}{l}\right) \geq \frac{\|y_0 - y^*\|^2}{32\mathbb{E}\left[\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2\right]}.$$

The next proof follow similar steps as in the proof of Theorem 2 from [123]. Let $\gamma_M = \frac{1-\cos\frac{\pi}{M}}{1+\cos\frac{\pi}{M}}$ be a decreasing sequence of positive numbers. Since $\gamma_2 = 1$ and $\lim_m \gamma_M = 0$, there exists $M \geq 2$ such that $\gamma_M \geq \chi^{-1} > \gamma_{M+1}$.

• If $M \geq 3$, let us consider linear graph of size M with vertexes $v_1, \dots v_M$, and weighted with $w_{1,2} = 1 - a$ and $w_{i,i+1} = 1$ for $i \geq 2$. Then we applied Lemmas 1 and 3 and get:

$$\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2 \geq q^{\frac{2K}{l}} \frac{\|y_0 - y^*\|^2}{32}.$$

If W_a is the normalized Laplacian of the weighted graph $\mathcal{G}$, one can note that with $a = 0$, $\chi^{-1}(W_a) = \gamma_M$, with $a = 1 - \chi^{-1}(W_a) = 0$. Hence, there exists $a \in (0; 1]$ such that $\chi^{-1}(W_a) = \chi^{-1}$. Then $\chi^{-1} \geq \gamma_{M+1} \geq \frac{2}{(M+1)^2}$, and $M \geq \sqrt{2\chi} - 1 \geq \frac{\sqrt{\chi}}{4}$. Finally, $l = M - 1 \geq \frac{15M}{16} - 1 \geq \frac{15}{16}\left(\sqrt{2\chi} - 1\right) - 1 \geq \frac{\sqrt{\chi}}{5}$ since $\chi^{-1} \leq \gamma_3 = \frac{1}{3}$. Hence,

$$\exp\left(80\frac{K}{\sqrt{\chi}} \cdot \frac{1}{1 + \sqrt{\frac{2\overline{L}^2}{n\mu^2} + 1}}\right) \geq \frac{\|y_0 - y^*\|^2}{32(\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2)}. \tag{A.8}$$

• If $M = 2$, we construct a totally connected network with 3 nodes with weight $w_{1,3} = a \in [0; 1]$. Let W_a is the normalized Laplacian. If $a = 0$, then the network is a linear graph and $\chi^{-1}(W_a) = \gamma_3 = \frac{1}{3}$. Hence, there exists $a \in [0; 1]$ such that $\chi^{-1}(W_a) = \chi^{-1}$. Finally, $B = \{v_1\}$, $\bar{B} = \{v_3\}$ and $l \geq 1 \geq \frac{1}{2\sqrt{\chi^{-1}}}$. Whence it follows that in this case (A.8) is also valid.

The same way we can work with (but without considering graph):

$$\left(\frac{1}{q}\right)^{\frac{2N}{n}} \geq \frac{\|y_0 - y^*\|^2}{32\mathbb{E}\left[\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2\right]}.$$

□

Time-varying network

Theorem A.2.9 (Theorem 2.2.6)**.** Let $\overline{L} > \mu > 0$, $n \in \mathbb{N}$ (with $\overline{L}/\mu \geq \sqrt{n}$), $\hat{\chi} \geq 3$ and $K, N \in \mathbb{N}$. There exists a distributed saddle-point problem over time-varying network (Assumption 2.1.4). For which the following statements are true:

- Assumption 2.1.4 holds with $\chi = \hat{\chi}$,
- $f = \sum\limits_{m=1}^{M} \frac{1}{n} \sum\limits_{i=1}^{n} f_{m.i} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is μ – strongly-convex-strongly-concave,
- f_m are $\overline{L}$-average smooth and $L = \frac{\overline{L}}{\sqrt{n}}$ - smooth,
- size $d \geq \max\left\{2\log_q\left(\frac{\alpha}{4\sqrt{2}}\right), 2K, 2N\right\}$, where $\alpha = \frac{2n\mu^2}{L^2}$ and $q = \frac{1}{2}\left(2 + \alpha - \sqrt{\alpha^2 + 4\alpha}\right) \in (0; 1)$,
- the solution of the problem is non-zero: $x^* \neq 0$, $y^* \neq 0$.

Then for any output $\hat{z}$ of any procedure (Definition A.2.2) with K communication rounds and N local computations, one can obtain the following estimate:

$$\|\hat{z} - z^*\|^2 = \Omega\left(\exp\left(-\frac{64}{1+\sqrt{\frac{2L^2}{\mu^2}+1}} \cdot \frac{K}{B\hat{\chi}}\right)\|y_0 - y^*\|^2\right);$$

$$\|\hat{z} - z^*\|^2 = \Omega\left(\exp\left(-\frac{16}{n+\sqrt{\frac{2n\bar{L}^2}{\mu^2}+n^2}} \cdot N\right)\|y_0 - y^*\|^2\right).$$

Proof. The same way as in the previous Theorem we can obtain

$$\exp\left(\frac{1}{1+\sqrt{\frac{2\bar{L}^2}{n\mu^2}+1}}\frac{16K}{l}\right) \geq \frac{\|y_0 - y^*\|^2}{32\mathbb{E}\left[\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2\right]}.$$

Following [139], we can consider the next sequences of graphs. Let us choose $M = \lfloor\hat{\chi}\rfloor$. Each communication t such that $t \neq 8\tau$ (for any $\tau \in \mathbb{N}$) we consider empty network. Each communication t such that $t = 8\tau$ (for some $\tau \in \mathbb{N}$) we construct star graph with vertex $[(\tau - 1) \mod (M-2)] + 2$ in the center of this star. It means that in the 8th communication vertex 2 is in the center; in the 16th communication vertex 3 is in the center etc. One can note that only vertexes with f_3 are in the center and they change sequentially. As matrices $W(t)$ we consider the normalized Laplacians. Then it holds $\chi = M = \lfloor\hat{\chi}\rfloor$. It holds that Assumption 2.1.4 is valid with $\hat{\chi}$.

It is left to estimate l. Suppose we need to transfer information from B to $\bar{B}$. At best:

- vertex j in the center, it means that we can transfer information to j;
- $B-1$ empty graphs – "empty" communications;
- vertex $j+1$ in the center, it means that we can transfer information to $j+1$;

...

- vertex $j-1$ in the center, it means that we can transfer information to $j-1$ (still there is no information in $\bar{B}$);
- $B-1$ empty graphs – "empty" communications;
- vertex j in the center, now we can transfer from j to $\bar{B}$.

In this (the best variant) we spend $M - 1 + (B-1)(M-2)$ communication rounds. It means that $l \geq M - 1 + (B-1)(M-2) \geq B(M-2) = B(\lfloor\hat{\chi}\rfloor - 2) \geq \frac{B\hat{\chi}}{4}$ (for $\hat{\chi} \geq 3$). Then we get

$$\exp\left(\frac{1}{1+\sqrt{\frac{2\bar{L}^2}{n\mu^2}+1}}\frac{64K}{B\hat{\chi}}\right) \geq \frac{\|y_0 - y^*\|^2}{32\mathbb{E}\left[\|\hat{x} - x^*\|^2 + \|\hat{y} - y^*\|^2\right]}.$$

□

A.2.2 Proof of Theorem 2.2.8

We start the proof from the following lemma on δ^k and $\Delta^{k+1/2}$ from Algorithm 6.

Lemma A.2.10. The following inequality holds:

$$\mathbb{E}_k\left[\left\|\delta^k - \mathbb{E}_k\left[\delta^k\right]\right\|^2\right] \leq \frac{2\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \alpha^2\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2. \tag{A.9}$$

$$\mathbb{E}_k\left[\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2\right] \leq \frac{2\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 + 2L^2\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2. \tag{A.10}$$

where $\mathbb{E}_k\left[\delta^k\right]$ is equal to

$$\mathbb{E}_k\left[\delta^k\right] = F(\mathbf{z}^k) + \alpha(F(\mathbf{z}^k) - F(\mathbf{z}^{k-1})). \tag{A.11}$$

Proof. Due to the fact that the batch S^k is generated uniformly and independently for all workers, we can make sure that (A.11) is correct. Then using definition of Δ^k from Algorithm 6, we get

$$\begin{aligned}\mathbb{E}_k\left[\left\|\delta^k - \mathbb{E}_k\left[\delta^k\right]\right\|^2\right] \leq\ & 2\mathbb{E}_k\left[\left\|\frac{1}{b}\sum_{j\in S^k}\left(\mathbf{F}_j(\mathbf{z}^k) - \mathbf{F}_j(\mathbf{w}^{k-1})\right) - \left(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{w}^{k-1})\right)\right\|^2\right] \\ & + 2\mathbb{E}_k\left[\left\|\frac{\alpha}{b}\sum_{j\in S^k}\left(\mathbf{F}_j(\mathbf{z}^k) - \mathbf{F}_j(\mathbf{z}^{k-1})\right) - \left(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1})\right)\right\|^2\right]\end{aligned}$$

By randomness and independence of indexes in S^k, we obtain

$$\begin{aligned}\mathbb{E}_k\left[\left\|\delta^k - \mathbb{E}_k\left[\delta^k\right]\right\|^2\right] =\ & \frac{2}{b^2}\mathbb{E}_k\left[\sum_{j\in S^k}\left\|\left(\mathbf{F}_j(\mathbf{z}^k) - \mathbf{F}_j(\mathbf{w}^{k-1})\right) - \left(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{w}^{k-1})\right)\right\|^2\right] \\ & + \frac{2\alpha^2}{b^2}\mathbb{E}_k\left[\sum_{j\in S^k}\left\|\left(\mathbf{F}_j(\mathbf{z}^k) - \mathbf{F}_j(\mathbf{z}^{k-1})\right) - \left(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1})\right)\right\|^2\right]\end{aligned}$$

The property of the second moment $\mathbb{E}\|\xi - \mathbb{E}\xi\|^2 = \mathbb{E}\|\xi\|^2 - \|\mathbb{E}\xi\|^2$ gives

$$\begin{aligned}\mathbb{E}_k\left[\left\|\delta^k - \mathbb{E}_k\left[\delta^k\right]\right\|^2\right] \leq & \frac{2}{b^2}\mathbb{E}_k\left[\sum_{j\in S^k}\left\|\mathbf{F}_j(\mathbf{z}^k) - \mathbf{F}_j(\mathbf{w}^{k-1})\right\|^2\right] \\ & + \frac{2\alpha^2}{b^2}\mathbb{E}_k\left[\sum_{j\in S^k}\left\|\mathbf{F}_j(\mathbf{z}^k) - F_j(\mathbf{z}^{k-1})\right\|^2\right]\end{aligned}$$

Again from the fact that S^k is generated uniformly and independently for all samples and workers, we can obtain that for each worker m indexes $j^k_{m,1}, \ldots j^k_{m,b}$ have the same uniform distribution,

that means

$$\begin{aligned}\mathbb{E}_k\left[\left\|\delta^k-\mathbb{E}_k\left[\delta^k\right]\right\|^2\right]\leq&\frac{2}{b^2}\mathbb{E}_k\left[\sum_{j\in S^k}\left\|\mathbf{F}_j(\mathbf{z}^k)-\mathbf{F}_j(\mathbf{w}^{k-1})\right\|^2\right]\\&+\frac{2\alpha^2}{b^2}\mathbb{E}_k\left[\sum_{j\in S^k}\left\|\mathbf{F}_j(\mathbf{z}^k)-F_j(\mathbf{z}^{k-1})\right\|^2\right]\\=&\frac{2}{mb}\sum_{j=1}^m\left(\left\|\mathbf{F}_j(\mathbf{z}^k)-\mathbf{F}_j(\mathbf{w}^{k-1})\right\|^2+\alpha^2\left\|\mathbf{F}_j(\mathbf{z}^k)-\mathbf{F}_j(\mathbf{z}^{k-1})\right\|^2\right).\end{aligned}$$

Using Assumption 3.1.24, we get

$$\mathbb{E}_k\left[\left\|\delta^k-\mathbb{E}_k\left[\delta^k\right]\right\|^2\right]\leq\frac{2\overline{L}^2}{b}\left(\left\|\mathbf{z}^k-\mathbf{w}^{k-1}\right\|^2+\alpha^2\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2\right).$$

This concludes the proof of (A.9).

The proof chain for (A.10) is very similar.

$$\begin{aligned}\mathbb{E}\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2&\leq2\mathbb{E}\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^{k+1})\right\|^2+2\mathbb{E}\left\|\mathbf{F}(\mathbf{z}^{k+1})-\mathbf{F}(\mathbf{z}^*)\right\|^2\\&=\mathbb{E}\frac{2}{b^2}\mathbb{E}_{k+1/2}\sum_{j\in S^{k+1/2}}\left\|\left(\mathbf{F}_j(\mathbf{z}^{k+1})-\mathbf{F}_j(\mathbf{w}^k)\right)-\left(\mathbf{F}(\mathbf{z}^{k+1})-\mathbf{F}(\mathbf{w}^k)\right)\right\|^2\\&\quad+2L^2\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2\\&\leq\mathbb{E}\frac{2}{b^2}\mathbb{E}_{k+1/2}\sum_{j\in S^{k+1/2}}\left\|\mathbf{F}_j(\mathbf{z}^{k+1})-\mathbf{F}_j(\mathbf{w}^k)\right\|^2+2L^2\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2\\&\leq\frac{2\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{w}^k\right\|^2+2L^2\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2.\end{aligned}$$

Here, we used Assumption 3.1.24. □

Before proving the main lemma of this section, let us introduce an auxiliary notation. Throughout the proof, we denote $W^\dagger:\text{range}W\to\text{range}W$ the inverse of the map $W:\text{range}W\to\text{range}W$. And we denote $\|\mathbf{y}\|^2_{(W^\dagger\otimes I)}=\langle(W^\dagger\otimes I)y;y\rangle$.

We define the following Lyapunov function:

$$\begin{aligned}\Psi^k=&\left(\frac{1}{\eta}+\frac{3\mu}{2}\right)\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\frac{1}{\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\&+2\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k+1})-(\mathbf{y}^k-\mathbf{y}^{k+1}),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle+\frac{1}{2\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\&+\frac{1}{8\eta}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2+\frac{\gamma+\frac{1}{2}\eta\mu}{p\eta}\left\|\mathbf{w}^{k+1}-\mathbf{z}^*\right\|^2+\frac{\gamma}{2\eta}\left\|\mathbf{w}^k-\mathbf{z}^{k+1}\right\|^2.\end{aligned}\tag{A.12}$$

Here we also use

$$\mathbf{y}^*=\mathbf{P}\mathbf{F}(\mathbf{z}^*)\tag{A.13}$$

with $\mathbf{P} \in \mathbb{R}^{nd \times nd}$, an orthogonal projection matrix onto the subspace $\mathcal{L}^\perp$, given as

$$\mathbf{P} = (I_M - \tfrac{1}{M}\mathbf{1}\mathbf{1}^\top) \otimes I,$$

where $\mathbf{1}_M = (1, \ldots, 1)^\top \in \mathbb{R}^M$.

Lemma A.2.11. Consider the problem (2.16) (or (2.1) + (2.7) + (2.14)) under Assumptions 2.0.4, 2.2.1 and 2.2.2 over a fixed connected graph $\mathcal{G}$ with a gossip matrix W. Let $\{\mathbf{z}^k\}$ be the sequence generated by Algorithm 6 with parameters

$$\gamma \leq \frac{1}{8}, \quad \eta \leq \min\left\{\frac{\sqrt{\alpha\gamma b}}{\sqrt{8}\cdot\overline{L}}, \frac{1}{16L}\right\}, \quad \beta \leq \min\left\{\frac{\mu}{4L^2}; \frac{b\gamma}{4\eta\overline{L}^2}\right\}, \quad \theta \leq \min\left\{\frac{1}{2\beta}; \frac{1}{16\eta}\right\},$$
$$\alpha = \max\left[\left(1 - \frac{\mu\eta}{4}\right); \left(1 - \beta\theta\chi^{-1}\right); \left(1 - \frac{p\eta\mu}{2\gamma + \eta\mu}\right)\right].$$

Then, after k iterations we get

$$\mathbb{E}\frac{1}{2\eta}\|\mathbf{z}^k - \mathbf{z}^*\|^2 \leq \max\left[\left(1 - \frac{\mu\eta}{4}\right); \left(1 - \beta\theta\chi^{-1}\right); \left(1 - \frac{p\eta\mu}{2\gamma + \eta\mu}\right)\right]^k \cdot \Psi^0.$$

Proof. **Part 1.** We start the proof from considering update of $\mathbf{y}^{k+1}$ in Algorithm 6.

$$\frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} = \frac{1}{\theta}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} - \frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2_{(W^\dagger \otimes I)}$$
$$- 2\langle (W \otimes I)(\mathbf{z}^{k+1} - \beta(\Delta^{k+1/2} - \mathbf{y}^k)), (W^\dagger \otimes I)(\mathbf{y}^{k+1} - \mathbf{y}^*)\rangle.$$

Next, we use the fact that $(W \otimes I)(W^\dagger \otimes I) = \mathbf{P}$ and obtain

$$\frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} = \frac{1}{\theta}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} - \frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2_{(W^\dagger \otimes I)}$$
$$- 2\langle \mathbf{P}(\mathbf{z}^{k+1} - \beta(\Delta^{k+1/2} - \mathbf{y}^k)), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle.$$

One can observe, that $\mathbf{z}^* \in \mathcal{L}$ and then $\mathbf{P}\mathbf{z}^* = 0$. Additionally, using update $\mathbf{y}^{k+1}$, we can note that $\mathbf{y}^k \in \mathcal{L}^\perp$ for all $k = 0, 1, 2, \ldots$. Hence,

$$\begin{aligned}
\frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} &= \frac{1}{\theta}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} - \frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2_{(W^\dagger \otimes I)} \\
&\quad - 2\langle \mathbf{P}(\mathbf{z}^{k+1} - \mathbf{z}^*), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + 2\beta\langle \mathbf{P}(\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&\quad - 2\beta\langle \mathbf{P}\mathbf{y}^k - \mathbf{y}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&= \frac{1}{\theta}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} - \frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2_{(W^\dagger \otimes I)} \\
&\quad - 2\langle \mathbf{P}(\mathbf{z}^{k+1} - \mathbf{z}^*), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + 2\beta\langle \mathbf{P}(\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&\quad - 2\beta\langle \mathbf{P}(\mathbf{y}^k - \mathbf{y}^*), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&= \frac{1}{\theta}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} - \frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2_{(W^\dagger \otimes I)}
\end{aligned}$$

$$
\begin{aligned}
&- 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^* \rangle + 2\beta \langle \Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*), \mathbf{y}^{k+1} - \mathbf{y}^* \rangle \\
&- 2\beta \langle \mathbf{y}^k - \mathbf{y}^*, \mathbf{y}^{k+1} - \mathbf{y}^* \rangle
\end{aligned}
$$

By the simple fact $\|a+b\|^2 = \|a\|^2 + 2\langle a; b\rangle + \|b\|^2$, we get

$$
\begin{aligned}
\frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} \leq & \frac{1}{\theta}\left\|\mathbf{y}^{k} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} - \frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2_{(W^\dagger \otimes I)} \\
&- 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^* \rangle + \beta\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 + \beta\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 \\
&- \beta\left\|\mathbf{y}^{k} - \mathbf{y}^*\right\|^2 - \beta\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \beta\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
= & \frac{1}{\theta}\left\|\mathbf{y}^{k} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} - \frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2_{(W^\dagger \otimes I)} \\
&- 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^* \rangle + \beta\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 - \beta\left\|\mathbf{y}^{k} - \mathbf{y}^*\right\|^2 \\
&+ \beta\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2.
\end{aligned}
$$

Using the fact, that $\lambda^+_{\min}(W) = \chi^{-1}$ and $\lambda_{\max}(W) = 1$, we can note $-\lambda^+_{\min}(W^\dagger) \leq -1$ and $\chi^{-1}\lambda_{\max}(W^\dagger) \leq 1$ and then get

$$
\begin{aligned}
\frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} \leq & \frac{1}{\theta}\left\|\mathbf{y}^{k} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} - \frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 - 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^* \rangle \\
&+ \beta\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 - \beta\chi^{-1}\left\|\mathbf{y}^{k} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} + \beta\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
= & \left(\frac{1}{\theta} - \beta\chi^{-1}\right)\left\|\mathbf{y}^{k} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} - \left(\frac{1}{\theta} - \beta\right)\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&- 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^* \rangle + \beta\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2.
\end{aligned}
$$

Then we take a full expectation and use (A.10):

$$
\begin{aligned}
\mathbb{E}\frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} \leq & \left(\frac{1}{\theta} - \beta\chi^{-1}\right)\mathbb{E}\left\|\mathbf{y}^{k} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} - \left(\frac{1}{\theta} - \beta\right)\mathbb{E}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&- 2\mathbb{E}\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^* \rangle + 2\beta L^2 \mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \\
&+ \frac{2\beta\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2.
\end{aligned} \tag{A.14}
$$

Here we stop and put aside (A.14). We will not return to it later.

Part 2. Now we work with

$$
\begin{aligned}
\frac{1}{\eta}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 = & \frac{1}{\eta}\left\|\mathbf{z}^{k} - \mathbf{z}^*\right\|^2 + \frac{2}{\eta}\langle \mathbf{z}^{k+1} - \mathbf{z}^k, \mathbf{z}^{k+1} - \mathbf{z}^* \rangle - \frac{1}{\eta}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
= & \frac{1}{\eta}\left\|\mathbf{z}^{k} - \mathbf{z}^*\right\|^2 + \frac{2\gamma}{\eta}\langle \mathbf{w}^{k} - \mathbf{z}^k, \mathbf{z}^{k+1} - \mathbf{z}^* \rangle - 2\langle \Delta^k - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^* \rangle \\
&- \frac{1}{\eta}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- 2\langle \tfrac{1}{\eta}(\mathbf{z}^k + \gamma(\mathbf{w}^k - \mathbf{z}^k) - \eta\Delta^k - \mathbf{z}^{k+1}) + (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^* \rangle.
\end{aligned}
$$

Optimality condition for (2.1)+(2.14) it follows, that

$$-F(z^*) \in \partial g(z^*).$$

Let us define $\Delta^* \in (\mathbb{R}^d)^M$ as

$$\Delta^* = \tfrac{1}{M}[F(z^*), \dots F(z^*)]^T.$$

It is to note that $-\Delta^* \in M\partial \mathbf{g}(\mathbf{z}^*)$. On the other hand

$$\Delta^* = (\mathbf{1}_M \mathbf{1}_M^\top \otimes I)\mathbf{F}(\mathbf{z}^*) = M(\mathbf{F}(\mathbf{z}^*) - \mathbf{P}\mathbf{F}(\mathbf{z}^*)) = M(\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^*))$$

This means that $-(\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^*) \in \partial \mathbf{g}(\mathbf{z}^*)$. From update for $\mathbf{z}^{k+1}$ of Algorithm 6 it follows, that

$$\mathbf{z}^k + \gamma(\mathbf{w}^k - \mathbf{z}^k) - \eta\Delta^k - \mathbf{z}^{k+1} \in \partial(\eta \mathbf{g})(\mathbf{z}^{k+1}).$$

Hence, from monotonicity of $\partial \mathbf{g}(\cdot)$ we get

$$\begin{aligned}
\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 &\leq \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{2\gamma}{\eta}\mathbb{E}\langle \mathbf{w}^k - \mathbf{z}^k, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\quad - 2\mathbb{E}\langle \Delta^k - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle - \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&= \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{2\gamma}{\eta}\mathbb{E}\langle \mathbf{w}^k - \mathbf{z}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\quad - \frac{2\gamma}{\eta}\mathbb{E}\langle \mathbf{z}^k - \mathbf{z}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle - 2\mathbb{E}\langle \Delta^k - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\quad - \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&= \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 + \left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 - \left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&\quad - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + \left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&\quad - 2\mathbb{E}\langle \Delta^k - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle - \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&= \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&\quad - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^{k+1}\right\|^2 - 2\mathbb{E}\langle \Delta^k - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\quad - \frac{1-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2.
\end{aligned}$$

In previous we also use the simple fact $\|a+b\|^2 = \|a\|^2 + 2\langle a; b\rangle + \|b\|^2$ twice. Small rearrangement gives

$$\begin{aligned}
\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 &\leq \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&\quad - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^{k+1}\right\|^2 - \frac{1-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2
\end{aligned}$$

$$
\begin{aligned}
&- 2\mathbb{E}\langle \mathbb{E}_k\left[\delta^k\right] - (\mathbf{y}^k + \alpha(\mathbf{y}^k - \mathbf{y}^{k-1})) - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&- 2\mathbb{E}\langle \delta^k - \mathbb{E}_k\left[\delta^k\right], \mathbf{z}^{k+1} - \mathbf{z}^k\rangle - 2\mathbb{E}\langle \delta^k - \mathbb{E}_k\left[\delta^k\right], \mathbf{z}^k - \mathbf{z}^*\rangle.
\end{aligned}
$$

Using the tower property of expectation we can obtain the following:

$$
\begin{aligned}
\mathbb{E}\langle \mathbb{E}_k\left[\delta^k\right] - \delta^k, \mathbf{z}^k - \mathbf{z}^*\rangle &= \mathbb{E}\mathbb{E}_k\left[\langle \mathbb{E}_k\left[\delta^k\right] - \delta^k, \mathbf{z}^k - \mathbf{z}^*\rangle\right] \\
&= \mathbb{E}\langle \mathbb{E}_k\left[\mathbb{E}_k\left[\delta^k\right] - \delta^k\right], \mathbf{z}^k - \mathbf{z}^*\rangle \\
&= \mathbb{E}\langle \mathbb{E}_k\left[\delta^k\right] - \mathbb{E}_k\left[\delta^k\right], \mathbf{z}^k - \mathbf{z}^*\rangle = 0.
\end{aligned}
$$

Hence, with (A.11)

$$
\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \\
&\quad\le \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}*\right\|^2 - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&\qquad - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^{k+1}\right\|^2 - \frac{1-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&\qquad - 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) + \alpha(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1})) - \mathbf{F}(\mathbf{z}^*) - (\mathbf{y}^k + \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\qquad + 2\mathbb{E}\langle \mathbb{E}_k\left[\delta^k\right] - \delta^k, \mathbf{z}^{k+1} - \mathbf{z}^k\rangle.
\end{aligned}
$$

Using the Young's inequality we get

$$
\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \\
&\quad\le \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&\qquad - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^{k+1}\right\|^2 - \frac{1-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&\qquad - 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) + \alpha(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1})) - \mathbf{F}(\mathbf{z}^*) - (\mathbf{y}^k + \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\qquad + 2\eta\mathbb{E}\left\|\mathbb{E}_k\left[\delta^k\right] - \delta^k\right\|^2 + \frac{1}{2\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&\quad= \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&\qquad - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^{k+1}\right\|^2 - \frac{1/2-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 + 2\eta\mathbb{E}\left\|\mathbb{E}_k\left[\delta^k\right] - \delta^k\right\|^2 \\
&\qquad - 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) + \alpha(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1})) - \mathbf{F}(\mathbf{z}^*) - (\mathbf{y}^k + \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle.
\end{aligned}
$$

By (A.9) we get

$$
\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \\
&\quad\le \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&\qquad - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^{k+1}\right\|^2 - \frac{1/2-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2
\end{aligned}
$$

$$
\begin{aligned}
&+ \frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\alpha^2\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 \\
&- 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) + \alpha(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1})) - \mathbf{F}(\mathbf{z}^*) - (\mathbf{y}^k + \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&= \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 \\
&- \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^{k+1}\right\|^2 - \frac{1/2-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&+ \frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\alpha^2\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 \\
&- 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k+1}) + \alpha(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1})) - (\mathbf{y}^k + \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle.
\end{aligned}
$$

Assumption 2.2.2 about μ-strong monotonicity of F gives

$$
\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \\
&\leq \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - 2\mu\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&- \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^{k+1}\right\|^2 - \frac{1/2-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&+ \frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\alpha^2\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 \\
&- 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k+1}) + \alpha(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1})) - (\mathbf{y}^k + \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - \mathbf{y}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&= \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - 2\mu\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&- \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^{k+1}\right\|^2 - \frac{1/2-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&+ \frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\alpha^2\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 \\
&- 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k+1}) - (\mathbf{y}^k - \mathbf{y}^{k+1}), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle + 2\mathbb{E}\langle \mathbf{y}^{k+1} - \mathbf{y}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^*\rangle \\
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^{k+1} - \mathbf{z}^k\rangle.
\end{aligned}
$$

Part 3. Now we are ready to consider (A.14). We sum up previous expression and (A.14). After small rearrangement, we have

$$
\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + \frac{1}{\theta}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} + 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k+1}) - (\mathbf{y}^k - \mathbf{y}^{k+1}), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\leq \frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - (2\mu - 2\beta L^2)\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + \left(1 - \beta\theta\chi^{-1}\right)\cdot\frac{1}{\theta}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2_{(W^\dagger \otimes I)} \\
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^*\rangle + \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&- \left(\frac{\gamma}{\eta} - \frac{2\beta\overline{L}^2}{b}\right)\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^{k+1}\right\|^2 - \frac{1/2-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 + \frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2
\end{aligned}
$$

$$+\frac{4\alpha^2\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2-2\alpha\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1}),\mathbf{z}^{k+1}-\mathbf{z}^k\rangle$$
$$+2\alpha\mathbb{E}\langle\mathbf{y}^k-\mathbf{y}^{k-1},\mathbf{z}^{k+1}-\mathbf{z}^k\rangle-\left(\frac{1}{\theta}-\beta\right)\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2.$$

With $\theta\leq\frac{1}{2\beta}$ (or $\beta\leq\frac{1}{2\theta}$) and Young's inequality we get

$$\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\frac{1}{\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}+2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k+1})-(\mathbf{y}^k-\mathbf{y}^{k+1}),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle\\
&\quad\leq\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2-(2\mu-2\beta L^2)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\left(1-\beta\theta\chi^{-1}\right)\cdot\frac{1}{\theta}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\
&\quad\quad-2\alpha\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^k-\mathbf{z}^*\rangle+\frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2-\frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2\\
&\quad\quad-\left(\frac{\gamma}{\eta}-\frac{2\beta\overline{L}^2}{b}\right)\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^{k+1}\right\|^2-\frac{1/2-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2+\frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{w}^{k-1}\right\|^2\\
&\quad\quad+\frac{4\alpha^2\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2+\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2+8\eta\alpha^2\mathbb{E}\left\|\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})\right\|^2\\
&\quad\quad+2\theta\alpha\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2+\frac{\alpha}{2\theta}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2-\frac{1}{2\theta}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2.
\end{aligned}$$

By choosing $\theta\leq\frac{1}{16\eta}$ and $\alpha\leq 1$

$$\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\frac{1}{\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}+2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k+1})-(\mathbf{y}^k-\mathbf{y}^{k+1}),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle\\
&\quad\leq\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2-(2\mu-2\beta L^2)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\left(1-\beta\theta\chi^{-1}\right)\cdot\frac{1}{\theta}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\
&\quad\quad-2\alpha\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^k-\mathbf{z}^*\rangle+\frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2-\frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2\\
&\quad\quad-\left(\frac{\gamma}{\eta}-\frac{2\beta\overline{L}^2}{b}\right)\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^{k+1}\right\|^2-\frac{1/4-\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2+\frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{w}^{k-1}\right\|^2\\
&\quad\quad+\frac{4\alpha^2\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2+8\eta\alpha^2\mathbb{E}\left\|\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})\right\|^2+\frac{\alpha}{2\theta}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2\\
&\quad\quad-\frac{1}{2\theta}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2.
\end{aligned}$$

With L-Lipshitzness of $\mathbf{F}$ (Assumption 2.0.4) and $\gamma\leq\frac{1}{8}$, we have

$$\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\frac{1}{\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}+2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k+1})-(\mathbf{y}^k-\mathbf{y}^{k+1}),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle\\
&\quad\leq\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2-(2\mu-2\beta L^2)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\left(1-\beta\theta\chi^{-1}\right)\cdot\frac{1}{\theta}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\
&\quad\quad-2\alpha\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^k-\mathbf{z}^*\rangle+\frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2-\frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2\\
&\quad\quad-\left(\frac{\gamma}{\eta}-\frac{2\beta\overline{L}^2}{b}\right)\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^{k+1}\right\|^2-\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2+\frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{w}^{k-1}\right\|^2
\end{aligned}$$

$$+\left(\frac{32\alpha^2\overline{L}^2\eta^2}{b}+64\eta^2\alpha^2L^2\right)\cdot\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2+\frac{\alpha}{2\theta}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2$$
$$-\frac{1}{2\theta}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2.$$

Small rearrangement gives

$$\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\frac{1}{\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}+2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k+1})-(\mathbf{y}^k-\mathbf{y}^{k+1}),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle\\
&\quad+\frac{1}{2\theta}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2\\
&\leq\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2-(2\mu-2\beta L^2)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\left(1-\beta\theta\chi^{-1}\right)\cdot\frac{1}{\theta}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\
&\quad-2\alpha\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^k-\mathbf{z}^*\rangle+\frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2-\frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2\\
&\quad-\left(\frac{\gamma}{\eta}-\frac{2\beta\overline{L}^2}{b}\right)\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^{k+1}\right\|^2+\frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{w}^{k-1}\right\|^2\\
&\quad+\left(\frac{32\alpha^2\overline{L}^2\eta^2}{b}+64\eta^2\alpha^2L^2\right)\cdot\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2+\frac{\alpha}{2\theta}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2.
\end{aligned}$$

With our choice of $\eta\leq\min\left\{\frac{\sqrt{\alpha\gamma b}}{\sqrt{8}\cdot\overline{L}},\frac{1}{16L}\right\}$ and $\alpha<1$, we get

$$\begin{aligned}
&\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\frac{1}{\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}+2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k+1})-(\mathbf{y}^k-\mathbf{y}^{k+1}),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle\\
&\quad+\frac{1}{2\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2\\
&\leq\frac{1}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2-(2\mu-2\beta L^2)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\left(1-\beta\theta\chi^{-1}\right)\cdot\frac{1}{\theta}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\
&\quad-\alpha\cdot2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^k-\mathbf{z}^*\rangle+\alpha\cdot\frac{1}{2\theta}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2\\
&\quad+\alpha\cdot\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2\\
&\quad+\frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2-\frac{\gamma}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2\\
&\quad-\left(\frac{\gamma}{\eta}-\frac{2\beta\overline{L}^2}{b}\right)\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^{k+1}\right\|^2+\frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{w}^{k-1}\right\|^2.
\end{aligned}$$

Now, we add $\frac{\mu}{2}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\frac{\gamma+\frac{1}{2}\eta\mu}{p\eta}\mathbb{E}\left\|\mathbf{w}^{k+1}-\mathbf{z}^*\right\|^2$ to both sides and use update for w^{k+1} of Algorithm 6

$$\frac{\gamma+\frac{1}{2}\eta\mu}{p\eta}\mathbb{E}\mathbb{E}_{w^{k+1}}\left\|\mathbf{w}^{k+1}-\mathbf{z}^*\right\|^2=\frac{\gamma+\frac{1}{2}\eta\mu}{\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+\frac{(\gamma+\frac{1}{2}\eta\mu)(1-p)}{\eta p}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2,$$

and get

$$\begin{aligned}
&\left(\frac{1}{\eta}+\frac{\mu}{2}\right)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\frac{1}{\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\
&\quad+2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k+1})-(\mathbf{y}^k-\mathbf{y}^{k+1}),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle+\frac{\gamma+\frac{1}{2}\eta\mu}{p\eta}\mathbb{E}\left\|\mathbf{w}^{k+1}-\mathbf{z}^*\right\|^2\\
&\quad+\frac{1}{2\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2+\left(\frac{\gamma}{\eta}-\frac{2\beta\overline{L}^2}{b}\right)\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^{k+1}\right\|^2\\
&\leq\left(\frac{1}{\eta}+\frac{\mu}{2}\right)\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2-\left(\frac{3}{2}\mu-2\beta L^2\right)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2\\
&\quad+\left(1-\beta\theta\chi^{-1}\right)\cdot\frac{1}{\theta}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}-\alpha\cdot2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^k-\mathbf{z}^*\rangle\\
&\quad+\alpha\cdot\frac{1}{2\theta}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2+\alpha\cdot\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2\\
&\quad+\left(1-p+\frac{p\gamma}{\gamma+\frac{1}{2}\eta\mu}\right)\frac{(\gamma+\frac{1}{2}\eta\mu)}{\eta p}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2+\frac{4\overline{L}^2\eta}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{w}^{k-1}\right\|^2.
\end{aligned}$$

Note that $\eta\leq\min\left\{\frac{\sqrt{\alpha\gamma b}}{\sqrt{8}\cdot\overline{L}},\frac{1}{16L}\right\}$, $\beta\leq\min\left\{\frac{\mu}{4L^2};\frac{b\gamma}{4\eta\overline{L}^2}\right\}$, then we get that

$$\left(\frac{\gamma}{\eta}-\frac{2\beta\overline{L}^2}{b}\right)\geq\frac{\gamma}{2\eta};\quad-\left(\frac{3}{2}\mu-2\beta L^2\right)\leq-\mu;\quad\frac{4\overline{L}^2\eta}{b}\leq\alpha\cdot\frac{\gamma}{2\eta}.$$

Hence, it holds

$$\begin{aligned}
&\left(\frac{1}{\eta}+\frac{3\mu}{2}\right)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\frac{1}{\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\
&\quad+2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k+1})-(\mathbf{y}^k-\mathbf{y}^{k+1}),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle+\frac{\gamma+\frac{1}{2}\eta\mu}{p\eta}\mathbb{E}\left\|\mathbf{w}^{k+1}-\mathbf{z}^*\right\|^2\\
&\quad+\frac{1}{2\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2+\frac{\gamma}{2\eta}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^{k+1}\right\|^2\\
&\leq\left(\frac{1}{\eta}+\frac{\mu}{2}\right)\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+\left(1-\beta\theta\chi^{-1}\right)\cdot\frac{1}{\theta}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\
&\quad-\alpha\cdot2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^k-\mathbf{z}^*\rangle\\
&\quad+\alpha\cdot\frac{1}{2\theta}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2+\alpha\cdot\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2\\
&\quad+\left(1-\frac{p\eta\mu}{2\gamma+\eta\mu}\right)\cdot\frac{\gamma+\frac{1}{2}\eta\mu}{p\eta}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2+\alpha\cdot\frac{\gamma}{2\eta}\mathbb{E}\left\|\mathbf{x}^k-\mathbf{w}^{k-1}\right\|^2.
\end{aligned}$$

With our η we have $\eta\mu\leq1$ and then

$$\left(\frac{1}{\eta}+\frac{\mu}{2}\right)\leq\left(\frac{1}{\eta}+\frac{3\mu}{2}\right)\left(1-\frac{\mu\eta}{4}\right).$$

It gives

$$\begin{aligned}
&\left(\frac{1}{\eta}+\frac{3\mu}{2}\right)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2+\frac{1}{\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\
&\quad+2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k+1})-(\mathbf{y}^k-\mathbf{y}^{k+1}),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle+\frac{\gamma+\frac{1}{2}\eta\mu}{p\eta}\mathbb{E}\left\|\mathbf{w}^{k+1}-\mathbf{z}^*\right\|^2\\
&\quad+\frac{1}{2\theta}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2+\frac{\gamma}{2\eta}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^{k+1}\right\|^2\\
&\leq\left(1-\frac{\mu\eta}{4}\right)\cdot\left(\frac{1}{\eta}+\frac{3\mu}{2}\right)\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+\left(1-\beta\theta\chi^{-1}\right)\cdot\frac{1}{\theta}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2_{(W^\dagger\otimes I)}\\
&\quad-\alpha\cdot2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^k-\mathbf{z}^*\rangle\\
&\quad+\alpha\cdot\frac{1}{2\theta}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2+\alpha\cdot\frac{1}{8\eta}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2\\
&\quad+\left(1-\frac{p\eta\mu}{2\gamma+\eta\mu}\right)\cdot\frac{\gamma+\frac{1}{2}\eta\mu}{p\eta}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2+\alpha\cdot\frac{\gamma}{2\eta}\mathbb{E}\left\|x^k-w^{k-1}\right\|^2.
\end{aligned}$$

Definition (A.12) of the Lyapunov function and the choice $\alpha=\max\left[\left(1-\frac{\mu\eta}{4}\right);\left(1-\beta\theta\chi^{-1}\right);\left(1-\frac{p\eta\mu}{2\gamma+\eta\mu}\right)\right]$ move us to

$$\mathbb{E}\Psi^{k+1}\leq\max\left[\left(1-\frac{\mu\eta}{4}\right);\left(1-\beta\theta\chi^{-1}\right);\left(1-\frac{p\eta\mu}{2\gamma+\eta\mu}\right)\right]\mathbb{E}\Psi^k.$$

It remains to show, that

$$\Psi^k\geq\frac{1}{2\eta}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2.$$

$$\begin{aligned}
\Psi^k&\geq\frac{1}{\eta}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+\frac{1}{8\eta}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2+\frac{1}{2\theta}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2\\
&\quad+2\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^*-\mathbf{z}^k\rangle\\
&\geq\frac{1}{\eta}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+\frac{1}{8\eta}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2+\frac{1}{2\theta}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2\\
&\quad-\frac{1}{2\theta}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2-2\theta\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2-\frac{1}{8\eta L^2}\left\|\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})\right\|^2-8\eta L^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2.
\end{aligned}$$

Using L-Lipschitzness of $\mathbf{F}$ we get

$$\Psi^k\geq\frac{1}{2\eta}\left(2-8\eta^2L^2-4\theta\eta\right)\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2.$$

With $\eta\leq\frac{1}{16L}$ and $\theta\leq\frac{1}{16\eta}$ we get

$$\Psi^k\geq\frac{1}{2\eta}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2.$$

□

Theorem A.2.12 (Theorem 2.2.8)**.** Consider the problem (2.16) (or (2.1) + (2.7) + (2.14)) under Assumptions 2.0.4, 2.2.1 and 2.2.2 over a fixed connected graph $\mathcal{G}$ with a gossip matrix W.

Let $\{\mathbf{z}^k\}$ be the sequence generated by Algorithm 6 with parameters

$$\gamma = p \leq \frac{1}{8}, \quad \eta = \min\left\{\frac{\sqrt{\gamma b}}{4\overline{L}}, \frac{1}{16L\sqrt{\chi}}\right\}, \quad \beta = \min\left\{\frac{\mu}{4L^2}; \frac{b\gamma}{4\eta\overline{L}^2}\right\}, \quad \theta = \min\left\{\frac{1}{2\beta}; \frac{1}{16\eta}\right\},$$
$$\alpha = \max\left[\left(1 - \frac{\mu\eta}{4}\right); \left(1 - \beta\theta\chi^{-1}\right); \left(1 - \frac{p\eta\mu}{2\gamma + \eta\mu}\right)\right].$$

Then, given $\varepsilon > 0$, the number of iterations for $\mathbb{E}[\|\mathbf{z}^k - \mathbf{z}^*\|^2] \leq \varepsilon$ is

$$O\left(\left[\frac{1}{p} + \chi + \frac{1}{\sqrt{pb}}\frac{\overline{L}}{\mu} + \sqrt{\chi}\frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right).$$

Proof. It is easy to check that $\alpha \geq \frac{1}{2}$, then also one can verify that the choice of γ, η, β, θ, α satisfies the conditions of Lemma A.2.11. We can get that the iteration complexity of Algorithm 6:

$$\begin{aligned}
O\left(\left[1 + \frac{1}{\eta\mu} + \frac{1}{\beta\theta\chi^{-1}} + \frac{\gamma + \eta\mu}{p\eta\mu}\right]\log\frac{1}{\varepsilon}\right) &= O\left(\left[\frac{1}{p} + \frac{1}{\eta\mu} + \frac{1}{\beta\theta\chi^{-1}}\right]\log\frac{1}{\varepsilon}\right) \\
&= O\left(\left[\frac{1}{p} + \chi + \frac{1}{\eta\mu} + \frac{\chi\eta}{\beta}\right]\log\frac{1}{\varepsilon}\right) \\
&= O\left(\left[\frac{1}{p} + \chi + \frac{1}{\eta\mu} + \frac{\chi\eta^2\overline{L}^2}{bp} + \frac{\chi\eta L^2}{\mu}\right]\log\frac{1}{\varepsilon}\right) \\
&= O\left(\left[\frac{1}{p} + \chi + \frac{1}{\sqrt{pb}}\frac{\overline{L}}{\mu} + \sqrt{\chi}\frac{L}{\mu}\right]\log\frac{1}{\varepsilon}\right).
\end{aligned}$$

□

A.2.3 Proof of Theorem 2.2.10

We start the proof from the following lemma on δ^k and $\delta^{k+1/2}$ from Algorithm 7.

Lemma A.2.13. The following inequality holds:

$$\mathbb{E}_k\left[\left\|\delta^k - \mathbb{E}_k\left[\delta^k\right]\right\|^2\right] \leq \frac{2\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \alpha^2\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2. \tag{A.15}$$

$$\mathbb{E}_k\left[\left\|\delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2\right] \leq \frac{2\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 + 2L^2\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2. \tag{A.16}$$

where $\mathbb{E}_k\left[\delta^k\right]$ is equal to

$$\mathbb{E}_k\left[\delta^k\right] = F(\mathbf{z}^k) + \alpha(F(\mathbf{z}^k) - F(\mathbf{z}^{k-1})). \tag{A.17}$$

Proof. The proof is the same as the proof for Lemma A.2.10. □

Before proving the main lemma of this section, let us introduce an auxiliary notation. Let $\hat{z}^k$ be defined for all $k = 0, 1, 2, \ldots$ as follows:

$$\hat{\mathbf{x}}^k = \mathbf{x}^k - \mathbf{P}m^k, \tag{A.18}$$

and the Lyapunov function Ψ^k be denoted by

$$\begin{aligned}\Psi^k = &\frac{1}{\eta_y}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 + \frac{2}{\eta_x}\left\|m^k\right\|_{\mathbf{P}}^2 + \frac{1}{2\eta_y}\left\|\mathbf{y}^k - \mathbf{y}^{k-1}\right\|^2\\ &+ \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^k + \mathbf{x}_f^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2 - 2\langle\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^*\rangle\\ &+ \frac{1}{4\eta_z}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 + \left(1 + \frac{3\mu\eta_z}{2}\right)\cdot\frac{1}{\eta_z}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{2\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2.\end{aligned}$$

Here we also use the next notation

$$\mathbf{y}^* = \mathbf{P}\mathbf{F}(\mathbf{z}^*) - \nu\mathbf{z}^*, \tag{A.19}$$

and:

$$\mathbf{x}^* = -\mathbf{P}\mathbf{F}(\mathbf{z}^*). \tag{A.20}$$

Lemma A.2.14. Consider the problem (2.16) (or (2.1) + (2.7) + (2.14)) under Assumptions 2.0.4, 2.2.1 and 2.2.2 over a sequence of time-varying graphs $\mathcal{G}(k)$ with gossip matrices $W(k)$. Let $\{\mathbf{z}^k\}$ be the sequence generated by Algorithm 7 with

$$T \geq B, \quad \omega = p \leq \frac{1}{16}; \quad \theta = \frac{1}{2}; \quad \beta = 5\gamma; \quad \nu \leq \frac{\mu}{4}; \quad \tau \in (0; 1);$$

$$\eta_z \leq \min\left\{\frac{1}{8L}, \frac{1}{32\eta_y}, \frac{\sqrt{\alpha b\omega}}{8\overline{L}}\right\}; \quad \eta_y \leq \min\left\{\frac{1}{4\gamma}, \frac{\nu}{8\tau}\right\}; \quad \eta_x \leq \min\left\{\frac{1}{900\chi(T)\gamma}, \frac{\nu}{36\tau\chi^2(T)}\right\};$$

$$\gamma \leq \min\left\{\frac{\mu}{16L^2}; \frac{b\omega}{24\overline{L}^2\eta_z}\right\},$$

$$\alpha = \max\left[\left(1 - \frac{p\eta_z\nu}{p + \eta_z\nu}\right); 1 - \eta_y\gamma; 1 - \eta_x\gamma; 1 - \frac{\mu\eta_z}{8}; 1 - \tau; 1 - \frac{1}{4\chi(T)}\right].$$

Let the choice of T guarantees contraction property (Assumption 2.1.4 point 4) with $\chi(T)$. Then, after k iterations we get

$$\mathbb{E}\frac{1}{4\eta_z}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \leq \max\left[\left(1 - \frac{p\eta_z\nu}{\omega + \eta_z\nu}\right); 1 - \eta_y\gamma; 1 - \eta_x\gamma, \left(1 - \frac{\mu\eta_z}{8}\right); 1 - \tau; \left(1 - \frac{1}{4\chi(T)}\right)\right]^k \cdot \Psi^0.$$

Proof. **Part 1.** We start from

$$\begin{aligned}
\frac{1}{\eta_z}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 &= \frac{1}{\eta_z}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{2}{\eta_z}\langle \mathbf{z}^{k+1} - \mathbf{z}^k, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle - \frac{1}{\eta_z}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&= \frac{1}{\eta_z}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{2\omega}{\eta_z}\langle \mathbf{w}^k - \mathbf{z}^k, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\quad - 2\langle \Delta_z^k - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle - \frac{1}{\eta_z}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&\quad - 2\langle \tfrac{1}{\eta_z}(\mathbf{z}^k + \omega(\mathbf{w}^k - \mathbf{z}^k) - \eta_z\Delta_z^k - \mathbf{z}^{k+1}) - (-(\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*)), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle.
\end{aligned}$$

Optimality condition for (2.1) it follows, that

$$-F(z^*) \in \partial g(z^*).$$

Let us define $\Delta^* \in (\mathbb{R}^d)^M$ as

$$\Delta^* = \tfrac{1}{M}[F(z^*), \dots F(z^*)]^T.$$

It is to note that $-\Delta^* \in M\partial\mathbf{g}(\mathbf{z}^*)$. On the other hand with notation (A.19), we get

$$\Delta^* = (\mathbf{1}\mathbf{1}^\top \otimes I)\mathbf{F}(\mathbf{z}^*) = M(\mathbf{F}(\mathbf{z}^*) - \mathbf{P}\mathbf{F}(\mathbf{z}^*)) = M(\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*)$$

It means that $-(\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*) \in \partial\mathbf{g}(\mathbf{z}^*)$. From update for $\mathbf{z}^{k+1}$ of Algorithm 7 it follows, that $\frac{1}{\eta_z}(\mathbf{z}^k + \omega(\mathbf{w}^k - \mathbf{z}^k) - \mathbf{z}^{k+1} - \eta_z\Delta_z^k) \in \partial\mathbf{g}(\mathbf{z}^{k+1})$. This together with $-(\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*) \in \partial\mathbf{g}(\mathbf{z}^*)$ and monotonicity of $\partial\mathbf{g}(\cdot)$ implies

$$\begin{aligned}
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 &\leq \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{2\omega}{\eta_z}\mathbb{E}\langle \mathbf{w}^k - \mathbf{z}^k, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\quad - 2\mathbb{E}\langle \Delta_z^k - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle - \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&= \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{2\omega}{\eta_z}\mathbb{E}\langle \mathbf{w}^k - \mathbf{z}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle
\end{aligned}$$

$$
\begin{aligned}
&- \frac{2\omega}{\eta_z}\mathbb{E}\langle \mathbf{z}^k - \mathbf{z}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle - 2\mathbb{E}\langle \Delta_z^k - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&- \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&= \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 + \left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 - \left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + \left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- 2\mathbb{E}\langle \Delta_z^k - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle - \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&= \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - 2\mathbb{E}\langle \Delta_z^k - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&- \frac{1-\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2.
\end{aligned}
$$

In previous we also use the simple fact $\|a+b\|^2 = \|a\|^2 + 2\langle a; b\rangle + \|b\|^2$ twice. Definition Δ_z^k and small rearrangement gives

$$
\begin{aligned}
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \leq{}& \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1-\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- 2\mathbb{E}\langle \mathbb{E}\delta^k - \nu\mathbf{z}^k - y^k - \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&+ 2\mathbb{E}\langle \mathbb{E}\delta^k - \delta^k, \mathbf{z}^{k+1} - \mathbf{z}^k\rangle + 2\mathbb{E}\langle \mathbb{E}\delta^k - \delta^k, \mathbf{z}^k - \mathbf{z}^*\rangle.
\end{aligned}
$$

Using the tower property of expectation we can obtain the following:

$$
\begin{aligned}
\mathbb{E}\langle \mathbb{E}_k\left[\delta^k\right] - \delta^k, \mathbf{z}^k - \mathbf{z}^*\rangle &= \mathbb{E}\mathbb{E}_k\left[\langle \mathbb{E}_k\left[\delta^k\right] - \delta^k, \mathbf{z}^k - \mathbf{z}^*\rangle\right] \\
&= \mathbb{E}\langle \mathbb{E}_k\left[\mathbb{E}_k\left[\delta^k\right] - \delta^k\right], \mathbf{z}^k - \mathbf{z}^*\rangle \\
&= \mathbb{E}\langle \mathbb{E}_k\left[\delta^k\right] - \mathbb{E}_k\left[\delta^k\right], \mathbf{z}^k - \mathbf{z}^*\rangle = 0.
\end{aligned}
$$

Then we have the following

$$
\begin{aligned}
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \leq{}& \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1-\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- 2\mathbb{E}\langle \mathbb{E}\delta^k - \nu\mathbf{z}^k - \mathbf{y}^k - \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&+ 2\mathbb{E}\langle \mathbb{E}\delta^k - \delta^k, \mathbf{z}^{k+1} - \mathbf{z}^k\rangle.
\end{aligned}
$$

Using the Young's inequality we get

$$
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \leq \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|w^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2
$$

$$
\begin{aligned}
&- \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1-\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- 2\mathbb{E}\langle \mathbb{E}\delta^k - \nu\mathbf{z}^k - \mathbf{y}^k - \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&+ 2\eta_z\mathbb{E}\left\|\mathbb{E}\delta^k - \delta^k\right\|^2 + \frac{1}{2\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
= &\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1/2-\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- 2\mathbb{E}\langle \mathbb{E}\delta^k - \nu\mathbf{z}^k - \mathbf{y}^k - \alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&+ 2\eta_z\mathbb{E}\left\|\mathbb{E}\delta^k - \delta^k\right\|^2.
\end{aligned}
$$

With (A.15) and (A.17) we have

$$
\begin{aligned}
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \leq &\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|w^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1/2-\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) + \alpha(\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1})) - \nu\mathbf{z}^k - \mathbf{y}^k, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&- 2\mathbb{E}\langle -\alpha(\mathbf{y}^k - \mathbf{y}^{k-1}) - (\mathbf{F}(\mathbf{z}^*) - \mathbf{y}^* - \nu\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&+ \frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\eta_z\alpha^2\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2.
\end{aligned}
$$

Small rearrangement gives

$$
\begin{aligned}
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \leq &\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1/2-\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^*), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle + 2\nu\mathbb{E}\langle \mathbf{z}^k - \mathbf{z}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&+ 2\mathbb{E}\langle \mathbf{y}^{k+1} - \mathbf{y}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&+ 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^k) - (\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^*\rangle \\
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^{k+1} - \mathbf{z}^k\rangle \\
&+ \frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\eta_z\alpha^2\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2.
\end{aligned}
$$

Using Assumption 2.2.2 of μ-strong monotonicity and Young's inequality we get

$$
\begin{aligned}
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \leq &\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1/2-\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- 2\mu\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + 2\nu\mathbb{E}\langle \mathbf{z}^k - \mathbf{z}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle
\end{aligned}
$$

$$
\begin{aligned}
&+ 2\mathbb{E}\langle \mathbf{y}^{k+1} - \mathbf{y}^*, \mathbf{z}^{k+1} - \mathbf{z}^* \rangle \\
&+ 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^k) - (\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{z}^{k+1} - \mathbf{z}^* \rangle \\
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^* \rangle \\
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^{k+1} - \mathbf{z}^k \rangle \\
&+ \frac{4\eta_z \overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\eta_z \alpha^2 \overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 \\
\leq\ & \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|w^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1/2 - \omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- 2\mu\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + \nu\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \nu\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \\
&+ 2\mathbb{E}\langle \mathbf{y}^{k+1} - \mathbf{y}^*, \mathbf{z}^{k+1} - \mathbf{z}^* \rangle \\
&+ 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^k) - (\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{z}^{k+1} - \mathbf{z}^* \rangle \\
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^* \rangle \\
&+ 2\alpha\mathbb{E}\|\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1})\|\|\mathbf{z}^{k+1} - \mathbf{z}^k\| + 2\alpha\mathbb{E}\|\mathbf{y}^k - \mathbf{y}^{k-1}\|\|\mathbf{z}^{k+1} - \mathbf{z}^k\| \\
&+ \frac{4\eta_z \overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\eta_z \alpha^2 \overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2.
\end{aligned}
$$

With L-Lipschitzness of $\mathbf{F}(\cdot)$ (Assumption 2.0.4) we have

$$
\begin{aligned}
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \leq\ & \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \left(\frac{\omega}{\eta_z} - \nu\right)\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1/2 - \omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- (2\mu - \nu)\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + 2\mathbb{E}\langle y^{k+1} - y^*, \mathbf{z}^{k+1} - \mathbf{z}^* \rangle \\
&+ 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^k) - (\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{z}^{k+1} - \mathbf{z}^* \rangle \\
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^* \rangle \\
&+ 2\alpha L\mathbb{E}\|\mathbf{z}^k - \mathbf{z}^{k-1}\|\|\mathbf{z}^{k+1} - \mathbf{z}^k\| + 2\alpha\mathbb{E}\|\mathbf{y}^k - \mathbf{y}^{k-1}\|\|\mathbf{z}^{k+1} - \mathbf{z}^k\| \\
&+ \frac{4\eta_z \overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\eta_z \alpha^2 \overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2
\end{aligned}
$$

Young's inequality we get

$$
\begin{aligned}
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \leq\ & \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \left(\frac{\omega}{\eta_z} - \nu\right)\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1/2 - \omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- (2\mu - \nu)\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + 2\mathbb{E}\langle y^{k+1} - y^*, \mathbf{z}^{k+1} - \mathbf{z}^* \rangle \\
&+ 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^k) - (y^{k+1} - y^k), \mathbf{z}^{k+1} - \mathbf{z}^* \rangle \\
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (y^k - y^{k-1}), \mathbf{z}^k - \mathbf{z}^* \rangle \\
&+ \alpha L\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 + \alpha L\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2
\end{aligned}
$$

$$
\begin{aligned}
&+ \frac{\alpha}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^k - \mathbf{y}^{k-1}\right\|^2 + 2\alpha\eta_y\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&+ \frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\eta_z\alpha^2\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2.
\end{aligned}
$$

Using the assumption on η_z we get that $L \leq \frac{1}{8\eta_z}$ and $\eta_y \leq \frac{1}{32\eta_z}$

$$
\begin{aligned}
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \leq{}& \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \left(\frac{\omega}{\eta_z} - \nu\right)\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1/2 - \omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- (2\mu - \nu)\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + 2\mathbb{E}\langle\mathbf{y}^{k+1} - \mathbf{y}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&+ 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^k) - (\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&- 2\alpha\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^*\rangle \\
&+ \frac{\alpha}{8\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 + \frac{\alpha}{8\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&+ \frac{\alpha}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^k - \mathbf{y}^{k-1}\right\|^2 + \frac{\alpha}{16\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&+ \frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{4\eta_z\alpha^2\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2.
\end{aligned}
$$

With $\alpha \leq 1$ and $\eta_z \leq \frac{\sqrt{\alpha b}}{8\overline{L}}$ we obtain

$$
\begin{aligned}
\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \leq{}& \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \left(\frac{\omega}{\eta_z} - \nu\right)\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{1/2 - \omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&- (2\mu - \nu)\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + 2\mathbb{E}\langle\mathbf{y}^{k+1} - \mathbf{y}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&+ 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^k) - (\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&- 2\alpha\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^*\rangle \\
&+ \frac{\alpha}{8\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 + \frac{1}{8\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&+ \frac{\alpha}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^k - \mathbf{y}^{k-1}\right\|^2 + \frac{1}{16\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&+ \frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \frac{\alpha}{16\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 \\
\leq{}& \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 \\
&- \left(\frac{\omega}{\eta_z} - \nu\right)\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \frac{5/16 - \omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&+ \frac{\alpha}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 - (2\mu - \nu)\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \\
&+ 2\mathbb{E}\langle\mathbf{y}^{k+1} - \mathbf{y}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&+ 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^k) - (\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle
\end{aligned}
$$

$$
\begin{aligned}
&- 2\alpha\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^*\rangle \\
&+ \frac{\alpha}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^k - \mathbf{y}^{k-1}\right\|^2 + \frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2.
\end{aligned}
$$

Choice of $\omega \leq \frac{1}{16}$ and small rearrangement give

$$
\begin{aligned}
&\frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + (2\mu - \nu)\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 + \frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 \\
&\quad - 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^k) - (\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\leq \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 - \left(\frac{\omega}{\eta_z} - \nu\right)\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&\quad + \alpha \cdot \frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 - \alpha \cdot 2\mathbb{E}\langle \mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^*\rangle \\
&\quad + \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 + 2\mathbb{E}\langle \mathbf{y}^{k+1} - \mathbf{y}^*, \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\quad + \frac{\alpha}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^k - \mathbf{y}^{k-1}\right\|^2 + \frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2.
\end{aligned}
$$

Part 2. Update for m^{k+1} and Assumption 2.1.4 on time-varying graph give

$$
\begin{aligned}
\left\|m^{k+1}\right\|_{\mathbf{P}}^2 &\leq (1 - \chi^{-1}(T))\left\|m^k + \eta_x\Delta_x^k\right\|_{\mathbf{P}}^2 \\
&\leq (1 - \chi^{-1}(T))\left(\left(1 + (2\chi(T))^{-1}\right)\left\|m^k\right\|_{\mathbf{P}}^2 + (1 + 2\chi(T))\left\|\eta_x\Delta_x^k\right\|_{\mathbf{P}}^2\right) \\
&\leq (1 - (2\chi(T))^{-1})\left\|m^k\right\|_{\mathbf{P}}^2 + 2\eta_x^2\chi(T)\left\|\Delta_x^k\right\|_{\mathbf{P}}^2.
\end{aligned}
$$

After rearranging we get

$$
\left\|m^k\right\|_{\mathbf{P}}^2 \leq (1 - (4\chi(T))^{-1})4\chi(T)\left\|m^k\right\|_{\mathbf{P}}^2 - 4\chi(T)\left\|m^{k+1}\right\|_{\mathbf{P}}^2 + 8\eta_x^2\chi^2(T)\left\|\Delta_x^k\right\|_{\mathbf{P}}^2. \tag{A.21}
$$

Part 3. Updates for $\mathbf{x}^{k+1}$ and m^{k+1} of Algorithm 7 imply $\hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^k - \eta_x\mathbf{P}\Delta_x^k$. By this together with update for $\mathbf{y}^{k+1}$ of Algorithm 7 we obtain

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1} - \mathbf{x}^*\right\|^2 \\
&\quad = \frac{1}{\eta_y}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \frac{1}{\eta_y}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k\right\|^2 \\
&\quad\quad + \frac{2}{\eta_y}\langle \mathbf{y}^{k+1} - \mathbf{y}^k, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + \frac{2}{\eta_x}\langle \hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^k, \hat{\mathbf{x}}^k - \mathbf{x}^*\rangle \\
&\quad \leq \frac{1}{\eta_y}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \frac{1}{\eta_y}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&\quad\quad - 2\langle \Delta_y^k, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle - 2\langle \mathbf{P}\Delta_x^k, \mathbf{x}^k - \mathbf{x}^*\rangle + 2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}} + \eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2.
\end{aligned}
$$

Using the definitions of Δ_y^k and Δ_x^k we get

$$\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\le \frac{1}{\eta_y}\left\|\mathbf{y}^{k}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k}-\mathbf{x}^*\right\|^2-\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\nu^{-1}(\mathbf{y}_c^k+\mathbf{x}_c^k)+\mathbf{z}^{k+1}+\gamma(\mathbf{y}^k+\mathbf{x}^k+\nu\mathbf{z}^k),\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-2\langle\nu^{-1}\mathbf{P}(\mathbf{y}_c^k+\mathbf{x}_c^k)+\beta\mathbf{P}(\mathbf{x}^k+\Delta^{k+1/2}),\mathbf{x}^k-\mathbf{x}^*\rangle\\
&\quad=\frac{1}{\eta_y}\left\|\mathbf{y}^{k}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k}-\mathbf{x}^*\right\|^2-\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\nu^{-1}(\mathbf{y}_c^k+\mathbf{x}_c^k)+\mathbf{z}^{k+1}+\gamma(\mathbf{y}^k+\mathbf{x}^k+\nu\mathbf{z}^k),\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-2\langle\nu^{-1}\mathbf{P}(\mathbf{y}_c^k+\mathbf{x}_c^k),\mathbf{x}^k-\mathbf{x}^*\rangle-2\langle\beta\mathbf{P}(\mathbf{x}^k+\Delta^{k+1/2}),\mathbf{x}^k-\mathbf{x}^*\rangle.
\end{aligned}$$

Definitions (A.19), (A.20) gives $\mathbf{PF}(\mathbf{z}^*)+\mathbf{x}^*=0\,\mathbf{P}(\mathbf{y}^*+\mathbf{x}^*)=0$, $\nu^{-1}(\mathbf{y}^*+\mathbf{x}^*)+\mathbf{z}^*=0$. Additionally, using $\mathbf{P}\mathbf{x}^k=\mathbf{x}^k$, we get

$$\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\le \frac{1}{\eta_y}\left\|\mathbf{y}^{k}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k}-\mathbf{x}^*\right\|^2-\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\nu^{-1}(\mathbf{y}_c^k+\mathbf{x}_c^k)+\mathbf{z}^{k+1}+\gamma(\mathbf{y}^k+\mathbf{x}^k+\nu\mathbf{z}^k),\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-2\langle\nu^{-1}(\mathbf{y}_c^k+\mathbf{x}_c^k),\mathbf{x}^k-\mathbf{x}^*\rangle-2\langle\beta(\mathbf{x}^k+\mathbf{P}\Delta^{k+1/2}),\mathbf{x}^k-\mathbf{x}^*\rangle\\
&\quad=\frac{1}{\eta_y}\left\|\mathbf{y}^{k}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k}-\mathbf{x}^*\right\|^2-\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\nu^{-1}(\mathbf{y}_c^k+\mathbf{x}_c^k)+\mathbf{z}^{k+1}-\nu^{-1}(\mathbf{y}^*+\mathbf{x}^*)-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-2\gamma\langle\mathbf{y}^k+\mathbf{x}^k+\nu\mathbf{z}^k-(\mathbf{y}^*+\mathbf{x}^*+\nu\mathbf{z}^*),\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-2\langle\nu^{-1}(\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*),\mathbf{x}^k-\mathbf{x}^*\rangle\\
&\qquad-2\beta\langle\mathbf{x}^k+\mathbf{P}\Delta^{k+1/2}-\mathbf{P}\mathbf{F}(\mathbf{z}^*)-\mathbf{x}^*,\mathbf{x}^k-\mathbf{x}^*\rangle.
\end{aligned}$$

And then

$$\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\le \frac{1}{\eta_y}\left\|\mathbf{y}^{k}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k}-\mathbf{x}^*\right\|^2-\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^{k+1}+\mathbf{x}^k-\mathbf{y}^*-\mathbf{x}^*\rangle\\
&\qquad-2\gamma\langle\mathbf{y}^k+\mathbf{x}^k+\nu\mathbf{z}^k-\mathbf{y}^*-\mathbf{x}^*-\nu\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-2\beta\langle\mathbf{x}^k+\mathbf{P}\Delta^{k+1/2}-\mathbf{x}^*-\mathbf{P}\mathbf{F}(\mathbf{z}^*),\mathbf{x}^k-\mathbf{x}^*\rangle.
\end{aligned}$$

Using parallelogram rule we get

$$\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2$$

$$
\begin{aligned}
&\leq \frac{1}{\eta_y}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2\\
&\quad-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^{k+1}+\mathbf{x}^k-\mathbf{y}^*-\mathbf{x}^*\rangle\\
&\quad-\gamma\left(\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2-\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\right)\\
&\quad+\gamma\left(\left\|\mathbf{x}^k-\mathbf{x}^*+\nu(\mathbf{z}^k-\mathbf{z}^*)\right\|^2+\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2\right)\\
&\quad-2\beta\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+\beta\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+\beta\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\leq \frac{1}{\eta_y}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\quad+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2\\
&\quad-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^{k+1}+\mathbf{x}^k-\mathbf{y}^*-\mathbf{x}^*\rangle\\
&\quad-\gamma\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2-(\beta-2\gamma)\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+\beta\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2.
\end{aligned}
$$

The definition of $\beta=5\gamma$ gives

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\leq \frac{1}{\eta_y}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^{k+1}+\mathbf{x}^k-\mathbf{y}^*-\mathbf{x}^*\rangle\\
&\qquad-\gamma\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2-3\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2.
\end{aligned}
$$

Using the definition (A.18) of $\hat{\mathbf{x}}^k$, we get

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\leq \frac{1}{\eta_y}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^{k+1}+\mathbf{x}^k-\mathbf{y}^*-\mathbf{x}^*\rangle\\
&\qquad-\gamma\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2-\gamma\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\left\|m^k\right\|_{\mathbf{P}}^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2\\
&\qquad+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\quad=\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+2\gamma\left\|m^k\right\|_{\mathbf{P}}^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^{k+1}+\mathbf{x}^k-\mathbf{y}^*-\mathbf{x}^*\rangle\\
&\qquad-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2.
\end{aligned}
$$

With updates of $\mathbf{x}_f^{k+1}$ and $\mathbf{y}_f^{k+1}$ from Algorithm 7 we have

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\leq\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+2\gamma\left\|m^k\right\|_{\mathbf{P}}^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\qquad-\frac{2\nu^{-1}}{\tau}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}_c^k-\mathbf{x}_c^k+\theta(W(k)\otimes I)(\mathbf{y}_c^k+\mathbf{x}_c^k)\rangle\\
&\qquad-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^k+\mathbf{x}^k-\mathbf{y}^*-\mathbf{x}^*\rangle\\
&\quad=\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+2\gamma\left\|m^k\right\|_{\mathbf{P}}^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\qquad-\frac{2\theta\nu^{-1}}{\tau}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,(W(k)\otimes I)(\mathbf{y}_c^k+\mathbf{x}_c^k)\rangle\\
&\qquad-\frac{2\nu^{-1}}{\tau}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}_c^k-\mathbf{x}_c^k\rangle\\
&\qquad-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^k+\mathbf{x}^k-\mathbf{y}^*-\mathbf{x}^*\rangle
\end{aligned}
$$

Using definitions (A.19), (A.20) we get $\mathbf{y}^*+\mathbf{x}^*\in\mathcal{L}$ and then

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\leq\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+2\gamma\left\|m^k\right\|_{\mathbf{P}}^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\qquad-\frac{2\theta\nu^{-1}}{\tau}\langle\mathbf{y}_c^k+\mathbf{x}_c^k,(W(k)\otimes I)(\mathbf{y}_c^k+\mathbf{x}_c^k)\rangle\\
&\qquad-\frac{2\nu^{-1}}{\tau}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}_c^k-\mathbf{x}_c^k\rangle\\
&\qquad-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^k+\mathbf{x}^k-\mathbf{y}^*-\mathbf{x}^*\rangle\\
&\quad=\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+2\gamma\left\|m^k\right\|_{\mathbf{P}}^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\qquad-\frac{2\theta\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k\right\|_{(W(k)\otimes I)}^2\\
&\qquad-\frac{2\nu^{-1}}{\tau}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}_c^k-\mathbf{x}_c^k\rangle
\end{aligned}
$$

$$- 2\nu^{-1}\langle \mathbf{y}_c^k + \mathbf{x}_c^k - \mathbf{y}^* - \mathbf{x}^*, \mathbf{y}^k + \mathbf{x}^k - \mathbf{y}^* - \mathbf{x}^* \rangle$$

By parallelogram rule we obtain

$$\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1} - \mathbf{x}^*\right\|^2 \\
&\quad\le \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x} - \gamma\right)\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&\qquad + 2\gamma\left\|m^k\right\|_{\mathbf{P}}^2 + 2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}} + \eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2 - 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&\qquad - \gamma\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2 + 2\gamma\nu^2\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + 5\gamma\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 \\
&\qquad - \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 + \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&\qquad + \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}_c^k - \mathbf{x}_c^k\right\|^2 \\
&\qquad - \frac{2\theta\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{(W(k)\otimes I)}^2 - 2\nu^{-1}\langle \mathbf{y}_c^k + \mathbf{x}_c^k - \mathbf{y}^* - \mathbf{x}^*, \mathbf{y}^k + \mathbf{x}^k - \mathbf{y}^* - \mathbf{x}^*\rangle \\
&\quad\le \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x} - \gamma\right)\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&\qquad + 2\gamma\left\|m^k\right\|_{\mathbf{P}}^2 + 2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}} + \eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2 - 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&\qquad - \gamma\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2 + 2\gamma\nu^2\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + 5\gamma\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 \\
&\qquad - \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 + \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&\qquad + \frac{2\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} - \mathbf{y}_c^k\right\|^2 + \frac{2\nu^{-1}}{\tau}\left\|\mathbf{x}_f^{k+1} - \mathbf{x}_c^k\right\|^2 - \frac{2\theta\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{(W(k)\otimes I)}^2 \\
&\qquad - 2\nu^{-1}\langle \mathbf{y}_c^k + \mathbf{x}_c^k - \mathbf{y}^* - \mathbf{x}^*, \mathbf{y}^k + \mathbf{x}^k - \mathbf{y}^* - \mathbf{x}^*\rangle
\end{aligned}$$

Again by expressions for $\mathbf{x}_f^{k+1}$ and $\mathbf{y}_f^{k+1}$

$$\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1} - \mathbf{x}^*\right\|^2 \\
&\quad\le \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x} - \gamma\right)\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&\qquad + 2\gamma\left\|m^k\right\|_{\mathbf{P}}^2 + 2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}} + \eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2 - 2\langle x^{k+1} - \mathbf{x}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&\qquad - \gamma\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2 + 2\gamma\nu^2\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + 5\gamma\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{x}^*)\right\|^2 \\
&\qquad - \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 + \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&\qquad + 2\nu^{-1}\tau\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 + \frac{2\theta^2\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{(W(k)\otimes I)^2}^2 - \frac{2\theta\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{(W(k)\otimes I)}^2 \\
&\qquad - 2\nu^{-1}\langle \mathbf{y}_c^k + \mathbf{x}_c^k - \mathbf{y}^* - \mathbf{x}^*, \mathbf{y}^k + \mathbf{x}^k - \mathbf{y}^* - \mathbf{x}^*\rangle.
\end{aligned}$$

Using the contraction property of the gossip matrix and the definition of $\theta = \frac{1}{2}$ we get

$$\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1} - \mathbf{x}^*\right\|^2$$

$$
\begin{aligned}
&\leq \left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma-2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\quad+2\gamma\left\|m^k\right\|_{\mathbf{P}}^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2-\frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k\right\|_{\mathbf{P}}^2\\
&\quad-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\quad-\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2+\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2\\
&\quad-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^k+\mathbf{x}^k-\mathbf{y}^*-\mathbf{x}^*\rangle.
\end{aligned}
$$

By updates for $\mathbf{x}_c^k$ and $\mathbf{y}_c^k$ we get

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\leq \left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma-2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+2\gamma\left\|m^k\right\|_{\mathbf{P}}^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2-\frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\qquad-\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2+\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2\\
&\qquad-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*\rangle\\
&\qquad-2\nu^{-1}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}^k+\mathbf{x}^k-\mathbf{y}_c^k-\mathbf{x}_c^k\rangle\\
&\quad=\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma-2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+2\gamma\left\|m^k\right\|_{\mathbf{P}}^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2-\frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\qquad-\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2+\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2\\
&\qquad-2\nu^{-1}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2\\
&\qquad+\frac{2\nu^{-1}(1-\tau)}{\tau}\langle\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*,\mathbf{y}_f^k+\mathbf{x}_f^k-\mathbf{y}_c^k-\mathbf{x}_c^k\rangle.
\end{aligned}
$$

Using parallelogram rule we obtain

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\leq \left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma-2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+2\gamma\left\|m^k\right\|_{\mathbf{P}}^2+2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}}+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2-\frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\qquad-\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2+\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2
\end{aligned}
$$

$$
\begin{aligned}
&- 2\nu^{-1}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&+ \frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k + \mathbf{x}_f^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2 - \frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&\leq \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x} - \gamma\right)\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \left(\frac{1}{\eta_y} - \gamma - 2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&+ 2\gamma\left\|m^k\right\|_{\mathbf{P}}^2 + 2\left\|\Delta_x^k\right\|_{\mathbf{P}}\left\|m^k\right\|_{\mathbf{P}} + \eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2 - \frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{\mathbf{P}}^2 \\
&- 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle - \gamma\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2 + 2\gamma\nu^2\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + 5\gamma\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 \\
&- \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 + \frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k + \mathbf{x}_f^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2.
\end{aligned}
$$

Using Young's inequality we get

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1} - \mathbf{x}^*\right\|^2 \\
&\quad\leq \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x} - \gamma\right)\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \left(\frac{1}{\eta_y} - \gamma - 2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&\quad\quad + 2\gamma\left\|m^k\right\|_{\mathbf{P}}^2 + 4\eta_x\chi(T)\left\|\Delta_x^k\right\|_{\mathbf{P}}^2 + (4\eta_x\chi(T))^{-1}\left\|m^k\right\|_{\mathbf{P}}^2 + \eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2 \\
&\quad\quad - \frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{\mathbf{P}}^2 - 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle - \gamma\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2 \\
&\quad\quad + 2\gamma\nu^2\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + 5\gamma\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 - \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&\quad\quad + \frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k + \mathbf{x}_f^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&\quad= \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x} - \gamma\right)\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \left(\frac{1}{\eta_y} - \gamma - 2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&\quad\quad + 2\gamma\left\|m^k\right\|_{\mathbf{P}}^2 + 4\eta_x\chi(T)\left\|\Delta_x^k\right\|_{\mathbf{P}}^2 + (4\eta_x\chi(T))^{-1}\left\|m^k\right\|_{\mathbf{P}}^2 + \eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2 \\
&\quad\quad - \frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{\mathbf{P}}^2 - 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle - \gamma\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2 \\
&\quad\quad + 2\gamma\nu^2\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + 5\gamma\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 - \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&\quad\quad + \frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k + \mathbf{x}_f^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2.
\end{aligned}
$$

By the assumption on $\eta_x \leq \frac{1}{8\chi(T)\gamma}$ we get that $2\gamma \leq \frac{1}{4\eta_x}$ and

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1} - \mathbf{x}^*\right\|^2 \\
&\quad\leq \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x} - \gamma\right)\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \left(\frac{1}{\eta_y} - \gamma - 2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&\quad\quad + 4\eta_x\chi(T)\left\|\Delta_x^k\right\|_{\mathbf{P}}^2 + (2\eta_x\chi(T))^{-1}\left\|m^k\right\|_{\mathbf{P}}^2 + \eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2 - \frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{\mathbf{P}}^2 \\
&\quad\quad - 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle - \gamma\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2 + 2\gamma\nu^2\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&\quad\quad + 5\gamma\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 - \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2
\end{aligned}
$$

$$+\frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k+\mathbf{x}_f^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2.$$

Using (A.21) we get

$$\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\le\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma-2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+8\eta_x\chi(T)\left\|\Delta_x^k\right\|_{\mathbf{P}}^2+(1-(4\chi(T))^{-1})\frac{2}{\eta_x}\left\|m^k\right\|_{\mathbf{P}}^2-\frac{2}{\eta_x}\left\|m^{k+1}\right\|_{\mathbf{P}}^2+\eta_x\left\|\Delta_x^k\right\|_{\mathbf{P}}^2\\
&\qquad-\frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k\right\|_{\mathbf{P}}^2-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2\\
&\qquad+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2-\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2\\
&\qquad+\frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k+\mathbf{x}_f^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2.
\end{aligned}$$

Using the definition of Δ_x^k with definition of $\beta=5\gamma$ we get

$$\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\le\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma-2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+9\eta_x\chi(T)\left\|\nu^{-1}(\mathbf{y}_c^k+\mathbf{x}_c^k)+5\gamma(\mathbf{x}^k+\Delta^{k+1/2})\right\|_{\mathbf{P}}^2+(1-(4\chi(T))^{-1})\frac{2}{\eta_x}\left\|m^k\right\|_{\mathbf{P}}^2\\
&\qquad-\frac{2}{\eta_x}\left\|m^{k+1}\right\|_{\mathbf{P}}^2-\frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k\right\|_{\mathbf{P}}^2-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2\\
&\qquad+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2-\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2\\
&\qquad+\frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k+\mathbf{x}_f^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2.
\end{aligned}$$

By definition $\mathbf{PF}(\mathbf{z}^*)+\mathbf{x}^*=0$ we obtain

$$\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\le\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma-2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+18\eta_x\nu^{-2}\chi(T)\left\|\mathbf{y}_c^k+\mathbf{x}_c^k\right\|_{\mathbf{P}}^2+450\eta_x\chi(T)\gamma^2\left\|\mathbf{x}^k-\mathbf{x}^*-\mathbf{PF}(\mathbf{z}^*)+\Delta^{k+1/2}\right\|_{\mathbf{P}}^2\\
&\qquad+(1-(4\chi(T))^{-1})\frac{2}{\eta_x}\left\|m^k\right\|_{\mathbf{P}}^2-\frac{2}{\eta_x}\left\|m^{k+1}\right\|_{\mathbf{P}}^2-\frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k+\mathbf{x}_c^k\right\|_{\mathbf{P}}^2\\
&\qquad-2\langle\mathbf{z}^{k+1}-\mathbf{z}^*,\mathbf{y}^{k+1}-\mathbf{y}^*\rangle-\gamma\left\|\mathbf{x}^k-\mathbf{x}^*\right\|^2+2\gamma\nu^2\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2+5\gamma\left\|\Delta^{k+1/2}-\mathbf{F}(\mathbf{z}^*)\right\|^2\\
&\qquad-\frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2+\frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k+\mathbf{x}_f^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2\\
&\quad\le\left(\frac{1}{\eta_y}-\gamma\right)\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+\left(\frac{1}{\eta_x}-\gamma\right)\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2-\left(\frac{1}{\eta_y}-\gamma-2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2
\end{aligned}$$

$$
\begin{aligned}
&+ 18\eta_x\nu^{-2}\chi(T)\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{\mathbf{P}}^2 + 900\eta_x\chi(T)\gamma^2\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{P}}^2 \\
&+ 900\eta_x\chi(T)\gamma^2\left\|\Delta^{k+1/2} - \mathbf{P}\mathbf{F}(\mathbf{z}^*)\right\|_{\mathbf{P}}^2 + (1-(4\chi(T))^{-1})\frac{2}{\eta_x}\left\|m^k\right\|_{\mathbf{P}}^2 \\
&- \frac{2}{\eta_x}\left\|m^{k+1}\right\|_{\mathbf{P}}^2 - \frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{\mathbf{P}}^2 - 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle - \gamma\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2 \\
&+ 2\gamma\nu^2\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + 5\gamma\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 \\
&- \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 + \frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k + \mathbf{x}_f^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2.
\end{aligned}
$$

With the assumption on $\eta_x \leq \frac{1}{900\chi(T)\gamma}; \frac{\nu}{36\tau\chi^2(T)}$ we get

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1} - \mathbf{x}^*\right\|^2 \\
&\quad\leq \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x} - \gamma\right)\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \left(\frac{1}{\eta_y} - \gamma - 2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&\quad\quad + \frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{\mathbf{P}}^2 + \gamma\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{P}}^2 + \gamma\left\|\Delta^{k+1/2} - \mathbf{P}\mathbf{F}(\mathbf{z}^*)\right\|_{\mathbf{P}}^2 \\
&\quad\quad + (1-(4\chi(T))^{-1})\frac{2}{\eta_x}\left\|m^k\right\|_{\mathbf{P}}^2 - \frac{2}{\eta_x}\left\|m^{k+1}\right\|_{\mathbf{P}}^2 - \frac{\nu^{-1}}{2\tau\chi(T)}\left\|\mathbf{y}_c^k + \mathbf{x}_c^k\right\|_{\mathbf{P}}^2 \\
&\quad\quad - 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle - \gamma\left\|\mathbf{x}^k - \mathbf{x}^*\right\|^2 + 2\gamma\nu^2\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + 5\gamma\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 \\
&\quad\quad - \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 + \frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k + \mathbf{x}_f^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2.
\end{aligned}
$$

With property of the projector: $\mathbf{P}\mathbf{P} = \mathbf{P}$, we get

$$
\begin{aligned}
&\frac{1}{\eta_y}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\left\|\hat{\mathbf{x}}^{k+1} - \mathbf{x}^*\right\|^2 \\
&\quad\leq \left(\frac{1}{\eta_y} - \gamma\right)\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x} - \gamma\right)\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 - \left(\frac{1}{\eta_y} - \gamma - 2\nu^{-1}\tau\right)\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&\quad\quad + (1-(4\chi(T))^{-1})\frac{2}{\eta_x}\left\|m^k\right\|_{\mathbf{P}}^2 - \frac{2}{\eta_x}\left\|m^{k+1}\right\|_{\mathbf{P}}^2 - 2\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&\quad\quad + 2\gamma\nu^2\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + 6\gamma\left\|\Delta^{k+1/2} - \mathbf{F}(\mathbf{z}^*)\right\|^2 \\
&\quad\quad - \frac{\nu^{-1}}{\tau}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 + \frac{\nu^{-1}(1-\tau)}{\tau}\left\|\mathbf{y}_f^k + \mathbf{x}_f^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2.
\end{aligned}
$$

Taking full expectation and using (A.16) we have

$$
\begin{aligned}
&\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^{k+1} - \mathbf{x}^*\right\|^2 \\
&\quad\leq \left(\frac{1}{\eta_y} - \gamma\right)\mathbb{E}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x} - \gamma\right)\mathbb{E}\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 \\
&\quad\quad - \left(\frac{1}{\eta_y} - \gamma - 2\nu^{-1}\tau\right)\mathbb{E}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 + (1-(4\chi(T))^{-1})\frac{2}{\eta_x}\mathbb{E}\left\|m^k\right\|_{\mathbf{P}}^2 \\
&\quad\quad - \frac{2}{\eta_x}\mathbb{E}\left\|m^{k+1}\right\|_{\mathbf{P}}^2 - 2\mathbb{E}\langle \mathbf{z}^{k+1} - \mathbf{z}^*, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + 2\gamma\nu^2\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2
\end{aligned}
$$

$$+ 12\gamma L^2\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2 + \frac{12\gamma\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{w}^k\right\|^2$$
$$- \frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2 + \frac{\nu^{-1}(1-\tau)}{\tau}\mathbb{E}\left\|\mathbf{y}_f^{k}+\mathbf{x}_f^{k}-\mathbf{y}^*-\mathbf{x}^*\right\|^2.$$

Using the definition of η_y we get that $\gamma \leq \frac{1}{4\eta_y}$ and $2\nu^{-1}\tau \leq \frac{1}{4\eta_y}$

$$\begin{aligned}
&\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2\\
&\quad\leq \left(\frac{1}{\eta_y}-\gamma\right)\mathbb{E}\left\|\mathbf{y}^{k}-\mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x}-\gamma\right)\mathbb{E}\left\|\hat{\mathbf{x}}^{k}-\mathbf{x}^*\right\|^2 - \frac{1}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+ (1-(4\chi(T))^{-1})\frac{2}{\eta_x}\mathbb{E}\left\|m^k\right\|_{\mathbf{P}}^2 - \frac{2}{\eta_x}\mathbb{E}\left\|m^{k+1}\right\|_{\mathbf{P}}^2 - 2\mathbb{E}\langle\mathbf{z}^{k+1}-\mathbf{z}^*, \mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad+ 2\gamma\nu^2\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2 + 12\gamma L^2\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2 + \frac{12\gamma\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{w}^k\right\|^2\\
&\qquad- \frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2 + \frac{\nu^{-1}(1-\tau)}{\tau}\mathbb{E}\left\|\mathbf{y}_f^{k}+\mathbf{x}_f^{k}-\mathbf{y}^*-\mathbf{x}^*\right\|^2.
\end{aligned}$$

Part 4. After combining parts 1 and 3 of this proof we get

$$\begin{aligned}
&\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2 + \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2 + (2\mu-\nu)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2\\
&\qquad+ \frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2 - 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^{k+1})-\mathbf{F}(\mathbf{z}^k)-(\mathbf{y}^{k+1}-\mathbf{y}^k), \mathbf{z}^{k+1}-\mathbf{z}^*\rangle\\
&\quad\leq \left(\frac{1}{\eta_y}-\gamma\right)\mathbb{E}\left\|\mathbf{y}^{k}-\mathbf{y}^*\right\|^2 + \left(\frac{1}{\eta_x}-\gamma\right)\mathbb{E}\left\|\hat{\mathbf{x}}^{k}-\mathbf{x}^*\right\|^2 - \frac{1}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+ (1-(4\chi(T))^{-1})\frac{2}{\eta_x}\mathbb{E}\left\|m^k\right\|_{\mathbf{P}}^2 - \frac{2}{\eta_x}\mathbb{E}\left\|m^{k+1}\right\|_{\mathbf{P}}^2 - 2\mathbb{E}\langle\mathbf{z}^{k+1}-\mathbf{z}^*, \mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
&\qquad+ 2\gamma\nu^2\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2 + 12\gamma L^2\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2 + \frac{12\gamma\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{w}^k\right\|^2\\
&\qquad- \frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2 + \frac{\nu^{-1}(1-\tau)}{\tau}\mathbb{E}\left\|\mathbf{y}_f^{k}+\mathbf{x}_f^{k}-\mathbf{y}^*-\mathbf{x}^*\right\|^2.\\
&\qquad+ \frac{1}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2 - \left(\frac{\omega}{\eta_z}-\nu\right)\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2\\
&\qquad+ \alpha\cdot\frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2 - \alpha\cdot 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}), \mathbf{z}^k-\mathbf{z}^*\rangle\\
&\qquad+ \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2 - \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{w}^k\right\|^2 + 2\mathbb{E}\langle\mathbf{y}^{k+1}-\mathbf{y}^*, \mathbf{z}^{k+1}-\mathbf{z}^*\rangle\\
&\qquad+ \frac{\alpha}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2 + \frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{w}^{k-1}\right\|^2.
\end{aligned}$$

Small rearrangement gives

$$\begin{aligned}
&\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2 + \frac{1}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\qquad+ \frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2 - 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^{k+1})-\mathbf{F}(\mathbf{z}^k)-(\mathbf{y}^{k+1}-\mathbf{y}^k), \mathbf{z}^{k+1}-\mathbf{z}^*\rangle
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{2}{\eta_x}\mathbb{E}\left\|m^{k+1}\right\|_{\mathbf{P}}^2 + \frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&+ \left(1 - \frac{12\gamma\overline{L}^2\eta_z}{b\omega}\right)\frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 + \left(\frac{1}{\eta_z} + 2\mu - \nu - 12\gamma L^2\right)\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \\
\leq\ & (1 - \eta_y\gamma)\cdot\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + (1 - \eta_x\gamma)\cdot\frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 + \alpha\cdot\frac{1}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^k - \mathbf{y}^{k-1}\right\|^2 \\
&+ \alpha\cdot\frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 - \alpha\cdot 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^*\rangle \\
&+ (1 - (4\chi(T))^{-1})\cdot\frac{2}{\eta_x}\mathbb{E}\left\|m^k\right\|_{\mathbf{P}}^2 + (1 - \tau)\cdot\frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^k + \mathbf{x}_f^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&+ \frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \left(\frac{1}{\eta_z} - \frac{\omega}{\eta_z} + \nu + 2\gamma\nu^2\right)\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&+ \frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2.
\end{aligned}
$$

Now, we add $\frac{\gamma+\eta\nu}{p\eta_z}\mathbb{E}\left\|\mathbf{w}^{k+1} - \mathbf{z}^*\right\|^2$ to both sides and use update for $\mathbf{w}^{k+1}$

$$
\frac{\omega + \eta_z\nu}{p\eta_z}\mathbb{E}\mathbb{E}_{\mathbf{w}^{k+1}}\left\|\mathbf{w}^{k+1} - \mathbf{z}^*\right\|^2 = \frac{\omega + \eta_z\nu}{\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 + \frac{(\omega + \eta_z\nu)(1-p)}{\eta_z p}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2,
$$

and then

$$
\begin{aligned}
&\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1} - \mathbf{y}^*\right\|^2 + \frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^{k+1} - \mathbf{x}^*\right\|^2 + \frac{1}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \\
&\quad + \frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|^2 - 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^{k+1}) - \mathbf{F}(\mathbf{z}^k) - (\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{z}^{k+1} - \mathbf{z}^*\rangle \\
&\quad + \frac{2}{\eta_x}\mathbb{E}\left\|m^{k+1}\right\|_{\mathbf{P}}^2 + \frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^{k+1} + \mathbf{x}_f^{k+1} - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&\quad + \left(1 - \frac{12\gamma\overline{L}^2\eta_z}{b\omega}\right)\frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{w}^k\right\|^2 + \left(\frac{1}{\eta_z} + 2\mu - \nu - 12\gamma L^2\right)\mathbb{E}\left\|\mathbf{z}^{k+1} - \mathbf{z}^*\right\|^2 \\
&\quad + \frac{\omega + \eta_z\nu}{p\eta_z}\mathbb{E}\left\|\mathbf{w}^{k+1} - \mathbf{z}^*\right\|^2 \\
\leq\ & (1 - \eta_y\gamma)\cdot\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^k - \mathbf{y}^*\right\|^2 + (1 - \eta_x\gamma)\cdot\frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^k - \mathbf{x}^*\right\|^2 + \alpha\cdot\frac{1}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^k - \mathbf{y}^{k-1}\right\|^2 \\
&\quad + \alpha\cdot\frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^{k-1}\right\|^2 - \alpha\cdot 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k) - \mathbf{F}(\mathbf{z}^{k-1}) - (\mathbf{y}^k - \mathbf{y}^{k-1}), \mathbf{z}^k - \mathbf{z}^*\rangle \\
&\quad + (1 - (4\chi(T))^{-1})\cdot\frac{2}{\eta_x}\mathbb{E}\left\|m^k\right\|_{\mathbf{P}}^2 + (1 - \tau)\cdot\frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^k + \mathbf{x}_f^k - \mathbf{y}^* - \mathbf{x}^*\right\|^2 \\
&\quad + \frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k - \mathbf{w}^{k-1}\right\|^2 + \left(\frac{1}{\eta_z} + 2\nu + 2\gamma\nu^2\right)\mathbb{E}\left\|\mathbf{z}^k - \mathbf{z}^*\right\|^2 \\
&\quad + \left(1 - \frac{p\eta_z\nu}{\omega + \eta_z\nu}\right)\cdot\frac{\omega + \eta_z\nu}{p\eta_z}\mathbb{E}\left\|\mathbf{w}^k - \mathbf{z}^*\right\|^2.
\end{aligned}
$$

With $\gamma \leq \frac{\mu}{48L^2}$ and $\nu \leq \frac{\mu}{4}$ we gat that

$$
\left(1 + 2\mu\eta_x - \nu\eta_x - 12\gamma L^2\eta_x\right) \geq 1 + \frac{3\mu\eta_x}{2},
$$

$$\left(\frac{1}{\eta_x}+2\nu+2\gamma\nu^2\right)\leq\left(1+\frac{\mu\eta_x}{2}+\frac{2\mu^3\eta_x}{16\cdot 48L^2}\right)\frac{1}{\eta_x}\leq(1+\mu\eta_x)\frac{1}{\eta_x}.$$

Additionally, by

$$(1+\mu\eta_x)\leq\left(1+\frac{3\mu\eta_x}{2}\right)\left(1-\frac{\mu\eta_x}{8}\right)$$

we obtain

$$\begin{aligned}
&\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2+\frac{1}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\quad+\frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2-2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^{k+1})-\mathbf{F}(\mathbf{z}^k)-(\mathbf{y}^{k+1}-\mathbf{y}^k),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle\\
&\quad+\frac{2}{\eta_x}\mathbb{E}\left\|m^{k+1}\right\|_{\mathbf{P}}^2+\frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2\\
&\quad+\left(1-\frac{12\gamma\overline{L}^2\eta_z}{b\omega}\right)\frac{\omega}{\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{w}^k\right\|^2+\left(1+\frac{3\mu\eta_x}{2}\right)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2\\
&\quad+\frac{\omega+\eta_z\nu}{p\eta_z}\mathbb{E}\left\|\mathbf{w}^{k+1}-\mathbf{z}^*\right\|^2\\
&\leq(1-\eta_y\gamma)\cdot\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+(1-\eta_x\gamma)\cdot\frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2+\alpha\cdot\frac{1}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2\\
&\quad+\alpha\cdot\frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2-\alpha\cdot 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^k-\mathbf{z}^*\rangle\\
&\quad+(1-(4\chi(T))^{-1})\cdot\frac{2}{\eta_x}\mathbb{E}\left\|m^k\right\|_{\mathbf{P}}^2+(1-\tau)\cdot\frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^k+\mathbf{x}_f^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2\\
&\quad+\frac{4\eta_z\overline{L}^2}{b}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{w}^{k-1}\right\|^2+\left(1-\frac{\mu\eta_x}{8}\right)\cdot\left(1+\frac{3\mu\eta_x}{2}\right)\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2\\
&\quad+\left(1-\frac{p\eta_z\nu}{\omega+\eta_z\nu}\right)\cdot\frac{\omega+\eta_z\nu}{p\eta_z}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2.
\end{aligned}$$

The choice $\eta_z\leq\frac{\sqrt{\alpha\omega b}}{4\overline{L}}$ and $\gamma\leq\frac{b\omega}{24\overline{L}^2\eta_z}$ gives

$$\begin{aligned}
&\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^*\right\|^2+\frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^{k+1}-\mathbf{x}^*\right\|^2+\frac{1}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^{k+1}-\mathbf{y}^k\right\|^2\\
&\quad+\frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^k\right\|^2-2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^{k+1})-\mathbf{F}(\mathbf{z}^k)-(\mathbf{y}^{k+1}-\mathbf{y}^k),\mathbf{z}^{k+1}-\mathbf{z}^*\rangle\\
&\quad+\frac{2}{\eta_x}\mathbb{E}\left\|m^{k+1}\right\|_{\mathbf{P}}^2+\frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^{k+1}+\mathbf{x}_f^{k+1}-\mathbf{y}^*-\mathbf{x}^*\right\|^2\\
&\quad+\frac{\omega}{2\eta_z}\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{w}^k\right\|^2+\left(1+\frac{3\mu\eta_x}{2}\right)\mathbb{E}\left\|\mathbf{z}^{k+1}-\mathbf{z}^*\right\|^2\\
&\quad+\frac{\omega+\eta_z\nu}{p\eta_z}\mathbb{E}\left\|\mathbf{w}^{k+1}-\mathbf{z}^*\right\|^2\\
&\leq(1-\eta_y\gamma)\cdot\frac{1}{\eta_y}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^*\right\|^2+(1-\eta_x\gamma)\cdot\frac{1}{\eta_x}\mathbb{E}\left\|\hat{\mathbf{x}}^k-\mathbf{x}^*\right\|^2+\alpha\cdot\frac{1}{2\eta_y}\mathbb{E}\left\|\mathbf{y}^k-\mathbf{y}^{k-1}\right\|^2\\
&\quad+\alpha\cdot\frac{1}{4\eta_z}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^{k-1}\right\|^2-\alpha\cdot 2\mathbb{E}\langle\mathbf{F}(\mathbf{z}^k)-\mathbf{F}(\mathbf{z}^{k-1})-(\mathbf{y}^k-\mathbf{y}^{k-1}),\mathbf{z}^k-\mathbf{z}^*\rangle\\
&\quad+(1-(4\chi(T))^{-1})\cdot\frac{2}{\eta_x}\mathbb{E}\left\|m^k\right\|_{\mathbf{P}}^2+(1-\tau)\cdot\frac{\nu^{-1}}{\tau}\mathbb{E}\left\|\mathbf{y}_f^k+\mathbf{x}_f^k-\mathbf{y}^*-\mathbf{x}^*\right\|^2
\end{aligned}$$

$$+\alpha\cdot\frac{\omega}{2\eta_z}\mathbb{E}\left\|\mathbf{z}^k-\mathbf{w}^{k-1}\right\|^2+\left(1-\frac{\mu\eta_x}{8}\right)\cdot\left(1+\frac{3\mu\eta_x}{2}\right)\mathbb{E}\left\|\mathbf{z}^k-\mathbf{z}^*\right\|^2$$
$$+\left(1-\frac{p\eta_z\nu}{\omega+\eta_z\nu}\right)\cdot\frac{\omega+\eta_z\nu}{p\eta_z}\mathbb{E}\left\|\mathbf{w}^k-\mathbf{z}^*\right\|^2.$$

Putting $p=\omega$ and using definition of the Lyapunov function, we get

$$\mathbb{E}[\Psi^{k+1}]\leq\max\left[\left(1-\frac{p\eta_z\nu}{p+\eta_z\nu}\right);1-\eta_y\gamma;1-\eta_x\gamma;1-\frac{\mu\eta_z}{8};1-\tau;1-\frac{1}{4\chi(T)}\right]\cdot\mathbb{E}[\Psi^k].$$

Finally,

$$\begin{aligned}\Psi^k&\geq\frac{1}{\eta_x}\left\|x^k-\mathbf{x}^*\right\|^2+\frac{1}{4\eta_x}\left\|x^{k-1}-x^k\right\|^2+\frac{1}{2\eta_y}\left\|y^{k-1}-y^k\right\|^2\\&\quad-2\langle\mathbf{F}(x^k)-\mathbf{F}(x^{k-1})-(y^k-y^{k-1}),x^k-\mathbf{x}^*\rangle\\&\geq\frac{1}{\eta_x}\left\|x^k-\mathbf{x}^*\right\|^2+\frac{1}{4\eta_x}\left\|x^{k-1}-x^k\right\|^2+\frac{1}{2\eta_y}\left\|y^{k-1}-y^k\right\|^2\\&\quad-\frac{1}{2\eta_y}\left\|y^k-y^{k-1}\right\|^2-2\eta_y\left\|x^k-\mathbf{x}^*\right\|^2-\frac{1}{4\eta_x}\left\|x^{k-1}-x^k\right\|^2-4\eta_xL^2\left\|x^k-\mathbf{x}^*\right\|^2\\&=\left(1-2\eta_x\eta_y-4\eta_x^2L^2\right)\frac{1}{\eta_x}\left\|x^k-\mathbf{x}^*\right\|^2\\&\geq\frac{1}{4\eta_x}\left\|x^k-\mathbf{x}^*\right\|^2.\end{aligned}$$

□

Theorem A.2.15 (Theorem 2.2.10)**.** Consider the problem (2.16) (or (2.1) + (2.7)+(2.14)) under Assumptions 2.0.4, 2.2.1 and 2.2.2 over a sequence of time-varying graphs $\mathcal{G}(k)$ with gossip matrices $W(k)$. Let $\{\mathbf{z}^k\}$ be the sequence generated by Algorithm 7 with parameters

$$T\geq B;\quad\omega=p\leq\frac{1}{16};\quad\theta=\frac{1}{2};\quad\beta=5\gamma;\quad\nu=\frac{\mu}{4};\quad\tau=\min\left\{\frac{\mu}{32L\chi(T)};\frac{\mu\sqrt{bp}}{32\overline{L}}\right\};$$

$$\eta_z=\min\left\{\frac{1}{8L\chi(T)},\frac{1}{32\eta_y},\frac{\sqrt{\alpha b\omega}}{8\overline{L}}\right\};\quad\eta_y=\min\left\{\frac{1}{4\gamma},\frac{\nu}{8\tau}\right\};\quad\eta_x=\min\left\{\frac{1}{900\chi(T)\gamma},\frac{\nu}{36\tau\chi^2(T)}\right\};$$

$$\gamma=\min\left\{\frac{\mu}{16L^2};\frac{b\omega}{24\overline{L}^2\eta_z}\right\},$$

$$\alpha=\max\left[\left(1-\frac{p\eta_z\nu}{p+\eta_z\nu}\right);1-\eta_y\gamma;1-\eta_x\gamma;1-\frac{\mu\eta_z}{8};1-\tau;1-\frac{1}{4\chi(T)}\right].$$

Let the choice of T guarantees contraction property (Assumption 2.1.4 point 4) with $\chi(T)$. Then, given $\varepsilon>0$, the number of iterations for $\mathbb{E}[\|\mathbf{z}^k-\mathbf{z}^*\|^2]\leq\varepsilon$ is

$$O\left(\left[\chi^2(T)+\frac{1}{p}+\chi(T)\frac{L}{\mu}+\frac{1}{\sqrt{bp}}\frac{\overline{L}}{\mu}\right]\log\frac{1}{\varepsilon}\right).$$

Proof. It is easy to check that $\alpha \geq \frac{1}{2}$, then also one can verify that the choice of ω, θ, β, ν, τ, η_z, η_y, η_x, γ, α satisfies the conditions of Lemma A.2.14. We can get that the iteration complexity of Algorithm 7:

$$O\left(\left[\chi(T) + \frac{1}{p} + \frac{1}{\eta_y\gamma} + \frac{1}{\eta_x\gamma} + \frac{1}{\eta_z\mu} + \frac{1}{\eta_z\nu} + \frac{1}{\tau} + \frac{1}{1-\alpha}\right]\log\frac{1}{\varepsilon}\right).$$

Substituting α, ν, ω, we get

$$O\left(\left[\chi(T) + \frac{1}{p} + \frac{1}{\eta_y\gamma} + \frac{1}{\eta_x\gamma} + \frac{1}{\eta_z\mu} + \frac{1}{\tau}\right]\log\frac{1}{\varepsilon}\right).$$

Putting η_z

$$O\left(\left[\chi(T) + \frac{1}{p} + \chi(T)\frac{L}{\mu} + \frac{1}{\sqrt{bp}}\frac{\overline{L}}{\mu} + \frac{1}{\eta_y\gamma} + \frac{1}{\eta_x\gamma} + \frac{\eta_y}{\mu} + \frac{1}{\tau}\right]\log\frac{1}{\varepsilon}\right).$$

Substituting η_y and η_x

$$O\left(\left[\chi(T) + \frac{1}{p} + \chi(T)\frac{L}{\mu} + \frac{1}{\sqrt{bp}}\frac{\overline{L}}{\mu} + \frac{\tau\chi^2(T)}{\mu\gamma} + \frac{\eta_y}{\mu} + \frac{1}{\tau}\right]\log\frac{1}{\varepsilon}\right).$$

Using $\eta_y \leq \frac{\nu}{8\tau}$

$$O\left(\left[\chi(T) + \frac{1}{p} + \chi(T)\frac{L}{\mu} + \frac{1}{\sqrt{bp}}\frac{\overline{L}}{\mu} + \frac{\tau\chi^2(T)}{\mu\gamma} + \frac{1}{\tau}\right]\log\frac{1}{\varepsilon}\right).$$

Substituting τ we get

$$O\left(\left[\chi(T) + \frac{1}{p} + \chi(T)\frac{L}{\mu} + \frac{1}{\sqrt{bp}}\frac{\overline{L}}{\mu} + \frac{\tau\chi^2(T)}{\mu\gamma}\right]\log\frac{1}{\varepsilon}\right).$$

Putting γ we obtain

$$O\left(\left[\chi(T) + \frac{1}{p} + \chi(T)\frac{L}{\mu} + \frac{1}{\sqrt{bp}}\frac{\overline{L}}{\mu} + \frac{\tau\overline{L}^2\eta_z\chi^2(T)}{\mu bp} + \frac{\tau L^2\chi^2(T)}{\mu^2}\right]\log\frac{1}{\varepsilon}\right).$$

With $\eta_z \leq \frac{\sqrt{abp}}{8\overline{L}}$ and $\tau \leq \frac{\mu\sqrt{bp}}{\overline{L}}$ we have

$$O\left(\left[\chi^2(T) + \frac{1}{p} + \chi(T)\frac{L}{\mu} + \frac{1}{\sqrt{bp}}\frac{\overline{L}}{\mu} + \frac{\tau L^2\chi^2(T)}{\mu^2}\right]\log\frac{1}{\varepsilon}\right).$$

Finally, $\tau \leq \frac{1}{8L\chi(T)}$

$$O\left(\left[\chi^2(T) + \frac{1}{p} + \chi(T)\frac{L}{\mu} + \frac{1}{\sqrt{bp}}\frac{\overline{L}}{\mu}\right]\log\frac{1}{\varepsilon}\right).$$

□

A.2.4 Proof of Theorem 2.4.5

We begin introducing some intermediate results.

Lemma A.2.16. Let $\{z^k\}$ be the sequence generated by Algorithm 9 over $\mathcal{G}$ with a master node. The following holds:

$$\begin{aligned}\left\|z^{k+1}-z^*\right\|^2 \leq (1-\gamma\mu)\left\|z^k-z^*\right\|^2 - (1-3\gamma\mu-4\gamma^2\delta^2)\left\|z^k-\hat{u}^k\right\|^2 \\ + \left(2+\frac{4\gamma\delta^2}{\mu}+\frac{4}{\gamma\mu}+4\gamma^2\delta^2\right)\left\|u^k-\hat{u}^k\right\|^2. \end{aligned} \tag{A.22}$$

Proof: We start from the simple algebra

$$\begin{aligned}\left\|z^{k+1}-z^*\right\|^2 &= \left\|z^k-z^*\right\|^2 + 2\langle z^{k+1}-z^k, z^k-z^*\rangle + \left\|z^{k+1}-z^k\right\|^2 \\ &= \left\|z^k-z^*\right\|^2 + 2\langle z^{k+1}-z^k, \hat{u}^k-z^*\rangle + 2\langle z^{k+1}-z^k, z^k-\hat{u}^k\rangle + \left\|z^{k+1}-z^k\right\|^2 \\ &= \left\|z^k-z^*\right\|^2 + 2\langle z^{k+1}-z^k, \hat{u}^k-z^*\rangle + \left\|z^{k+1}-\hat{u}^k\right\|^2 - \left\|z^k-\hat{u}^k\right\|^2 \\ &= \left\|z^k-z^*\right\|^2 + 2\langle u^k+\gamma\cdot(F(z^k)-F_1(z^k)-F(u^k)+F_1(u^k))-z^k, \hat{u}^k-z^*\rangle \\ &\quad + \left\|z^{k+1}-\hat{u}^k\right\|^2 - \left\|z^k-\hat{u}^k\right\|^2 \\ &= \left\|z^k-z^*\right\|^2 + 2\langle u^k+\gamma\cdot(F(z^k)-F_1(z^k))-z^k, \hat{u}^k-z^*\rangle \\ &\quad - 2\gamma\langle F(u^k)-F_1(u^k), \hat{u}^k-z^*\rangle + \left\|z^{k+1}-\hat{u}^k\right\|^2 - \left\|z^k-\hat{u}^k\right\|^2.\end{aligned}$$

Substituting the expression of v^k, we have

$$\begin{aligned}\left\|z^{k+1}-z^*\right\|^2 &\leq \left\|z^k-z^*\right\|^2 + 2\langle u^k-v^k, \hat{u}^k-z^*\rangle - 2\gamma\langle F(u^k)-F_1(u^k), \hat{u}^k-z^*\rangle \\ &\quad + \left\|z^{k+1}-\hat{u}^k\right\|^2 - \left\|z^k-\hat{u}^k\right\|^2 \\ &= \left\|z^k-z^*\right\|^2 + 2\langle \hat{u}^k-v^k, \hat{u}^k-z^*\rangle - 2\gamma\langle F(u^k)-F_1(u^k), \hat{u}^k-z^*\rangle \\ &\quad + 2\langle u^k-\hat{u}^k, \hat{u}^k-z^*\rangle + \left\|z^{k+1}-\hat{u}^k\right\|^2 - \left\|z^k-\hat{u}^k\right\|^2.\end{aligned}$$

Invoking the optimality of $\hat{u}^k$, $\langle \gamma F_1(\hat{u}^k)+\hat{u}^k-v^k, \hat{u}^k-z\rangle + g(\hat{u}^k)-g(z)\leq 0$ (for all $z\in\mathcal{Z}$), yields:

$$\begin{aligned}\left\|z^{k+1}-z^*\right\|^2 &\leq \left\|z^k-z^*\right\|^2 - 2\gamma\langle F_1(\hat{u}^k), \hat{u}^k-z^*\rangle - 2\gamma\langle F(u^k)-F_1(u^k), \hat{u}^k-z^*\rangle \\ &\quad + 2\langle u^k-\hat{u}^k, \hat{u}^k-z^*\rangle + \left\|w^k-\hat{u}^k\right\|^2 - \left\|z^k-\hat{u}^k\right\|^2 \\ &\quad + 2\gamma g(z^*) - 2\gamma g(\hat{u}^k) \\ &= \left\|z^k-z^*\right\|^2 - 2\gamma\langle F_1(\hat{u}^k), \hat{u}^k-z^*\rangle - 2\gamma\langle F(\hat{u}^k)-F_1(\hat{u}^k), \hat{u}^k-z^*\rangle \\ &\quad + 2\langle \gamma(F(\hat{u}^k)-F_1(\hat{u}^k)-F(u^k)+F_1(u^k))+u^k-\hat{u}^k, \hat{u}^k-z^*\rangle \\ &\quad + \left\|w^k-\hat{u}^k\right\|^2 - \left\|z^k-\hat{u}^k\right\|^2 \\ &\quad + 2\gamma g(z^*) - 2\gamma g(\hat{u}^k).\end{aligned}$$

Invoking the optimality of the solution z^*: $\langle \gamma F(z^*), z^* - z\rangle + g(z^*) - g(z) \leq 0$ (for all $z \in \mathcal{Z}$) along with the μ-strong convexity-strong concavity of f, we obtain

$$\begin{aligned}
\left\|z^{k+1} - z^*\right\|^2 &\leq \left\|z^k - z^*\right\|^2 - 2\gamma\langle F(\hat{u}^k) - F(z^*), \hat{u}^k - z^*\rangle \\
&\quad + 2\langle \gamma(F(\hat{u}^k) - F_1(\hat{u}^k) - F(u^k) + F_1(u^k)) + u^k - \hat{u}^k, \hat{u}^k - z^*\rangle \\
&\quad + \left\|w^k - \hat{u}^k\right\|^2 - \left\|z^k - \hat{u}^k\right\|^2 \\
&\leq \left\|z^k - z^*\right\|^2 - 2\gamma\mu \left\|\hat{u}^k - z^*\right\|^2 \\
&\quad + 2\langle \gamma(F(\hat{u}^k) - F_1(\hat{u}^k) - F(u^k) + F_1(u^k)) + u^k - \hat{u}^k, \hat{u}^k - z^*\rangle \\
&\quad + \left\|w^k - \hat{u}^k\right\|^2 - \left\|z^k - \hat{u}^k\right\|^2.
\end{aligned}$$

By Young's inequality, we have

$$\begin{aligned}
\left\|z^{k+1} - z^*\right\|^2 &\leq \left\|z^k - z^*\right\|^2 - 2\gamma\mu \left\|\hat{u}^k - z^*\right\|^2 \\
&\quad + \frac{2}{\gamma\mu}\left\|\gamma(F(\hat{u}^k) - F_1(\hat{u}^k) - F(u^k) + F_1(u^k)) + u^k - \hat{u}^k\right\|^2 + \frac{\gamma\mu}{2}\left\|\hat{u}^k - z^*\right\|^2 \\
&\quad + \left\|w^k - \hat{u}^k\right\|^2 - \left\|z^k - \hat{u}^k\right\|^2 \\
&\leq \left\|z^k - z^*\right\|^2 - \frac{3\gamma\mu}{2}\left\|\hat{u}^k - z^*\right\|^2 \\
&\quad + \frac{4\gamma}{\mu}\left\|F(\hat{u}^k) - F_1(\hat{u}^k) - F(u^k) + F_1(u^k)\right\|^2 + \frac{4}{\gamma\mu}\left\|u^k - \hat{u}^k\right\|^2 \\
&\quad + \left\|u^k + \gamma\cdot(F(z^k) - F_1(z^k) - F(u^k) + F_1(u^k)) - \hat{u}^k\right\|^2 - \left\|z^k - \hat{u}^k\right\|^2 \\
&= \left\|z^k - z^*\right\|^2 - \frac{3\gamma\mu}{2}\left\|\hat{u}^k - z^*\right\|^2 \\
&\quad + \frac{4\gamma}{\mu}\left\|F(\hat{u}^k) - F_1(\hat{u}^k) - F(u^k) + F_1(u^k)\right\|^2 + \frac{4}{\gamma\mu}\left\|u^k - \hat{u}^k\right\|^2 \\
&\quad + 2\left\|u^k - \hat{u}^k\right\|^2 + 2\gamma^2\left\|F(z^k) - F_1(z^k) - F(u^k) + F_1(u^k)\right\|^2 - \left\|z^k - \hat{u}^k\right\|^2.
\end{aligned}$$

Note that the operator $F - F_1$ is δ-Lipschitz, therefore,

$$\begin{aligned}
\left\|z^{k+1} - z^*\right\|^2 &\leq \left\|z^k - z^*\right\|^2 - \frac{3\gamma\mu}{2}\left\|\hat{u}^k - z^*\right\|^2 \\
&\quad + \frac{4\gamma\delta^2}{\mu}\left\|u^k - \hat{u}^k\right\|^2 + \frac{4}{\gamma\mu}\left\|u^k - \hat{u}^k\right\|^2 \\
&\quad + 2\left\|u^k - \hat{u}^k\right\|^2 + 2\gamma^2\delta^2\left\|z^k - u^k\right\|^2 - \left\|z^k - \hat{u}^k\right\|^2 \\
&\leq \left\|z^k - z^*\right\|^2 - \frac{3\gamma\mu}{2}\left\|\hat{u}^k - z^*\right\|^2 - (1 - 4\gamma^2\delta^2)\left\|z^k - \hat{u}^k\right\|^2 \\
&\quad + \left(2 + \frac{4\gamma\delta^2}{\mu} + \frac{4}{\gamma\mu} + 4\gamma^2\delta^2\right)\left\|u^k - \hat{u}^k\right\|^2.
\end{aligned}$$

Finally, using $\|a+b\|^2 \geq \frac{2}{3}\|a\|^2 - 2\|b\|^2$, we obtain the desired result (A.22).

□

Theorem A.2.17. Let $\{z^k\}$ the sequence generated by Algorithm 9 (in the setting of Theorem 3.2.11) with the step-size γ given by

$$\gamma = \min\left\{\frac{1}{12\mu}, \frac{1}{4\delta}\right\}. \tag{A.23}$$

Let each subproblem (2.23) be solved up to (relative) precision $\tilde{e}$,

$$\tilde{e} = \frac{1}{2\left(2 + \frac{4\gamma\delta^2}{\mu} + \frac{4}{\gamma\mu} + 4\gamma^2\delta^2\right)}. \tag{A.24}$$

Then, $\left\|z^K - z^*\right\|^2 \leq \varepsilon$ after

$$K = \mathcal{O}\left(\left(1 + \frac{\delta}{\mu}\right)\log\frac{\left\|z^0 - z^*\right\|^2}{\varepsilon}\right) \quad \text{iterations/communications.}$$

Proof: The output u^k produced by inner method satisfies

$$\left\|u^k - \hat{u}^k\right\|^2 \leq \tilde{e}\left\|z^k - \hat{u}^k\right\|^2.$$

Combining this fact and Lemma A.2.16 yields

$$\begin{aligned}\left\|z^{k+1} - z^*\right\|^2 &\leq (1-\gamma\mu)\left\|z^k - z^*\right\|^2 - (1 - 3\gamma\mu - 4\gamma^2\delta^2)\left\|z^k - \hat{u}^k\right\|^2 \\ &\quad + \left(2 + \frac{4\gamma\delta^2}{\mu} + \frac{4}{\gamma\mu} + 4\gamma^2\delta^2\right)\tilde{e}\left\|z^k - \hat{u}^k\right\|^2 \\ &\overset{(A.24)}{\leq} (1-\gamma\mu)\left\|z^k - z^*\right\|^2 - \left(\frac{1}{2} - 3\gamma\mu - 4\gamma^2\delta^2\right)\left\|z^k - \hat{u}^k\right\|^2.\end{aligned}$$

The proof is completed by choosing γ according to (A.23).

□

Corollary A.2.18. Let we solve the subproblem (2.23) via Extragradient method with starting point z^k and

$$T = \mathcal{O}\left((1 + \gamma L)\log\frac{1}{\tilde{e}}\right) \tag{A.25}$$

iterations. Then we can estimate the total number local iterations at the server side by

$$\mathcal{O}\left(\left(1 + \frac{\delta}{\mu} + \frac{L}{\mu}\right)\log\frac{1}{\tilde{e}}\log\frac{\left\|z^0 - z^*\right\|^2}{\varepsilon}\right).$$

Proof: Firstly, one can note that after T iterations of the Extragradient method from (A.25) we can achieve $\tilde{e}$ precision. It follows readily from the convergence of the Extragradient method and the fact that the objective operator in (2.23) is 1-strongly monotone and $(1 + \gamma L)$-Lipschitz. Then

we can estimate the total number of local iterations at the server side, namely:

$$\begin{aligned} K \cdot T &= \mathcal{O}\left(\frac{1}{\gamma\mu}\left(1+\gamma L\right)\log\frac{1}{\tilde{e}}\log\frac{\left\|z^0-z^*\right\|^2}{\varepsilon}\right) \\ &= \mathcal{O}\left(\left(\frac{1}{\gamma\mu}+\frac{L}{\mu}\right)\log\frac{1}{\tilde{e}}\log\frac{\left\|z^0-z^*\right\|^2}{\varepsilon}\right) \\ &= \mathcal{O}\left(\left(1+\frac{\delta}{\mu}+\frac{L}{\mu}\right)\log\frac{1}{\tilde{e}}\log\frac{\left\|z^0-z^*\right\|^2}{\varepsilon}\right). \end{aligned}$$

□

A.2.5 Proof of Theorem 2.5.4

Let us introduce the useful notation:

$$\hat{z}^k = z^k - e^k - \frac{1}{M}\sum_{m=1}^M e_m^k, \quad \hat{z}^{k+1/2} = z^{k+1/2} - e^k - \frac{1}{M}\sum_{m=1}^M e_m^k, \quad \hat{w}^k = w^k - e^k - \frac{1}{M}\sum_{m=1}^M e_m^k.$$

It is easy to verify that such sequences have a very useful property:

$$\begin{aligned}
\hat{z}^{k+1} &= z^{k+1} - e^{k+1} - \frac{1}{M}\sum_{m=1}^M e_m^{k+1} \\
&= z^{k+1/2} - C^{\text{serv}}\left[\frac{1}{M}\sum_{m=1}^M C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k) + e^k\right] \\
&\quad - e^k - \frac{1}{M}\sum_{m=1}^M C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k) \\
&\quad + C^{\text{serv}}\left[\frac{1}{M}\sum_{m=1}^M C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k) + e^k\right] \\
&\quad - \frac{1}{M}\sum_{m=1}^M \left[e_m^k + \gamma \cdot F_m(z^{k+1/2}) - \gamma \cdot F_m(w^k) - C_m^{\text{dev}}(\gamma \cdot F_m(z^{k+1/2}) - \gamma \cdot F_m(w^k) + e_m^k)\right] \\
&= z^{k+1/2} - e^k - \frac{1}{M}\sum_{m=1}^M e_m^k - \gamma \cdot (F(z^{k+1/2}) - F(w^k)) \\
&= \hat{z}^{k+1/2} - \gamma \cdot (F(z^{k+1/2}) - F(w^k)).
\end{aligned} \tag{A.26}$$

The following theorem gives the convergence of `MASHA2`.

Theorem A.2.19 (Theorem 2.5.4)**.** Let distributed variational inequality (2.1) + (2.7) is solved by Algorithm 12 with $\tau \geq \frac{3}{4}$ and biased compressor operators (2.26): on server with δ^{serv} parameter, on devices with δ^{dev}. Let Assumption 2.0.4 and one case of Assumption 2.0.5 are satisfied. Then the following estimates holds

- in strongly-monotone case with $\gamma \leq \min\left[\frac{1-\tau}{8\mu}; \frac{\sqrt{1-\tau}}{2L+165\delta^{\text{serv}}\delta^{\text{dev}}\tilde{L}}\right]$:

$$\mathbb{E}\left[\|\hat{z}^K - z^*\|^2\right] \leq \left(1 - \frac{\mu\gamma}{2}\right)^K \cdot 2\|z^0 - z^*\|^2;$$

- in monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2L+165\delta^{\text{serv}}\delta^{\text{dev}}\tilde{L}}$:

$$\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1/2}\right)\right] \leq \frac{2\max_{z\in\mathcal{C}}\|z^0 - z\|^2 + 4\|z^0 - z^*\|^2}{\gamma K};$$

- in non-monotone case with $\gamma \leq \frac{\sqrt{1-\tau}}{2L+165\delta^{\text{serv}}\delta^{\text{dev}}\tilde{L}}$:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\|F(w^k)\|^2\right] \leq \frac{32\mathbb{E}\|z^0 - z^*\|^2}{\gamma^2 K}.$$

Proof. We start from the following equalities for any z:

$$\|\hat{z}^{k+1} - z\|^2 = \|z^{k+1/2} - z\|^2 + 2\langle \hat{z}^{k+1} - z^{k+1/2}, z^{k+1/2} - z\rangle + \|\hat{z}^{k+1} - z^{k+1/2}\|^2,$$

$$\|z^{k+1/2} - z\|^2 = \|\hat{z}^{k} - z\|^2 + 2\langle z^{k+1/2} - \hat{z}^{k}, z^{k+1/2} - z\rangle - \|z^{k+1/2} - \hat{z}^{k}\|^2.$$

Summing up, we obtain

$$\begin{aligned}\|\hat{z}^{k+1} - z\|^2 = \|\hat{z}^{k} - z\|^2 &+ 2\langle \hat{z}^{k+1} - \hat{z}^{k}, z^{k+1/2} - z\rangle \\ &+ \|\hat{z}^{k+1} - z^{k+1/2}\|^2 - \|z^{k+1/2} - \hat{z}^{k}\|^2. \end{aligned} \tag{A.27}$$

Using that (A.26), we get

$$\begin{aligned}\|\hat{z}^{k+1} - z^{k+1/2}\|^2 &\leq 2\|\hat{z}^{k+1} - \hat{z}^{k+1/2}\|^2 + 2\|\hat{z}^{k+1/2} - z^{k+1/2}\|^2 \\ &= 2\gamma^2 \cdot \|F(z^{k+1/2}) - F(w^k)\|^2 + 2\left\|e^k - \frac{1}{M}\sum_{m=1}^{M} e_m^k\right\|^2 \\ &\leq 2\gamma^2 L^2 \cdot \|z^{k+1/2} - w^k\|^2 + 4\|e^k\|^2 + \frac{4}{M}\sum_{m=1}^{M}\left\|e_m^k\right\|^2 \\ &\leq 2\gamma^2 L^2 \cdot \|z^{k+1/2} - w^k\|^2 + 4\|e^k\|^2 + \frac{4}{M}\sum_{m=1}^{M}\left\|e_m^k\right\|^2. \end{aligned} \tag{A.28}$$

Additionally, here we use that F is L-Lipschitz (Assumption 2.10). Next, (A.27) with (A.28) gives

$$\begin{aligned}\|\hat{z}^{k+1} - z\|^2 \leq \|\hat{z}^{k} - z\|^2 &+ 2\langle \hat{z}^{k+1} - \hat{z}^{k}, z^{k+1/2} - z\rangle \\ &+ 2\gamma^2 L^2 \cdot \|z^{k+1/2} - w^k\|^2 + 4\|e^k\|^2 + \frac{4}{M}\sum_{m=1}^{M}\left\|e_m^k\right\|^2 \\ &- \|z^{k+1/2} - \hat{z}^{k}\|^2. \end{aligned} \tag{A.29}$$

Now we consider the inner product $\langle \hat{z}^{k+1} - \hat{z}^{k}, z^{k+1/2} - z\rangle$. Using that

$$\begin{aligned}\hat{z}^{k+1} - \hat{z}^{k} &= \hat{z}^{k+1} - \hat{z}^{k+1/2} + \hat{z}^{k+1/2} - \hat{z}^{k} = -\gamma \cdot (F(z^{k+1/2}) - F(w^k)) + z^{k+1/2} - z^k \\ &= -\gamma \cdot F(z^{k+1/2}) + \bar{z}^k - z^k,\end{aligned}$$

and using the definition of $\bar{z}^k$ (line 4), we get

$$\begin{aligned}
2\langle \hat{z}^{k+1} - \hat{z}^k, z^{k+1/2} - z\rangle &= 2\langle -\gamma \cdot F(z^{k+1/2}) + \bar{z}^k - z^k, z^{k+1/2} - z\rangle \\
&= -2\gamma\langle F(z^{k+1/2}), z^{k+1/2} - z\rangle + 2\langle \bar{z}^k - z^k, z^{k+1/2} - z\rangle \\
&= -2\gamma\langle F(z^{k+1/2}), z^{k+1/2} - z\rangle + 2(1-\tau)\langle w^k - z^k, z^{k+1/2} - z\rangle.
\end{aligned}$$

Substituting in (A.29), we obtain

$$\begin{aligned}
\|\hat{z}^{k+1} - z\|^2 \leq \|\hat{z}^k - z\|^2 &- 2\gamma\langle F(z^{k+1/2}), z^{k+1/2} - z\rangle + 2(1-\tau)\langle w^k - z^k, z^{k+1/2} - z\rangle \\
&+ 2\gamma^2 L^2 \cdot \|z^{k+1/2} - w^k\|^2 + 4\|e^k\|^2 + \frac{4}{M}\sum_{m=1}^{M} \left\|e_m^k\right\|^2 \\
&- \|z^{k+1/2} - \hat{z}^k\|^2.
\end{aligned}$$

The equality $2\langle a, b\rangle = \|a+b\|^2 - \|a\|^2 - \|b\|^2$ gives

$$\begin{aligned}
\|\hat{z}^{k+1} - z\|^2 &\leq \|\hat{z}^k - z\|^2 - 2\gamma\langle F(z^{k+1/2}), z^{k+1/2} - z\rangle \\
&\quad + 2(1-\tau)\langle w^k - z^{k+1/2}, z^{k+1/2} - z\rangle \\
&\quad + 2(1-\tau)\langle z^{k+1/2} - z^k, z^{k+1/2} - z\rangle \\
&\quad + 2\gamma^2 L^2 \cdot \|z^{k+1/2} - w^k\|^2 + 4\|e^k\|^2 + \frac{4}{M}\sum_{m=1}^{M} \left\|e_m^k\right\|^2 - \|z^{k+1/2} - \hat{z}^k\|^2 \\
&= \|\hat{z}^k - z\|^2 - 2\gamma\langle F(z^{k+1/2}), z^{k+1/2} - z\rangle \\
&\quad + (1-\tau)\|w^k - z\|^2 - (1-\tau)\|w^k - z^{k+1/2}\|^2 - (1-\tau)\|z^{k+1/2} - z\|^2 \\
&\quad + (1-\tau)\|z^{k+1/2} - z^k\|^2 + (1-\tau)\|z^{k+1/2} - z\|^2 - (1-\tau)\|z^k - z\|^2 \\
&\quad + 2\gamma^2 L^2 \cdot \|z^{k+1/2} - w^k\|^2 + 4\|e^k\|^2 + \frac{4}{M}\sum_{m=1}^{M} \left\|e_m^k\right\|^2 - \|z^{k+1/2} - \hat{z}^k\|^2 \\
&= \|\hat{z}^k - z\|^2 - (1-\tau)\|z^k - z^*\|^2 + (1-\tau)\|w^k - z\|^2 \\
&\quad - 2\gamma\langle F(z^{k+1/2}), z^{k+1/2} - z\rangle - (1-\tau)\|w^k - z^{k+1/2}\|^2 \\
&\quad + 2\gamma^2 L^2 \|w^k - z^{k+1/2}\|^2 + 4\|e^k\|^2 + \frac{4}{M}\sum_{m=1}^{M} \left\|e_m^k\right\|^2 \\
&\quad - \|z^{k+1/2} - \hat{z}^k\|^2 + (1-\tau)\|z^{k+1/2} - z^k\|^2 \\
&\leq \|\hat{z}^k - z\|^2 - (1-\tau)\|z^k - z\|^2 + (1-\tau)\|w^k - z\|^2 \\
&\quad - 2\gamma\langle F(z^{k+1/2}), z^{k+1/2} - z^*\rangle - (1-\tau)\|w^k - z^{k+1/2}\|^2 \\
&\quad + 2\gamma^2 L^2 \|w^k - z^{k+1/2}\|^2 + 4\|e^k\|^2 + \frac{4}{M}\sum_{m=1}^{M} \left\|e_m^k\right\|^2 \\
&\quad - \frac{1}{2}\|z^{k+1/2} - z^k\|^2 + \|z^k - \hat{z}^k\|^2 + (1-\tau)\|z^{k+1/2} - z^k\|^2.
\end{aligned}$$

With definition of $\hat{z}^k$ we get

$$\begin{aligned}\|\hat{z}^{k+1} - z\|^2 \leq \|\hat{z}^k - z\|^2 - (1-\tau)\|z^k - z\|^2 + (1-\tau)\|w^k - z\|^2 \\ - 2\gamma\langle F(z^{k+1/2}), z^{k+1/2} - z\rangle - (1 - \tau - 2\gamma^2 L^2)\|w^k - z^{k+1/2}\|^2 \\ + 6\|e^k\|^2 + \frac{6}{M}\sum_{m=1}^{M}\left\|e_m^k\right\|^2 - \left(\tau - \frac{1}{2}\right)\|z^{k+1/2} - z^k\|^2. \end{aligned} \tag{A.30}$$

Next we will consider three cases of monotonicity separately.

Strongly-monotone

We continue with (A.30) by putting $z = z^*$ and using optimality condition: $\langle F(z^*), z^{k+1/2} - z^*\rangle \leq 0$.

$$\begin{aligned}\|\hat{z}^{k+1} - z^*\|^2 \leq \|\hat{z}^k - z^*\|^2 - (1-\tau)\|z^k - z^*\|^2 + (1-\tau)\|w^k - z^*\|^2 \\ - 2\gamma\langle F(z^{k+1/2}) - F(z^*), z^{k+1/2} - z^*\rangle - (1 - \tau - 2\gamma^2 L^2)\|w^k - z^{k+1/2}\|^2 \\ + 6\|e^k\|^2 + \frac{6}{M}\sum_{m=1}^{M}\left\|e_m^k\right\|^2 - \left(\tau - \frac{1}{2}\right)\|z^{k+1/2} - z^k\|^2.\end{aligned}$$

Taking a full mathematical expectation, we obtain

$$\begin{aligned}\mathbb{E}\|\hat{z}^{k+1} - z^*\|^2 \leq \mathbb{E}\|\hat{z}^k - z^*\|^2 - (1-\tau)\mathbb{E}\|z^k - z^*\|^2 + (1-\tau)\mathbb{E}\|w^k - z^*\|^2 \\ - 2\gamma\mathbb{E}\left[\langle F(z^{k+1/2}) - F(z^*), z^{k+1/2} - z^*\rangle\right] - (1 - \tau - 2\gamma^2 L^2)\mathbb{E}\|w^k - z^{k+1/2}\|^2 \\ + 6\mathbb{E}\|e^k\|^2 + \frac{6}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2 - \left(\tau - \frac{1}{2}\right)\mathbb{E}\|z^{k+1/2} - z^k\|^2.\end{aligned}$$

Next, we take into account strong-monotonicity (Assumption 2.0.5 (SM)):

$$\begin{aligned}\mathbb{E}\|\hat{z}^{k+1} - z^*\|^2 \leq \mathbb{E}\|\hat{z}^k - z^*\|^2 - (1-\tau)\mathbb{E}\|z^k - z^*\|^2 + (1-\tau)\mathbb{E}\|w^k - z^*\|^2 \\ - 2\gamma\mu\mathbb{E}\|z^{k+1/2} - z^*\|^2 - (1 - \tau - 2\gamma^2 L^2)\mathbb{E}\|w^k - z^{k+1/2}\|^2 \\ + 6\mathbb{E}\|e^k\|^2 + \frac{6}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2 - \left(\tau - \frac{1}{2}\right)\mathbb{E}\|z^{k+1/2} - z^k\|^2.\end{aligned}$$

Taking a full mathematical expectation, we obtain

$$\begin{aligned}\mathbb{E}\|\hat{z}^{k+1} - z^*\|^2 \leq \mathbb{E}\|\hat{z}^k - z^*\|^2 - (1-\tau)\mathbb{E}\|z^k - z^*\|^2 + (1-\tau)\mathbb{E}\|w^k - z^*\|^2 \\ - 2\gamma\mu\mathbb{E}\|z^{k+1/2} - z^*\|^2 - (1 - \tau - 2\gamma^2 L^2)\mathbb{E}\|w^k - z^{k+1/2}\|^2 \\ + 6\mathbb{E}\|e^k\|^2 + \frac{6}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2 - \left(\tau - \frac{1}{2}\right)\mathbb{E}\|z^{k+1/2} - z^k\|^2.\end{aligned}$$

Then we use choice of w^{k+1} (lines 12, 17, 20) and get

$$\mathbb{E}\|w^{k+1} - z^*\|^2 = \mathbb{E}\left[\mathbb{E}_{w^{k+1}}\|w^{k+1} - z^*\|^2\right] = \tau\mathbb{E}\left\|w^k - z^*\right\|^2 + (1-\tau)\mathbb{E}\|z^k - z^*\|^2,$$

Summing up the two previous expressions gives

$$\begin{aligned}
&\mathbb{E}\|\hat{z}^{k+1} - z^*\|^2 + \mathbb{E}\|w^{k+1} - z^*\|^2 \\
&\qquad \leq \mathbb{E}\|\hat{z}^k - z^*\|^2 + \mathbb{E}\|w^k - z^*\|^2 - 2\gamma\mu\mathbb{E}\|z^{k+1/2} - z^*\|^2 \\
&\qquad\quad - (1 - \tau - 2\gamma^2L^2)\mathbb{E}\|w^k - z^{k+1/2}\|^2 - \left(\tau - \frac{1}{2}\right)\mathbb{E}\|z^{k+1/2} - z^k\|^2 \\
&\qquad\quad + 6\mathbb{E}\|e^k\|^2 + \frac{6}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2.
\end{aligned}$$

Then we can weight previous expression by p^k and get

$$\begin{aligned}
&\sum_{k=0}^{K-1} p^k\mathbb{E}\|\hat{z}^{k+1} - z^*\|^2 + \sum_{k=0}^{K-1} p^k\mathbb{E}\|w^{k+1} - z^*\|^2 \\
&\quad \leq \sum_{k=0}^{K-1} p^k\mathbb{E}\|\hat{z}^k - z^*\|^2 + \sum_{k=0}^{K-1} p^k\mathbb{E}\|w^k - z^*\|^2 - 2\gamma\mu\sum_{k=0}^{K-1} p^k\mathbb{E}\|z^{k+1/2} - z^*\|^2 \\
&\quad\quad - (1 - \tau - 2\gamma^2L^2)\cdot\sum_{k=0}^{K-1} p^k\mathbb{E}\|w^k - z^{k+1/2}\|^2 - \left(\tau - \frac{1}{2}\right)\cdot\sum_{k=0}^{K-1} p^k\mathbb{E}\|z^{k+1/2} - z^k\|^2 \\
&\quad\quad + 6\cdot\sum_{k=0}^{K-1} p^k\mathbb{E}\|e^k\|^2 + 6\cdot\sum_{k=0}^{K-1} p^k\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2. \qquad \text{(A.31)}
\end{aligned}$$

Next we will estimate "error" term:

$$\begin{aligned}
\mathbb{E}\|e^{k+1}\|^2 &= \mathbb{E}\Bigg\|e^k + \frac{1}{M}\sum_{m=1}^{M} C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k) \\
&\qquad - C^{\text{serv}}\left[\frac{1}{M}\sum_{m=1}^{M} C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k) + e^k\right]\Bigg\|^2 \\
&\leq \left(1 - \frac{1}{\delta^{\text{serv}}}\right)\mathbb{E}\left\|e^k + \frac{1}{M}\sum_{m=1}^{M} C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k)\right\|^2 \\
&\leq (1+c)\left(1 - \frac{1}{\delta^{\text{serv}}}\right)\mathbb{E}\left\|e^k\right\|^2 \\
&\quad + \left(1 + \frac{1}{c}\right)\left(1 - \frac{1}{\delta^{\text{serv}}}\right)\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|C_m^{\text{dev}}(\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k)\right\|^2.
\end{aligned}$$

Here we use definition of biased compression (2.26) and inequality $\|a+b\|^2 \leq (1+c)\|a\|^2 + (1 + 1/c)\|b\|^2$ (for $c > 0$). Is is easy to prove that for baised compressor C_m^{dev} from (2.26) it holds that

$\|C_m^{\text{dev}}(x)\|^2 \leq 4\|x\|^2$ (see Chapter 5). Then

$$\begin{aligned}
\mathbb{E}\|e^{k+1}\|^2 &\leq (1+c)\left(1-\frac{1}{\delta^{\text{serv}}}\right)\mathbb{E}\left\|e^k\right\|^2 \\
&\quad + \left(1+\frac{1}{c}\right)\left(1-\frac{1}{\delta^{\text{serv}}}\right)\frac{4}{M}\sum_{m=1}^{M}\mathbb{E}\left\|\gamma F_m(z^{k+1/2}) - \gamma F_m(w^k) + e_m^k\right\|^2 \\
&\leq (1+c)\left(1-\frac{1}{\delta^{\text{serv}}}\right)\mathbb{E}\left\|e^k\right\|^2 \\
&\quad + \gamma^2\left(1+\frac{1}{c}\right)\left(1-\frac{1}{\delta^{\text{serv}}}\right)\frac{8}{M}\sum_{m=1}^{M}\mathbb{E}\left\|F_m(z^{k+1/2}) - F_m(w^k)\right\|^2 \\
&\quad + \left(1+\frac{1}{c}\right)\left(1-\frac{1}{\delta^{\text{serv}}}\right)\frac{8}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2 \\
&\leq (1+c)\left(1-\frac{1}{\delta^{\text{serv}}}\right)\mathbb{E}\left\|e^k\right\|^2 + 8\gamma^2\tilde{L}^2\left(1+\frac{1}{c}\right)\left(1-\frac{1}{\delta^{\text{serv}}}\right)\mathbb{E}\left\|z^{k+1/2}-w^k\right\|^2 \\
&\quad + \left(1+\frac{1}{c}\right)\left(1-\frac{1}{\delta^{\text{serv}}}\right)\frac{8}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2.
\end{aligned}$$

In the last we use Assumption 2.10 and definition of $\tilde{L}$ from this Assumption. With $c = \frac{1}{2(\delta-1)}$ we get

$$\begin{aligned}
\mathbb{E}\|e^{k+1}\|^2 &\leq \left(1-\frac{1}{2\delta^{\text{serv}}}\right)\mathbb{E}\left\|e^k\right\|^2 + 16\delta^{\text{serv}}\gamma^2\tilde{L}^2\cdot\mathbb{E}\|z^{k+1/2}-w^k\|^2 + 16\delta^{\text{serv}}\cdot\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2 \\
&\leq 16\delta^{\text{serv}}\gamma^2\tilde{L}^2\sum_{j=0}^{k}\left(1-\frac{1}{2\delta^{\text{serv}}}\right)^{k-j}\cdot\left\|z^{j+1/2}-w^j\right\|^2 \\
&\quad + 16\delta^{\text{serv}}\sum_{j=0}^{k}\left(1-\frac{1}{2\delta^{\text{serv}}}\right)^{k-j}\cdot\frac{1}{M}\sum_{m=1}^{M}\left\|e_m^j\right\|^2.
\end{aligned}$$

We weigh the sequence as follows $\sum\limits_{k=0}^{K} p^k\mathbb{E}\left\|e^k\right\|^2$. Here we also assume p such that $p^k \leq p^j(1 + 1/4\delta^{\text{serv}})^{k-j}$. Then

$$\begin{aligned}
\sum_{k=0}^{K-1} p^k\mathbb{E}\left\|e^k\right\|^2 &\leq 16\delta^{\text{serv}}\gamma^2\tilde{L}^2\sum_{k=0}^{K-1}p^k\sum_{j=0}^{k-1}\left(1-\frac{1}{2\delta^{\text{serv}}}\right)^{k-j-1}\cdot\mathbb{E}\left\|z^{j+1/2}-w^j\right\|^2 \\
&\quad + 16\delta^{\text{serv}}\sum_{k=0}^{K-1}p^k\sum_{j=0}^{k-1}\left(1-\frac{1}{2\delta^{\text{serv}}}\right)^{k-j-1}\cdot\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^j\right\|^2 \\
&\leq \frac{16\delta^{\text{serv}}\gamma^2\tilde{L}^2}{(1-1/2\delta^{\text{serv}})}\sum_{k=0}^{K-1}\sum_{j=0}^{k-1}p^j\left(1+\frac{1}{4\delta^{\text{serv}}}\right)^{k-j}\left(1-\frac{1}{2\delta^{\text{serv}}}\right)^{k-j}\cdot\mathbb{E}\left\|z^{j+1/2}-w^j\right\|^2 \\
&\quad + \frac{16\delta^{\text{serv}}}{(1-1/2\delta^{\text{serv}})}\sum_{k=0}^{K-1}\sum_{j=0}^{k-1}p^j\left(1+\frac{1}{4\delta^{\text{serv}}}\right)^{k-j}\left(1-\frac{1}{2\delta^{\text{serv}}}\right)^{k-j}\cdot\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^j\right\|^2
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{16\delta^{\mathrm{serv}}\gamma^2\tilde{L}^2}{(1-1/2\delta^{\mathrm{serv}})}\sum_{k=0}^{K-1}\sum_{j=0}^{k-1}p^j\left(1-\frac{1}{4\delta^{\mathrm{serv}}}\right)^{k-j}\cdot\mathbb{E}\left\|z^{j+1/2}-w^j\right\|^2\\
&\quad+\frac{16\delta^{\mathrm{serv}}}{(1-1/2\delta^{\mathrm{serv}})}\sum_{k=0}^{K-1}\sum_{j=0}^{k-1}p^j\left(1-\frac{1}{4\delta^{\mathrm{serv}}}\right)^{k-j}\cdot\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^j\right\|^2\\
&\leq \frac{16\delta^{\mathrm{serv}}\gamma^2\tilde{L}^2}{(1-1/2\delta^{\mathrm{serv}})}\sum_{k=0}^{K-1}p^k\mathbb{E}\left\|z^{k+1/2}-w^k\right\|^2\cdot\sum_{j=0}^{\infty}\left(1-\frac{1}{4\delta^{\mathrm{serv}}}\right)^j\\
&\quad+\frac{16\delta^{\mathrm{serv}}}{(1-1/2\delta^{\mathrm{serv}})}\sum_{k=0}^{K-1}p^k\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2\cdot\sum_{j=0}^{\infty}\left(1-\frac{1}{4\delta^{\mathrm{serv}}}\right)^j\\
&\leq 128(\delta^{\mathrm{serv}})^2\gamma^2\tilde{L}^2\sum_{k=0}^{K-1}p^k\mathbb{E}\left\|z^{k+1/2}-w^k\right\|^2\\
&\quad+128(\delta^{\mathrm{serv}})^2\sum_{k=0}^{K-1}p^k\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2. \qquad \text{(A.32)}
\end{aligned}
$$

Combining (A.31) with (A.32), we obtain

$$
\begin{aligned}
&\sum_{k=0}^{K-1}p^k\left(\mathbb{E}\|\hat{z}^{k+1}-z^*\|^2+\mathbb{E}\|w^{k+1}-z^*\|^2\right)\\
&\quad\leq\sum_{k=0}^{K-1}p^k\mathbb{E}\|\hat{z}^k-z^*\|^2+\sum_{k=0}^{K-1}p^k\mathbb{E}\|w^k-z^*\|^2-2\gamma\mu\sum_{k=0}^{K-1}p^k\mathbb{E}\|z^{k+1/2}-z^*\|^2\\
&\quad\quad-(1-\tau-2\gamma^2L^2)\cdot\sum_{k=0}^{K-1}p^k\mathbb{E}\|w^k-z^{k+1/2}\|^2-\left(\tau-\frac{1}{2}\right)\cdot\sum_{k=0}^{K-1}p^k\mathbb{E}\|z^{k+1/2}-z^k\|^2\\
&\quad\quad+768(\delta^{\mathrm{serv}})^2\gamma^2\tilde{L}^2\cdot\sum_{k=0}^{K-1}p^k\mathbb{E}\left\|z^{k+1/2}-w^k\right\|^2\\
&\quad\quad+768(\delta^{\mathrm{serv}})^2\cdot\sum_{k=0}^{K-1}p^k\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2+6\cdot\sum_{k=0}^{K-1}p^k\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2\\
&\quad\leq\sum_{k=0}^{K-1}p^k\mathbb{E}\|\hat{z}^k-z^*\|^2+\sum_{k=0}^{K-1}p^k\mathbb{E}\|w^k-z^*\|^2-2\gamma\mu\sum_{k=0}^{K-1}p^k\mathbb{E}\|z^{k+1/2}-z^*\|^2\\
&\quad\quad-(1-\tau-2\gamma^2L^2-768(\delta^{\mathrm{serv}})^2\gamma^2\tilde{L}^2)\cdot\sum_{k=0}^{K-1}p^k\mathbb{E}\|w^k-z^{k+1/2}\|^2\\
&\quad\quad-\left(\tau-\frac{1}{2}\right)\cdot\sum_{k=0}^{K-1}p^k\mathbb{E}\|z^{k+1/2}-z^k\|^2+775(\delta^{\mathrm{serv}})^2\cdot\sum_{k=0}^{K-1}p^k\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|e_m^k\right\|^2.
\end{aligned}
$$

Using $-\|a\|^2\leq-\frac{1}{2}\|a+b\|^2+\|b\|^2$, we get

$$
\sum_{k=0}^{K-1}p^k\left(\mathbb{E}\|\hat{z}^{k+1}-z^*\|^2+\mathbb{E}\|w^{k+1}-z^*\|^2\right)
$$

$$
\begin{aligned}
&\leq \sum_{k=0}^{K-1} p^k \left(1 - \frac{\mu\gamma}{2}\right) \left(\mathbb{E}\|\hat{z}^k - z^*\|^2 + \mathbb{E}\|w^k - z^*\|^2\right) \\
&\quad - (1 - \tau - \gamma\mu - 2\gamma^2 L^2 - 768(\delta^{\text{serv}})^2\gamma^2\tilde{L}^2) \cdot \sum_{k=0}^{K-1} p^k \mathbb{E}\|w^k - z^{k+1/2}\|^2 \\
&\quad - \left(\tau - \frac{1}{2}\right) \cdot \sum_{k=0}^{K-1} p^k \mathbb{E}\|z^{k+1/2} - z^k\|^2 + 775(\delta^{\text{serv}})^2 \cdot \sum_{k=0}^{K-1} p^k \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}\left\|e_m^k\right\|^2 \\
&\quad + \gamma\mu \cdot \sum_{k=0}^{K-1} p^k \mathbb{E}\|z^{k+1/2} - \hat{z}^k\|^2 \\
&\leq \sum_{k=0}^{K-1} p^k \left(1 - \frac{\mu\gamma}{2}\right) \left(\mathbb{E}\|\hat{z}^k - z^*\|^2 + \mathbb{E}\|w^k - z^*\|^2\right) \\
&\quad - (1 - \tau - \gamma\mu - 2\gamma^2 L^2 - 768(\delta^{\text{serv}})^2\gamma^2\tilde{L}^2) \cdot \sum_{k=0}^{K-1} p^k \mathbb{E}\|w^k - z^{k+1/2}\|^2 \\
&\quad - \left(\tau - 2\gamma\mu - \frac{1}{2}\right) \cdot \sum_{k=0}^{K-1} p^k \mathbb{E}\|z^{k+1/2} - z^k\|^2 \\
&\quad + (775(\delta^{\text{serv}})^2 + 2\mu\gamma) \cdot \sum_{k=0}^{K-1} p^k \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}\left\|e_m^k\right\|^2.
\end{aligned}
$$

With $\gamma \leq \frac{1}{2\mu}$ we get

$$
\begin{aligned}
&\sum_{k=0}^{K-1} p^k \left(\mathbb{E}\|\hat{z}^{k+1} - z^*\|^2 + \mathbb{E}\|w^{k+1} - z^*\|^2\right) \\
&\quad \leq \sum_{k=0}^{K-1} p^k \left(1 - \frac{\mu\gamma}{2}\right) \left(\mathbb{E}\|\hat{z}^k - z^*\|^2 + \mathbb{E}\|w^k - z^*\|^2\right) \\
&\qquad - (1 - \tau - \gamma\mu - 2\gamma^2 L^2 - 768(\delta^{\text{serv}})^2\gamma^2\tilde{L}^2) \cdot \sum_{k=0}^{K-1} p^k \mathbb{E}\|w^k - z^{k+1/2}\|^2 \\
&\qquad - \left(\tau - 2\gamma\mu - \frac{1}{2}\right) \cdot \sum_{k=0}^{K-1} p^k \mathbb{E}\|z^{k+1/2} - z^k\|^2 \\
&\qquad + 776(\delta^{\text{serv}})^2 \cdot \sum_{k=0}^{K-1} p^k \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}\left\|e_m^k\right\|^2.
\end{aligned} \tag{A.33}
$$

Next, we work with the other "error" term. The same way as for (A.32) we get

$$
\begin{aligned}
&\frac{1}{M} \sum_{m=1}^{M} \mathbb{E}\left\|e_m^{k+1}\right\|^2 \\
&\quad = \frac{1}{M} \sum_{m=1}^{M} \left\|e_m^k + \gamma \cdot F_m(z^{k+1/2}) - \gamma \cdot F_m(w^k) - C_m^{\text{dev}}(\gamma \cdot F_m(z^{k+1/2}) - \gamma \cdot F_m(w^k) + e_m^k)\right\|^2
\end{aligned}
$$

$$\leq \frac{1}{M}\sum_{m=1}^{M}\left(1-\frac{1}{\delta^{\text{dev}}}\right)\left\|e_m^k+\gamma\cdot F_m(z^{k+1/2})-\gamma\cdot F_m(w^k)\right\|^2$$
$$\leq \frac{1}{M}\sum_{m=1}^{M}(1+c)\left(1-\frac{1}{\delta^{\text{dev}}}\right)\left\|e_m^k\right\|^2+\left(1+\frac{1}{c}\right)\left(1-\frac{1}{\delta^{\text{dev}}}\right)\gamma^2\cdot\left\|F_m(z^{k+1/2})-F_m(w^k)\right\|^2.$$

With $c=\frac{1}{2(\delta^{\text{dev}}-1)}$

$$\begin{aligned}\frac{1}{M}\sum_{m=1}^{M}\left\|e_m^{k+1}\right\|^2 &\leq \frac{1}{M}\sum_{m=1}^{M}\left(1-\frac{1}{2\delta^{\text{dev}}}\right)\left\|e_m^k\right\|^2+2\delta^{\text{dev}}\gamma^2\cdot\left\|F_m(z^{k+1/2})-F_m(w^k)\right\|^2\\ &\leq \left(1-\frac{1}{2\delta^{\text{dev}}}\right)\cdot\frac{1}{M}\sum_{m=1}^{M}\left\|e_m^k\right\|^2+2\delta^{\text{dev}}\gamma^2\tilde{L}^2\cdot\left\|z^{k+1/2}-w^k\right\|^2\\ &\leq 2\delta^{\text{dev}}\gamma^2\tilde{L}^2\sum_{j=0}^{k}\left(1-\frac{1}{2\delta^{\text{dev}}}\right)^{k-j}\cdot\left\|z^{j+1/2}-w^j\right\|^2.\end{aligned}$$

We weigh the sequence as follows $\sum_{k=0}^{K}p^k\frac{1}{M}\sum_{m=1}^{M}\left\|e_m^k\right\|^2$. Here we assume that p such that $p^k\leq p^j(1+1/4\delta^{\text{dev}})^{k-j}$. Then

$$\begin{aligned}\sum_{k=0}^{K-1}p^k\frac{1}{M}\sum_{m=1}^{M}\left\|e_m^k\right\|^2 &\leq 2\delta^{\text{dev}}\gamma^2\tilde{L}^2\sum_{k=0}^{K-1}p^k\sum_{j=0}^{k-1}\left(1-\frac{1}{2\delta^{\text{dev}}}\right)^{k-j-1}\cdot\left\|z^{j+1/2}-w^j\right\|^2\\ &\leq \frac{2\delta^{\text{dev}}\gamma^2\tilde{L}^2}{(1-1/2\delta^{\text{dev}})}\sum_{k=0}^{K-1}\sum_{j=0}^{k-1}p^j\left(1+\frac{1}{4\delta^{\text{dev}}}\right)^{k-j}\left(1-\frac{1}{2\delta^{\text{dev}}}\right)^{k-j}\cdot\left\|z^{j+1/2}-w^j\right\|^2\\ &\leq \frac{2\delta^{\text{dev}}\gamma^2\tilde{L}^2}{(1-1/2\delta^{\text{dev}})}\sum_{k=0}^{K-1}\sum_{j=0}^{k-1}p^j\left(1-\frac{1}{4\delta^{\text{dev}}}\right)^{k-j}\cdot\left\|z^{j+1/2}-w^j\right\|^2\\ &\leq \frac{2\delta^{\text{dev}}\gamma^2\tilde{L}^2}{(1-1/2\delta^{\text{dev}})}\sum_{k=0}^{K-1}p^k\left\|z^{k+1/2}-w^k\right\|^2\cdot\sum_{j=0}^{\infty}\left(1-\frac{1}{4\delta^{\text{dev}}}\right)^{j}\\ &\leq 16(\delta^{\text{dev}})^2\gamma^2\tilde{L}^2\sum_{k=0}^{K-1}p^k\left\|z^{k+1/2}-w^k\right\|^2.\end{aligned}\tag{A.34}$$

(A.33) together with (A.34) gives

$$\begin{aligned}&\sum_{k=0}^{K-1}p^k\left(\mathbb{E}\|\hat{z}^{k+1}-z^*\|^2+\mathbb{E}\|w^{k+1}-z^*\|^2\right)\\ &\quad\leq\sum_{k=0}^{K-1}p^k\left(1-\frac{\mu\gamma}{2}\right)\left(\mathbb{E}\|\hat{z}^k-z^*\|^2+\mathbb{E}\|w^k-z^*\|^2\right)\\ &\quad\quad-(1-\tau-\gamma\mu-2\gamma^2L^2-13200(\delta^{\text{serv}})^2(\delta^{\text{dev}})^2\gamma^2\tilde{L}^2)\cdot\sum_{k=0}^{K-1}p^k\mathbb{E}\|w^k-z^{k+1/2}\|^2\\ &\quad\quad-\left(\tau-2\gamma\mu-\frac{1}{2}\right)\cdot\sum_{k=0}^{K-1}p^k\mathbb{E}\|z^{k+1/2}-z^k\|^2.\end{aligned}$$

With $\tau \geq \frac{3}{4}$, $\gamma \leq \min\left[\frac{1-\tau}{8\mu}; \frac{\sqrt{1-\tau}}{2L+165\delta^{\text{serv}}\delta^{\text{dev}}\tilde{L}}\right]$ we obtain

$$\sum_{k=0}^{K-1} p^k \left(\mathbb{E}\|\hat{z}^{k+1} - z^*\|^2 + \mathbb{E}\|w^{k+1} - z^*\|^2\right) \leq \sum_{k=0}^{K-1} p^k \left(1 - \frac{\mu\gamma}{2}\right) \left(\mathbb{E}\|\hat{z}^{k+1} - z^*\|^2 + \mathbb{E}\|w^{k+1} - z^*\|^2\right).$$

Then we just need to take $p = 1/(1 - \mu\gamma/2)$ (easy to check that $p^k \leq p^j(1 + 1/8\delta^{\text{serv}})^{k-j}$ and $p^k \leq p^j(1 + 1/8\delta^{\text{dev}})^{k-j}$ work with our $\gamma \leq \frac{\sqrt{1-\tau}}{2L+150\delta^{\text{serv}}\delta^{\text{dev}}\tilde{L}}$) and get

$$\mathbb{E}\left(\|\hat{z}^K - z^*\|^2 + \|w^K - z^*\|^2\right) \leq \left(1 - \frac{\mu\gamma}{2}\right)^K \left(\|\hat{z}^0 - z^*\|^2 + \|w^0 - z^*\|^2\right).$$

This ends the proof for strongly-monotone case. □

A.2.6 Proof of Theorems 3.1.2, 3.1.5, 3.1.7

Theorem A.2.20 (Theorem 3.1.2)**.** Let F be μ-quasi-strongly monotone with $\mu > 0$ and Assumption 3.1.1 hold. Assume that

$$0 < \gamma \leq \min\left\{\frac{1}{\mu}, \frac{1}{2(A+CM)}\right\}$$

for some $M > {}^{B}/_{\rho}$. Then for the Lyapunov function $V_k = \|z^k - z^{*,k}\|^2 + M\gamma^2\sigma_k^2$, and for all $k \geq 0$ we have

$$\mathbb{E}[V_k] \leq \left(1 - \min\left\{\gamma\mu, \rho - \frac{B}{M}\right\}\right)^k \mathbb{E}[V_0] + \frac{\gamma^2(D_1 + MD_2)}{\min\left\{\gamma\mu, \rho - {}^{B}/_{M}\right\}}.$$

Proof. First of all, we recall a well-known fact about proximal operators: for any solution z^* of (3.1) we have

$$z^* = \mathrm{prox}_{\gamma g}(z^* - \gamma F(z^*)).$$

Using this and non-expansiveness of proximal operator, we derive

$$\begin{aligned}
\|z^{k+1} - z^{*,k+1}\|^2 &\leq \|z^{k+1} - z^{*,k}\|^2 \\
&= \left\|\mathrm{prox}_{\gamma g}(z^k - \gamma g^k) - \mathrm{prox}_{\gamma R}(z^{*,k} - \gamma F(z^{*,k}))\right\|^2 \\
&\leq \left\|z^k - \gamma g^k - z^{*,k} - \gamma F(z^{*,k})\right\|^2 \\
&= \|z^k - z^{*,k}\|^2 - 2\gamma\left\langle z^k - z^{*,k}, g^k - F(z^{*,k})\right\rangle + \gamma^2\|g^k - F(z^{*,k})\|^2.
\end{aligned}$$

Next, we take an expectation $\mathbb{E}_k[\cdot]$ w.r.t. the randomness at iteration k and use (3.6)

$$\begin{aligned}
\mathbb{E}_k\left[\|z^{k+1} - z^{*,k+1}\|^2\right] =& \|z^k - z^{*,k}\|^2 - 2\gamma\left\langle z^k - z^{*,k}, F(z^k) - F(z^{*,k})\right\rangle \\
&+ \gamma^2\mathbb{E}_k\left[\left\|g^k - F(z^{*,k})\right\|^2\right] \\
\leq& \|z^k - z^{*,k}\|^2 - 2\gamma\left\langle z^k - z^*, F(z^k) - F(z^{*,k})\right\rangle \\
&+ \gamma^2\left(2A\left\langle z^k - z^{*,k}, F(z^k) - F(z^{*,k})\right\rangle + B\sigma_k^2 + D_1\right).
\end{aligned}$$

Summing up this inequality with (3.7) multiplied by $M\gamma^2$, we obtain

$$\begin{aligned}
&\mathbb{E}_k\left[\|z^{k+1} - z^{*,k+1}\|^2\right] + M\gamma^2\mathbb{E}_k[\sigma_{k+1}^2] \\
&\qquad\leq \|z^k - z^{*,k}\|^2 - 2\gamma\left\langle z^k - z^{*,k}, F(z^k) - F(z^{*,k})\right\rangle \\
&\qquad\quad + \gamma^2\left(2A\left\langle z^k - z^{*,k}, F(z^k) - F(z^{*,k})\right\rangle + B\sigma_k^2 + D_1\right) \\
&\qquad\quad + M\gamma^2\left(2C\left\langle z^k - z^{*,k}, F(z^k) - F(z^{*,k})\right\rangle + (1-\rho)\sigma_k^2 + D_2\right) \\
&\qquad= \|z^k - z^{*,k}\|^2 + M\gamma^2\left(1 - \rho + \frac{B}{M}\right)\sigma_k^2 + \gamma^2(D_1 + MD_2) \\
&\qquad\quad - 2\gamma\left(1 - \gamma(A+CM)\right)\left\langle z^k - z^{*,k}, F(z^k) - F(z^{*,k})\right\rangle. \qquad \text{(A.35)}
\end{aligned}$$

Since $\gamma \leq \frac{1}{2(A+CM)}$ the factor $-2\gamma\left(1 - \gamma(A+CM)\right)$ is non-positive. Therefore, applying strong

quasi-monotonicity of F, we derive

$$\mathbb{E}_k\left[\|z^{k+1}-z^{*,k+1}\|^2+M\gamma^2\sigma_{k+1}^2\right]\leq(1-2\gamma\mu\left(1-\gamma(A+CM)\right))\|z^k-z^{*,k}\|^2 + M\gamma^2\left(1-\rho+\frac{B}{M}\right)\sigma_k^2+\gamma^2(D_1+MD_2).$$

Using $\gamma\leq\frac{1}{2(A+CM)}$ and the definition $V_k=\|z^k-z^{*,k}\|^2+M\gamma^2\sigma_k^2$, we get

$$\begin{aligned}\mathbb{E}_k\left[V_{k+1}\right]&\leq(1-\gamma\mu)\|z^k-z^{*,k}\|^2+M\gamma^2\left(1-\rho+\frac{B}{M}\right)\sigma_k^2+\gamma^2(D_1+MD_2)\\&\leq\left(1-\min\left\{\gamma\mu,\rho-\frac{B}{M}\right\}\right)V_k+\gamma^2(D_1+MD_2).\end{aligned}$$

Next, we take the full expectation from the above inequality and establish the following recurrence:

$$\mathbb{E}\left[V_{k+1}\right]\leq\left(1-\min\left\{\gamma\mu,\rho-\frac{B}{M}\right\}\right)\mathbb{E}[V_k]+\gamma^2(D_1+MD_2).$$

Unrolling the recurrence, we derive

$$\begin{aligned}\mathbb{E}\left[V_k\right]&\leq\left(1-\min\left\{\gamma\mu,\rho-\frac{B}{M}\right\}\right)^k\mathbb{E}[V_0]+\gamma^2(D_1+MD_2)\sum_{t=0}^{k-1}\left(1-\min\left\{\gamma\mu,\rho-\frac{B}{M}\right\}\right)^t\\&\leq\left(1-\min\left\{\gamma\mu,\rho-\frac{B}{M}\right\}\right)^k\mathbb{E}[V_0]+\gamma^2(D_1+MD_2)\sum_{t=0}^{\infty}\left(1-\min\left\{\gamma\mu,\rho-\frac{B}{M}\right\}\right)^t\\&=\left(1-\min\left\{\gamma\mu,\rho-\frac{B}{M}\right\}\right)^k\mathbb{E}[V_0]+\frac{\gamma^2(D_1+MD_2)}{\min\left\{\gamma\mu,\rho-{}^{B}/_{M}\right\}},\end{aligned}$$

which finishes the proof. □

Theorem A.2.21 (Theorem 3.1.5)**.** Let F be monotone, ℓ-star-cocoercive and Assumptions 3.1.1, 3.1.4 hold. Assume that

$$0<\gamma\leq\frac{1}{2(A+{}^{BC}/_{\rho})}.$$

Then for the function $\mathrm{Gap}(z)$ from (4.3) and for all $K\geq0$ we have

$$\begin{aligned}\mathbb{E}\left[\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K}z^k\right)\right]\leq&\frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K}\\&+\frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K}+(4A+\ell+{}^{8BC}/_{\rho})\cdot\frac{\|z^0-z^{*,0}\|^2}{K}\\&+(4+(4A+\ell+{}^{8BC}/_{\rho})\,\gamma)\,\frac{\gamma B\sigma_0^2}{\rho K}\\&+\gamma(2+\gamma\,(4A+\ell+{}^{8BC}/_{\rho}))(D_1+{}^{2BD_2}/_{\rho})\\&+9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2.\end{aligned}$$

Proof. First, we apply the classical result about proximal operators (Lemma A.1.3) with $x^+ = z^{k+1}$, $x = z^k - \gamma g^k$, and $z = u$ for arbitrary point $u \in \mathbb{R}^d$:

$$\langle z^{k+1} - z^k + \gamma g^k, u - z^{k+1} \rangle \geq \gamma \left(g(z^{k+1}) - g(u) \right).$$

Multiplying by the factor of 2 and making small rearrangement, we get

$$2\gamma \langle g^k, u - z^k \rangle + 2\langle z^{k+1} - z^k, u - z^k \rangle + 2\langle z^{k+1} - z^k + \gamma g^k, z^k - z^{k+1} \rangle \geq 2\gamma \left(g(z^{k+1}) - g(u) \right)$$

implying

$$\begin{aligned} 2\gamma \left(\langle F(z^k), z^k - u \rangle + g(z^{k+1}) - g(u) \right) \leq & 2\langle z^{k+1} - z^k, u - z^k \rangle + 2\gamma \langle F(z^k) - g^k, z^k - u \rangle \\ & + 2\langle z^{k+1} - z^k, z^k - z^{k+1} \rangle + 2\gamma \langle g^k, z^k - z^{k+1} \rangle. \end{aligned}$$

Next, we use a squared norm decomposition $\|a + b\|^2 = \|a\|^2 + \|b\|^2 + 2\langle a, b \rangle$, and obtain

$$\begin{aligned} 2\gamma \left(\langle F(z^k), z^k - u \rangle + g(z^{k+1}) - g(u) \right) \leq & \|z^{k+1} - z^k\|^2 + \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma \langle F(z^k) - g^k, z^k - u \rangle \\ & - 2\|z^{k+1} - z^k\|^2 + 2\gamma \langle g^k, z^k - z^{k+1} \rangle. \end{aligned} \tag{A.36}$$

Then, due to $2\langle a, b \rangle \leq \|a\|^2 + \|b\|^2$ we have

$$\begin{aligned} 2\gamma \left(\langle F(z^k), z^k - u \rangle + g(z^{k+1}) - g(u) \right) \leq & \|z^{k+1} - z^k\|^2 + \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma \langle F(z^k) - g^k, z^k - u \rangle \\ & - 2\|z^{k+1} - z^k\|^2 + \gamma^2 \|g^k\|^2 + \|z^k - z^{k+1}\|^2 \\ = & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma \langle F(z^k) - g^k, z^k - u \rangle + \gamma^2 \|g^k\|^2. \end{aligned}$$

Monotonicity of F implies $\langle F(u), z^k - u \rangle \leq \langle F(z^k), z^k - u \rangle$, allowing us to continue our derivation as follows:

$$\begin{aligned} 2\gamma \left(\langle F(u), z^k - u \rangle + g(z^{k+1}) - g(u) \right) \leq & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma \langle F(z^k) - g^k, z^k - u \rangle + \gamma^2 \|g^k\|^2 \\ = & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma \langle F(z^k) - g^k, z^k - u \rangle \\ & + \gamma^2 \|g^k - g^{*,k} + g^{*,k}\|^2 \\ \leq & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma \langle F(z^k) - g^k, z^k - u \rangle \\ & + 2\gamma^2 \|g^k - g^{*,k}\|^2 + 2\gamma^2 \|g^{*,k}\|^2. \end{aligned}$$

Summing up the above inequality for $k = 0, 1, \ldots, K-1$, we get

$$\begin{aligned}
2\gamma \sum_{k=0}^{K-1} \left(\langle F(u), z^k - u \rangle + g(z^{k+1}) - g(u) \right) \leq & \sum_{k=0}^{K-1} \|z^k - u\|^2 - \sum_{k=0}^{K-1} \|z^{k+1} - u\|^2 \\
& + 2\gamma^2 \sum_{k=0}^{K-1} \|g^{*,k}\|^2 \\
& + 2\gamma \sum_{k=0}^{K-1} \langle F(z^k) - g^k, z^k - u \rangle \\
& + 2\gamma^2 \sum_{k=0}^{K-1} \|g^k - g^{*,k}\|^2 \\
= & \|z^0 - u\|^2 - \|z^K - u\|^2 + 2\gamma^2 \sum_{k=0}^{K-1} \|g^{*,k}\|^2 \\
& + 2\gamma \sum_{k=0}^{K-1} \langle F(z^k) - g^k, z^k - u \rangle \\
& + 2\gamma^2 \sum_{k=0}^{K-1} \|g^k - g^{*,k}\|^2.
\end{aligned}$$

Next, we divide both sides by $2\gamma K$

$$\begin{aligned}
\frac{1}{K} \sum_{k=0}^{K-1} \left(\langle F(u), z^k - u \rangle + g(z^{k+1}) - g(u) \right) \leq & \frac{\|z^0 - u\|^2 - \|z^K - u\|^2}{2\gamma K} + \frac{\gamma}{K} \sum_{k=0}^{K-1} \|g^{*,k}\|^2 \\
& + \frac{1}{K} \sum_{k=0}^{K-1} \langle F(z^k) - g^k, z^k - u \rangle \\
& + \frac{\gamma}{K} \sum_{k=0}^{K-1} \|g^k - g^{*,k}\|^2
\end{aligned}$$

and, after small rearrangement, we obtain

$$\begin{aligned}
\frac{1}{K} \sum_{k=0}^{K-1} \left(\langle F(u), z^{k+1} - u \rangle + g(z^{k+1}) - g(u) \right) \leq & \frac{\|z^0 - u\|^2 - \|z^K - u\|^2}{2\gamma K} + \frac{\langle F(u), z^K - z^0 \rangle}{K} \\
& + \frac{\gamma}{K} \sum_{k=0}^{K-1} \|g^{*,k}\|^2 \\
& + \frac{1}{K} \sum_{k=0}^{K-1} \langle F(z^k) - g^k, z^k - u \rangle \\
& + \frac{\gamma}{K} \sum_{k=0}^{K-1} \|g^k - g^{*,k}\|^2.
\end{aligned}$$

Applying Jensen's inequality for convex function g, we get $g\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1}\right) \leq \frac{1}{K}\sum_{k=0}^{K-1} g(z^{k+1})$.

Plugging this in the previous inequality, we derive for u^* being a projection of u on $\mathcal{Z}^*$

$$
\begin{aligned}
&\left\langle F(u), \left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1}\right) - u \right\rangle + g\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1}\right) - g(u) \\
&\quad\leq \frac{\|z^0 - u\|^2 - \|z^K - u\|^2}{2\gamma K} + \frac{\langle F(u), z^K - z^0\rangle}{K} + \frac{\gamma}{K}\sum_{k=0}^{K-1}\|g^{*,k}\|^2 \\
&\qquad + \frac{1}{K}\sum_{k=0}^{K-1}\langle F(z^k) - g^k, z^k - u\rangle + \frac{\gamma}{K}\sum_{k=0}^{K-1}\|g^k - g^{*,k}\|^2 \\
&\quad\leq \frac{\|z^0 - u\|^2 - \|z^K - u\|^2}{2\gamma K} + \frac{\|z^K - z^0\|^2}{4\gamma K} + \frac{4\gamma}{K}\|F(u) - F(u^*) + F(u^*)\|^2 \\
&\qquad + \frac{\gamma}{K}\sum_{k=0}^{K-1}\|g^{*,k}\|^2 + \frac{1}{K}\sum_{k=0}^{K-1}\langle F(z^k) - g^k, z^k - u\rangle + \frac{\gamma}{K}\sum_{k=0}^{K-1}\|g^k - g^{*,k}\|^2 \\
&\quad\leq \frac{\|z^0 - u\|^2 - \|z^K - u\|^2}{2\gamma K} + \frac{\|z^0 - u\|^2 + \|z^K - u\|^2}{2\gamma K} + \frac{8\gamma}{K}\|F(u) - F(u^*)\|^2 \\
&\qquad + \frac{\gamma}{K}\sum_{k=0}^{K-1}\|g^{*,k}\|^2 + 8\gamma\|F(u^*)\|^2 + \frac{1}{K}\sum_{k=0}^{K-1}\langle F(z^k) - g^k, z^k - u\rangle \\
&\qquad + \frac{\gamma}{K}\sum_{k=0}^{K-1}\|g^k - g^{*,k}\|^2 \\
&\quad\leq \frac{\|z^0 - u\|^2}{\gamma K} + \frac{8\gamma\ell^2\|u - u^*\|^2}{K} + 9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\
&\qquad + \frac{1}{K}\sum_{k=0}^{K-1}\langle F(z^k) - g^k, z^k - u\rangle + \frac{\gamma}{K}\sum_{k=0}^{K-1}\|g^k - g^{*,k}\|^2.
\end{aligned}
$$

Next, we take maximum from the both sides in $u \in \mathcal{C}$, which gives $\text{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)$ in the left-hand side by definition of the gap function, and take the expectation of the result:

$$
\begin{aligned}
\mathbb{E}\left[\text{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq& \frac{\mathbb{E}\left[\max_{u\in\mathcal{C}}\|z^0 - u\|^2\right]}{\gamma K} + \frac{8\gamma\ell^2\mathbb{E}\left[\max_{u\in\mathcal{C}}\|u - u^*\|^2\right]}{K} \\
&+ 9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\
&+ \frac{1}{K}\mathbb{E}\left[\max_{u\in\mathcal{C}}\sum_{k=0}^{K-1}\langle F(z^k) - g^k, z^k - u\rangle\right] + \frac{\gamma}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|g^k - g^{*,k}\|^2\right] \\
\leq& \frac{\mathbb{E}\left[\max_{u\in\mathcal{C}}\|z^0 - u\|^2\right]}{\gamma K} + \frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K} + 9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\
&+ \frac{1}{K}\mathbb{E}\left[\max_{u\in\mathcal{C}}\sum_{k=0}^{K-1}\langle F(z^k) - g^k, z^k - u\rangle\right] \\
&+ \frac{\gamma}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|g^k - g^{*,k}\|^2\right].
\end{aligned} \tag{A.37}
$$

In the last step, we also use that $\mathcal{Z}^* \subset C$ and $\Omega_{\mathcal{C}} := \max_{x,y\in C}\|x - y\|$ (Assumption 3.1.4).

It remains to upper bound the terms from the last two lines of (A.37). We start with the first one. Since

$$\mathbb{E}\left[\sum_{k=0}^{K-1}\langle F(z^k)-g^k, z^k\rangle\right] = \mathbb{E}\left[\sum_{k=0}^{K-1}\langle \mathbb{E}[F(z^k)-g^k \mid z^k], z^k\rangle\right] = 0,$$
$$\mathbb{E}\left[\sum_{k=0}^{K-1}\langle F(z^k)-g^k, z^0\rangle\right] = \sum_{k=0}^{K-1}\left\langle \mathbb{E}[F(z^k)-g^k], z^0\right\rangle = 0,$$

we have

$$\begin{aligned}
&\frac{1}{K}\mathbb{E}\Big[\max_{u\in\mathcal{C}}\sum_{k=0}^{K-1}\langle F(z^k)-g^k, z^k-u\rangle\Big] \\
&\quad= \frac{1}{K}\mathbb{E}\left[\sum_{k=0}^{K-1}\langle F(z^k)-g^k, z^k\rangle\right] \\
&\qquad+ \frac{1}{K}\mathbb{E}\left[\max_{u\in\mathcal{C}}\sum_{k=0}^{K-1}\langle F(z^k)-g^k, -u\rangle\right] \\
&\quad= \frac{1}{K}\mathbb{E}\left[\max_{u\in\mathcal{C}}\sum_{k=0}^{K-1}\langle F(z^k)-g^k, -u\rangle\right] \\
&\quad= \frac{1}{K}\mathbb{E}\left[\sum_{k=0}^{K-1}\langle F(z^k)-g^k, z^0\rangle\right] \\
&\qquad+ \frac{1}{K}\mathbb{E}\left[\max_{u\in\mathcal{C}}\sum_{k=0}^{K-1}\langle F(z^k)-g^k, -u\rangle\right] \\
&\quad= \mathbb{E}\left[\max_{u\in C}\left\langle \frac{1}{K}\sum_{k=0}^{K-1}(F(z^k)-g^k), z^0-u\right\rangle\right] \\
&\quad\le \mathbb{E}\left[\max_{u\in C}\left\{\frac{\gamma K}{2}\left\|\frac{1}{K}\sum_{k=0}^{K-1}(F(z^k)-g^k)\right\|^2 + \frac{1}{2\gamma K}\|z^0-u\|^2\right\}\right] \\
&\quad= \frac{\gamma}{2K}\mathbb{E}\left[\left\|\sum_{k=0}^{K-1}(F(z^k)-g^k)\right\|^2\right] + \frac{1}{2\gamma K}\max_{u\in C}\|z^0-u\|^2.
\end{aligned}$$

We notice that $\mathbb{E}[F(z^k)-g^k \mid F(z^0)-g^0, \ldots, F(z^{k-1})-g^{k-1}] = 0$ for all $k \ge 1$, i.e., conditions of Lemma A.1.4 are satisfied. Therefore, applying Lemma A.1.4, we get

$$\begin{aligned}
\frac{1}{K}\mathbb{E}\left[\max_{u\in\mathcal{C}}\sum_{k=0}^{K-1}\langle F(z^k)-g^k, z^k-u\rangle\right] \le& \frac{\gamma}{2K}\sum_{k=0}^{K-1}\mathbb{E}[\|F(z^k)-g^k\|^2] \\
&+ \frac{1}{2\gamma K}\max_{u\in C}\|z^0-u\|^2. \qquad \text{(A.38)}
\end{aligned}$$

Combining (A.37) and (A.38), we derive

$$\begin{aligned}\mathbb{E}\left[\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq & \frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K} + \frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K} + 9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\ & + \frac{\gamma}{2K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|g^k - F(z^k)\|^2\right] \\ & + \frac{\gamma}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|g^k - g^{*,k}\|^2\right] \\ \leq & \frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K} + \frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K} + 9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\ & + \frac{\gamma}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|F(z^k) - g^{*,k}\|^2\right] + \frac{2\gamma}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|g^k - g^{*,k}\|^2\right].\end{aligned} \tag{A.39}$$

Using ℓ-star-cocoercivity of F together with the first part of Assumption 3.1.1, we continue our derivation as follows:

$$\begin{aligned}\mathbb{E}\left[\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq & \frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K} + \frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K} + 2\gamma D_1 + 9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\ & + \frac{\gamma(4A+\ell)}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\langle F(z^k) - g^{*,k}, z^k - z^{*,k}\rangle\right] + \frac{2\gamma B}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\sigma_k^2\right] \\ = & \frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K} + \frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K} + 2\gamma D_1 + 9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\ & + \frac{\gamma(4A+\ell)}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\langle F(z^k) - g^{*,k}, z^k - z^{*,k}\rangle\right] \\ & + \frac{2\gamma B}{K}\left(1+\frac{1}{\rho}\right)\sum_{k=0}^{K-1}\mathbb{E}\left[\sigma_k^2\right] - \frac{2\gamma B}{\rho K}\sum_{k=0}^{K-1}\mathbb{E}\left[\sigma_k^2\right].\end{aligned}$$

Next, we use the second part of Assumption 3.1.1 and get

$$\begin{aligned}\mathbb{E}\left[\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq & \frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K} + \frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K} + 2\gamma D_1 + 9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\ & + \frac{\gamma(4A+\ell)}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\langle F(z^k) - g^{*,k}, z^k - z^{*,k}\rangle\right] \\ & + \frac{2\gamma B}{K}\left(1+\frac{1}{\rho}\right)\sum_{k=1}^{K-1}\mathbb{E}\left[2C\langle F(z^{k-1}) - g^{*,k-1}, z^{k-1} - z^{*,k-1}\rangle\right] \\ & + \frac{2\gamma B}{K}\left(1+\frac{1}{\rho}\right)\sum_{k=1}^{K-1}\mathbb{E}\left[(1-\rho)\sigma_{k-1}^2 + D_2\right] \\ & + \frac{2\gamma B}{K}\left(1+\frac{1}{\rho}\right)\sigma_0^2 - \frac{2\gamma B}{\rho K}\sum_{k=0}^{K-1}\mathbb{E}\left[\sigma_k^2\right]\end{aligned}$$

$$
\begin{aligned}
\leq & \frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K}+\frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K}+\frac{2\gamma B(1+1/\rho)}{K}\sigma_0^2 \\
& +2\gamma\left(D_1+B(1+1/\rho)D_2\right) \\
& +9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2+\frac{\gamma(4A+\ell)}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\langle F(z^k)-g^{*,k},z^k-z^{*,k}\rangle\right] \\
& +\frac{2\gamma B}{K}\left(1+\frac{1}{\rho}\right)\sum_{k=0}^{K-2}\mathbb{E}\left[2C\langle F(z^k)-g^{*,k},z^k-z^{*,k}\rangle+(1-\rho)\sigma_k^2\right] \\
& -\frac{2\gamma B}{\rho K}\sum_{k=0}^{K-1}\mathbb{E}\left[\sigma_k^2\right] \\
\leq & \frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K}+\frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K}+\frac{2\gamma B(1+1/\rho)}{K}\sigma_0^2 \\
& +2\gamma\left(D_1+B(1+1/\rho)D_2\right)+9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\
& +\left(4A+\ell+4BC(1+1/\rho)\right)\frac{\gamma}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\langle F(z^k)-g^{*,k},z^k-z^{*,k}\rangle\right] \\
& +\frac{2\gamma B}{K}(1-\rho)\left(1+\frac{1}{\rho}\right)\sum_{k=0}^{K-2}\mathbb{E}\left[\sigma_k^2\right]-\frac{2\gamma B}{\rho K}\sum_{k=0}^{K-1}\mathbb{E}\left[\sigma_k^2\right].
\end{aligned}
$$

Since $(1-\rho)(1+1/\rho)=-\rho+1/\rho\leq 1/\rho$, the last row is non-positive and we have

$$
\begin{aligned}
\mathbb{E}\left[\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K}z^k\right)\right]\leq & \frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K}+\frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K}+\frac{2\gamma B(1+1/\rho)}{K}\sigma_0^2 \\
& +2\gamma\left(D_1+B(1+1/\rho)D_2\right)+9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\
& +\frac{\gamma\left(4A+\ell+4BC(1+1/\rho)\right)}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\langle F(z^k)-g^{*,k},z^k-z^{*,k}\rangle\right].
\end{aligned}
\tag{A.40}
$$

Note that inequality (A.35) from the proof of Theorem 3.1.2 is derived using Assumption 3.1.1 only. With $M=B/\rho$ it gives

$$
\begin{aligned}
\mathbb{E}\left[\|z^{k+1}-z^{*,k+1}\|^2\right]+\frac{\gamma^2 B}{\rho}\mathbb{E}[\sigma_{k+1}^2]\leq & \mathbb{E}\left[\|z^k-z^{*,k}\|^2\right]+\frac{\gamma^2 B}{\rho}\mathbb{E}\left[\sigma_k^2\right]+\gamma^2(D_1+BD_2/\rho) \\
& -2\gamma\left(1-\gamma(A+BC/\rho)\right)\mathbb{E}\left[\left\langle z^k-z^{*,k},F(z^k)-g^{*,k}\right\rangle\right].
\end{aligned}
$$

Since $\gamma\leq 1/2(A+BC/\rho)$ we obtain

$$
\begin{aligned}
\gamma\mathbb{E}\left[\left\langle z^k-z^{*,k},F(z^k)-g^{*,k}\right\rangle\right]\leq & \mathbb{E}\left[\|z^k-z^{*,k}\|^2\right]+\frac{\gamma^2 B}{\rho}\mathbb{E}\left[\sigma_k^2\right]-\mathbb{E}\left[\|z^{k+1}-z^{*,k+1}\|^2\right] \\
& -\frac{\gamma^2 B}{\rho}\mathbb{E}[\sigma_{k+1}^2]+\gamma^2(D_1+BD_2/\rho).
\end{aligned}
$$

Plugging this inequality in (A.40), we derive

$$\begin{aligned}
\mathbb{E}\left[\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq & \frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K} + \frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K} + \frac{2\gamma B(1+1/\rho)}{K}\sigma_0^2 \\
& + 2\gamma\left(D_1 + B(1+1/\rho)D_2\right) + 9\gamma\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2 \\
& + (4A+\ell+4BC(1+1/\rho))\cdot\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|z^k - z^{*,k}\|^2\right] \\
& - (4A+\ell+4BC(1+1/\rho))\cdot\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|z^{k+1} - z^{*,k+1}\|^2\right] \\
& + (4A+\ell+4BC(1+1/\rho))\cdot\frac{\gamma^2 B}{\rho K}\sum_{k=0}^{K-1}\mathbb{E}\left[\sigma_k^2 - \sigma_{k+1}^2\right] \\
& + \gamma^2(4A+\ell+4BC(1+1/\rho))\cdot(D_1 + BD_2/\rho) \\
\leq & \frac{3\left[\max_{u\in\mathcal{C}}\|z^0-u\|^2\right]}{2\gamma K} + \frac{8\gamma\ell^2\Omega_{\mathcal{C}}^2}{K} + (4A+\ell+8BC/\rho)\cdot\frac{\|z^0 - z^{*,0}\|^2}{K} \\
& + \left(4 + (4A+\ell+8BC/\rho)\,\gamma\right)\frac{\gamma B\sigma_0^2}{\rho K} \qquad \text{(A.41)} \\
& + \gamma\left((2+\gamma(4A+\ell+8BC/\rho))(D_1 + 2BD_2/\rho) + 9\max_{z^*\in\mathcal{Z}^*}\|F(z^*)\|^2\right),
\end{aligned}$$

where in the last inequality we use $1 + 1/\rho \leq 2/\rho$. □

Theorem A.2.22 (Theorem 3.1.7)**.** Let F be ℓ-cocoercive and Assumptions 3.1.1, 3.1.4 hold. Assume that

$$0 < \gamma \leq \min\left\{\frac{1}{\ell}, \frac{1}{2(A + {}^{BC}/_{\rho})}\right\}.$$

Then for the function $\mathrm{Gap}(z)$ and for all $K \geq 0$ we have

$$\begin{aligned}\mathbb{E}\left[\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq & \frac{3\max_{u\in\mathcal{C}}\|z^0 - u\|^2}{2\gamma K} + (6A + 3\ell + {}^{12BC}/_{\rho}) \cdot \frac{\|z^0 - z^{*,0}\|^2}{K} \\ & + (6 + (6A + 3\ell + {}^{12BC}/_{\rho})\,\gamma)\,\frac{\gamma B\sigma_0^2}{\rho K} \\ & + \gamma(3 + \gamma\,(6A + 3\ell + {}^{12BC}/_{\rho}))(D_1 + {}^{2BD_2}/_{\rho}).\end{aligned}$$

Proof. We start the proof from (A.36).

$$\begin{aligned}2\gamma\left(\langle F(z^k), z^k - u\rangle + g(z^{k+1}) - g(u)\right) \leq & \|z^{k+1} - z^k\|^2 + \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma\langle F(z^k) - g^k, z^k - u\rangle \\ & - 2\|z^{k+1} - z^k\|^2 + 2\gamma\langle g^k, z^k - z^{k+1}\rangle \\ = & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma\langle F(z^k) - g^k, z^k - u\rangle \\ & - \|z^{k+1} - z^k\|^2 + 2\gamma\langle F(u), z^k - z^{k+1}\rangle \\ & + 2\gamma\langle g^k - F(u), z^k - z^{k+1}\rangle.\end{aligned}$$

Then, due to $2\langle a, b\rangle \leq \|a\|^2 + \|b\|^2$ we have

$$\begin{aligned}2\gamma\left(\langle F(z^k), z^k - u\rangle + g(z^{k+1}) - g(u)\right) \leq & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma\langle F(z^k) - g^k, z^k - u\rangle \\ & - \|z^{k+1} - z^k\|^2 + 2\gamma\langle F(u), z^k - z^{k+1}\rangle \\ & + \gamma^2\|g^k - F(u)\|^2 + \|z^k - z^{k+1}\|^2 \\ = & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma\langle F(z^k) - g^k, z^k - u\rangle \\ & + 2\gamma\langle F(u), z^k - z^{k+1}\rangle + \gamma^2\|g^k - F(u)\|^2.\end{aligned}$$

Next, we add $2\gamma\left(\langle F(u), z^{k+1} - u\rangle - \langle F(z^k), z^k - u\rangle\right)$ to both sides of the previous inequality.

$$\begin{aligned}2\gamma\left(\langle F(u), z^{k+1} - u\rangle + g(z^{k+1}) - g(u)\right) \leq & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & + 2\gamma\langle F(u) - g^k, z^k - u\rangle + \gamma^2\|g^k - F(u)\|^2 \\ \leq & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\ & - 2\gamma\langle F(z^k) - F(u), z^k - u\rangle \\ & - 2\gamma\langle g^k - F(z^k), z^k - u\rangle\end{aligned}$$

$$+ 2\gamma^2\|g^k - F(z^k)\|^2 + 2\gamma^2\|F(z^k) - F(u)\|^2.$$

Using that F is ℓ-co-cocoercive, we get

$$\begin{aligned}
2\gamma\left(\langle F(u), z^{k+1} - u\rangle + g(z^{k+1}) - g(u)\right) \leq & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\
& - \frac{2\gamma}{\ell}\|F(z^k) - F(u)\|^2 \\
& - 2\gamma\langle g^k - F(z^k), z^k - u\rangle \\
& + 2\gamma^2\|g^k - F(z^k)\|^2 + 2\gamma^2\|F(z^k) - F(u)\|^2 \\
= & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\
& - \frac{2\gamma}{\ell}\left(1 - \gamma\ell\right)\|F(z^k) - F(u)\|^2 \\
& - 2\gamma\langle g^k - F(z^k), z^k - u\rangle \\
& + 2\gamma^2\|g^k - F(z^k)\|^2.
\end{aligned}$$

With $\gamma \leq \frac{1}{\ell}$, we have

$$\begin{aligned}
2\gamma\left(\langle F(u), z^{k+1} - u\rangle + g(z^{k+1}) - g(u)\right) \leq & \|z^k - u\|^2 - \|z^{k+1} - u\|^2 \\
& - 2\gamma\langle g^k - F(z^k), z^k - u\rangle \\
& + 2\gamma^2\|g^k - F(z^k)\|^2.
\end{aligned}$$

Summing up the above inequality for $k = 0, 1, \ldots, K-1$, we get

$$\begin{aligned}
2\gamma\sum_{k=0}^{K-1}\left(\langle F(u), z^{k+1} - u\rangle + g(z^{k+1}) - g(u)\right) \leq & \sum_{k=0}^{K-1}\|z^k - u\|^2 - \sum_{k=0}^{K-1}\|z^{k+1} - u\|^2 \\
& + 2\gamma^2\sum_{k=0}^{K-1}\|g^k - F(z^k)\|^2 \\
& + 2\gamma\sum_{k=0}^{K-1}\langle F(z^k) - g^k, z^k - u\rangle.
\end{aligned}$$

Next, we divide both sides by $2\gamma K$

$$\begin{aligned}
\frac{1}{K}\sum_{k=0}^{K-1}\left(\langle F(u), z^{k+1} - u\rangle + g(z^{k+1}) - g(u)\right) \leq & \frac{\|z^0 - u\|^2 - \|z^K - u\|^2}{2\gamma K} \\
& + \frac{\gamma}{K}\sum_{k=0}^{K-1}\|g^k - F(z^k)\|^2 \\
& + \frac{1}{K}\sum_{k=0}^{K-1}\langle F(z^k) - g^k, z^k - u\rangle.
\end{aligned}$$

Applying Jensen's inequality for convex function g, we get $g\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1}\right) \leq \frac{1}{K}\sum_{k=0}^{K-1} g(z^{k+1})$.

$$\begin{aligned}\left\langle F(u), \left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1}\right) - u\right\rangle + g\left(\frac{1}{K}\sum_{k=0}^{K-1} z^{k+1}\right) - g(u) \leq & \frac{\|z^0 - u\|^2 - \|z^K - u\|^2}{2\gamma K} \\ & + \frac{\gamma}{K}\sum_{k=0}^{K-1}\|g^k - F(z^k)\|^2 \\ & + \frac{1}{K}\sum_{k=0}^{K-1}\langle F(z^k) - g^k, z^k - u\rangle.\end{aligned}$$

Next, we take maximum from the both sides in $u \in \mathcal{C}$, which gives $\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)$ in the left-hand side by definition of the gap function, and take the expectation of the result:

$$\begin{aligned}\mathbb{E}\left[\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq & \frac{\mathbb{E}\left[\max_{u\in\mathcal{C}}\|z^0 - u\|^2\right]}{\gamma K} + \frac{\gamma}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|g^k - F(z^k)\|^2\right] \\ & + \frac{1}{K}\mathbb{E}\left[\max_{u\in\mathcal{C}}\sum_{k=0}^{K-1}\langle F(z^k) - g^k, z^k - u\rangle\right].\end{aligned}$$

Using the estimate (A.38), we get

$$\begin{aligned}\mathbb{E}\left[\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq & \frac{\max_{u\in\mathcal{C}}\|z^0 - u\|^2}{\gamma K} + \frac{\gamma}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|g^k - F(z^k)\|^2\right] \\ & + \frac{\gamma}{2K}\sum_{k=0}^{K-1}\mathbb{E}[\|F(z^k) - g^k\|^2] + \frac{1}{2\gamma K}\max_{u\in C}\|z^0 - u\|^2 \\ \leq & \frac{3\max_{u\in\mathcal{C}}\|z^0 - u\|^2}{2\gamma K} + \frac{3\gamma}{2K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|g^k - F(z^k)\|^2\right].\end{aligned}$$

It remains to estimate $\frac{1}{K}\sum\limits_{k=0}^{K-1}\mathbb{E}\left[\|g^k - F(z^k)\|^2\right]$. This was done in the previous proof (see from (A.39) to (A.41)). Then, we finally have

$$\begin{aligned}\mathbb{E}\left[\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K} z^k\right)\right] \leq & \frac{3\max_{u\in\mathcal{C}}\|z^0 - u\|^2}{2\gamma K} + (6A + 3\ell + {}^{12BC}/_{\rho})\cdot\frac{\|z^0 - z^{*,0}\|^2}{K} \\ & + \left(6 + (6A + 3\ell + {}^{12BC}/_{\rho})\,\gamma\right)\frac{\gamma B\sigma_0^2}{\rho K} \\ & + \gamma(3 + \gamma\,(6A + 3\ell + {}^{12BC}/_{\rho}))(D_1 + {}^{2BD_2}/_{\rho}).\end{aligned}$$

□

A.2.7 Proof of Theorems 4.3.2, 4.3.3

Our further analysis relies heavily on work conducted in [194], but it seems to us that there is a typo in the main part. Furthermore, as we will deal with nonreducible "noises", we will need to adjust proofs by a little. For convenience of the proof, we split the result into three following lemmas:

Lemma A.2.23. Consider two non-negative sequences $\{r_k\}$, $\{\eta_k\}$, that satisfy

$$r_{k+1} \leq (1-\eta_k)r_k + a\eta_k^2 + b, \tag{A.42}$$

for all $k \geq 0$, constants $a \geq 0$, $b \geq 0$ and for positive stepsizes $\{\eta_k\}$ with $\eta_k \leq \frac{1}{d}$, $d > 1$. Then there exists η, such that $\forall k \geq 0$, $\eta_k \equiv \eta$

$$r_K < r_0 \exp\left[-\frac{K}{d}\right] + \frac{a}{d} + bK. \tag{A.43}$$

Proof. Set $\eta_k \equiv \eta$, $\forall k \geq 0$. Unroll inequalities:

$$r_K \leq (1-\eta)r_{K-1} + a\eta^2 + b \leq (1-\eta)^K r_0 + a\eta^2 \sum_{i=0}^{K-1}(1-\eta)^i + bK$$
$$\leq (1-\eta)^T + a\eta^2 \sum_{k=0}^{\infty}(1-\eta)^k + bK \leq (1-\eta)^K r_0 + a\eta + bK.$$

Put $\eta = \frac{1}{d}$ and use, that for $\forall x > 0,\ 0 < a < 1 \to (1-a)^x < \exp(-ax)$. Hence,

$$r_K < r_0 \exp\left[-\frac{K}{d}\right] + \frac{a}{d} + bK.$$

□

Lemma A.2.24. Let the non-negative sequence $\{r_k\}$ satisfy the conditions of Lemma A.2.23. Then there are decreasing stepsizes $\eta_k = \frac{2}{2d+k}$, such that

$$r_{K+1} \leq \frac{4d^2 r_0}{(2d+K)^2} + 4a\frac{K+1}{(2d+K)^2} + \frac{b}{6}\frac{(K+1)(24d^2+12dK+2K^2+K)}{(2d+K)^2}.$$

Proof. Rearrange (A.42):

$$0 \leq (1-\eta_k)r_k - r_{k+1} + a\eta_k^2 + b.$$

Divide both parts by η_k:

$$0 \leq \frac{1-\eta_k}{\eta_k}r_k - \frac{r_{k+1}}{\eta_k} + a\eta_k + \frac{b}{\eta_k} = (2d+k-2)\frac{r_k}{2} - (2d+k)\frac{r_{k+1}}{2} + a\frac{2}{2d+k} + b\frac{2d+k}{2}.$$

Multiply both parts by $2 \cdot (2d+k)$ and use, that $x \cdot (x-2) \leq (x-1)^2$:

$$0 \le (2d+k-2)(2d+k)r_k - (2d+k)^2 r_{k+1} + 4a + b(2d+k)^2$$
$$\le (2d+k-1)^2 r_k - (2d+k)^2 r_{k+1} + 4a + b(2d+k)^2.$$

We obtain a telescoping sum, hence:

$$0 \le (2d-1)^2 r_0 - (2d+k)^2 r_{K+1} + 4a(K+1) + b\sum_{i=0}^{K}(2d+i)^2$$
$$\le 4d^2 r_0 - (2d+k)^2 r_{K+1} + 4a(K+1) + \frac{b(K+1)(24d^2+12dK+2K^2+K)}{6}.$$

Finally, we can gain the original inequality by rearranging the terms. □

Lemma A.2.25. Let $\{r_k\}$ satisfy (A.42). Then there exist stepsizes η_k,

$$\begin{aligned} &\text{if} \quad K \le d, && \eta_k = \tfrac{1}{d}, \\ &\text{if} \quad K > d \ \text{ and } \ k < k_0, && \eta_k = \tfrac{1}{d}, \\ &\text{if} \quad K > d \ \text{ and } \ k \ge k_0, && \eta_k = \tfrac{2}{2d+k-k_0}, \end{aligned}$$

where $k_0 = \lfloor \frac{K}{2} \rfloor$, such that

$$r_{K+1} = \mathcal{O}\left(r_0 \exp\left(-\frac{K}{2d}\right) + \frac{a}{d+K} + bK\right). \tag{A.44}$$

Proof. If $K \le d$ we'll take $\eta_k \equiv \frac{1}{d}$, hence using Lemma A.2.23 we obtain

$$r_K \le r_0 \exp\left[-\frac{K}{d}\right] + \frac{a}{d} + bK \le r_0 \exp\left[-\frac{K}{d}\right] + \frac{a}{K} + bK.$$

As $\frac{a}{K} / \frac{a}{d+K} = \frac{d+K}{K} \le \frac{2d}{K} \le 2d$ and $\exp\left(-\frac{K}{d}\right) / \exp\left(-\frac{K}{2d}\right) = \exp\left(-\frac{K}{2d}\right) < 1$, thus (A.44) is correct for $K \le d$.

If $K > d$ for $k \le k_0 := \lceil \frac{K}{2} \rceil$ we'll take $\eta_k \equiv \frac{1}{d}$ and for $k > k_0$: $\eta_k = \frac{2}{2d+k-k_0}$. Therefore, according to (A.43) we obtain

$$r_{k_0} \le r_0 \exp\left(-\frac{k_0}{d}\right) + \frac{a}{d} + bk_0.$$

Using Lemma A.2.24 with $k_1 = K - k_0$ we get

$$r_{K+1} \le \frac{4d^2 r_{k_0}}{(2d+k_1)^2} + 4a\frac{K+1}{(2d+k_1)^2} + \frac{b}{6}\frac{(k_1+1)(24d^2+12dk_1+2k_1^2+k_1)}{(2d+k_1)^2}.$$

Combining inequalities we derive

$$\frac{4d^2 r_{k_0}}{(2d+k_1)^2} \le \frac{4d^2}{(2d+k_1)^2}\left(r_0 \exp\left(-\frac{k_0}{d}\right) + \frac{a}{d} + bk_0\right) \le r_0 \exp\left(-\frac{k_0}{d}\right) + \frac{4ad}{(2d+k_1)^2} + \frac{4bk_0 d^2}{(2d+k_1)^2}.$$

Since $\frac{K}{2} \leq \lceil \frac{K}{2} \rceil < K$ and $\frac{K}{4} < \lfloor \frac{K}{2} \rfloor < K$ we can replace all k_0, k_1 with K in fractions in O-notation. We gain the replacement in the exponent, as $-\lceil \frac{K}{2} \rceil \leq -\frac{K}{2}$. Thus,

$$\begin{aligned} r_{K+1} &= \mathcal{O}\left(r_0 \exp\left(-\frac{K}{2d}\right) + \frac{ad}{(d+K)^2} + \frac{d^2bK}{(d+K)^2} + \frac{aK}{(d+K)^2} + bK\frac{d^2+K^2}{(d+K)^2}\right) \\ &= \mathcal{O}\left(r_0 \exp\left(-\frac{K}{2d}\right) + \frac{a}{K+d} + bK\frac{d^2+K^2}{(d+K)^2}\right), \end{aligned}$$

where second equality is implied by $d, K > 0$, $d^2 + K^2 \leq (d+K)^2$. □

Lemma A.2.26. If x^k is upgraded due to (4.17), then for all $\alpha > 0$:

$$\begin{aligned} \mathbb{E}[f(x^{k+1}) - f(x^*)] \leq &(1-\eta_k)\mathbb{E}[f(x^k) - f(x^*)] + \frac{\alpha}{L}\mathbb{E}[\|\nabla f(x^k) - g^k\|^2] \\ &+ \frac{L\eta_k^2}{\alpha}\mathbb{E}[\|s^k - x^*\|^2] + \frac{L\eta_k^2}{2}\mathbb{E}[\|s^k - x^k\|^2]. \end{aligned}$$

Proof. First, we assume that $f(x)$ satisfies Assumptions 4.0.1 and 4.0.2:

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2.$$

With update of x^{k+1}, according to Algorithm (4.17):

$$\begin{aligned} f(x^{k+1}) - f(x^*) \leq &f(x^k) - f(x^*) + \eta_k\langle \nabla f(x^k), s^k - x^k \rangle + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2 \\ = &f(x^k) - f(x^*) + \eta_k\langle g^k, s^k - x^k \rangle + \eta_k\langle \nabla f(x^k) - g^k, s^k - x^k \rangle \\ &+ \frac{L\eta_k^2}{2}\|s^k - x^k\|^2. \end{aligned}$$

The optimal choice of s^k in (4.17) gives $\langle g^k, s^k - x^k \rangle \leq \langle g^k, x^* - x^k \rangle$. Then

$$\begin{aligned} f(x^{+1}) - f(x^*) \leq &f(x^k) - f(x^*) + \eta_k\langle g^k, x^* - x^k \rangle + \eta_k\langle \nabla f(x^k) - g^k, s^k - x^k \rangle \\ &+ \frac{L\eta_k^2}{2}\|s^k - x^k\|^2 \\ = &f(x^k) - f(x^*) + \eta_k\langle \nabla f(x^k), x^* - x^k \rangle + \eta_k\langle g^k - \nabla f(x^k), x^* - x^k \rangle \\ &+ \eta_k\langle \nabla f(x^k) - g^k, s^k - x^k \rangle + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2 \\ = &f(x^k) - f(x^*) + \eta_k\langle \nabla f(x^k), x^* - x^k \rangle + \eta_k\langle \nabla f(x^k) - g^k, s^k - x^* \rangle \\ &+ \frac{L\eta_k^2}{2}\|s^k - x^k\|^2. \end{aligned}$$

Using Young's inequality here we state for any positive α:

$$f(x^{k+1}) - f(x^*) \leq f(x^k) - f(x^*) + \eta_k\langle \nabla f(x^k), x^* - x^k \rangle + \frac{\alpha}{L}\|\nabla f(x^k) - g^k\|^2$$

$$+\frac{L\eta_k^2}{\alpha}\|s^k-x^*\|^2+\frac{L\eta_k^2}{2}\|s^k-x^k\|^2$$

Since f satisfies Assumption 4.0.2 we have $\langle\nabla f(x^k),x^*-x^k\rangle\leq-(f(x^k)-f(x^*))$. Thus,

$$\begin{aligned}f(x^{k+1})-f(x^*)\leq&f(x^k)-f(x^*)-\eta_k(f(x^k)-f(x^*))+\frac{\alpha}{L}\|\nabla f(x^k)-g^k\|^2\\&+\frac{L\eta_k^2}{\alpha}\|s^k-x^*\|^2+\frac{L\eta_k^2}{2}\|s^k-x^k\|^2\\=&(1-\eta_k)(f(x^k)-f(x^*))+\frac{\alpha}{L}\|\nabla f(x^k)-g^k\|^2\\&+\frac{L\eta_k^2}{\alpha}\|s^k-x^*\|^2+\frac{L\eta_k^2}{2}\|s^k-x^k\|^2.\end{aligned}$$

Taking the full mathematical expectation finishes the proof. □

Theorem A.2.27. (Theorem 4.3.2) Let Assumptions 4.0.1, 4.0.2 and 4.3.1 be satisfied. Then there exist $\eta_k\leq\min(\rho_1,\rho_2)$ for (4.17) and constants M_1, M_2, α such that:

$$r_{K+1}=\mathcal{O}\Bigg(r_0\exp\left(-\frac{K\min(\rho_1,\rho_2)}{4}\right)+\frac{LD^2}{K+\frac{1}{\min(\rho_1,\rho_2)}}+D^2\sqrt{\frac{B\rho_2+AE}{\rho_1\rho_2\left(K+\frac{1}{\min(\rho_1,\rho_2)}\right)^2}+\frac{K}{K+\frac{1}{\min(\rho_1,\rho_2)}}\frac{C\rho_2}{\rho_1\rho_2D^2}}\Bigg),$$

where $r_k=\mathbb{E}_k[f(x^k)-f(x_*)+M_1\|g^k-\nabla f(x^k)\|^2+M_2\sigma_k^2]$.

Proof. Since Assumption 4.3.1 holds, we can derive the following inequality with Lemma A.2.26:

$$\begin{aligned}r_{k+1}=&\mathbb{E}[f(x^{k+1})-f(x^*)+M_1\|\nabla f(x^{k+1})-g^{k+1}\|^2+M_2\sigma_{k+1}^2]\\\leq&(1-\eta_k)\mathbb{E}[f(x^k)-f(x^*)]\\&+\frac{\alpha}{L}\mathbb{E}[\|\nabla f(x^k)-g^k\|^2]+\left(\frac{L}{\alpha}+\frac{L}{2}\right)D^2\eta_k^2+M_1(1-\rho_1)\|g^k-\nabla f(x^k)\|^2\\&+M_1A\sigma_k^2+M_1\eta_k^2BD^2+M_1C+M_2(1-\rho_2)\sigma_k^2+M_2\eta_k^2ED^2\\=&(1-\eta_k)\mathbb{E}[f(x^k)-f(x^*)]+\left(\frac{\alpha}{L}+M_1(1-\rho_1)\right)\mathbb{E}\left[\|\nabla f(x^k)-g^k\|^2\right]\\&+\left(M_1A+M_2(1-\rho_2)\right)\sigma_k^2\\&+\left(\frac{L}{\alpha}+\frac{L}{2}+M_1B+M_2E\right)\eta_k^2D^2+M_1C.\end{aligned}$$

With constants $M_1=\frac{2\alpha}{\rho_1L}$, $M_2=\frac{2M_1A}{\rho_2}$ we get

$$\mathbb{E}[f(x^{k+1})-f(x^*)]+M_1\mathbb{E}[\|\nabla f(x^{k+1})-g^{k+1}\|^2]+M_2\sigma_{k+1}^2$$

$$\leq (1-\eta_k)\mathbb{E}[f(x^k)-f(x^*)] + \left(1-\frac{\rho_1}{2}\right) M_1 \mathbb{E}[\|\nabla f(x^k) - g^k\|^2]$$
$$+ \left(1-\frac{\rho_2}{2}\right) M_2 \sigma_k^2 + \left(\frac{L}{2} + \frac{L}{\alpha} + \frac{2B}{\rho_1 L}\alpha + \frac{4AE}{\rho_1\rho_2 L}\alpha\right)\eta_k^2 D^2 + \frac{2C}{\rho_1 L}\alpha.$$

Using Lemma A.2.25 with $d = \frac{2}{\min(\rho_1,\rho_2)}$ and $r_k = \mathbb{E}_k[f(x^k) - f(x_*) + M_1\|\nabla f(x^k) - g^k\|^2 + M_2\sigma_k^2]$ we obtain

$$r_{K+1} = \mathcal{O}\left(r_0 \exp\left(-\frac{K}{2d}\right) + \frac{LD^2}{2(K+d)} + \frac{\frac{L}{\alpha} + \alpha\frac{2B\rho_2+4AE}{\rho_1\rho_2 L}}{K+d} D^2 + K\alpha\frac{2C\rho_2}{\rho_1\rho_2 L}\right).$$

This estimation holds for any $\alpha > 0$, and thus to obtain optimal estimation on the K-th iteration we shall minimize this to α. It is easy to see, that minimum of $x\alpha + \frac{y}{\alpha}$ is located at $\alpha = \sqrt{\frac{y}{x}}$ and equals to $2\sqrt{xy}$. Therefore taking optimal α as

$$\alpha = \sqrt{\frac{\frac{LD^2}{K+d}}{\frac{2B\rho_2+4AE}{\rho_1\rho_2 L(K+d)}D^2 + K\frac{2C\rho_2}{\rho_1\rho_2 L}}},$$

we get

$$r_{K+1} = \mathcal{O}\left(r_0 \exp\left(-\frac{K}{2d}\right) + \frac{LD^2}{K+d} + D^2\sqrt{\frac{2B\rho_2+4AE}{\rho_1\rho_2(K+d)^2} + \frac{K}{K+d}\frac{2C\rho_2}{\rho_1\rho_2 D^2}}\right).$$

It completes the proof. □

Theorem A.2.28. (Theorem 4.3.3) Let the Assumptions 4.0.1 and 4.3.1 be satisfied. Then, there exist η_k for (4.17), that

$$\mathbb{E}\left[\min_{0\leq k\leq K-1} \mathbf{gap}(x^k)\right] = \mathcal{O}\left(\frac{r_0}{\sqrt{K}} + \frac{D^2}{\sqrt{K}}\left[L + \sqrt{\frac{B\rho_2+AE}{\rho_1\rho_2} + K\frac{C\rho_2}{D^2\rho_1\rho_2}}\right]\right).$$

Proof. With Assumption 4.0.1 we have:

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k\rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2.$$

With update of x^{k+1}, according to (4.17):

$$f(x^{k+1}) - f(x^*) \leq f(x^k) - f(x^*) + \eta_k\langle \nabla f(x^k), s^k - x^k\rangle + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2$$
$$= f(x^k) - f(x^*) + \eta_k\langle g^k, s^k - x^k\rangle + \eta_k\langle \nabla f(x^k) - g^k, s^k - x^k\rangle + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2.$$

The optimal choice of s^k in (4.17) gives $\langle g^k, s^k - x^k\rangle \leq \langle g^k, x - x^k\rangle$ for all $x \in \mathcal{X}$. Then,

$$f(x^{+1}) - f(x^*) \leq f(x^k) - f(x^*) + \eta_k\langle g^k, x - x^k\rangle + \eta_k\langle \nabla f(x^k) - g^k, s^k - x^k\rangle$$

$$
\begin{aligned}
&+ \frac{L\eta_k^2}{2}\|s^k - x^k\|^2 \\
=&f(x^k) - f(x^*) + \eta_k\langle \nabla f(x^k), x - x^k\rangle + \eta_k\langle g^k - \nabla f(x^k), x - x^k\rangle \\
&+ \eta_k\langle \nabla f(x^k) - g^k, s^k - x^k\rangle + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2 \\
=&f(x^k) - f(x^*) + \eta_k\langle \nabla f(x^k), x - x^k\rangle + \eta_k\langle \nabla f(x^k) - g^k, s^k - x\rangle \\
&+ \frac{L\eta_k^2}{2}\|s^k - x^k\|^2.
\end{aligned}
$$

For any positive α we have:

$$
\begin{aligned}
f(x^{k+1}) - f(x^*) \leq& f(x^k) - f(x^*) + \eta_k\langle \nabla f(x^k), x - x^k\rangle + \frac{\alpha}{L}\|\nabla f(x^k) - g^k\|^2 \\
&+ \frac{L\eta_k^2}{\alpha}\|s^k - x\|^2 + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2.
\end{aligned}
$$

After small rearrangements one can get:

$$
\begin{aligned}
\eta_k\langle \nabla f(x^k), x^k - x\rangle \leq& f(x^k) - f(x^*) - \big(f(x^{k+1}) - f(x^*)\big) + \frac{\alpha}{L}\|\nabla f(x^k) - g^k\|^2 \\
&+ \frac{L\eta_k^2}{\alpha}\|s^k - x\|^2 + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2.
\end{aligned}
$$

Maximizing over $\mathcal{X}$, taking the full mathematical expectation and bounding the distances by diameter, we get:

$$
\begin{aligned}
\eta_k\mathbb{E}\Big[\max_{x\in\mathcal{X}}\langle \nabla f(x^k), x^k - x\rangle\Big] \leq& \mathbb{E}\Big[f(x^k) - f(x^*)\Big] - \mathbb{E}\Big[f(x^{k+1}) - f(x^*))\Big] \\
&+ \frac{\alpha}{L}\mathbb{E}\Big[\|\nabla f(x^k) - g^k\|^2\Big] + \eta_k^2\frac{LD^2}{\alpha} + \eta_k^2\frac{LD^2}{2}.
\end{aligned}
$$

After multiplying Assumption (4.3.1) by the positive constants M_1, M_2 (which we will define below) and summarizing with previous inequality we have:

$$
\begin{aligned}
\eta_k\mathbb{E}\Big[\max_{x\in\mathcal{X}}\langle \nabla f(x^k), x^k - x\rangle\Big] \leq& \mathbb{E}\Big[f(x^k) - f(x^*) + \left(1 - \rho_1 + \frac{\alpha}{M_1 L}\right) M_1\|\nabla f(x^k) - g^k\|^2 \\
&+ \left(1 - \rho_2 + \frac{M_1 A}{M_2}\right) M_2\sigma_k^2\Big] \\
&- \mathbb{E}\Big[f(x^{k+1}) - f(x^*)) + M_1\|\nabla f(x^{k+1}) - g^k\|^2 + M_2\sigma_{k+1}^2\Big] \\
&+ D^2\eta_k^2\left(\frac{L}{2} + \frac{L}{\alpha} + M_1 B + M_2 E\right) + M_1 C.
\end{aligned}
$$

With $M_1 = \frac{\alpha}{L\rho_1}, M_2 = \frac{M_1 A}{\rho_2}$ and summarizing over all k from 0 to $K-1$ we have:

$$
\sum_{k=0}^{K-1}\eta_k\mathbb{E}\Big[\max_{x\in\mathcal{X}}\langle \nabla f(x^k), x^k - x\rangle\Big] \leq f(x^0) - f(x^*) + \|\nabla f(x^0) - g^0\|^2 + \|\sigma_0\|^2
$$

$$+ D^2 \left(\frac{L}{2} + \frac{L}{\alpha} + \alpha \left(\frac{B}{\rho_1 L} + \frac{AE}{\rho_1 \rho_2 L} \right) \right) \sum_{k=0}^{K-1} \eta_k^2 + K M_1 C.$$

Assuming $\min_{k \leq K} \eta_k = \eta_{min}$, we gain:

$$\begin{aligned} \sum_{k=0}^{K-1} \eta_k \mathbb{E}\Big[\max_{x \in \mathcal{X}} \langle \nabla f(x^k), x^k - x \rangle \Big] \leq & f(x^0) - f(x^*) + \|\nabla f(x^0) - g^0\|^2 + \|\sigma_0\|^2 \\ & + D^2 \left(\frac{L}{2} + \frac{L}{\alpha} + \alpha \left(\frac{B}{\rho_1 L} + \frac{AE}{\rho_1 \rho_2 L} + \frac{C}{\rho_1 L \eta_{min}^2 D^2} \right) \right) \sum_{k=0}^{K-1} \eta_k^2. \end{aligned}$$

With $\alpha = \sqrt{\frac{LD^2}{\frac{1}{\rho_1 L}(BD^2 + \frac{C}{\eta_{min}^2}) + \frac{A}{\rho_1 \rho_2 L} E D^2}}$ one can obtain:

$$\begin{aligned} \sum_{k=0}^{K-1} \eta_k \mathbb{E}\Big[\max_{x \in \mathcal{X}} \langle \nabla f(x^k), x^k - x \rangle \Big] \leq & f(x^0) - f(x^*) + \|\nabla f(x^0) - g^0\|^2 \\ & + D^2 \left(\frac{L}{2} + 2 \sqrt{\frac{B\rho_2 + AE}{\rho_1 \rho_2} + \frac{1}{\eta_{min}^2} \frac{C \rho_2}{D^2 \rho_1 \rho_2}} \right) \sum_{k=0}^{K-1} \eta_k^2. \end{aligned}$$

If we take $\eta_k = \frac{1}{\sqrt{K}}$, then $\eta_{min} = \frac{1}{\sqrt{K}}$. Divide both sides by $\sqrt{K}$, then:

$$\begin{aligned} \mathbb{E}\Big[\frac{1}{K} \max_{x \in \mathcal{X}} \langle \nabla f(x^k), x^k - x \rangle \Big] \leq & \frac{f(x^0) - f(x^*) + \|\nabla f(x^0) - g^0\|^2 + \|\sigma_0\|^2}{\sqrt{K}} \\ & + \frac{D^2}{\sqrt{K}} \Big(\frac{L}{2} + 2 \sqrt{\frac{B\rho_2 + AE}{\rho_1 \rho_2} + K \frac{C \rho_2}{D^2 \rho_1 \rho_2}} \Big). \end{aligned}$$

Finally, we obtain the needed estimation. □

A.2.8 Proof of Theorem 5.3.7

We preface the proof by two technical Lemmas.

Lemma A.2.29. Assume Assumption 5.3.1 and Assumption 5.3.2. Then for the iterates of Algorithm 51 with $\theta = (p\eta^{-1}-1)/(\beta p\eta^{-1}-1)$, $\theta > 0$, $\eta \geq 1$, $p > 0$, it holds that

$$\begin{aligned}\mathbb{E}_k[\|x^{k+1}-x^*\|^2] \leq &(1+\alpha p\gamma\eta)(1-\beta)\|x^k-x^*\|^2 + (1+\alpha p\gamma\eta)\beta\|x_g^k-x^*\|^2 \\ &+ (1+\alpha p\gamma\eta)(\beta^2-\beta)\|x^k-x_g^k\|^2 + p^2\eta^2\gamma^2\mathbb{E}_k[\|g^k\|^2] \\ &- 2\eta^2\gamma\langle\nabla f(x_g^k), x_g^k + (p\eta^{-1}-1)x_f^k - \eta^{-1}px^*\rangle \\ &+ \frac{p\eta\gamma}{\alpha}\|\mathbb{E}_k[g^k]-\nabla f(x_g^k)\|^2, \end{aligned} \tag{A.45}$$

where $\alpha > 0$ is any positive constant.

Proof. We start with lines 6 and 5 of Algorithm 51:

$$\begin{aligned}\|x^{k+1}-x^*\|^2 =&\|\eta x_f^{k+1} + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\|^2 \\ =&\|\eta x_g^k - p\eta\gamma g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\|^2 \\ =&\|\eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\|^2 + p^2\gamma^2\eta^2\|g^k\|^2 \\ &- 2p\gamma\eta\langle g^k, \eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\rangle.\end{aligned}$$

Using straightforward algebra, we get

$$\begin{aligned}\|x^{k+1}-x^*\|^2 =&\|\eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\|^2 + p^2\gamma^2\eta^2\|g^k\|^2 \\ &- 2p\gamma\eta\langle\nabla f(x_g^k), \eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\rangle \\ &- 2p\gamma\eta\langle\mathbb{E}_k[g^k]-\nabla f(x_g^k), \eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\rangle \\ &- 2p\gamma\eta\langle g^k-\mathbb{E}_k[g^k], \eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\rangle \\ \leq&(1+\alpha p\eta\gamma)\|\eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\|^2 \\ &- 2p\gamma\eta\langle\nabla f(x_g^k), \eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\rangle \\ &- 2p\gamma\eta\langle g^k-\mathbb{E}_k[g^k], \eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\rangle \\ &+ p^2\gamma^2\eta^2\|g^k\|^2 + \frac{p\gamma\eta}{\alpha}\|\mathbb{E}_k[g^k]-\nabla f(x_g^k)\|^2.\end{aligned}$$

In the last step we also applied Cauchy-Schwartz inequality in the form (A.1) with $\alpha > 0$. Taking the conditional expectation, we get

$$\begin{aligned}\mathbb{E}_k[\|x^{k+1}-x^*\|^2] \leq&(1+\alpha p\eta\gamma)\|\eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\|^2 \\ &- 2p\gamma\eta\langle\nabla f(x_g^k), \eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\rangle \\ &+ p^2\gamma^2\eta^2\mathbb{E}_k[\|g^k\|^2] + \frac{p\gamma\eta}{\alpha}\|\mathbb{E}_k[g^k]-\nabla f(x_g^k)\|^2.\end{aligned} \tag{A.46}$$

Now let us handle expression $\|\eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\|^2$ for a while.

Taking into account line 2 and the choice of θ such that $\theta = (p\eta^{-1} - 1)/(\beta p\eta^{-1} - 1)$ (in particular, $(p\eta^{-1} - 1) = (\beta p\eta^{-1} - 1)\theta$ and $\eta(1 - \beta p\eta^{-1})(1 - \theta) = p(1 - \beta)$), we get

$$\begin{aligned}
&\eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k \\
&\qquad = (\eta + (1-p)\beta)x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k \\
&\qquad = (\eta + (1-p)\beta)x_g^k + \eta(p\eta^{-1} - 1)x_f^k + (1-p)(1-\beta)x^k \\
&\qquad = (\eta + (1-p)\beta)x_g^k + \eta(\beta p\eta^{-1} - 1)\theta x_f^k + (1-p)(1-\beta)x^k \\
&\qquad = (\eta + (1-p)\beta)x_g^k + \eta(\beta p\eta^{-1} - 1)(x_g^k - (1-\theta)x^k) + (1-p)(1-\beta)x^k \\
&\qquad = \beta x_g^k - \eta(\beta p\eta^{-1} - 1)(1-\theta)x^k + (1-p)(1-\beta)x^k \\
&\qquad = \beta x_g^k + p(1-\beta)x^k + (1-p)(1-\beta)x^k \\
&\qquad = \beta x_g^k + (1-\beta)x^k .
\end{aligned}$$

Substituting into $\|\eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\|^2$, we get

$$\begin{aligned}
&\|\eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^*\|^2 \\
&\qquad = \|\beta x_g^k + (1-\beta)x^k - x^*\|^2 \\
&\qquad = \|x^k - x^* + \beta(x_g^k - x^k)\|^2 \\
&\qquad = \|x^k - x^*\|^2 + 2\beta\langle x^k - x^*, x_g^k - x^k\rangle + \beta^2\|x^k - x_g^k\|^2 \\
&\qquad = \|x^k - x^*\|^2 + \beta\left(\|x_g^k - x^*\|^2 - \|x^k - x^*\|^2 - \|x_g^k - x^k\|^2\right) + \beta^2\|x^k - x_g^k\|^2 \\
&\qquad = (1-\beta)\|x^k - x^*\|^2 + \beta\|x_g^k - x^*\|^2 + (\beta^2 - \beta)\|x^k - x_g^k\|^2. \qquad \text{(A.47)}
\end{aligned}$$

Again with line 2 and the choice of θ such that $\theta = (p\eta^{-1} - 1)/(\beta p\eta^{-1} - 1)$ (in particular, $\eta^{-1}p(1-\beta) = (1 - \beta p\eta^{-1})(1-\theta)$ and $(\beta p\eta^{-1} - 1)\theta = (p\eta^{-1} - 1)$), one can also note

$$\begin{aligned}
&\eta x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k - x^* \\
&\quad = (\eta + (1-p)\beta)x_g^k + (p-\eta)x_f^k + (1-p)(1-\beta)x^k - x^* \\
&\quad = \eta p^{-1}\left((p + (1-p)\eta^{-1}p\beta)x_g^k + (p\eta^{-1} - 1)px_f^k + (1-p)(1-\beta)p\eta^{-1}x^k - \eta^{-1}px^*\right) \\
&\quad = \eta p^{-1}\left((p + (1-p)\eta^{-1}p\beta)x_g^k + (p\eta^{-1} - 1)px_f^k + (1-p)(1-\beta p\eta^{-1})(1-\theta)x^k - \eta^{-1}px^*\right) \\
&\quad = \eta p^{-1}\left((p + (1-p)\eta^{-1}p\beta)x_g^k + (p\eta^{-1} - 1)px_f^k + (1-p)(1-\beta p\eta^{-1})(x_g^k - \theta x_f^k) - \eta^{-1}px^*\right) \\
&\quad = \eta p^{-1}\left(x_g^k + (p\eta^{-1} - 1)px_f^k - (1-p)(1-\beta p\eta^{-1})\theta x_f^k - \eta^{-1}px^*\right) \\
&\quad = \eta p^{-1}\left(x_g^k + (p\eta^{-1} - 1)px_f^k + (1-p)(p\eta^{-1} - 1)x_f^k - \eta^{-1}px^*\right) \\
&\quad = \eta p^{-1}\left(x_g^k + (p\eta^{-1} - 1)x_f^k - \eta^{-1}px^*\right). \qquad \text{(A.48)}
\end{aligned}$$

Combining (A.47) and (A.48) with (A.46), we finish the proof. □

Lemma A.2.30. Assume Assumption 5.3.1-Assumption 5.3.2. Let problem (5.23) be solved by

Algorithm 51. Then for any $u \in \mathbb{R}^d$, we get

$$\begin{aligned}\mathbb{E}_k[f(x_f^{k+1})] \leq & f(u) - \langle \nabla f(x_g^k), u - x_g^k \rangle - \frac{\mu}{2}\|u - x_g^k\|^2 - \frac{\gamma}{2}\|\nabla f(x_g^k)\|^2 \\ & + \frac{\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 + \frac{L\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2].\end{aligned}$$

Proof. Using Assumption 5.3.1 with $x = x_f^{k+1}$, $y = x_g^k$ and line 5 of Algorithm 51, we get

$$\begin{aligned}f(x_f^{k+1}) \leq & f(x_g^k) + \langle \nabla f(x_g^k), x_f^{k+1} - x_g^k \rangle + \frac{L}{2}\|x_f^{k+1} - x_g^k\|^2 \\ = & f(x_g^k) - p\gamma\langle \nabla f(x_g^k), g^k \rangle + \frac{Lp^2\gamma^2}{2}\|g^k\|^2 \\ = & f(x_g^k) - p\gamma\langle \nabla f(x_g^k), \nabla f(x_g^k) \rangle - p\gamma\langle \nabla f(x_g^k), \mathbb{E}_k[g^k] - \nabla f(x_g^k) \rangle \\ & - p\gamma\langle \nabla f(x_g^k), g^k - \mathbb{E}_k[g^k] \rangle + \frac{Lp^2\gamma^2}{2}\|g^k\|^2 \\ \leq & f(x_g^k) - p\gamma\|\nabla f(x_g^k)\|^2 + \frac{p\gamma}{2}\|\nabla f(x_g^k)\|^2 + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 \\ & - p\gamma\langle \nabla f(x_g^k), g^k - \mathbb{E}_k[g^k] \rangle + \frac{Lp^2\gamma^2}{2}\|g^k\|^2.\end{aligned}$$

Here we also used Cauchy Schwartz inequality (A.1) with $a = \nabla f(x_g^k)$, $b = \nabla f(x_g^k) - \mathbb{E}_k[g^k]$ and $c = 1$. Taking the conditional expectation, we get

$$\mathbb{E}_k[f(x_f^{k+1})] \leq f(x_g^k) - \frac{p\gamma}{2}\|\nabla f(x_g^k)\|^2 + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 + \frac{Lp^2\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2].$$

Using Assumption 5.3.2 with $x = u$ and $y = x_g^k$, one can conclude that for any $u \in \mathbb{R}^d$ it holds

$$\begin{aligned}\mathbb{E}_k[f(x_f^{k+1})] \leq & f(u) - \langle \nabla f(x_g^k), u - x_g^k \rangle - \frac{\mu}{2}\|u - x_g^k\|^2 - \frac{p\gamma}{2}\|\nabla f(x_g^k)\|^2 \\ & + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 + \frac{Lp^2\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2].\end{aligned}$$

□

Theorem A.2.31 (Theorem 5.3.7)**.** Assume Assumption 5.3.1 – Assumption 5.3.4. Let problem (5.23) be solved by Algorithm 51. Then for any $b \in \mathbb{N}$, $\gamma \in (0; \frac{3}{4L}]$, and $\beta, \theta, \eta, p, M, B$ satisfying

$$\begin{gathered}p = \left[1 + 2\left(1 + \gamma L\right)\left(1 + 4\left[C_1\tau b^{-1} + (C_1 + 1)\tau^2 b^{-2}\right]\delta^2\right)\right]^{-1}, \\ \beta = \sqrt{\tfrac{4p^2\mu\gamma}{3}}, \quad \eta = \tfrac{3\beta}{2p\mu\gamma} = \sqrt{\tfrac{3}{\mu\gamma}}, \quad \theta = \tfrac{p\eta^{-1}-1}{\beta p\eta^{-1}-1}, \\ M = \max\{2; \sqrt{C_2 p^{-1}(1 + 2p/\beta)}\}, \quad B = \lceil b\log_2 M\rceil.\end{gathered}$$

it holds that

$$\mathbb{E}\left[\|x^N - x^*\|^2 + \frac{6}{\mu}(f(x_f^N) - f(x^*))\right]$$

$$= \mathcal{O}\left(\exp\left(-N\sqrt{\frac{p^2\mu\gamma}{3}}\right)\left[\|x^0 - x^*\|^2 + \frac{6}{\mu}(f(x^0) - f(x^*))\right] + \frac{p\sqrt{\gamma}}{\mu^{3/2}}\left(\sigma^2\tau b^{-1} + \sigma^2\tau^2 b^{-2}\right)\right).$$

Proof. Using Lemma A.2.30 with $u = x^*$ and $u = x_f^k$, we get

$$\begin{aligned}\mathbb{E}_k[f(x_f^{k+1})] \leq & f(x^*) - \langle \nabla f(x_g^k), x^* - x_g^k\rangle - \frac{\mu}{2}\|x^* - x_g^k\|^2 - \frac{p\gamma}{2}\|\nabla f(x_g^k)\|^2 \\ & + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 + \frac{Lp^2\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2],\end{aligned}$$

$$\begin{aligned}\mathbb{E}_k[f(x_f^{k+1})] \leq & f(x_f^k) - \langle \nabla f(x_g^k), x_f^k - x_g^k\rangle - \frac{\mu}{2}\|x_f^k - x_g^k\|^2 - \frac{p\gamma}{2}\|\nabla f(x_g^k)\|^2 \\ & + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 + \frac{Lp^2\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2].\end{aligned}$$

Summing the first inequality with coefficient $2p\gamma\eta$, the second with coefficient $2\gamma\eta(\eta - p)$ and (A.45), we obtain

$$\begin{aligned}&\mathbb{E}_k[\|x^{k+1} - x^*\|^2 + 2\gamma\eta^2 f(x_f^{k+1})] \\ &\quad \leq (1 + \alpha p\gamma\eta)(1 - \beta)\|x^k - x^*\|^2 + (1 + \alpha p\gamma\eta)\beta\|x_g^k - x^*\|^2 \\ &\qquad + (1 + \alpha p\gamma\eta)(\beta^2 - \beta)\|x^k - x_g^k\|^2 - 2\eta^2\gamma\langle \nabla f(x_g^k), x_g^k + (p\eta^{-1} - 1)x_f^k - \eta^{-1}px^*\rangle \\ &\qquad + p^2\eta^2\gamma^2\mathbb{E}_k[\|g^k\|^2] + \frac{p\eta\gamma}{\alpha}\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 \\ &\qquad + 2p\gamma\eta\Big(f(x^*) - \langle \nabla f(x_g^k), x^* - x_g^k\rangle - \frac{\mu}{2}\|x^* - x_g^k\|^2 - \frac{p\gamma}{2}\|\nabla f(x_g^k)\|^2 \\ &\qquad + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 + \frac{Lp^2\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2]\Big) \\ &\qquad + 2\gamma\eta(\eta - p)\Big(f(x_f^k) - \langle \nabla f(x_g^k), x_f^k - x_g^k\rangle - \frac{\mu}{2}\|x_f^k - x_g^k\|^2 - \frac{p\gamma}{2}\|\nabla f(x_g^k)\|^2 \\ &\qquad + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 + \frac{Lp^2\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2]\Big) \\ &\quad = (1 + \alpha p\gamma\eta)(1 - \beta)\|x^k - x^*\|^2 + 2\gamma\eta\left(\eta - p\right)f(x_f^k) + 2p\gamma\eta f(x^*) \\ &\qquad + \left((1 + \alpha p\gamma\eta)\beta - p\gamma\eta\mu\right)\|x_g^k - x^*\|^2 \\ &\qquad + (1 + \alpha p\gamma\eta)(\beta^2 - \beta)\|x^k - x_g^k\|^2 - p\gamma^2\eta^2\|\nabla f(x_g^k)\|^2 \\ &\qquad + \left(\frac{p\eta\gamma}{\alpha} + p\gamma^2\eta^2\right)\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 + \left(p^2\eta^2\gamma^2 + p^2\gamma^3\eta^2 L\right)\mathbb{E}_k[\|g^k\|^2] \\ &\quad \leq (1 + \alpha p\gamma\eta)(1 - \beta)\|x^k - x^*\|^2 + 2\gamma\eta\left(\eta - p\right)f(x_f^k) + 2p\gamma\eta f(x^*) \\ &\qquad + \left((1 + \alpha p\gamma\eta)\beta - p\gamma\eta\mu\right)\|x_g^k - x^*\|^2 \\ &\qquad + (1 + \alpha p\gamma\eta)(\beta^2 - \beta)\|x^k - x_g^k\|^2 - p\gamma^2\eta^2\|\nabla f(x_g^k)\|^2 \\ &\qquad + p\eta\gamma\left(\frac{1}{\alpha} + \gamma\eta\right)\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2 + 2p^2\eta^2\gamma^2\left(1 + \gamma L\right)\mathbb{E}_k[\|g^k - \nabla f(x_g^k)\|^2] \\ &\qquad + 2p^2\eta^2\gamma^2\left(1 + \gamma L\right)\mathbb{E}_k[\|\nabla f(x_g^k)\|^2].\end{aligned}$$

In the last step we also used (A.2) with $c = 1$. Since $\gamma \leq \frac{3}{4L}$, the choice of $\alpha = \frac{\beta}{2p\eta\gamma}$, $\beta = \sqrt{4p^2\mu\gamma/3}$,

and $p\mu\gamma\eta = 3\beta/2$ gives

$$\begin{aligned}
&\beta = \sqrt{4p^2\mu\gamma/3} \le \sqrt{p^2\mu/L} \le 1,\\
&(1+\alpha p\eta\gamma)(1-\beta) = \left(1+\frac{\beta}{2}\right)(1-\beta) \le \left(1-\frac{\beta}{2}\right),\\
&((1+\alpha p\eta\gamma)\beta - p\mu\gamma\eta) = \left(\beta + \frac{\beta^2}{2} - p\mu\gamma\eta\right) \le \left(\frac{3\beta}{2} - p\mu\gamma\eta\right) \le 0,
\end{aligned}$$

and, therefore,

$$\begin{aligned}
&\mathbb{E}_k\left[\|x^{k+1}-x^*\|^2 + 2\gamma\eta^2 f(x_f^{k+1})\right]\\
&\qquad \le (1-\beta/2)\|x^k - x^*\|^2 + 2\gamma\eta\left(\eta - p\right) f(x_f^k) + 2p\gamma\eta f(x^*)\\
&\qquad\quad + p\eta^2\gamma^2\left(1+2p/\beta\right)\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2\\
&\qquad\quad + 2p^2\eta^2\gamma^2\left(1+\gamma L\right)\mathbb{E}_k[\|g^k - \nabla f(x_g^k)\|^2]\\
&\qquad\quad - p\gamma^2\eta^2(1-2p(1+\gamma L))\|\nabla f(x_g^k)\|^2.
\end{aligned}$$

Subtracting $2\gamma\eta^2 f(x^*)$ from both sides, we get

$$\begin{aligned}
&\mathbb{E}_k\left[\|x^{k+1}-x^*\|^2 + 2\gamma\eta^2(f(x_f^{k+1}) - f(x^*))\right]\\
&\qquad \le (1-\beta/2)\|x^k - x^*\|^2 + (1-p/\eta)\cdot 2\gamma\eta^2(f(x_f^k) - f(x^*))\\
&\qquad\quad + p\eta^2\gamma^2\left(1+2p/\beta\right)\|\mathbb{E}_k[g^k] - \nabla f(x_g^k)\|^2\\
&\qquad\quad + 2p^2\eta^2\gamma^2\left(1+\gamma L\right)\mathbb{E}_k[\|g^k - \nabla f(x_g^k)\|^2]\\
&\qquad\quad - p\gamma^2\eta^2(1-2p(1+\gamma L))\|\nabla f(x_g^k)\|^2.
\end{aligned}$$

Applying Lemma 5.3.6, one can obtain

$$\begin{aligned}
&\mathbb{E}_k\left[\|x^{k+1}-x^*\|^2 + 2\gamma\eta^2(f(x_f^{k+1}) - f(x^*))\right]\\
&\quad \le (1-\beta/2)\|x^k - x^*\|^2 + (1-p/\eta)\cdot 2\gamma\eta^2(f(x_f^k) - f(x^*))\\
&\qquad + p\eta^2\gamma^2\left(1+2p/\beta\right)\cdot C_2\tau^2 M^{-2}B^{-2}(\sigma^2 + \delta^2\|\nabla f(x_g^k)\|^2)\\
&\qquad + 2p^2\eta^2\gamma^2\left(1+\gamma L\right)\cdot\left(4C_1\tau B^{-1}\log_2 M + (4C_1+2)\tau^2B^{-2}\right)(\sigma^2 + \delta^2\|\nabla f(x_g^k)\|^2)\\
&\qquad - p\gamma^2\eta^2(1-2p(1+\gamma L))\|\nabla f(x_g^k)\|^2.
\end{aligned}$$

With $M \ge \sqrt{C_2 p^{-1}(1+2p/\beta)}$, we have

$$\begin{aligned}
&\mathbb{E}_k\left[\|x^{k+1}-x^*\|^2 + 2\gamma\eta^2(f(x_f^{k+1}) - f(x^*))\right]\\
&\quad \le (1-\beta/2)\|x^k - x^*\|^2 + (1-p/\eta)\cdot 2\gamma\eta^2(f(x_f^k) - f(x^*))\\
&\qquad + p^2\eta^2\gamma^2\tau^2B^{-2}(\sigma^2 + \delta^2\|\nabla f(x_g^k)\|^2)\\
&\qquad + 2p^2\eta^2\gamma^2\left(1+\gamma L\right)\cdot\left(4C_1\tau B^{-1}\log_2 M + (4C_1+2)\tau^2B^{-2}\right)(\sigma^2 + \delta^2\|\nabla f(x_g^k)\|^2)\\
&\qquad - p\gamma^2\eta^2(1-2p(1+\gamma L))\|\nabla f(x_g^k)\|^2\\
&\quad \le (1-\beta/2)\|x^k - x^*\|^2 + (1-p/\eta)\cdot 2\gamma\eta^2(f(x_f^k) - f(x^*))
\end{aligned}$$

$$+ 8p^2\eta^2\gamma^2\left(1+\gamma L\right)\cdot\left(C_1\tau B^{-1}\log_2 M + (C_1+1)\tau^2 B^{-2}\right)\sigma^2$$
$$- p\gamma^2\eta^2\left[1 - 2p\left(1+\gamma L\right)\left(1+4\left[C_1\tau B^{-1}\log_2 M + (C_1+1)\tau^2 B^{-2}\right]\delta^2\right)\right]\|\nabla f(x_g^k)\|^2.$$

Since $p = \left[1+2\left(1+\gamma L\right)\left(1+4\left[C_1\tau b^{-1}+(C_1+1)\tau^2 b^{-2}\right]\delta^2\right)\right]^{-1}$, $B = \lceil b\log_2 M\rceil$ and $M \geq 2$, we obtain

$$\begin{aligned} p &= \left[1+2\left(1+\gamma L\right)\left(1+4\left[C_1\tau b^{-1}+(C_1+1)\tau^2 b^{-2}\right]\delta^2\right)\right]^{-1} \\ &\leq \left[1+2\left(1+\gamma L\right)\left(1+4\left[C_1\tau B^{-1}\log_2 M+(C_1+1)\tau^2 B^{-2}\right]\delta^2\right)\right]^{-1}, \end{aligned}$$

and then,

$$\begin{aligned} &\mathbb{E}_k\left[\|x^{k+1}-x^*\|^2 + 2\gamma\eta^2(f(x_f^{k+1})-f(x^*))\right] \\ &\qquad\leq (1-\beta/2)\|x^k-x^*\|^2 + (1-p/\eta)\cdot 2\gamma\eta^2(f(x_f^k)-f(x^*)) \\ &\qquad\quad + 8p^2\eta^2\gamma^2\left(1+\gamma L\right)\cdot\left(C_1\tau B^{-1}\log_2 M+(C_1+1)\tau^2 B^{-2}\right)\sigma^2 \\ &\qquad\leq \max\left\{(1-\beta/2),(1-p/\eta)\right\}\left[\|x^k-x^*\|^2+2\gamma\eta^2(f(x_f^k)-f(x^*))\right] \\ &\qquad\quad + 8p^2\eta^2\gamma^2\left(1+\gamma L\right)\cdot\left(C_1\tau B^{-1}\log_2 M+(C_1+1)\tau^2 B^{-2}\right)\sigma^2. \end{aligned}$$

Using that $p\eta\gamma = 3\beta/(2\mu)$, $\beta/2 = p/\eta$, $B = \lceil b\log_2 M\rceil$ and $\gamma \leq L^{-1}$, we have

$$\begin{aligned} &\mathbb{E}_k\left[\|x^{k+1}-x^*\|^2 + 2\gamma\eta^2(f(x_f^{k+1})-f(x^*))\right] \\ &\qquad\leq (1-\beta/2)\left[\|x^k-x^*\|^2+2\gamma\eta^2(f(x_f^k)-f(x^*))\right] \\ &\qquad\quad + 36\beta^2\mu^{-2}\left(C_1\tau b^{-1}+(C_1+1)\tau^2 b^{-2}\right)\sigma^2. \end{aligned}$$

Here we also took into account that $M \geq 2$. Finally, we perform the recursion and substitute $\beta = \sqrt{4p^2\mu\gamma/3}$

$$\begin{aligned} &\mathbb{E}\left[\|x^K-x^*\|^2 + 2\gamma\eta^2(f(x_f^K)-f(x^*))\right] \\ &\qquad\leq \left(1-\sqrt{\frac{p^2\mu\gamma}{3}}\right)^N\left[\|x^0-x^*\|^2+2\gamma\eta^2(f(x_f^0)-f(x^*))\right] \\ &\qquad\quad + 72\beta\mu^{-2}\left(C_1\tau b^{-1}+(C_1+1)\tau^2 b^{-2}\right)\sigma^2 \\ &\qquad\leq \exp\left(-\sqrt{\frac{p^2\mu\gamma N^2}{3}}\right)\left[\|x^0-x^*\|^2+2\gamma\eta^2(f(x_f^0)-f(x^*))\right] \\ &\qquad\quad + \frac{144p\sqrt{\gamma}}{\sqrt{3}\mu^{3/2}}\left(C_1\sigma^2\tau b^{-1}+(C_1+1)\sigma^2\tau^2 b^{-2}\right). \end{aligned}$$

Substituting of $\eta = \sqrt{\frac{3}{\mu\gamma}}$ concludes the proof. □

List of Figures

2.1 Comparison communication complexities of Algorithm 6, EGD-GT, EGD-Con and Sliding on (2.17) over fixed networks. 50

2.2 Comparison communication complexities of Algorithm 7, EGD-GT, EGD-Con and Sliding on (2.17) over time-varying networks. 50

2.3 Convergence of Algorithm 8 with constant stepsize in the presence of stochastic noise in strongly monotone (left) and monotone (right) cases. We observe linear convergence up to an error floor depending on the noise variance and problem parameters (Theorem 2.3.4). 58

2.4 Verifying the $\mathcal{O}\big(\frac{D^2}{p^2K^2}\big)$ convergence rate for the strongly monotone noiseless ($\sigma^2 = 0$) case. 58

2.5 Convergence of Algorithm 8 with the decreasing stepsizes in the noiseless (left) and stochastic (right) cases. 59

2.6 Comparison of the three network topologies in DCGAN distributed decentralized learning on CIFAR-10. FID Score and Inception Score vs the number of communications (two top), and the same Scores vs the number of local epochs (two bottom). The experiment was repeated 5 times on different random data splitting, the maximum and minimum deviations are depicted in the plots by the shade. 61

2.7 Simulated data, 25 workers, ambient dimension = 40 67

2.8 a9a dataset 67

2.9 Comparison MASHA1 (Algorithm 11) and MASHA2 (Algorithm 12) with Error Feedback, QGD and Compressed Extra Gradient (`CEG`) in iterations and in Mbytes for (2.30). 84

2.10 (upper left) ALBERT training objective convergence rate with different compression algorithms; (upper right) ALBERT training objective convergence rate with different compression algorithms (zoomed); (lower) Average wall time per communication round with standard deviation over 5 repetitions and downstream evaluation scores on GLUE benchmark tasks after at 80 billion training tokens ($\approx 10^4$ optimizer steps). 86

2.11 Comparison of state-of-the-art methods for distributed variational inequalities. The comparison is made on synthetic datasets with small, medium, and large δ. The x-axis denotes the number of full operators transmitted by one of the devices. 102

2.12 Comparison of state-of-the-art methods for distributed variational inequalities. The comparison is made on the real datasets a9a, w7a, w8a from LibSVM. The x-axis denotes the number of full operators transmitted by one of the devices. 103

3.1 Comparison of Uniform Sampling (US) vs Importance Sampling (IS). Left: the result for the problem without constraints, right: with constraints. As expected by theory IS converges faster and to a smaller neighborhood than US. 123

3.2 Comparison of variance-reduced methods. Left: the result for the problem without constraints, right: with constraints. Note that L-SVRGDA is very competitive, and outperforms all the other methods. 124
3.3 Comparison of algorithms in distributed setting. Left: number of oracle calls, right: number of bits communicated. 124
3.4 Comparison of computational complexities for Algorithm 30, EG-Mal22-1 (Algorithm 1 [214]), EG-Mal22-2 (Algorithm 2 [214]), EG-Car19 (Algorithm 1+2 from [281]) on (3.16) with policeman and burglar matrix from [289]. 133
3.5 Comparison of computational complexities for Algorithm 30, EG-Mal22-1 (Algorithm 1 [214]), EG-Mal22-2 (Algorithm 2 [214]), EG-Car19 (Algorithm 1+2 from [281]) on (3.16) with policeman and burglar matrix from [289]. 134
3.6 Comparison of computational complexities for Algorithm 31, MP-Car19 (Algorithm 1+2 from [281]), and MP-Alac21 (Algorithm 2 [214]) with different batch sizes on test matrix from [289]. 135
3.7 Comparison of computational complexities for Algorithm 31, MP-Car19 (Algorithm 1+2 from [281]), and MP-Alac21 (Algorithm 2 [214]) with different batch sizes on policeman and burglar matrix from [289]. 136
3.8 Comparison of computational complexities for Algorithm 31(with adjusted parameters), MP-Car19 (Algorithm 1+2 from [281]), and MP-Alac21 (Algorithm 2 [214]) with different batch sizes on first test matrix from [289]. 137
3.9 Comparison of computational complexities for Algorithm 31(with adjusted parameters), MP-Car19 (Algorithm 1+2 from [281]), and MP-Alac21 (Algorithm 2 [214]) with different batch sizes on policeman and burglar matrix from [289]. 138

4.1 Comparison of state-of-the-art projection free methods with small batches for (4.6) with $R = 2000$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. In the modified plots (the right plots in the first three lines), we left only every 100th point for [320], [322], Algorithm 32 and Algorithm 33. 150
4.2 Comparison of state-of-the-art projection free methods with small batches for (4.6) with $R = 2000$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. In the modified plots (the right plots in the first three lines), we left only every 100th point for [320], [322], Algorithm 32 and Algorithm 33. 151
4.3 Comparison of state-of-the-art projection free methods with small batches for (4.7) with $R = 2000$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. In the modified plots (the right plots in the first three lines), we left only every 100th point for [320], [322], Algorithm 32 and Algorithm 33. 152

4.4 Comparison of state-of-the-art projection free methods with small batches for (4.6) with $R = 200$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. 153
4.5 Comparison of state-of-the-art projection free methods with small batches for (4.6) with $R = 20$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. 154
4.6 Comparison of state-of-the-art projection free methods with small batches for (4.6) with $R = 2$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. 155
4.7 Deterministic FW Algorithm. 166
4.8 Stochastic FW Algorithm. 167
4.9 GD Algorithm. 168
4.10 Comparison of methods for solving (4.19) in the stochastic case. SARAH FW, SAGA SARAH FW are considered. The comparison is made on LibSVM datasets w1a, rcv1. 184
4.11 Comparison of stochastic methods with baselines. SARAH FW, SAGA SARAH FW are considered. The comparison is made on LibSVM datasets w1a, rcv1. 185
4.12 Comparison of Frank-Wolfe-based algorithms and methods with projection for the stochastic problem. L-SVRG FW, SAGA SARAH FW as well as SGD and SVRG with projection are considered. The comparison is made on LibSVM dataset mushrooms. 185
4.13 Comparison of methods for solving (4.19) in the coordinate case. SEGA FW, JAGUAR are considered. The comparison is made on LibSVM datasets mushrooms, w1a. 186
4.14 Comparison of methods for solving (4.19) in the distributed case. MARINA FW, VR-MARINA FW, EF21 FW, DIANA FW are considered. The comparison is made on LibSVM datasets mushrooms, rcv1. 186

5.1 Trajectories on quadratic function. 194
5.2 Convergence of SARAH-type methods on various LiBSVM datasets. Convergence on the function. 196
5.3 Convergence of SARAH-type methods on various LiBSVM datasets. Convergence on the distance to the solution. 197
5.4 Convergence of SARAH-type methods on various LIBSVM datasets. Convergence on the norm og the gradient. 198
5.5 $\|v^k - \nabla f(x^k)\|^2$ changes. 199

5.6 The comparison of Top-k and Rand-k sparsifiers with respect to normalized variance and the number of encoding bits used for each coordinate on average. Each point/marker represents a single $d = 10^4$ dimensional vector drawn form Gaussian distribution and then compressed by the specified operator. Each curve was obtained by varying the free parameter $k \in \{1, 2, \ldots, d\}$. Plots for different d look very similar. Notice that, for random sparsification the normalized variance is perfectly linear with respect to the number of bit per coordinate. Letting b be the total number of bits to encode the compressed vector (say in *binary32* system), the normalized variance produced by random sparsifier is almost $1 - \frac{b/d}{32}$. However, greedy sparsifier achieves exponentially lower variance $\approx 0.86^{b/d}$ utilizing the same amount of bits. 209

5.7 Calculations of the Rand-5 and Top-5 energy "saving" for practical gradient distributions ((a),(b),(c): quadratic problem, (d): logistic regression). The results of Top-5 are 3–5× better. 210

5.8 Comparison of various compressors (with and without free design parameters) with respect to the parameter $\delta \geq 1$ in $\log_{10}$ −scale and the number of encoding bits used for each coordinate on average. Each point/marker represents a single $d = 10^4$ dimensional vector x drawn from Gaussian distribution and then compressed by the specified operator. Each curve was obtained by varying free parameters ($k \in \{1, 2, \ldots, d\}$ for sparsifiers, $s \geq 1$ for ditherings) of the specified compressor. Parameter-free compressors, such as ternary quantization and natural compression, have fixed communication budgets which explains the vertical arrangements of the points. 211

5.9 Comparison of empirical variance $\|C(x) - x\|^2 / \|x\|^2$ during training procedure for two pairs of method– deterministic with classic/unbiased C_{nat} and Top-k with Rand-k with Top-1/5 of coordinates. Both of the plots were produced using ResNet18, GoogleNet, and VGG19 on CIFAR10 dataset. 218

5.10 Training/Test loss and accuracy for VGG19 on CIFAR10 distributed among 4 nodes for 4 different compression operators. 219

5.11 Training loss and test accuracy for VGG11 on CIFAR10 distributed among 4 nodes for 5 different compression operators. 220

5.12 Theoretical vs. Practical Convergence of Compressed Gradient Descent on Quadratics problem with different condition number κ for Top-5 and Rand-5 compression operators. 221

5.13 Theoretical vs. Practical Convergence of Compressed Gradient Descent on Linear Regression problem for *Boston* and *Diabetes* datasets with Top-5 and Rand-5 compression operators. 222

5.14 Training objective value for ALBERT-large with different compression operators. 223

6.1 Algorithm 54 (green), first-order Mirror Descent (orange) and zero-order Mirror Descent (blue) applied to solve (6.17) in distributed form (6.18) with different network topologies. 250

6.2 Evolution of $\|\nabla f(x_t)\|$ for different datasets, regularization parameter and different level of solving the subproblem (6.20) (larger T means we optimize the subproblem better). 260

6.3 Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, levels of λ and number of devices $n \in \{25, 100\}$ with cyclic network. 260
6.4 Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, levels of λ and number of devices $n \in \{25, 100\}$ with grid network. 261
6.5 Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, levels of λ and number of devices $n \in \{25, 100\}$ with erdos network. 261
6.6 Comparison of average accuracy of local models on local data for various communication networks and levels of λ (r). 262
6.7 Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, number of devices $n \in \{25, 100\}$ and different probability of keeping the communication edge p with the erdos network. 263
6.8 Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, number of devices $n \in \{25, 100\}$ and different probability of keeping the communication edge p with grid network. 263
6.9 Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, number of devices $n \in \{25, 100\}$ and different probability p of keeping the device in the communication with erdos network. 264
6.10 Evolution of $\|\nabla f(x_t)\|$ and $x_t^T W x_t$ for various datasets, number of devices $n \in \{25, 100\}$ and different probability p of keeping the device in the communication with grid network. 264
6.11 Evolution of $\|\nabla f(x_t)\|$, $x_t^T W x_t$ and average accuracy of local models for rcv1 datasets and erdos and grid network. We compare with two levels of regularization: low with $\lambda = 0.125 \cdot \frac{L}{\lambda_{\max}(W)}$ and high with $\lambda = 16 \cdot \frac{L}{\lambda_{\max}(W)}$. Note that, although both regularization values give comparable $\|\nabla f(x_t)\|$, the average accuracy for the case with smaller penalization is better. 265

List of Tables

2.1 Lower and upper bounds for distributed smooth general stochastic strongly monotone (sm) or monotone (`c) variational inequalities in the centralized and decentralized (constant and time-varying) cases.` 32

2.2 Comparison of approaches for distributed strongly monotone (SM), monotone (M), and non-monotone (NM) variational inequalities or, respectively, strongly convex-strongly concave, convex-concave, non-convex-non-concave saddle point problems. 52

2.3 Hyperparameters for DCGAN training. 60

2.4 Summary of our iteration complexity results for finding an ε-solution for problem (2.1) in the deterministic (i.e., (2.7)) with only device compression, deterministic with bidirectional (device-server) compression, stochastic (i.e., (2.7)+(2.28)) and federated learning/partial participation (i.e., (2.7)+(2.29)) setups. In the strongly monotone (strongly convex - strongly convex) case, convergence is measured by the distance to the solution. In the monotone(convex-concave) case, convergence is measured in terms of the gap function (2.8). In non-monotone (non-convex-non-concave) case convergence is measured in terms of the norm of the operator. 70

2.5 Summary of the results on the number of transmitted information in different approaches to communication bottleneck for distributed variational inequalities and saddle point problems. 88

3.1 Summary of the complexity results for variance reduced methods for solving (3.1). By complexity we mean the number of oracle calls required for the method to find z such that $\mathbb{E}[\|z - z^*\|^2] \leq \varepsilon$. Dependencies on numerical and logarithmic factors are hidden. By default, operator F is assumed to be μ-strongly monotone and, as the result, the solution is unique. Our results rely on μ-quasi strong monotonicity of F (3.3), but we also assume uniqueness of the solution. 115

3.2 Summary of the complexity results for distributed methods with unbiased compression for solving distributed (3.1) with $F(z) = \frac{1}{n}\sum_{i=1}^n F_i(z)$. By complexity we mean the number of communication rounds required for the method to find z such that $\mathbb{E}[\|z - z^*\|^2] \leq \varepsilon$. Dependencies on numerical and logarithmic factors are hidden. Our results rely on μ-quasi strong monotonicity of F (3.3), but we also assume the uniqueness of the solution. 119

3.3 Summary of the complexity results for zeroth-order methods with two-points feedback oracles for solving (3.1). By complexity we mean the number of oracle calls required for the method to find z such that $\mathbb{E}[\|z - z^*\|^2] \leq \varepsilon$. By default, operator F is assumed to be μ-strongly monotone and, as the result, the solution is unique. Our results rely on μ-quasi strong monotonicity of F (3.3). 120

3.4 Summary complexities for finding an ε-solution for monotone stochastic (finite-sum) variational inequality (3.1)+(3.13) with Lipschitz operators. Convergence is measured by gap function. 127

4.1 Summary of the results on projection free methods for stochastic constrained minimization problems. 142
4.2 Datasets from LibSVM in experiments. 149
4.3 Adjustment of the various zero-order and coordinate FW methods. 157
4.4 Summary of complexity results for finding an ε-solution stochastic finite-sum non-distributed constrained minimization problems (4.4) with n terms by projection free methods. Convergence is measured by the functional distance to the solution in the convex case and by the gap function in the non-convex case. Complexities are given in terms of the number of stochastic gradient calls. 175
4.5 Summary of complexity results for finding an ε-solution constrained minimization problems with dimension d by coordinate projection free methods. Convergence is measured by the functional distance to the solution in the convex case and by the gap function in the non-convex case. Complexities are given in terms of the number of coordinates derivatives computed. 177
4.6 Summary of complexity results for finding an ε-solution distributed constrained minimization problems (4.4) with n devices by projection free methods with compression with parameters ω and δ (see Definitions 4.3.14 and 4.3.19). Convergence is measured by the functional distance to the solution in the convex case and by the gap function in the non-convex case. Complexities are given in terms of the number of transmitted coordinates if we choose RandK and TopK as particular cases of compressors. 181

5.1 Summary of results on solving a smooth strongly convex finite-sum optimization problem. The results are given both for independent generation of batch numbers and for different shuffling techniques. In the case of independent generation, the results are given for reference and make no claim to completeness. 190
5.2 Summary of datasets. 195
5.3 Optimal parameters for SARAH. 195
5.4 Complexity results for GD with biased compression. The identity compressor $C(x) \equiv x$ belongs to all classes with $\alpha = \beta = \gamma = \delta = 1$; all three results recover standard rate of Gradient Descent. 201
5.5 Ergodic convergence of distributed SGD with biased compression and error-feedback (Algorithm 50) for L-smooth and μ-strongly convex functions (K communications). Details are given in Theorem 5.2.26. 202
5.6 Compressors C described in Section 5.2.3 and their membership in $\mathbb{B}^1(\alpha, \beta)$, $\mathbb{B}^2(\gamma, \beta)$, $\mathbb{B}^3(\delta)$ and $\mathbb{U}(\zeta)$. 205
5.7 Information savings of greedy and random sparsifiers for $k = 3$ and $k = 5$. 209

5.8 Comparison of average time per one communication round and downstream evaluation scores on GLUE benchmark tasks. 223

5.9 This table summarizes our results on first-order method with Markovian noise. The columns of the table indicate whether the authors consider optimization over bounded domain, potentially unbounded gradients, and whether or not they assume additional restrictions on the Markovian noise (finite state space or reversibility). For ease of comparison we provide the respective results on SGD and ASGD (accelerated SGD) in the i.i.d. setting. 225

5.10 This table summarizes the findings on methods for solving stochastic (strongly) monotone variational inequalities with a Lipschitz operator and (un)bounded stochasticity. The columns of the table indicate whether the authors consider variational inequalities or only certain saddle point problems, the arbitrariness of the sets, and the use of additional composite terms. The columns on stochasticity provide information on the assumptions made with respect to the stochastic operator, such as bounded variance and the Markovian noise setting. Note that with the exception of our work, all other studies assume the independent noise. 227

6.1 Summary of complexity results on the number of ∇g and f calls for finding an ε-solution of (6.5) in the different geometric setups: Euclidean and non-Euclidean. Convergence is measured by the function distance to the solution. 243

6.2 Summary of complexity results on communications (comm) and local computations (local) for finding an ε-solution of centralized (6.2)+(6.3) and decentralized (6.4) distributed problems. Convergence is measured by the function distance to the solution. 243

6.3 Summary of complexity results (upper and lower bounds) on communications (comm) and local computations (local) for finding an ε-optimal solution of (6.4) in the deterministic (gradient) case. 251

6.4 The number of features and number of samples for each dataset used in the numerical experiments. 257