Part I. Nonlinear Programming

Lecture 1.

Introduction.

– About this course.

– General formulation of the problem.

– Important examples.

– Black Box and Iterative Methods.

– Analytical and Arithmetical Complexity.

– Uniform Grid Method.

– Lower complexity bounds.

– Lower bounds for Global Optimization.

– Rules of the Game.

About this course.

What we are going to do:

- Give a description of the modern optimization theory.
- Take into account the historical aspect in order to understand the logic of the development.
- Develop a "computational sense", which helps us to understand what we *can* and what we *cannot* expect from a numerical method.

Why that is important:

- Optimization formulations are very popular among practitioners.
- Optimization Theory is simple and easy to learn.
- Optimization is an excellent example of a *complete* theory.

General formulation of the problem.

Let x be an n-dimensional real vector:

$$x = (x^{(1)}, \ldots, x^{(n)}) \in R^n,$$

S be a subset of R^n: $S \subseteq R^n$,

$f_0(x) \ldots f_m(x)$ are some real-valued function of x.

Problem formulation:

$$\begin{array}{rl} & \min \; f_0(x) \qquad (\equiv -\max(-f_0(x)) \\ \text{s.t.:} & f_j(x) \left(\begin{array}{c} = \\ \leq \end{array} \right) 0, j = 1 \ldots m, \\ & x \in S. \end{array}$$

Terminology:

$f_0(x)$ - objective function,

$f(x) = (f_1(x), \ldots, f_m(x))$ - functional constraints,

S - basic feasible set.

Q - feasible set:

$$Q = \{x \in S \mid \; f_j(x) \leq 0, \; j = 1 \ldots m\}.$$

Types of minimization problems:

– *Constrained problems*: $Q \subset R^n$,

– *Unconstrained problems*: $Q \equiv R^n$,

– *Smooth problems*: all $f_j(x)$ are differentiable,

– *Nonsmooth problems*: there is a nondifferentiable component $f_k(x)$,

– *Linearly constrained problems*: all functional constraints are linear:

$$f_j(x) = \sum_{i=1}^{n} a_{i,j} x_i + b_j \equiv \langle a_j, x \rangle + b_j, \; j = 1 \ldots m,$$

$\uparrow$ *inner product*

and S is a polyhedron.

If $f_0(x)$ is also linear then that is a *Linear Programming Problem.*

If $f_0(x)$ is quadratic then that is a *Quadratic Programming Problem.*

Feasibility:

– Problem is *feasible* if $Q \neq \emptyset$.
– Problem is *strictly feasible* if $\exists x \in \operatorname{int} Q$ such that

$$f_j(x) < 0$$

for all inequality constraints and

$$f_j(x) = 0$$

for all equality constraints.

Extremum:

– x^* is an optimal *global solution* to the problem if

$$f_0(x^*) \leq f_0(x) \ \text{ for all } x \in Q$$

(*global minimum*). Then $f_0(x^*)$ is called the *optimal value* of the problem.

– x^* is an optimal *local solution* to the problem if

$$f_0(x^*) \leq f_0(x) \ \text{ for all } x \in \operatorname{int} \bar{Q} \subset Q$$

(*local minimum*).

Example of the problem, 1.

Let $x^{(1)} \dots x^{(n)}$ be our *design variables*.

Then we can fix some *characteristics* of our decision:

$$f_0(x), \dots, f_m(x).$$

That could be:

- The price of the project,
- Amount of the required resources,
- Reliability of the system,

and many others.

We fix the most important characteristics, $f_0(x)$, as our *objective*.

For all others we impose some bounds:

$$a_j \le f_j(x) \le b_j.$$

Thus, we come to the problem:

$$\begin{aligned} &\min\ f_0(x), \\ \text{s.t.:}\ & a_j \le f_j(x) \le b_j,\ j = 1 \dots m, \\ & x \in S, \end{aligned}$$

where S stands for the *structural* constraints (for example, positiveness of some variables)

Example of the problem, 2^*.

Let our initial problem be as follows:
Find $x \in R^n$:

$$\begin{aligned} f_1(x) &= a_1, \\ &\dots \\ f_m(x) &= a_m. \end{aligned} \tag{1}$$

Then we can consider the problem:

$$\min_x \sum_{j=1}^{m} (f_j(x) - a_j)^2$$

(may be with some additional constraints on x).

Note:
The problem (1) is almost *universal.*

It covers:

- Ordinary differential equations,
- Partial differential equations,
- Problems, arising in Game Theory,

and many other fields.

Example of the problem, 3^{**}.

Let our decision variable $x^{(1)} \dots x^{(n)}$ must be *integer*.

That can be described by the constraint:

$$\sin(\pi x^{(i)}) = 0, \quad i = 1 \dots n.$$

Thus, we could treat also the *Integer Programming* Problems:

$$\begin{aligned} & \min \ f_0(x), \\ \text{s.t.:}\ & a_j \le f_j(x) \le b_j,\ j = 1 \dots m, \\ & x \in S, \\ & \sin(\pi x^{(i)}) = 0,\ i = 1 \dots n. \end{aligned}$$

Conclusion:

1955:

– Nonlinear Optimization is very important. It covers almost all fields of Numerical Analysis.

1975:

– In general, optimization problems are *unsolvable*.

Performance of a Numerical Method

$$\text{Numerical Method } \mathcal{M} \quad \Longleftrightarrow \quad \text{Problem } \mathcal{P}$$

What we can say about the performance of $\mathcal{M}$?

Observations:

1. Performance of $\mathcal{M}$ with respect to a *single* problem $\mathcal{P}$ is a silly notion.

(All methods are worse than the trivial one returning the solution of $\mathcal{P}$ all the time.)

2. Therefore we need:

- Description of a *class* of problems $\mathcal{F} \supset \mathcal{P}$.
- Description of an *oracle* $\mathcal{O}$, which provides $\mathcal{M}$ by some information about $\mathcal{P}$.

The *class* and the *oracle* compose the *model* of our problem. (Not unique !)

We can define the *performance* of $\mathcal{M}$ on $(\mathcal{F}, \mathcal{O})$ as its performance on the *worst* $\mathcal{P}_w$ from $\mathcal{F}$.

(Note that this P_w can be bad only for $\mathcal{M}$!)

Performance of $\mathcal{M}$ on $\mathcal{P}$:

The total amount of *Computational Efforts*

which is required by method $\mathcal{M}$

to *Solve the Problem* $\mathcal{P}$.

To Solve the Problem could mean:

1. Find the *exact* solution.
(Impossible to find in finite time even for the simplest nonlinear problems.)

2. Find an *approximate* solution with a small accuracy $\epsilon > 0$.
(For that apply an *iterative process*.)

General Iterative Scheme.

Input:

- A starting point x_0.
- An accuracy $\epsilon > 0$.

Initialization. Set $k = 0$, $I_{-1} = \emptyset$.

k is the iteration counter.

I_k is the *informational set* accumulated after k iterations.

Main Loop.

1. Call the oracle $\mathcal{O}$ at x_k.

2. Update the informational set:

$$I_k = I_{k-1} \bigcup (x_k, \mathcal{O}(x_k)).$$

3. Apply the rules of method $\mathcal{M}$ to I_k and form the new test point x_{k+1}.

4. Check the stopping criterion. If **yes** then form an output $\bar{x}$. Otherwise set $k = k + 1$ and go to 1.

Computational Efforts:

1. *Analytical complexity:*

   The number of calls of the oracle, which is required to solve the problem $\mathcal{P}$ upto the accuracy ϵ.

2. *Arithmetical complexity:*

   The total number of the arithmetic operations (including the work of the oracle and the method), which is required to solve the problem $\mathcal{P}$ upto the accuracy ϵ.

Note: The meaning of the words *upto the accuracy* $\epsilon > 0$ must be *exact*.

Black Box Concept:

1. The only information available from the oracle is its answer. No intermediate results are available.

2. The oracle is *local*:
A small variation of the problem far enough from the test point x does not change the answer at x.

Note: This concept is extremely popular, but it is not inevitable. We will see that later.

Examples of the oracle:

1. *Zero-order* oracle.

Input: test point x.

Output: the value $f(x)$.

2. *First-order* oracle.

Input: test point x.

Output: the value $f(x)$ and the gradient $f'(x)$.

3. *Second-order* oracle.

Input: test point x.

Output: the value $f(x)$, the gradient $f'(x)$ and the Hessian $f''(x)$.

Uniform Grid Method.

Problem Formulation:

$$\min \quad f(x),$$
$$x \in B_n$$

where B_n is an n-dimensional box in R^n:

$$B_n = \{x \in R^n \mid \ 0 \le x_i \le 1,\ i = 1, \ldots, n\}.$$

Assumption: (≡ Problem Class)

The objective function $f(x)$ is *Lipshitz continuous* on B_n:

$$\forall x, y \in B_n : \quad \mid f(x) - f(y) \mid \le L \parallel x - y \parallel$$

with some constant L (*Lipshitz constant*).

Here and in the sequel $\parallel \cdot \parallel$ is the *Euclidean norm* on R^n:

$$\parallel x \parallel = \langle x, x \rangle = \sqrt{\sum_{i=1}^{n} (x_i)^2}.$$

Scheme of the method $\mathcal{G}(p)$.
(p is an integer input parameter)

1. Form p^n points

$$x_{(i_1,i_2,\ldots,i_n)} = \left(\frac{i_1}{p} - \frac{1}{2p}, \frac{i_2}{p} - \frac{1}{2p}, \ldots, \frac{i_n}{p} - \frac{1}{2p}\right),$$

where

$$\begin{array}{c} i_1 = 1, \ldots, p, \\ i_2 = 1, \ldots, p, \\ \ldots \\ i_n = 1, \ldots, p. \end{array}$$

2. Among all points $x_{(\ldots)}$ find the point $\bar{x}$ with the minimal value of the objective function.

3. Return the pair $(\bar{x}, f(\bar{x}))$ as the result.

Note: 1. This method can be treated as an iterative process without any influence of the accumulated information on the sequence of test points.

2. This is a zero-order method.

Theorem 1.1. *Let f^* be the global optimal value of our problem. Then*

$$f(\bar{x}) - f^* \leq L\frac{\sqrt{n}}{2p}.$$

Proof. Let x^* be the global minimum of our problem. Then there exists a "number"

$$(i_1, i_2, \ldots, i_n)$$

such that

$$x \equiv x_{(i_1, i_2, \ldots, i_n)} \leq x^* \leq x_{(i_1+1, i_2+1, \ldots, i_n+1)} \equiv y$$

(here and in the sequel we write $x \leq y$ for $x, y \in R^n$ iff $x_i \leq y_i$, $i = 1, \ldots, n$).

Note that $\quad y_i - x_i = 1/p$, $i = 1, \ldots, n$,

and $\quad x_i^* \in [x_i, y_i]$, $i = 1, \ldots, n$,

Denote $\hat{x} = (x + y)/2$.

Let us form a point $\tilde{x}$ as follows:

$$\tilde{x}_i = \begin{cases} y_i, & \text{if } x_i^* \geq \hat{x}_i, \\ x_i, & \text{otherwise.} \end{cases}$$

It is clear that

$$| \tilde{x}_i - x_i^* | \leq \frac{1}{2p}, \quad i = 1, \dots, n.$$

Therefore

$$\| \tilde{x} - x^* \|^2 = \sum_{i=1}^{n} (\tilde{x}_i - x_i^*)^2 \leq \frac{n}{4p^2}.$$

Since $\tilde{x}$ belongs to our grid, we conclude that

$$\begin{aligned} f(\bar{x}) - f(x^*) &\leq f(\tilde{x}) - f(x^*) \\ &\leq L \| \tilde{x} - x^* \| \leq L \tfrac{\sqrt{n}}{2p} \end{aligned}$$

□

Approximate solution:

Find $\bar{x} \in B_n : f(\bar{x}) - f^* \leq \epsilon$.

Corollary 1.1. *The analytical complexity of the method $\mathcal{G}$ is as follows:*

$$\mathcal{A}(\mathcal{G}) = \left(\left] L\frac{\sqrt{n}}{2\epsilon} \right[+ 1 \right)^n$$

(here $]a[$ is the integer part of a).

Proof. Indeed, let us take

$$p = \left] L\frac{\sqrt{n}}{2\epsilon} \right[+ 1.$$

Then

$$p \geq L\frac{\sqrt{n}}{2\epsilon},$$

and therefore, in view of T.1.1,

$$f(\bar{x}) - f^* \leq L\frac{\sqrt{n}}{2p} \leq \epsilon.$$

□

Questions:

- How good is this estimate?
- How good is this method?

Lower complexity bounds.

1. Are based on the *Black Box* concept.

2. Can be derived for a specific class of problems $\mathcal{F}$ equipped by an oracle $\mathcal{O}$.

3. Are valid for *any* iterative scheme.

4. Provide us with a lower bound for the *analytical complexity* of the class $\mathcal{F}$.

5. Use the idea of *resisting* oracle.

Resisting Oracle:

1. It is trying to create a *worst* problem for each concrete method.

2. It starts from an "empty" function and it tries to answer each call of the method in the worst possible way.

3. However, the answers must be *compatible* with

- Previous answers.
- Description of the problem class.

Note that:

- After the termination of the method, it is possible to *reconstruct* the created problem.
- If we launch the method on this problem, it will reproduce the same sequence of the test points.

Lower bounds for Global Optimization.

Problem Formulation:

$$\min \quad f(x),$$
$$x \in B_n$$

where

$$B_n = \{x \in R^n \mid \; 0 \le x_i \le 1, \; i = 1, \dots, n\}.$$

Problem Class:

The objective function $f(x)$ is *Lipshitz continuous* on B_n.

Approximate solution:

Find $\bar{x} \in B_n : f(\bar{x}) - f^* \le \epsilon$.

Theorem 1.2. *The analytical complexity of this class for the* 0*-order methods is at least*

$$\left(\left] \frac{L}{2\epsilon} \right[\right)^n .$$

Proof. Assume that there exists a method, which needs no more than

$$N < p^n, \quad p = \left] \frac{L}{2\epsilon} \right[\ (\geq 1),$$

calls of oracle to solve any problem of our class upto accuracy $\epsilon > 0$.

Let us apply this method to a resisting oracle:

It reports that $f(x) = 0$ at any test point.

Therefore this method can find only

$$\bar{x} \in B_n : \quad f(\bar{x}) = 0.$$

Note that there exists $\hat{x} \in B_n$ such that

$$\hat{x} + \frac{1}{p} e \in B_n, \quad e = (1, \ldots, 1),$$

and there were no test points in the box

$$B = \{x \mid \hat{x} \leq x \leq \hat{x} + \frac{1}{p} e\}.$$

Denote $\tilde{x} = \hat{x} + \frac{1}{2p} e$ and consider the function

$$\bar{f}(x) = \min\{0, L \parallel x - \tilde{x} \parallel_\infty - \epsilon\},$$

where $\parallel a \parallel_\infty = \max_{1 \leq i \leq n} \mid a_i \mid$.

Note that:

1. Function $\bar{f}(x)$ is Lipshitz continuous.
(Check that using the inequality $\| a \|_\infty \leq \| a \|$).

2. The optimal value of $\bar{f}(\cdot)$ is $-\epsilon$.

3. $B \equiv \{x \mid \| x - \tilde{x} \|_\infty \leq \frac{1}{2p}\}$.

4. Function $\bar{f}(x)$ differs from zero only inside the box

$$B' = \{x \mid \| x - \tilde{x} \|_\infty \leq \frac{\epsilon}{L}\}.$$

5. Since $2p \leq L/\epsilon$, we conclude that

$$B' \subseteq B.$$

Therefore $\bar{f}(x)$ is equal to zero *at all test points* of our method.

Since the accuracy of the result of our method is ϵ, we come to the following conclusion:

> *If the number of calls of the oracle is less than p^n then the accuracy of the result cannot be less than ϵ.*

□

What we can say now?

	$\mathcal{G}$	Optimal
Complexity Estimate	$\left(L\frac{\sqrt{n}}{2\epsilon}\right)^n$	$\left(\frac{L}{2\epsilon}\right)^n$

1. The dependence in ϵ is *optimal*.

2. The dependence in n is *not optimal*

Note: Our conclusion depends on the *problem class*.

Exercise: Prove that for the problem class

$$\forall x, y \in B_n : \quad | f(x) - f(y) | \leq L \parallel x - y \parallel_\infty$$

the Uniform Grid Method is *optimal* with the efficiency estimate:

$$\left(\frac{L}{2\epsilon}\right)^n .$$

What does it mean: unsolvable?

Lower complexity bound: $\left(\frac{L}{2\epsilon}\right)^n$.

Example:

$$L = 2,$$
$$n = 10, \quad \text{(very small size)},$$
$$\epsilon = 0.01, \quad (\ 1\%\ \text{accuracy}).$$

Lower bound:	10^{20} calls of oracle,
Complexity of the oracle:	n a.o.,
Total complexity:	10^{21} a.o.,
Sun Station:	10^6 a.o. per second,
Total time:	10^{15} seconds,
1 year:	less than $3.2 \cdot 10^7$ sec.
We need:	32 000 000 years.

Note:

$(n \to n+1) \Rightarrow$ Multiply complexity by 100.

But:

$(\epsilon \to 2\epsilon) \quad \Rightarrow \quad$ Divide complexity by 1000.

$\epsilon = 8\% \quad \Rightarrow \quad$ 2 weeks.

Why this works in another fields?

Example: Integration.

Problem: Compute the integral $\mathcal{I} = \int_0^1 f(x)dx$.

Discrete Sum:

$$S_n = \frac{1}{N}\sum_{i=1}^{n} f(x_i), \quad x_i = \frac{i}{N}, \; i = 1, \ldots, N.$$

Result: If $f(x)$ is Lipshitz continuous then

$$N = L/\epsilon \quad \Rightarrow \quad | \mathcal{I} - S_N | \leq \epsilon.$$

This approach is standard. Why?

The reason of the difference is in the *dimension* !

Integration: $1 - 3$,

Optimization: $1 - 10\ 000\ 000$.

What is the next?

Reasons to stop:

– We have already proved everything.

– This problem is too difficult to solve. We cannot wait for 32 000 000 years. Forget it.

Reasons to continue:

– We need to solve these problems.

– We know that people have already solved a lot of them and they are enjoyed by the results.

– May be we want too much?

Rules of the Game

Primary:

- Description of the goals.
- Description of the problem class.
- Description of the oracle.

Secondary:

- Desired properties of the methods.

Global Optimization
(Lecture 1)

Goals: Find a global minimum.

Problem Class: Continuous functions.

Oracle: $0 - 1 - 2$ order black box.

Desired properties: Convergence to a global minimum.

Features:

- This game is too short.
- We always lose it.

Problem Sizes: Sometimes people pretend to solve problems with several thousands of variables. No guarantee for success even for very small problems.

History:

- Starts from 1955.
- Several local peaks of interest (simulated annealing, neural networks, genetic algorithms).

Nonlinear Optimization

Goals: Find a local minimum.

Problem Class: Differentiable functions.

Oracle: $1-2$ order black box.

Desired properties: Convergence to a local minimum. Fast convergence.

Features:

- Variability of approaches.
- Most widespread software.
- The goal is not always acceptable.

Problem Sizes: upto 1000 variables.

History:

- Starts from 1955.
- Peak period: 1965 – 1975.
- Theoretical activity now is rather low.

Convex Optimization
(Lectures 2 – 5)

Goals: Find a global minimum.

Problem Class: Convex sets and functions.

Oracle: 1st order black box.

Desired properties: Convergence to a global minimum. Rate of convergence depends on the dimension.

Features:

- Very reach and interesting theory.
- Complete complexity theory.
- Efficient practical methods.
- The problem class is sometimes restrictive.

Problem Sizes: upto 1000 variables.

History:

- Starts from 1970.
- Peak period: 1975 – 1985.
- Theoretical activity now is rather high.

Interior-Point Polynomial Methods
(Lectures 6 – 8)

Goals: Find a global minimum.

Problem Class: Convex sets and functions.

Oracle: 2nd order oracle which is not a black box.

Desired properties: Fast convergence to a global minimum. Rate of convergence depends on the structure of the problem.

Features:

- Very new and perspective theory.
- Avoid the black box concept.
- The problem class is the same as in Convex Programming.

Problem Sizes: Sometimes up to 10 000 000 variables.

History:

- Starts from 1984.
- Peak period: $1990 - \ldots$
- Very high theoretical activity just now.