На правах рукописи

ГАФАРОВ Евгений Рашидович

АЛГОРИТМЫ РЕШЕНИЯ NP-ТРУДНЫХ ЗАДАЧ МИНИМИЗАЦИИ СУММАРНОГО ЗАПАЗДЫВАНИЯ И МИНИМИЗАЦИИ ВРЕМЕНИ ВЫПОЛНЕНИЯ ПРОЕКТА И ИХ ПРИМЕНЕНИЕ В КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ.

специальность 01.01.09 – дискретная математика и математическая кибернетика

Автореферат

диссертации на соискание ученой степени
кандидата физико-математических наук

Москва — 2008

Работа выполнена в отделе математических проблем распознавания и методов комбинаторного анализа Вычислительного центра им. А.А. Дородницына Российской академии наук.

Научный руководитель: доктор физико-математических наук,
доцент Лазарев Александр Алексеевич

Официальные оппоненты: доктор технических наук,
профессор Коган Дмитрий Израилевич,
Московский государственный университет
приборостроения и информатики,

кандидат физико-математических наук
Посыпкин Михаил Анатольевич,
Институт системного анализа РАН

Ведущая организация: Центральный экономико-математический
институт РАН

Защита диссертации состоится "27" июня 2008 г. в 9 часов на заседании диссертационного совета Д 212.156.05 при Московском физико-техническом институте (государственном университете).

С диссертацией можно ознакомиться в библиотеке МФТИ (ГУ).

Автореферат разослан "21" мая 2008 г.

Ученый секретарь
диссертационного совета Федько О.С.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы.

Диссертационная работа посвящена исследованию NP–трудных задач теории расписаний "Минимизация суммарного запаздывания для одного прибора" (далее $1 \mid\mid \sum T_j$) и "Минимизация времени выполнения проекта" (далее RCPSP, как принято в англоязычной литературе), а также алгоритмам решения классических комбинаторных задач "Разбиение" и "Ранец" [1].

В теории расписаний основное внимание уделяется вопросам оптимального распределения и упорядочения конечного множества требований, обслуживаемых детерминированными системами одного или нескольких приборов при различных предположениях относительно характера их обслуживания. Изучаемые в теории расписаний задачи выбора очередности обслуживания имеют самый общий характер и возникают при различных видах целенаправленной деятельности, например, при календарном планировании производства, строительства, транспортных перевозок, обучения, информационно–вычислительных процессов.

Большинство задач теории расписаний являются NP–трудными, поэтому важное направление исследований заключается в разработке подходов к их решению и соответствующих алгоритмов. При этом часто используются идеи метода ветвей и границ, динамического программирования, методов нахождения приближенного решения, идеи метаэвристических алгоритмов. Алгоритмы и идеи, полученные при исследовании классических задач теории расписаний "с одним прибором", применяются при решении классических комбинаторных задач и практических задач теории расписаний (например, в задаче построения расписания проекта или в задачах построения расписаний для многостадийных систем). В свою очередь, алгоритмы решения классических задач "Разбиение" и "Ранец" используются в качестве вычислительных процедур при нахождении верхних и нижних оценок функционалов в задачах теории расписаний. Этому направлению исследований и посвящена диссертация.

[1] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи: Пер. с англ. – М.: Мир, 1982. – 416 с.

Цель работы заключается в исследовании задач $1 \,||\, \sum T_j$, RCPSP для отыскания новых свойств оптимальных расписаний, построения на их основе алгоритмов решения и применения полученных результатов для классических NP–трудных задач комбинаторной оптимизации "Разбиение" и "Ранец".

Научная новизна.

1. Для задачи "Минимизация суммарного запаздывания для одного прибора" предложены новые точные (полиномиальные и псевдополиномиальные) и метаэвристические алгоритмы решения. Проведены экспериментальные исследования алгоритмов. Определена трудоемкость известных точных алгоритмов других исследователей. Представлено доказательство NP-трудности одного частного канонического случая задачи.

2. Для классических комбинаторных задач "Разбиение" и "Ранец" предложены новые точные "графические" алгоритмы решения, позволяющие решать примеры с нецелочисленными или большими числовыми параметрами быстрее, чем алгоритмы динамического программирования.

3. Для задачи "Минимизация времени выполнения проекта" проведен анализ известных нижних оценок функционала. Получены новые оценки целевой функции. Выдвинута гипотеза о том, что оптимальное значение целевой функции для задачи, когда запрещены прерывания при обслуживании требований, не более чем в два раза превышает оптимальное значение функционала в задаче, когда прерывания разрешены. Гипотеза доказана для некоторых частных случаев.

4. Показано, что для любого проекта (задача "Минимизация времени выполнения проекта") существует соответствующий проект с планарным сетевым графиком отношений предшествования.

5. Предложен используемый на практике метаэвристический алгоритм решения задачи "Минимизация времени выполнения проекта", основанный на методе "Муравьиные колонии".

Методы исследования. При формулировке и доказательстве результатов использованы методы дискретной математики и

математической кибернетики, в частности теории расписаний, теории оптимизации и математического программирования. Достоверность результатов диссертации подтверждена строгими доказательствами всех сформулированных теорем и лемм и вычислительными экспериментами.

Теоретическая и практическая значимость. Полученные в работе результаты для задачи $1 \mid\mid \sum T_j$ могут быть использованы для построения алгоритмов решения многоприборных и многостадийных задач выбора очередности обслуживания, возникающих на практике. Предложенные методы и подходы могут быть применены к исследованию других задач теории расписаний и комбинаторной оптимизации. Результаты, полученные для задачи RCPSP, могут без значительных изменений использоваться на практике в программных продуктах MS Project, Spyder Project, Primavera. В частности, результаты были применены в пакете "1С: Управление строительной организацией".

Апробация работы. Результаты диссертационной работы докладывались и обсуждались на XLIII и XLIV Международных студенческих конференциях «Студент и научно-технический прогресс» (Новосибирск, НГУ, 2005, 2006 гг.), на XII Международной конференции «INCOM 2006. IFAC Symposium on Information Control Problems in Manufacturing» (Сент-Этьен, Франция, 2006 г.), на XVIII Международной конференции «ECCO'05 European Chapter on Combinatorial Optimization» (Минск, Беларусь, 2005 г.), на XIX Международной конференции «ECCO'06 European Chapter on Combinatorial Optimization» (Порто, Португалия, 2006 г.), на Международной конференции «OR'2006. International Conference on Operation Research» (Карлсруэ, Германия, 2006 г.), на V Московской международной конференция по исследованию операций «ORM'2007. Moscow International Conference on Operation Research» (Москва, 2007 г.), на научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук» (Долгопрудный, МФТИ, 2006 г.), на научных семинарах Института проблем управления РАН (руководитель семинара профессор Д.А. Новиков, 2005–2007 гг.), на научных семинарах отдела математических проблем распознавания и методов комбинаторного анализа Вычислительного Центра РАН (2005–2007 гг.).

Публикации. По теме диссертации опубликовано 13 печатных работ,

в том числе три в изданиях из списка, рекомендованного ВАК РФ [1,2,3].

Структура и объем работы. Диссертация состоит из введения, трех глав, заключения, указателя основных обозначений и списка использованных источников (110 наименований). Содержит 21 таблицу и 39 иллюстраций. Общий объем диссертации составляет 181 страница.

СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обосновывается актуальность темы диссертационной работы, приводится обзор литературы, описывается структура диссертации и кратко излагается содержание работы.

В **первой главе** исследуется классическая задача теории расписаний – минимизация суммарного запаздывания для одного прибора ($1 \mid\mid \sum T_j$). Предложены полиномиальные и псевдополиномиальный алгоритмы решения частных случаев задачи. Приводится доказательство NP-трудности в обычном смысле частного случая задачи. Показано, что известные точные алгоритмы[2,3], для которых не получена оценка трудоемкости, имеют экспоненциальную трудоемкость. На основе точного алгоритма и метаэвристического *алгоритма "Муравьиные колонии"* построен *Гибридный алгоритм*. Приводятся результаты экспериментальных исследований на трех классах примеров.

В параграфе 1.1 рассматривается постановка исследуемой задачи, вводятся необходимые понятия и обозначения, описаны основные результаты исследования задачи, полученные другими исследователями.

Необходимо обслужить n требований на одном приборе. Прерывания при обслуживании и обслуживание более одного требования в любой момент времени запрещены. Для требования $j \in N = \{1, 2, \ldots, n\}$ заданы продолжительность обслуживания $p_j > 0$ и директивный срок окончания обслуживания d_j, где N – множество требований, которые необходимо обслужить. Задан момент освобождения прибора

[2]Лазарев А.А. Решение NP-трудной задачи теории расписаний минимизации суммарного запаздывания // Журнал Вычислительной математики и математической физиики – 2007. том 47, N.6. – С. 1087–1098

[3]Szwarc W., Grosso A., Della Croce F. Algorithmic paradoxes of the single machine total tardiness problem // Journal of Scheduling. – 2001. – V.4. – P. 93–104.

t_0, с которого прибор готов начать обслуживание требований. Все требования поступают на обслуживание одновременно в момент времени t_0. Расписание обслуживания требований π строится с момента времени t_0 и однозначно задается перестановкой элементов множества N.

Требуется построить расписание π^* обслуживания требований множества N, при котором достигается минимум функции $F(\pi, t_0) = \sum_{j=1}^{n} \max\{0, c_j(\pi) - d_j\}$, где $c_j(\pi)$– момент завершения обслуживания требования j при расписании π. Пусть $\pi = (j_1, j_2, \ldots, j_n)$, тогда $c_{j_1}(\pi) = t_0 + p_{j_1}$ и $c_{j_k}(\pi) = c_{j_{k-1}}(\pi) + p_{j_k}$ для $k = 2, 3, \ldots, n$. Величина $T_j(\pi, t_0) = \max\{0, c_j(\pi) - d_j\}$ называется *запаздыванием* требования j при расписании π, а $F(\pi, t_0)$ - *суммарным запаздыванием* требований множества N при расписании π, построенном с момента времени t_0. В случае, когда $t_0 = 0$, будем обозначать $T_j(\pi)$ и $F(\pi)$, соответственно. Сформулированная задача является NP–трудной в обычном смысле[4].

В параграфе 1.2 описаны алгоритмы решения примеров рассматриваемой задачи, основанные на декомпозиционных свойствах задачи. Приводится точный *алгоритм A*, использующий при построении расписания правила исключения 1–4[2,3]. В процессе работы *алгоритма A* строится дерево поиска решения. Ветвление в дереве поиска возникает в ситуации, когда для очередного требования j^* с наибольшей продолжительностью обслуживания перебирается список из "подходящих позиций". После постановки требования j^* на "подходящую позицию", исходный пример разбивается на два подпримера, которые решаются аналогичным образом.

В параграфе 1.3 рассмотрен частный случай **B** исследуемой задачи и алгоритмы его решения. Приводится псевдополиномиальный *алгоритм B-1* решения частного случая **B-1** задачи, при котором выполняются ограничения: $d_1 \leq \ldots \leq d_n$, $p_1 \geq \ldots, \geq p_n$, $d_n - d_1 \leq p_n$. Далее кратко описана идея "Графической реализации" *алгоритма B-1*, с использованием которой становится возможным решать примеры с нецелочисленными параметрами и примеры "большого масштаба". Идея "Графической реализации" подробно рассматривается во второй главе данной диссертации.

[4]Du J., Leung J. Y.-T. Minimizing total tardiness on one processor is NP-hard // Math. Oper. Res. – 1990. – N.15. – P. 483–495.

Для частного случая **B-1 general**, при котором выполняются ограничения: $d_{max} - d_{min} \leq p_{min}$, приводятся новые свойства оптимальных расписаний.

Расписание $\pi = (j_1, j_2, \ldots, j_n)$ будем называть *SPT-расписанием* (short processing time), если выполняется $p_{j_k} \leq p_{j_{i+k}}$ (для $p_{j_k} = p_{j_{k+1}}$ имеем $d_{j_k} \leq d_{j_{k+1}}$), $k = 1, 2, \ldots, n-1$. Расписание π называют *LPT-расписанием* (large processing time), если $p_{j_k} \geq p_{j_{i+k}}$ (для $p_{j_k} = p_{j_{k+1}}$ выполняется $d_{j_k} \leq d_{j_{k+1}}$), $k = 1, 2, \ldots, n-1$. Расписание $\pi = (j_1, j_2, \ldots, j_n)$ будем называть *EDD-расписанием* (early due date), если $d_{j_k} \leq d_{j_{k+1}}$ (для $d_{j_k} = d_{j_{k+1}}$ выполняется $p_{j_k} \leq p_{j_{k+1}}$), $k = 1, 2, \ldots, n-1$.

Лемма 1 *Для случая* **B-1 general** *существует оптимальное расписание вида* (π_{EDD}, π_{SPT}), π_{EDD} *и* π_{SPT} *– частичные расписания, построенные по правилам EDD и SPT, соответственно. Кроме того,* $\{\pi_{EDD}\} \bigcap \{\pi_{SPT}\} = \emptyset$, $\{\pi_{EDD}\} \bigcup \{\pi_{SPT}\} = N$.

Лемма 2 *Для случая* **B-1 general** *существует оптимальное расписание вида* $(\pi_{LPT}, l, \pi_{SPT})$, π_{LPT} *и* π_{SPT} *– частичные расписания, построенные по правилам LPT и SPT, соответственно,* $\{\pi_{EDD}\} \bigcap \{\pi_{SPT}\} = \emptyset$, $l \in N, \{\pi_{EDD}\} \bigcup \{\pi_{SPT}\} \bigcup \{l\} = N$.

На основании данных двух свойств построен точный *алгоритм B-1 general* решения частного случая **B-1 general**, трудоемкость которого составляет $O(n^2 \sum p_j)$ операций.

В параграфе 1.4 рассматриваются 2 частных NP–трудных случая задачи: *Канонические DL-примеры* [4] и *Канонические LG-примеры*. Показано, что частный случай **B-1** задачи $1||\sum T_j$ является NP-трудным в обычном смысле.

Доказательство проведено сведением модифицированного примера задачи **Четно-Нечетное Разбиение** (**ЧНР**) к частному случаю **B-1**.

Задача **Четно-Нечетное Разбиение**: Задано упорядоченное множество из $2n$ положительных целых чисел $B = \{b_1, b_2, \ldots, b_{2n}\}$, $b_i > b_{i+1}$, для всех $1 \leq i \leq 2n-1$. Требуется определить, существует ли разбиение множества B на два подмножества B_1 и B_2, такое, что выполняется $\sum_{b_i \in B_1} b_i = \sum_{b_i \in B_2} b_i$ и для каждой пары чисел $\{b_{2i-1}, b_{2i}\}$, $i = 1, 2, \ldots, n$, подмножество B_1 (следовательно, и B_2) содержит в точности один элемент из пары $\{b_{2i-1}, b_{2i}\}$.

Обозначим $\delta_i = b_{2i-1} - b_{2i},\ i = 1, \ldots, n$, $\delta = \sum_{i=1}^{n} \delta_i$.

Построим модифицированный пример задачи **ЧНР**.

$$\begin{cases} a_{2n} = M + b, \\ a_{2i} = a_{2i+2} + b,\ i = n-1, \ldots, 1, \\ a_{2i-1} = a_{2i} + \delta_i,\ i = n, \ldots, 1, \end{cases} \tag{1}$$

где $b \gg n\delta,\ M \geq n^3 b$.

Очевидно выполняется $a_i > a_{i+1},\ \forall i = 1, 2, \ldots, 2n-1$, кроме того $\delta_i = b_{2i-1} - b_{2i} = a_{2i-1} - a_{2i},\ i = 1, \ldots, n$.

Лемма 3 *Исходный пример* **ЧНР** *имеет решение "ДА" тогда и только тогда, когда модифицированный пример* **ЧНР** *имеет решение "ДА".*

Приведем полиномиальную схему сведения модифицированного примера **ЧНР** к частному случаю **B-1**.

Количество требований в конструируемом примере равно $2n + 1$. Требования обозначим следующим образом:

$$V_1, V_2, V_3, V_4, \ldots, V_{2i-1}, V_{2i}, \ldots, V_{2n-1}, V_{2n}, V_{2n+1},$$

$N = \{1, 2, \ldots, 2n, 2n+1\}$. Для упрощения будем использовать следующие обозначения параметров требований $p_{V_i} = p_i,\ d_{V_i} = d_i,\ T_{V_i} = T_i,\ c_{V_i} = c_i,\ i = 1, \ldots, 2n+1$. Пример, удовлетворяющий следующим ограничениям, будем называть *Каноническим LG-примером.*

$$\begin{cases} p_1 > p_2 > \ldots > p_{2n+1}, & (2.1) \\ d_1 < d_2 < \ldots < d_{2n+1}, & (2.2) \\ d_{2n+1} - d_1 < p_{2n+1}, & (2.3) \\ p_{2n+1} = M = n^3 b, & (2.4) \\ p_{2n} = p_{2n+1} + b = a_{2n}, & (2.5) \\ p_{2i} = p_{2i+2} + b = a_{2i},\ i = n-1, \ldots, 1, & (2.6) \\ p_{2i-1} = p_{2i} + \delta_i = a_{2i-1},\ i = n, \ldots, 1, & (2.7) \\ d_{2n+1} = \sum_{i:=1}^{n} p_{2i} + p_{2n+1} + \frac{1}{2}\delta, & (2.8) \\ d_{2n} = d_{2n+1} - \delta, & (2.9) \\ d_{2i} = d_{2i+2} - (n-i)b + \delta,\ i = n-1, \ldots, 1, & (2.10) \\ d_{2i-1} = d_{2i} - (n-i)\delta_i - \varepsilon\delta_i,\ i = n, \ldots, 1, & (2.11) \end{cases} \tag{2}$$

где $b = n^2\delta$, $0 < \varepsilon < \frac{\min_i \delta_i}{\max_i \delta_i}$.

Лемма 4 *Для случая (2) при любом расписании количество запаздывающих требований равно или n, или $n+1$.*

Расписание вида

$$(V_{1,1}, V_{2,1}, \ldots, V_{i,1}, \ldots, V_{n,1}, V_{2n+1}, V_{n,2}, \ldots, V_{i,2}, \ldots, V_{2,2}, V_{1,2})$$

будем называть *Каноническим LG-расписанием*, где $\{V_{i,1}, V_{i,2}\} = \{V_{2i-1}, V_{2i}\}$, $i = 1, 2, \ldots, n$.

Теорема 1 *Для случая (2) все оптимальные расписания являются Каноническими или могут быть преобразованы к Каноническим расписаниям применением правила EDD к $(n+1)$ требованиям, обслуживаемым вначале расписания.*

Теорема 2 *Решение примера* **ЧНР** *будет "ДА" тогда и только тогда, когда при оптимальном Каноническом расписании $c_{2n+1}(\pi) = d_{2n+1}$.*

В параграфе 1.5 показано, что точный *алгоритм A*, использующий при построении расписания правила исключения 1-4[3] имеет экспоненциальную трудоемкость для некоторых частных случаев задачи $1 \mid\mid \sum T_j$.

Известно, что алгоритм, приведенный в работе Шварца и др.[3] имеет трудоемкость, не превосходящую $O(n^4 \sum p_j)$, но более точной оценки трудоемкости не получено. По результатам проведенных экспериментов с использованием классической схемы генерации примеров Поттса и ван Вассенхова предложенный в работе Шварца и др. алгоритм позволяет решать примеры размерностью до $n = 600$ требований. Нами показано, что для двух частных случаев задачи, для которых алгоритм имеет экспоненциальную трудоемкость, возможность решать примеры размерностью $n = 600$ представляется не реальной.

Схема работы алгоритма, приведенного в работе Шварца и др., совпадает со схемой работы *алгоритма A*. Как было сказано, в процессе работы *алгоритма A* конструируется дерево поиска решения.

Теорема 3 *Для случая* SD *алгоритмы, использующие только следующие способы сокращения перебора: Правила исключения 1-4, использование* E_j *и* L_j*, построение модифицированного примера, имеют трудоемкость не меньше* $O(n2^{\frac{n-1}{3}-1})$ *операций.*

Теорема 4 *Для случая* BF *алгоритмы, использующие только следующие правила сокращения перебора: правила исключения 1-4, использование* E_j *и* L_j*, построение модифицированного примера, имеют трудоемкость* $O(n2^{n/2})$*.*

Случай *SD* является подслучаем *Канонических DL-примеров*, а случай *BF* – подслучаем случая **B-1**, при котором количество запаздывающих требований при любом расписаний постоянно. Для случая *BF* существует точный полиномиальный алгоритм решения.

В параграфе 1.6 приводятся *Гибридный алгоритм* решения задачи и результаты сравнительного анализа его эффективности и эффективности метаэвристического *алгоритма "Муравьиные колонии".*

Гибридный алгоритм основан на точном *алгоритме A* и метаэвристическом *алгоритме "Муравьиные колонии"* . В *Гибридном алгоритме* используется дерево поиска решения, конструируемое *алгоритмом A*, но при этом перебираются не все подходящие позиции для очередного требования j^*, а только одна позиция, выбранная из множества подходящих согласно накопленной статистике о том, насколько хорошим было "поставить" требование j^* на данную позицию при предыдущих запусках алгоритма. Идея "накопления статистики" о "качестве" выбора той или иной позиции и многократный запуск алгоритма – основа метаэвристического *алгоритма "Муравьиные колонии"* .

В этом же параграфе приводятся результаты экспериментального сравнения эффективности *Гибридного алгоритма* и *алгоритма "Муравьиные колонии"* . По результатам экспериментов, *Гибридный алгоритм* существенно эффективнее *алгоритма "Муравьиные колонии"* на тестовых примерах Поттса и ван Вассенхова. Для 99.5% примеров *Гибридным алгоритмом* найдено точное решение. Относительная погрешность не превосходит 0.5 %. Среднее необходимое "количество муравьев" не превосходит 5.

На "трудных" примерах случая **B-1** *Гибридный алгоритм* уступает в точности *алгоритму "Муравьиные колонии"* , но находит точное решение более чем для 99% примеров. Относительная погрешность не превосходит 0.01 %.

Для NP-трудных *Канонических DL-примеров* "хорошая эффективность" обоих алгоритмов достигается за счет локального поиска, когда построенное расписание "улучшается" за счет попарных перестановок требований.

Во **второй главе** приводятся два алгоритма решения классических NP–трудных задач "Разбиение" и "Ранец", основанные на идее *Графического алгоритма*, описанного в первой главе и применяемого для решения задачи минимизация суммарного запаздывания. Приводятся результаты экспериментальной оценки трудоемкости построенных алгоритмов.

Будем рассматривать следующие формулировки задач.

Задача Разбиение (в оптимизационной постановке). Задано упорядоченное множество из n положительных целых чисел $B = \{b_1, b_2, \ldots, b_n\}$. Требуется разбить множество B, на два подмножества B_1 и B_2, так чтобы минимизировать значение

$$|\sum_{b_i \in B_1} b_i - \sum_{b_i \in B_2} b_i|.$$

Задачу "Ранец" мы можем представить в виде задачи целочисленного линейного программирования с булевыми переменными.

$$\begin{cases} f(x) = \sum_{i:=1}^{n} c_i x_i \longrightarrow \max \\ \sum_{i:=1}^{n} a_i x_i \leq A, \\ x_i \in \{0,1\},\ i = 1, \ldots, n. \end{cases} \tag{3}$$

Предлагаемые в данной главе алгоритмы являются "графической" реализацией алгоритмов динамического программирования, при которой рассматриваются не все целочисленные точки из некоторого интервала, а лишь точки, в которых изменяется значение целевой функции. На каждом шаге графического алгоритма происходит "сдвиг" функции, полученной на предыдущем шаге.

В параграфе 2.1 описан известный алгоритм динамического программирования[5] для задачи "Ранец" трудоемкости $O(nA)$ операций.

В параграфе 2.2 представлен *Графический алгоритм* решения задачи "Ранец". *Графический алгоритм* имеет ряд преимуществ перед алгоритмом динамического программирования. С помощью *Графического алгоритма* можно решать примеры с нецелочисленными параметрами ($a_i \notin Z,\ i = 1, \ldots, n,\ A \notin Z$), примеры большого масштаба. Известно, что при увеличении значений $A,\ a_i,\ i = 1, \ldots, n$, в k раз трудоемкость алгоритма динамического программирования основанного на принципе оптимальности Беллмана возрастет так же в k раз (особенно если исключить возможность обратного масштабирования небольшим изменением параметров). В *Графическом алгоритме* увеличение значений исходных параметров в k раз не окажет влияние на трудоемкость. В *Графическом алгоритме* так же учитываются некоторые "внутренние свойства задачи". К примеру, предметы с минимальным соотношением c_i/a_i могут не оказывать влияние на решение.

Трудоемкость *Графического алгоритма* обсуждается в параграфе 2.3. Трудоемкость *Графического алгоритма* для примеров с целочисленными параметрами не превосходит $O(nA)$, в силу того, что на каждом шаге $i = 1, \ldots, n$ рассматривается не более A точек. В работе[6] предложен альтернативный алгоритм динамического программирования для задачи о ранце с трудоемкостью $O(nf_{max})$ операций. Трудоемкость *Графического алгоритма* не превосходит $O(n \min\{A, f_{max}\})$ операций.

В параграфе 2.4 представлен *Графический алгоритм* для задачи "Разбиение". Трудоемкость *Графического алгоритма* не превосходит $O(n \sum_{b_j \in B} b_j)$ для целочисленных примеров, что соответствует трудоемкости алгоритма динамического программирования для задачи "Разбиение", представленного в работе Гэри и Джонсона.

Необходимо заметить, что существует класс целочисленных примеров, на котором количество рассматриваемых в *Графическом алгоритме*

[5]Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование: теория и вычислительные алгоритмы.// М.: Физматлит, 2002. – 240 с.

[6]Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность.// М.: Мир, 1985, – 512 с.

точек t от шага к шагу растет экспоненциальным образом. Например: $B = \{b_1, b_2, \ldots, b_n\} = \{M, M-1, M-2, \ldots, 1, 1, \ldots, 1\}$, где M – достаточно большое число, а сумма единиц в примере равна $M(M+1)/2$;

В параграфе 2.6 представлены результаты экспериментального исследования трудоемкости *Графического алгоритма* для задачи "Разбиение".

В первой серии экспериментов предпринята попытка оценить количество точек t, рассматриваемых на каждом шаге алгоритма и выяснить, превосходит ли количество точек n^2 или $b_{max} = \max_{b_j \in B} b_j$.

В ходе поставленного эксперимента рассматривались примеры задачи "Разбиение", размерностью $n = 4, 5, \ldots, 10$. Для каждого значения n перебирались все примеры с целочисленными параметрами, для которых: $30 \geq b_1 \geq b_2 \geq \ldots \geq b_n \geq 1$. Для каждого примера с помощью *Графического алгоритма* подсчитывалось количество рассмотренных точек.

По результатам экспериментов, для всех рассмотренных примеров количество рассмотренных точек t на каждом шаге не превосходит n^2. Для 99% примеров количество рассмотренных алгоритмом точек не превосходит nb_{max}.

В следующей серии экспериментов проведен сравнительный анализ эффективности *алгоритма Balsub*[7] и *Графического алгоритма*.

Трудоемкость алгоритма динамического *программирования Balsub*, для целочисленных примеров составляет $O(nb_{max})$ операций и считается наилучшей теоретической оценкой для данной задачи

Рассматривались все примеры размерности $n = 4, 5, \ldots, 10$, с целочисленными параметрами, для которых $40 \geq b_1 \geq b_2 \geq \ldots \geq b_n \geq 1$.

Во всех рассмотренных примерах трудоемкость *алгоритма Balsub* была больше, чем у *Графического алгоритма*, за исключением 38 примеров для $n = 4$ и 2 примеров для $n = 5$.

Необходимо также заметить, что *Графический алгоритм* находит решение и для примеров с отрицательными параметрами, что не могут алгоритмы, основанные на методе динамического программирования.

[7]Keller H., Pferschy U., Pisinger D. Knapsack problems// Springer, 2004, – 546 p.

В **третьей главе** рассматриваются задачи построения расписания на быстродействие (минимизация общего момента окончания обслуживания всего множества требований) с ресурсными ограничениями и отношениями предшествования. Данные задачи достаточно часто возникают на практике. Например, при строительстве того или иного объекта на разных стадиях строительства необходимо разное количество трудовых ресурсов, строительной техники, материалов и т.п. Между отдельными стадиями строительства существуют отношения предшествования обусловленные технологией строительства. Требуется составить расписание выполнения работ, при котором не нарушаются отношения предшествования, ресурсные ограничения, и при этом срок окончания строительства минимален. Для решения данной задачи используются алгоритмы, основанные на методе ветвей и границ. В диссертации проведен анализ существующих алгоритмов вычисления нижних и верхних оценок целевой функции. Показано, что для задачи "Минимизация времени выполнения проекта" (RCPSP) известные нижние оценки имеют относительную погрешность порядка размерности задачи или их нахождение является NP-трудной задачей. Выдвинута гипотеза, что оптимальные значения целевой функции для задачи RCPSP с прерываниями в обслуживании требований и без прерываний отличаются не более чем в 2 раза. Доказательство (или опровержение) данной гипотезы может оказать существенное влияние на построение эффективных алгоритмов решения задачи RCPSP.

Показано, что любой пример задачи RCPSP можно преобразовать в эквивалентный пример, граф отношений предшествования которого будет планарным, причем "мощность" (сумма количества вершин и ребер) графа не увеличится. Некоторые полученные результаты были доведены до практической реализации в программном продукте "1С:УСО" и получили хорошее экспериментальное подтверждение.

В параграфе 3.1 представлена постановка задачи "Минимизация времени выполнения проекта"(RCPSP).

Дано множество требований $N = \{1, \ldots, n\}$ и K возобновляемых ресурсов $k = 1, \ldots, K$. В каждый момент времени t доступно Q_k единиц ресурса k. Заданы продолжительности обслуживания $p_i \in \mathbb{Z}^+$ каждого требования $i = 1, \ldots, n$. Во время обслуживания требования i требуется $q_{ik} \leq Q_k$ единиц ресурса $k = 1, \ldots, K$. После завершения

обслуживания требования, освобожденные ресурсы в полном объеме могут быть мгновенно назначены на выполнение других требований.

Между некоторыми парами требований заданы ограничения предшествования: $i \rightarrow j$ означает, что обслуживание требования j начинается не раньше окончания требования i.

Обслуживание требований начинается в момент времени $t = 0$. Прерывание при обслуживании требований запрещены.

Необходимо определить моменты времени начала обслуживания требований S_i, $i = 1, \ldots, n$, так, чтобы минимизировать момент окончания выполнения всего проекта $C_{max} = \max_{i=1,\ldots,n}\{C_i\}, C_i = S_i + p_i$. При этом должны быть соблюдены следующие ограничения:

1. в каждый момент времени $t \in [0, C_{max})$ выполняется $\sum_{i:=1}^{n} q_{ik}\varphi_i(t) \leq Q_k$, $k = 1, \ldots, K$, где $\varphi_i(t) = 1$, если требование i обслуживается в момент времени t и $\varphi_i(t) = 0$, в противном случае. То есть требования в процессе своего выполнения должны быть полностью обеспечены ресурсами;

2. не нарушаются отношения предшествования, т. е. $S_i + p_i \leq S_j$, если $i \rightarrow j$, $i, j \in N$.

Данную задачу будем называть задачей *минимизация времени выполнения проекта* и обозначать $RCPSP$ (Resource-constrained Project Scheduling Problem, как принято называть в англоязычной литературе). К данной задаче за полиномиальное время сводится NP-трудная задача о многомерном ранце.

Решение задачи RCPSP представляется в виде набора моментов начала обслуживания требований $S = (S_1, \ldots, S_n)$. Решение, удовлетворяющее ресурсным ограничениям и ограничениям предшествования, будем называть допустимым.

Можем представить структуру проекта как *требования-в-вершинах* ориентированного графа $G = (V, A)$, где каждой вершине из $V = \{1, \ldots, n\}$ соответствует некоторое требование множества $N = \{1, \ldots, n\}$, а множество дуг $A = \{(i, j) | i, j \in V; i \rightarrow j\}$ соответствует ограничениям предшествования. Очевидно, что допустимое решение существует только когда граф предшествования ацикличен.

Обычно в рассмотрение вводят два фиктивных требования 0 и $n+1$ с продолжительностями обслуживания $p_0 = p_{n+1} = 0$. Отношения предшествования $0 \to j \to n+1,\ j = 1, \ldots, n,\ q_{0k} = q_{n+1k} = 0, k = 1, \ldots, K$.

Будем обозначать через UB верхнюю границу для оптимального значения C_{max}. К примеру, можем принять $UB = \sum_{i:=1}^{n} p_i$.

Для каждого требования $i \in N$ определим временно́е окно $[r_i, d_i]$, в котором требование i должно быть обслужено при любом допустимом расписании S:

$$r_0 = 0,\ r_i = \max_{j|(j,i)\in A} \{r_j + p_j\},\ i = 1, \ldots, n+1;$$

$$d_{n+1} = UB,\ d_i = \min_{j|(i,j)\in A} d_j - p_j,\ i = n, n-1, \ldots, 0.$$

Обозначим C^*_{max} – оптимальное значение целевой функции. Пусть $C_{max}(S^*)$ – значение целевой функции при оптимальном расписании S^*. Также будем обозначать $C^*_{max}(pmtn.)$ – оптимальное значение целевой функции для задачи RCPSP с прерываниями при обслуживании требований.

В параграфе 3.2 приведены результаты анализа известных нижних оценок функционала для задачи RCPSP и представлена новая нижняя оценка. Показано, что рассмотренные нижние оценки (LB_0, LB_1, LB_s) не эффективны, либо их (LB_{LG}, LB_M) нахождение является в общем случае NP-трудной задачей.

Путь соединяющий вершины 0 и $n+1$ в сетевом графике и имеющий наибольшую длину называется критическим путем (длина складывается из продолжительности обслуживания требований, входящих в этот путь). Длина критического пути будет равна $C_{max}(S^*)$, когда нет ресурсных ограничений. То есть длина критического пути является нижней оценкой для $C_{max}(S^*)$. Эту нижнюю оценку принято обозначать LB_0.

Лемма 5 *Существует пример RCPSP, для которого* $\frac{C_{max}(S^*)}{LB_0} = n$.

Другая нижняя оценка LB_1 может быть найдена за $O(nK)$ операций при рассмотрении каждого ресурса отдельно. Назовем величину

$\sum_{i=1}^{n} q_{ik}p_i$ суммарной загрузкой ресурса k.

Очевидно, что $\sum_{i=1}^{n} q_{ik}p_i \leq Q_k \cdot C_{max}(S^*), k = 1, \ldots, n$. Тогда значение

$$LB_1 = \max_{k=1}^{K} \lceil \sum_{i=1}^{n} q_{ik}p_i/Q_k \rceil$$

является нижней оценкой для $C_{max}(S^*)$.

Лемма 6 *Существует пример RCPSP, для которого* $C_{max}(S^*) - LB_1 = \sum_{i=1}^{n} p_i - 1$.

Обозначим множество требований, принадлежащих критическому пути через CP. Для каждого требования $i \notin CP$ обозначим через e_i максимальную длину интервала, входящего в $[r_i, d_i]$, в котором требование i может обслуживаться параллельно с требованиями критического пути без нарушения ограничений на ресурсы. Если выполняется $e_i < p_i$, то не существует допустимого расписания, при котором $C_{max}(S^*) = LB_0$. Тогда значение

$$LB_S = LB_0 + \max_{i \notin CP}\{max\{p_i - e_i, 0\}\}$$

также является нижней оценкой для $C_{max}(S^*)$.

Лемма 7 *Существует пример RCPSP, для которого* $\frac{C_{max}(S^*)}{LB_S} = n/2$.

В лучшем известном на данный момент алгоритме используется нижняя оценка LB_M[8].

Лемма 8 *Вычисление оценки* LB_M *является NP–трудной задачей.*

Лемма 9 *Существует пример RCPSP, для которого* $\frac{C_{max}(S^*)}{LB_M} \approx 2$.

В диссертационной работе выдвинута гипотеза, что значения целевой функции для задачи RCPSP на быстродействие с прерываниями обслуживания требований и без прерываний отличаются не более чем

[8]Mingozzi A., Maniezzo V., Ricciardelli S., Bianco L. An exact alforithm for project scheduling with recource constraints based on new mathematical formulation// Management Science, 1998, – V44 – P. 714–729.

в 2 раза. Набросок доказательства гипотезы, а так же рассуждения о ее значении представлены в параграфе 3.3.

В этом же параграфе рассматривается NP–трудный частный случай задачи RCPSP с одним кумулятивным ресурсом мощности Q_1 и пустым графом отношений предшествования. Для данного частного случая доказана следующая теорема.

Теорема 5 $2 \cdot C^*_{max}(pmtn.) > C^*_{max} - p_{max}$, *где* $p_{max} = max_{j \in N} p_j$.

В параграфе 3.4 приводятся свойства частного случая задачи RCPSP – задачи построения расписания для параллельных машин (PMS). Представлены новые свойства частного случая задачи PMS, когда продолжительности обслуживания всех требований равны, т.е. $p_j = p$.

Частный случай задачи RCPSP с одним кумулятивным ресурсом $Q_1 = m$ и $p_j = p,\ j = 1, \ldots, n$, рассматривается в параграфе 3.5.

В параграфе 3.6 представлены общие свойства задач построения расписания с отношениями предшествования, исследуются свойства планарности сетевого графика, и решение "обратного" примера. Описан алгоритм укладки сетевого графика на плоскости без пересечения ребер.

Два примера задачи RCPSP будем называть *эквивалентными*, если первый пример сводится ко второму введением "пустых" требований (с нулевой продолжительностью) и удалением "лишних" связей, так что если между вершинами i и j был хотя бы один путь, то путь между ними останется. Если пути не было, то он и не появится. Значения целевой функции для эквивалентных примеров равны.

Теорема 6 *Для любого примера задачи RCPSP с* n *требованиями и* e *связями в графе* $G(V, E)$ *отношений предшествования существует эквивалентный пример с планарным графом* $G'(V', E')$, n' *требованиями и* e' *связями, причем* $n + e \geq n' + e'$.

Доказательство теоремы основано на следующих леммах.

Лемма 10 *Если в графе отношений предшествования встречается подграф* $K_{3,3}$, *то можно его преобразовать, удаляя лишние дуги и добавляя "пустые" вершины.*

Лемма 11 *Если в графе отношений предшествования встречается подграф K_5, то можно его преобразовать, удаляя лишние дуги.*

В параграфе 3.7 представлен метаэвристический *алгоритм ACO* (основанный на методе "Муравьиные колонии") решения задачи RCP-SP, применяемый в рамках программного продукта "1С:Управление строительной организацией" (1С:УСО).

В феврале 2007 года в продажу поступил программный продукт 1С:УСО. Цель создания программы – комплексная автоматизация основных подразделений строительной компании. В 1С:УСО в единую систему объединены модули: бухгалтерский учет, управление финансами, закупками, запасами и т.д. Ядро системы – модуль "управление строительным производством", в котором реализованы базовые принципы управления проектами с учетом специфики строительного производства. К маю 2008 года внедрение 1С:УСО было осуществлено более чем на 140 предприятиях с численностью автоматизированных рабочих мест, превышающим 2400.

В **заключении** изложены основные результаты и выводы работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ ДИССЕРТАЦИИ

1. Предложены новые точные алгоритмы решения частных случаев задачи "Минимизация суммарного запаздывания для одного прибора", а также *Гибридный алгоритм* решения общего случая задачи, для которого приведены экспериментальные оценки его эффективности.

2. Доказано, что частный случай задачи "Минимизация суммарного запаздывания для одного прибора", при котором максимальная разница директивных сроков не превосходит минимального времени обслуживания требований, является NP–трудным в обычном смысле. Доказательство проведено сведением известной задачи "Разбиение" к этому частному случаю.

3. Предложены два *Графических алгоритма* решения классических NP–трудных задач "Разбиение" и "Ранец". Доказано, что трудоемкость *Графических алгоритмов* для целочисленных

примеров не превосходит трудоемкости соответствующих алгоритмов динамического программирования, но при этом с помощью *Графических алгоритмов* можно решать примеры с нецелочисленными или большими числовыми параметрами.

4. Проведен анализ известных нижних оценок функционала задачи "Минимизация времени выполнения проекта" и предложена новая нижняя оценка. Показано, что известные оценки не эффективны, либо их нахождение является в общем случае NP-трудной задачей.

5. Выдвинута гипотеза о том, что значения целевой функции для задачи "Минимизация времени выполнения проекта" на быстродействие с прерываниями в обслуживании требований и без прерываний отличаются не более чем в 2 раза. Приведены аргументы в пользу этой гипотезы.

6. Показано, что любой пример задачи "Минимизация времени выполнения проекта" можно преобразовать в соответствующий пример, граф отношений предшествования которого будет планарным, при этом сумма числа вершин и числа ребер графа не увеличивается.

7. Ряд полученных результатов доведен до практической реализации в программном продукте "1С:Управление строительной организацией" и получил хорошее экспериментальное подтверждение.

СПИСОК РАБОТ, ОПУБЛИКОВАННЫХ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Гафаров Е.Р. Гибридный алгоритм решения задачи минимизации суммарного запаздывания для одного прибора // Информационные технологии, 2007, – №1 – С. 30–37.

2. Гафаров Е.Р., Лазарев А.А. Доказательство NP-трудности частного случая задачи минимизация суммарного запаздывания для одного прибора // Известия РАН: Теория и системы управления. – 2006. – №.3. – С. 120–128.

3. Лазарев А.А., Кварацхелия А.Г., Гафаров Е.Р. Алгоритмы решения NP-трудной проблемы минимизации суммарного запаздывания для одного прибора. // Доклады Академии Наук, 2007. – Том 412, № 6. – С. 739–742.

4. Лазарев А.А., Гафаров Е.Р. Теория расписаний. Минимизация суммарного запаздывания для одного прибора. // Научное издание, М.: Вычислительный центр им. А.А. Дороницына РАН,2006.- 134 с.

5. Лазарев А.А., Гафаров Е.Р. Теория расписаний. Исследование задач с отношениями предшествования и ресурсными ограничениями. // Научное издание, М.: Вычислительный центр им. А.А. Дороницына РАН, 2007. – 80 с.

6. Гафаров Е.Р. Алгоритмы решения проблемы $1 \mid\mid \sum T_j$ и чётно-нечётного разбиения // Материалы XLIII Международной научной студенческой конференции «Студент и научно-технический прогресс»: Математика. – Новосиб. гос. ун-т. Новосибирск, 2005. – С. 201–202.

7. Гафаров Е.Р. Гибридный алгоритм решения проблемы $1 \mid\mid \sum T_j$ // Материалы XLIV Международной научной студенческой конференции «Студент и научно-технический прогресс»: Математика. – Новосиб. гос. ун-т. Новосибирск, 2006. – С. 190–191.

8. Lazarev A., Kvaratskhelia A., Gafarov E. Algorithms for solving problems $1 \mid\mid \sum Tj$ and Even-Odd Partition. // In Book of Abstracts of XVIII International Conference «European Chapters on Combinatorial Optimization». – Minsk, 26–28.V.2005. – P. 32–33.

9. Lazarev A.A., Gafarov E.R. Graphical approach for solving combinatorial problems // Abstract Guide of International Conference on Operation Research OR 2006, Karlsruhe 6.09-8.09, Germany, P.59.

10. Lazarev A.A., Gafarov E.R. Algorithms for the single machine total tardiness problem // Abstract Guide of International Conference on Operation Research OR 2006, Karlsruhe 6.09-8.09, Germany, P.285.

11. Lazarev A.A., Gafarov E.R. Estimation of lower bounds for resources constrained project scheduling problem // Proceedings of V Moscow

International Conference on Operation Research (ORM2007), Moscow, 2007, P.236–238.

12. Гафаров Е.Р. Задачи теории расписаний. Алгоритмы и применение. // Труды 49 научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук»: Управление и прикладная математика, Москва-Долгопрудный, 2006. – С. 82–83.

13. Гафаров Е.Р. Алгоритмы решения NP-трудных задач теории расписаний и разбиения // Труды всероссийской конференции студентов, аспирантов и молодых ученых «Технологии Microsoft в теории и практике программирования», Москва, 2006. – С. 120–121.

В работах с соавторами лично Гафаровым Е.Р. сделано следующее. В работах [2,3,4] доказана NP трудность частного случая задачи "Минимизация суммарного запаздывания для одного прибора", в работе [4] определена трудоемкость известных точных алгоритмов, разработан алгоритм решения частного случая задачи B-1 general и обоснована его эффективность. В работах [1,4,7] предложен Гибридный алгоритм, а в работах [6,8,10] алгоритмы решения частных случаев задачи. На основе идеи Лазарева А.А. о сокращении перебора в алгоритмах динамического программирования Гафаровым Е.Р. разработаны алгоритмы решения задач "Разбиение" и "Ранец" [9,12]. В работах [5,11,12] проведены исследования оценок функционала для задачи "Минимизация времени выполнения проекта", предложено доказательство теоремы о планарности сетевого графика.

Выражаю признательность д.ф.-м.н. В.К. Леонтьеву, д.т.н. Сигалу И.Х., д.т.н. Буркову В.И., а также своему научному руководителю д.ф.-м.н. Лазареву А.А. за помощь в подготовке диссертации.

.

Гафаров Евгений Рашидович

АЛГОРИТМЫ РЕШЕНИЯ NP-ТРУДНЫХ ЗАДАЧ
МИНИМИЗАЦИИ СУММАРНОГО ЗАПАЗДЫВАНИЯ И
МИНИМИЗАЦИИ ВРЕМЕНИ ВЫПОЛНЕНИЯ ПРОЕКТА
И ИХ ПРИМЕНЕНИЕ В КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ.

Подписано в печать 16.05.08 Формат 60x84 1/16. Бумага офсетная. Усл. печ. л. 1,0. Тираж 90 экз. Заказ № 329

Московский физико-технический институт (государственный университет) НИЧ МФТИ 141700, Московская обл., Долгопрудный, Институтский пер., 9