

На правах рукописи

КВАРАЦХЕЛИЯ Александр Гонерович

**МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ МИНИМИЗАЦИИ
СУММАРНОГО ЗАПАЗДЫВАНИЯ ДЛЯ ОДНОГО
ПРИБОРА И ЗАДАЧИ РАЗБИЕНИЯ**

01.01.09 – дискретная математика и математическая кибернетика

Автореферат

диссертации на соискание ученой степени
кандидата физико-математических наук

Москва — 2007

Работа выполнена на кафедре экономической кибернетики Казанского государственного университета им. В.И. Ульянова–Ленина.

Научный руководитель: кандидат физико-математических наук,
доцент А.А. Лазарев

Официальные оппоненты: доктор технических наук,
профессор И.Х. Сигал,

кандидат физико-математических наук
М.А. Посыпкин

Ведущая организация: Институт проблем управления РАН, г. Москва

Защита диссертации состоится "____"_____ 2007 г. в ____ часов
на заседании диссертационного совета Д.002.017.02 при Вычислитель-
ном центре им. А.А. Дородницына Российской академии наук (119991, г.
Москва, ул. Вавилова 40, конф.зал).

С диссертацией можно ознакомиться в библиотеке ВЦ РАН.

Автореферат разослан "____"_____ 2007 г.

Ученый секретарь
диссертационного совета, д.ф.-м.н.

В.В. Рязанов

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Диссертационная работа посвящена исследованию классической NP-трудной (в обычном смысле) задачи теории расписаний — минимизации суммарного запаздывания для одного прибора (далее — задача $1 \parallel \sum T_j$) — с целью отыскания новых свойств оптимальных расписаний общего и некоторых частных случаев задачи, построения на их основе новых алгоритмов решения этой задачи и применения полученных результатов для построения алгоритма решения NP-полной (в обычном смысле) задачи РАЗБИЕНИЕ¹.

Изучаемые в теории расписаний задачи выбора очередности обслуживания имеют самый общий характер и возникают при различных видах целенаправленной деятельности, например, при календарном планировании производства, транспортных перевозок, обучения, информационно-вычислительных процессов, исследовании структуры молекулы ДНК².

В теории расписаний особое значение имеют задачи для одного прибора. Результаты, получаемые при исследовании данных задач, могут быть использованы для построения алгоритмов решения более сложных многоприборных и многостадийных задач, возникающих на практике. При этом, задачи для одного прибора могут быть использованы как для построения эвристических алгоритмов решения многоприборных и многостадийных задач, так и для получения оценок оптимального значения. Кроме того, решение некоторых задач для многих приборов можно свести к решению некоторой совокупности задач для одного прибора.

Большинство задач теории расписаний являются NP-трудными, т.е. не существует полиномиальных алгоритмов их решения при предположении, что классы $\mathcal{P}$ и $\mathcal{NP}$ не совпадают. Поэтому актуальным является получение новых свойств оптимальных расписаний и построение на их основе переборных и псевдополиномиальных алгоритмов решения, получение оценок оптимального значения целевой функции, а также нахождение новых полиномиально разрешимых частных случаев данных задач. Исследования в этом направлении и проведены в диссертации.

Цель работы заключается в исследовании задачи $1 \parallel \sum T_j$ для отыс-

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

²В зарубежной литературе данное направление известно как *DNA-sequencing*

кания новых свойств оптимальных расписаний общего и некоторых частных случаев задачи, построения на их основе новых алгоритмов решения и применения полученных результатов для некоторых NP-полных задач разбиения.

Методы исследования. При формулировке и доказательстве результатов используются методы дискретной математики и математической кибернетики, в частности понятия и утверждения теории расписаний и математического программирования. Достоверность результатов диссертации подтверждается доказательствами всех теорем и лемм, сформулированных в работе.

Теоретическая и практическая значимость. Полученные в работе результаты 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.

Апробация работы. Результаты диссертационной работы докладывались и обсуждались на VII Международном семинаре «Дискретная математика и её приложения» (Москва, МГУ, 29 января – 2 февраля 2001 г.), на XXIII Конференции молодых ученых мехмата МГУ (Москва, 9–14 апреля 2001 г.), на Международной конференции «MAPSP’01 Models and Algorithms in Planning and Scheduling Problems» (Асса, Франция, 17–22 июня 2001 г.), на XL Международной студенческой конференции «Студент и научно-технический прогресс» (Новосибирск, НГУ, 16–18 апреля 2002 г.), на XIII Международной конференции «Проблемы теоретической кибернетики» (Казань, КГУ, 27–31 мая 2002 г.), на Международной конференции «DAOR’2002 Дискретный анализ и исследование операций» (Новосибирск, 24–28 июня 2002 г.), на Международной конференции «MAPSP’03 Models and Algorithms for Planning and Scheduling Problems» (Асса, Франция, 30 марта–4 апреля 2003 г.), на Международной летней школе «Doctoral Courses in Discrete Systems and Optimization» (Гренобль, Франция, 22 июня–4 июля 2003 г.), на VIII Международном семинаре «Дискретная математика и её приложения» (Москва, МГУ, 2–6 февраля 2004 г.), на III Международной конференции «PSC’04 Parallel Computing Systems» (Колима, Мексика, 19–22 сентября 2004 г.), на VI Международной конференции «Дискретные модели в теории управляющих систем» (Москва, МГУ, 7–11 декабря 2004 г.), на XVIII Между-

народной конференции «ЕССО'05 European Chapter on Combinatorial Optimization» (Минск, Беларусь, 26–28 мая 2005 г.), на итоговых научных конференциях Казанского государственного университета и научных семинарах кафедры экономической кибернетики Казанского государственного университета.

Публикации. Основные результаты диссертации изложены в одиннадцати публикациях, в том числе две из них в центральной печати, шесть в материалах и три в тезисах докладов на международных и российских конференциях. Список публикаций автора по теме диссертации приведен в конце автореферата.

Структура и объем работы. Диссертационная работа состоит из введения, трех глав, заключения, указателя основных обозначений и списка литературы (86 наименований). Общий объем диссертации – 126 страниц.

СОДЕРЖАНИЕ РАБОТЫ

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

В первой главе рассматриваются комбинаторные свойства задачи минимизации суммарного запаздывания для одного прибора $1 \parallel \sum T_j$. Получены свойства оптимальных расписаний, на основе которых построены алгоритмы решения задачи. Даны некоторые оценки оптимального значения целевой функции.

В параграфе 1.1 приводится постановка рассматриваемой задачи, вводятся необходимые понятия и обозначения.

Необходимо обслужить n требований на одном приборе. Прерывания требований, искусственные простои прибора при обслуживании и обслуживание более одного требования в любой момент времени запрещены. Требования пронумерованы числами $1, 2, \dots, n$. Множество индексов $N = \{1, 2, \dots, n\}$ назовем *множеством требований*.

Для требования $j \in N$ заданы следующие *параметры*: *продолжительность обслуживания* $p_j > 0$, $p_j \in \mathbb{Z}^+$, и *директивный срок окончания обслуживания* d_j . Задан *момент начала обслуживания* t_0 , с которого прибор готов начать обслуживание требований. Все требования поступают на обслуживание одновременно в момент времени t_0 .

Расписание обслуживания требований множества N задается кусочно-постоянной непрерывной слева функцией $s : \mathbb{R} \rightarrow \{0, 1, 2, \dots, n\}$. Если $s(t) = 0$, то в момент времени t прибор простаивает; если $s(t) = j$, $j \in N$, то в момент времени t прибор обслуживает требование j . Поскольку в рассматриваемой задаче требования поступают на обслуживание одновременно, обслуживаются без прерываний и искусственных простоев прибора, то расписание однозначно задается перестановкой π элементов множества N . Далее понятия расписания и перестановки элементов множества требований будем отождествлять и называть перестановку π *расписанием*.

Для обозначения *индивидуального примера* I задачи с заданными множеством требований N , продолжительностями обслуживания p_j , директивными сроками d_j и моментом начала обслуживания t_0 будем использовать запись $I = \langle \{p_j, d_j\}_{j \in N}, t_0 \rangle$. Краткая форма этой записи $I = \{N, t_0\}$ применяется в случае, когда параметры требований p_j, d_j фиксированы (однозначно определены контекстом). Пусть $\Pi(I)$ — множество всех возможных $n!$ расписаний для примера I .

Через $(i \rightarrow j)_\pi$ будем обозначать взаимный порядок обслуживания двух требований i и j при расписании π ; данная запись означает то, что требование i обслуживается ранее требования j . Основной характеристикой обслуживания требования $j \in N$ при расписании π является *момент окончания обслуживания* $c_j(\pi)$, равный $t_0 + \sum_{i:(i \rightarrow j)_\pi} p_i + p_j$. Величину $T_j(\pi) = \max\{0, c_j(\pi) - d_j\}$ называют *запаздыванием* требования j при расписании π , а величину $F(\pi) = \sum_{j \in N} T_j(\pi)$ — *суммарным запаздыванием* требований при расписании π .

Задача минимизации суммарного запаздывания для одного прибора заключается в построении оптимального расписания π^* , при котором выполняется неравенство $F(\pi^*) \leq F(\pi)$ для всех $\pi \in \Pi(I)$.

Пусть $\Pi^*(I)$ — множество оптимальных расписаний для примера I . Сформулированная задача является NP-трудной в обычном смысле³.

В параграфе 1.2 рассматриваются алгоритмы решения примеров рассматриваемой задачи, основанные на декомпозиционных свойствах задачи. Такие свойства позволяют рассматривать процесс нахождения оптимального расписания как процесс разбиения примера на подпримеры,

³Du J., Leung J. Y.-T. Minimizing total tardiness on one processor is NP-hard // Math. Oper. Res. — 1990. — N.15. — P. 483–495.

которые могут быть решены независимо друг от друга. Одно из первых декомпозиционных свойств задачи было получено Лаулером, который предложил алгоритм решения примеров задачи с оценкой трудоемкости $O(n^4 \sum p_j)$ операций⁴.

Перенумеруем требования множества N в порядке неубывания директивных сроков $d_1 \leq \dots \leq d_n$ (при этом, если $d_j = d_{j+1}$, то $p_j \leq p_{j+1}$, $j = 1, 2, \dots, n-1$). Пусть j^* будет требованием с максимальной продолжительностью (если таких требований несколько, то выберем требование с наибольшим номером), т.е. $j^* = \max\{j \in N : p_j = \max_{i \in N} p_i\}$. Пусть $S_k = t_0 + \sum_{j=1}^k p_j$ ($k = 1, 2, \dots, n$).

Через $L(I)$ обозначим множество всех таких индексов $k \geq j^*$, что выполнены неравенства $d_j + p_j < S_k$ ($j = j^* + 1, \dots, k$) и $S_k \leq d_{k+1}$, доопределив $d_{n+1} = +\infty$. Множество $L(I)$ будем называть *множеством подходящих позиций* для требования j^* .

Теорема 1 *Для любого примера I множество подходящих позиций $L(I)$ для требования j^* не пусто.*

Декомпозиционные свойства задачи выражены следующей теоремой.

Теорема 2⁵ *Для любого примера I существует оптимальное расписание $\pi^* \in \Pi^*(I)$, при котором требование j^* обслуживается на месте $k \in L(I)$. При этом для подходящего места k выполняется*

$$(j \rightarrow j^*)_{\pi^*} \text{ для } j \in \{1, 2, \dots, k\} \setminus \{j^*\} \text{ и } (j^* \rightarrow j)_{\pi^*} \text{ для } j \in \{k+1, \dots, n\}.$$

Приведем схему решения произвольного примера задачи $1 \parallel \sum T_j$, на основе приведенных декомпозиционных свойств.

Рассмотрим пример $I = \{N, t_0\}$. Для него найдем требование j^* и построим множество подходящих позиций $L(I)$. Пусть $m = |L(I)|$. Разобьем пример I на $2m$ подпримеров следующим образом. Для каждого $k \in L(I)$ построим два подпримера $I'_k = \{N'_k, t'_k\}$ и $I''_k = \{N''_k, t''_k\}$, где $N'_k = \{1, 2, \dots, k\} \setminus \{j^*\}$, $t'_k = t_0$ и $N''_k = \{k+1, \dots, n\}$, $t''_k = S_k$. Решим построенные примеры, т.е. построим оптимальные расписания π'_k и

⁴Lawler E.L. A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness // Ann. Discrete Math.- 1977.- No 1.- P. 331–342.

⁵Лазарев А.А. Алгоритмы декомпозиционного типа решения задачи минимизации суммарного запаздывания // Исследования по прикладной математике. – Казань: Изд-во Казан. гос. ун-та, 1990. – Вып. 17. – С. 71–78.

π_k'' для примеров I_k' и I_k'' ($k \in L(I)$). Пусть $\pi_k = (\pi_k', j^*, \pi_k'')$ ($k \in L(I)$). Тогда, согласно теореме 2 расписание π_{k^*} , где $k^* = \arg \min_{k \in L(I)} F(\pi_k)$, будет оптимальным расписанием для примера I . Построение примеров I_k' и I_k'' и нахождение для них оптимальных расписаний будем называть *декомпозицией* примера I по подходящей позиции k .

Идея алгоритма решения произвольного примера задачи с помощью декомпозиционных свойств (далее, алгоритм A) основана на обходе дерева подпримеров "в глубину" с использованием рекурсивного вызова описанной выше процедуры декомпозиции примеров на подпримеры. Данная процедура принимает в качестве параметров множество требований и момент начала обслуживания, и результатом её работы является построение оптимального расписания для данного примера (подпримера).

Отметим, что если в двух различных вершинах дерева подпримеров будет находиться один и тот же подпример (одинаковы множества требований и моменты начала обслуживания), то алгоритм A построит оптимальное расписание для данного подпримера дважды. Чтобы избежать данного недостатка, предлагается использовать алгоритм SBA , процесс выполнения которого разбивается на два этапа. На первом этапе осуществляется декомпозиция исходного примера в список всех его подпримеров вплоть до получения подпримеров из одного требования. Список подпримеров организован как хэш-таблица. Значение хэш-функции, соответствующее подпримеру, вычисляется с использованием момента начала обслуживания требований данного примера. При формировании данного списка подпример включается в список только в том случае, если в списке нет аналогичного примера. На втором этапе осуществляется "сборка" оптимального расписания на основе построенного списка подпримеров.

В заключение параграфа приводятся результаты экспериментального сравнения времени выполнения алгоритмов A и SBA на некотором классе тестовых примеров.

В параграфе 1.3 рассматриваются свойства оптимальных расписаний и предлагается алгоритм решения примеров I в случае, когда при любом расписании $\pi \in \Pi(I)$ запаздывает одно и то же количество требований m ($0 \leq m \leq n$). При этом запаздывающие требования упорядочены в конце расписания π (т.е. расписание π можно представить в виде $\pi = (\pi_1, \pi_2)$, где подрасписание π_2 содержит все m запаздывающих требований).

Через $D(\pi)$ обозначим множество запаздывающих расписаний при расписании π . В рассматриваемом случае верны следующие утверждения.

Лемма 1 *При любом оптимальном расписании π^* требования множества $D(\pi^*)$ обслуживаются в SPT порядке⁶.*

Далее, перенумеруем требования примера I в порядке невозрастания продолжительностей обслуживания $p_1 \geq p_2 \geq \dots \geq p_n$, при этом, если $p_j = p_{j+1}$ ($j = 1, \dots, n-1$), то $d_j \geq d_{j+1}$.

Лемма 2 *Существует оптимальное расписание π^* такое, что если для некоторого $k \in N$ выполняется $k \in D(\pi^*)$, то $(j \rightarrow k)_{\pi^*}$ для всех требований $j \in \{k+1, \dots, n\}$.*

Свойства оптимальных расписаний в рассматриваемом случае, выраженные в леммах 1 и 2, позволяют построить алгоритм нахождения оптимального расписания за $O(n^3)$ операций (далее, алгоритм FA). Построение оптимального расписания алгоритмом FA сводится к обходу дерева, каждая ветка которого соответствует некоторому порядку обслуживания запаздывающих требований. Количество вершин этого дерева экспоненциально зависит от величин m и n , однако полученные свойства позволяют выполнить определенную "склейку" вершин дерева так, что в нем остается только полиномиальное количество вершин.

В параграфе 1.4 приводятся две оценки оптимального значения суммарного запаздывания, основанные на свойствах SPT-расписаний. При этом полученные оценки позволяют получить абсолютную погрешность значения суммарного запаздывания при SPT-расписании⁷.

Перенумеруем требования множества N в порядке неубывания продолжительностей обслуживания, то есть $p_1 \leq p_2 \leq \dots \leq p_n$. SPT-расписание $(1, 2, \dots, n)$ будем обозначать через π_{spt} . Рассмотрим произвольный пример задачи $I = \langle \{p_j, d_j\}_{j \in N}, t_0 \rangle$ с оптимальным значением целевой функции $F^*(I)$. Пусть $d_{\min} = \min_{j \in N} d_j$ и $d_{\max} = \max_{j \in N} d_j$.

⁶SPT – Shortest Processing Time. Расписание π называется SPT-расписанием, если требования упорядочены при расписании π в порядке неубывания продолжительностей обслуживания.

⁷Относительная погрешность значения суммарного запаздывания при EDD-расписании (расписании, требования при котором упорядочены в порядке невозрастания значений директивных сроков) установлена Лаулером в 1977 г.

Выберем величины C и δ такими, что

$$d_{\min} \leq C \leq d_{\max} \text{ и } \delta = \max\{d_{\max} - C, C - d_{\min}\}.$$

Пусть $S = t_0 + \sum_{j=1}^n p_j$, и рассмотрим пример $I' = \langle \{p_j, d_j' = C\}_{j \in N}, t_0 \rangle$. Справедлива следующая теорема.

Теорема 3 (первая оценка). Для выбранных значений C и δ верно неравенство

$$|F^*(I) - F^*(I')| \leq n\delta \left(2 - 2\frac{C}{S} + \frac{\delta}{S} \right) + \delta,$$

Отметим, что SPT-расписание π_{spt} является оптимальным расписанием для любого примера, директивные сроки которого равны некоторой константе. Следовательно, величина $F^*(I')$ равна суммарному запаздыванию требований примера I' при расписании π_{spt} .

Для формулировки второй оценки рассмотрим функцию

$$f(t) = \sum_{j=1}^n \max\{0, t_0 + \sum_{i=1}^j p_i - t\}.$$

Значение этой функции в точке t равно оптимальному значению суммарного запаздывания для параметрического примера $I(t) = \langle \{p_j, t\}_{j \in N}, t_0 \rangle$, при котором директивные сроки всех требований одновременно равны t . Данное утверждение справедливо потому, что оптимальным расписанием для данного примера при любом значении t будет SPT-расписание π_{spt} . Функция f является кусочно-линейной, невозрастающей, непрерывной и неотрицательной функцией, принимающей значения 0 для всех $t \geq t_0 + \sum_{j \in N} p_j$.

Теорема 4 (вторая оценка). Для любого примера I справедливо неравенство $f(d_{\max}) \leq F^*(I) \leq f(d_{\min})$.

В заключение параграфа приводится схема построения такого примера задачи 1 || $\sum T_j$, при котором оптимальное значение целевой функции равняется заранее заданному значению. Схема построения такого примера основана на свойствах функции f .

Параграф 1.5 посвящен результатам экспериментальных исследований задачи.

Во второй главе диссертации рассматривается частный случай задачи минимизации суммарного запаздывания для одного прибора, когда параметры требований удовлетворяют условиям:

$$\begin{cases} p_1 \geq p_2 \geq \dots \geq p_n, \\ d_1 \leq d_2 \leq \dots \leq d_n. \end{cases} \quad (1)$$

Предлагается подход к решению задачи в случае (1), при котором исходное множество требований N разбивается на $\mathbb{k}$ таких непересекающихся подмножеств $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_{\mathbb{k}}$, что $N = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \dots \cup \mathcal{M}_{\mathbb{k}}$ и $\max_{i,j \in \mathcal{M}_\nu} |d_i - d_j| \leq \min_{j \in \mathcal{M}_\nu} p_j$ ($\nu = 1, \dots, \mathbb{k}$), а затем на основе этого разбиения строится оптимальное расписание.

Получены новые свойства оптимальных расписаний в случае (1). На основе этих свойств построен алгоритм решения с трудоемкостью не более $O(n^2 \sum p_j)$ операций. Выделены один псевдополиномиально разрешимый (за $O(n \sum p_j)$ операций) и два полиномиально разрешимых (за $O(n^2)$ операций) подслучая рассматриваемого случая с соответствующими алгоритмами решения. Отметим, что случай (1) является NP-трудным⁸.

В параграфе 2.1 формулируется и доказываются две леммы, отражающие свойства оптимальных расписаний в случае (1).

Лемма 3 *Существует оптимальное расписание π^* , при котором выполняется либо $(k \rightarrow i)_{\pi^*}$, либо $(j \rightarrow k)_{\pi^*}$ для любой такой тройки требований $i, j, k \in N$, что $|d_i - d_j| \leq \min\{p_i, p_j\}$, $k < \min\{i, j\}$ и $(i \rightarrow j)_{\pi^*}$.*

Лемма 4 *Существует оптимальное расписание π^* , при котором выполняется либо $(k \rightarrow j)_{\pi^*}$, либо $((k+1) \rightarrow k)_{\pi^*}$ для любой такой пары требований $k, j \in N$, что $k < j$.*

На основе полученных свойств оптимальных расписаний в параграфе 2.2 приводится описание подхода к решению примеров задачи в случае (1).

На первом этапе производится разбиение множества требований N на непересекающиеся подмножества $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_{\mathbb{k}}$, такие, что разность

⁸Лазарев А.А., Гафаров Е.Р. Доказательство NP-трудности частного случая задачи минимизации суммарного запаздывания // Известия РАН. Теория и системы управления. – 2006. – № 3. – С. 120–128. А также [8]

между максимальным и минимальным директивными сроками требований каждого подмножества не превышает величины минимальной продолжительности требований этого подмножества⁹.

Определим *параметрический пример* $I_k(t) = \langle \{p_j, d_j(t)\}_{j \in N_k}, 0 \rangle$, где $N_k = \{k, k+1, \dots, n\}$ и $d_j(t) = d_j - d_n + t - t_0$. $I_k(t)$ – это индивидуальный пример исследуемой задачи обслуживания требований множества N_k с моментом начала обслуживания, равным нулю, и директивными сроками $d_j(t)$, линейно зависящими от параметра t . Исходный пример I равен параметрическому примеру $I_1(t)$ при $t = d_n$, т.е. множества оптимальных расписаний для примеров I и $I_1(d_n)$ совпадают.

Идея предлагаемого подхода к решению примеров в случае (1) основана на методе динамического программирования и заключается в построении оптимальных расписаний $\pi_k^*(t)$ для примеров $I_k(t)$ в порядке $k = n, n-1, \dots, 1$ для всех целочисленных точек $t \in [t_0; d_n]$. Расписания $\pi_k^*(t)$ строятся на основе построенных на предыдущих шагах расписаний $\pi_l^*(t')$, $l > k$, $t' \in [t_0; d_n]$. Оптимальные расписания для примеров $I_n(t)$, множество требований которых включает в себя только одно требование с номером n , равны расписанию (n) .

Согласно леммам 3 и 4, при построении оптимального расписания для исходного примера I можно исключить из рассмотрения все расписания π , при которых

- (а) существуют такие требования $i, j, k \in N$, что $k \leq \min\{i, j\}$, требования i, j принадлежат некоторому подмножеству M_ν , требование k принадлежит подмножеству с меньшим, чем ν , номером, и выполняется $(i \rightarrow k \rightarrow j)_\pi$;
или
- (б) существуют такие требования $j, k \in N$, что $k < j$, и выполняется $(j \rightarrow k \rightarrow (k+1))_\pi$.

При построении оптимальных расписаний для примеров $I_k(t)$, отталкиваясь от структуры полученных на ранних шагах оптимальных расписаний примеров $I_l(t')$, $l > k$, можно определить позиции для требования k , на которых оно может быть обслужено при некотором оптимальном расписании. Показано, что таких позиций может быть не больше, чем

⁹Такое разбиение множества требований может быть выполнено за $O(n)$ операций.

количество подмножеств $\mathcal{M}$. Накапливая информацию об оптимальных значениях целевой функции в ходе работы алгоритма, можно найти оптимальное расписание для исходного примера I за $O(\mathbb{k}n \sum p_j)$ операций.

В параграфе 2.3 рассматривается случай $\mathbb{k} = 1$, то есть, параметры требований удовлетворяют условиям

$$\begin{cases} p_1 \geq p_2 \geq \dots \geq p_n, \\ d_1 \leq d_2 \leq \dots \leq d_n, \\ d_n - d_1 \leq p_n. \end{cases} \quad (2)$$

В этом случае при построении оптимального расписания $\pi_k^*(t)$ с помощью описанного выше подхода для каждой точки t требуется просмотреть только две позиции для требования k : до всех требований множества N_{k+1} и после всех требований этого множества. При постановке требования k на обслуживание до всех требований множества N_{k+1} получаем расписание $(k, \pi_{k+1}^*(t - p_k))$, после всех требований – расписание $(\pi_{k+1}^*(t), k)$. Расписание с наименьшим значением суммарного запаздывания среди двух полученных расписаний будет оптимальным расписанием для примера $I_k(t)$. Алгоритм, реализующий данный подход к построению расписания в случае (2), назовем *алгоритмом B-1*.

Теорема 5 *Алгоритм B-1 находит оптимальное расписание для случая (2) за $O(n \sum p_j)$ операций.*

В заключение параграфа приводится схема нахождения полиномиально разрешимых подслучаев случая (2), для которых алгоритм FA (параграф 1.3) находит решение за $O(n^3)$ операций.

В параграфе 2.4 рассматривается случай для произвольного значения $\mathbb{k}$. Приводится формальное описание алгоритма $B\text{-}\mathbb{k}$ нахождения оптимального расписания в случае (1). Необходимо отметить, что в случае $\mathbb{k} = 1$ шаги алгоритма $B\text{-}\mathbb{k}$ совпадают с шагами алгоритма $B\text{-}1$. Трудоемкость нахождения решения с помощью алгоритма $B\text{-}\mathbb{k}$ равна $O(\mathbb{k}n \sum p_j)$ операций.

Далее, в параграфах 2.5 и 2.6 формулируются и доказываются свойства оптимальных расписаний для двух частных случаев задачи, позволяющие построить алгоритмы нахождения решения примеров данных случаев с трудоемкостью $O(n^2)$ операций.

В параграфе 2.5 рассматриваются примеры, параметры требований которых удовлетворяют условиям

$$\begin{cases} d_1 \leq d_2 \leq \dots \leq d_n, \\ d_n - d_1 \leq 1, \\ t_0 \in \mathbb{Z} \end{cases} \quad (3)$$

На основе полученных свойств оптимальных расписаний в случае (3) построен алгоритм С-1, который находит решение за $O(n^2)$ операций. Идея данного алгоритма основана на построении расписания "с конца". Процесс работы алгоритма разбит на 3 этапа.

Действия первого этапа заключаются в упорядочении требований с наиболее большими продолжительностями обслуживания в конце расписания до тех пор, пока среди неупорядоченных требований не появится такого требования, которое будучи поставленным на обслуживание перед упорядоченными ранее требованиями, своим моментом начала обслуживания не окажется некоторым образом "близким" к группе директивных сроков. После завершения работы первого этапа начинается этап перебора всех возможных пар из оставшихся неупорядоченных требований, которые будут обслужены "вблизи" группы директивных сроков. Оставшиеся после двух первых этапов неупорядоченные требования помещаются на обслуживание в произвольном порядке в начале расписания.

Алгоритм С-1 находит оптимальное расписание для случая (3) за $O(n^2)$ операций. Отметим, что данный алгоритм можно использовать для решения примеров в случае, когда в условиях (3) второе неравенство заменено на $d_n - d_1 \leq \text{НОД}(p_1, \dots, p_n)$.

В параграфе 2.6 рассматриваются примеры, параметры требований которых удовлетворяют условиям

$$d_j - d_{j-1} > p_j, \quad j = 2, 3, \dots, n. \quad (4)$$

Поскольку $p_j > 0$, $j \in N$, то выполняется $d_1 \leq \dots \leq d_n$. Отметим, что случай (4) является "предельным" подслучаем случая (1), когда количество подмножеств $\mathbb{K}$ равно количеству требований, т.е. каждое подмножество $\mathcal{M}_\nu$ ($\nu = 1, \dots, \mathbb{K}$) состоит ровно из одного требования.

Показано, что для примеров в случае (4) существует оптимальное расписание $\pi^* \in \Pi^*(I)$, при котором требование j^* обслуживается на первой позиции из множества $L(I)$. Данное свойство позволяет построить

алгоритм B - n решения примеров рассматриваемого случая за $O(n^2)$ операций. Этот алгоритм представляет собой модификацию алгоритма A , в которой перебираются не все возможные позиции для требования j^* , а только первая из них.

В третьей главе диссертации исследуются свойства алгоритма B -1, показано, что данный алгоритм может быть использован для нахождения решения следующих NP-полных задач разбиения: РАЗБИЕНИЕ, ЧЁТНО–НЕЧЁТНОЕ РАЗБИЕНИЕ, ЧЁТНО–НЕЧЁТНОЕ РАЗБИЕНИЕ С ОГРАНИЧЕНИЯМИ. Предложены модификации алгоритма B -1 и показано, что процесс построения расписания этим алгоритмом можно свести к процессу построения кусочно–линейной функции специального вида.

В параграфе 3.1 вводится определение расписания, обладающего свойством B -1 и показывается, что если для примера I существует оптимальное расписание, обладающее свойством B -1, то алгоритм B -1 находит оптимальное расписание для данного примера I .

Расписание π назовем расписанием, обладающим *свойством B -1*, если для всех требований $k \in \{1, 2, \dots, n-1\}$ при расписании π выполняется либо $(k \rightarrow j)_\pi$, либо $(j \rightarrow k)_\pi$ для всех $j \in \{k+1, \dots, n\}$.

Множество всех расписаний, обладающих *свойством B -1* для примера I , будем обозначать через $\Pi_{B-1}(I)$. Пусть $\Pi_{B-1}^*(I) = \Pi_{B-1}(I) \cap \Pi^*(I)$. Таким образом, множество $\Pi_{B-1}^*(I)$ является множеством всех оптимальных расписаний, обладающих свойством B -1. Получено следующее необходимое и достаточное условие того, что алгоритмом B -1 для некоторого примера I будет построено оптимальное расписание (даже если параметры требований примера I не удовлетворяют условиям (2)).

Теорема 6 *Алгоритм B -1 находит оптимальное расписание для некоторого примера I тогда и только тогда, когда $\Pi_{B-1}^*(I) \neq \emptyset$.*

В заключение параграфа 3.1 приводится один частный случай задачи, для которого показывается, что для любого примера этого случая существует оптимальное расписание, обладающее свойством B -1. Таким образом, для решения примеров этого частного случая можно использовать алгоритм B -1.

Параграф 3.2 посвящен классической NP-полной (в обычном смысле) задаче РАЗБИЕНИЕ, рассматривается схема полиномиального сведения

примеров этой задачи к примерам задачи теории расписаний $1 \parallel \sum T_j$. Приводится алгоритм решения задачи РАЗБИЕНИЕ.

В подпараграфе 3.2.1 приводится постановка задачи РАЗБИЕНИЕ и указывается схема полиномиального сведения данной задачи к двум NP-полным в обычном смысле задачам ЧЕТНО-НЕЧЕТНОЕ РАЗБИЕНИЕ (далее, ЧНР) и ЧЕТНО-НЕЧЕТНОЕ РАЗБИЕНИЕ С ОГРАНИЧЕНИЯМИ (далее, ОЧНР).

В классической задаче РАЗБИЕНИЕ требуется определить, существует ли такое разбиение множества m целых положительных чисел на два подмножества, что суммы чисел в обоих подмножествах равны. Задача ЧНР представляет собой модификацию задачи разбиения, когда на разбиение множества накладывается ограничение, что соседние числа (первое и второе, третье и четвертое и т. д.) должны оказаться в разных подмножествах. Задача ОЧНР представляет собой задачу ЧНР с дополнительными ограничениями на значения чисел исходного множества.

Известна¹⁰ схема полиномиального сведения примеров задачи ОЧНР к задаче $1 \parallel \sum T_j$ (таким образом устанавливается NP-трудность задачи $1 \parallel \sum T_j$). Основная идея схемы сведения заключается в построении по заданному примеру задачи ОЧНР примера задачи $1 \parallel \sum T_j$ (такие примеры задачи суммарного запаздывания называются *каноническими*). При этом доказывается, что для полученных примеров задачи $1 \parallel \sum T_j$ существует оптимальное расписание специального вида, называемое *каноническим*. Исходный пример задачи ОЧНР имеет искомое разбиение на два подмножества тогда и только тогда, когда соответствующий канонический пример задачи $1 \parallel \sum T_j$ имеет оптимальное значения суммарного запаздывания, равное определенной величине.

В подпараграфе 3.2.2 показано, что любое (в том числе, и оптимальное) расписание канонического вида обладает свойством $B-1$. Таким образом, мы можем использовать алгоритм $B-1$ для нахождения решения канонических примеров задачи $1 \parallel \sum T_j$ и, соответственно, задач РАЗБИЕНИЕ, ЧНР, ОЧНР, хотя канонические примеры не принадлежат случаю (2). С учетом особенностей канонических примеров предложен алгоритм $B-1$ –*канонический*, трудоемкость которого не хуже известного алгоритма Гэри и Джонсона¹¹ для задачи РАЗБИЕНИЕ. Алгоритм $B-$

¹⁰Du J., Leung J. Y.-T. Minimizing total tardiness on one processor is NP-hard // Math. Oper. Res. – 1990.- N.15. – P. 483–495.

¹¹Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи: Пер. с англ. – М.:

1-канонический отличается от алгоритма $B-1$ тем, что на каждом шаге алгоритма просматриваются не все точки t из интервала $t \in [t_0, d_n]$, а лишь некоторые из них.

В параграфе 3.3 приводится модификация алгоритма $B-1$, которая позволяет снизить трудоемкость решения примеров в случае (2). Идея алгоритма $B-1$ -модифицированный заключается в том, что процесс построения оптимального расписания сведен к процессу построения кусочно-линейной функции специального вида. Рассматривается функция $f(t)$: кусочно-линейная непрерывная монотонно-невозрастающая функция, обращаяющаяся в нуль для всех t , больших или равных некоторой величины t' , и строго убывающая для всех $t < t'$. При этом коэффициент наклона функции $f(t)$ по модулю не превышает значения n . Рассматриваются три операции над функциями такого вида: сдвиг функции, сложение двух функций и нахождение функции минимума двух функций.

Полученные свойства рассматриваемых функций использованы для построения алгоритма $B-1$ -модифицированный. Преимущество этого алгоритма заключается в том, что при построении решения для примеров случая (2) нет необходимости рассматривать все целочисленные точки интервала $t \in [t_0, d_n]$. Трудоемкость решения полиномиально зависит от максимального количества точек изменения наклона функций, рассматриваемых в ходе решения. Преимуществом Алгоритма $B-1$ -модифицированный является то, что при умножении одновременно всех параметров примера в константное количество раз трудоемкость решения не меняется. Также данный алгоритм может быть использован для решения задач $1 \parallel \sum T_j$ (в случае (2)), РАЗБИЕНИЕ, ЧНР и ОЧНР с нецелочисленными параметрами примеров.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ ДИССЕРТАЦИОННОЙ РАБОТЫ

- Найдены новые свойства задачи $1 \parallel \sum T_j$, использование которых позволяет сократить перебор при построении решения с помощью одного известного алгоритма декомпозиционного типа. На основе полученных свойств предложен полиномиальный алгоритм решения примеров в случае, когда при любом возможном расписании количество запаздывающих требований является константой;

- Получены свойства расписаний, требования при которых упорядочены в порядке возрастания продолжительностей обслуживания, на их основе даны две оценки оптимального значения целевой функции;
- Доказаны новые свойства оптимальных расписаний NP-трудного частного случая, когда параметры требований удовлетворяют условиям $p_1 \geq p_2 \geq \dots \geq p_n$, $d_1 \leq d_2 \leq \dots \leq d_n$, и предложен алгоритм решения примеров этого случая за $O(n^2 \sum p_j)$ операций. Выделен один псевдополиномиально разрешимый ($O(n \sum p_j)$ операций) и два полиномиально ($O(n^2)$ операций) разрешимых подслучая этого случая;
- Получены необходимое и достаточное условие того, что один псевдополиномиальный алгоритм находит оптимальное решение для некоторого примера задачи. На основе полученных свойств показано, как данный алгоритм может быть использован для нахождения решения NP-полных задач разбиения;
- Показано, что процесс построения расписания с помощью алгоритма из предыдущего результата можно свести к процессу построения кусочно-линейной функции специального вида. Предложена модификация этого алгоритма, которая позволяет решать задачи разбиения с нецелочисленными параметрами.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

- [1] Кварацхелия А.Г. К исследованию NP-сложной задачи теории расписаний минимизации суммарного запаздывания для одного прибора / А.Г. Кварацхелия // Материалы XL Международной научной студенческой конференции «Студент и научно-технический прогресс»: Математика. – Новосиб. гос. ун-т. Новосибирск, 2002. – С. 124-125.
- [2] Лазарев А.А. Стохастическое исследование методов ветвей и границ для NP-трудной задачи теории расписаний минимизации суммарного запаздывания для одного прибора / А.А. Лазарев, А.Г. Кварацхелия // Материалы VII Международного семинара «Дискретная математика и ее приложения» (29 января – 2 февраля 2001 г.). Часть III. – М.: Изд-во центра прикладных исследований при мех.-мат. факультете МГУ, 2001 г. – С. 379–381.
- [3] Лазарев А.А. Исследование проблемы теории расписаний $1 || \sum T_j$ / А.А. Лазарев, А.Г. Кварацхелия // Проблемы теоретической кибернетики. Тезисы докладов XIII Международной конференции (Казань, 27–31 мая 2002 г.). Часть II. – М.: Изд-во

центра прикладных исследований при механико-математическом факультете МГУ, 2002. – С. 107.

- [4] Лазарев А.А. Исследование проблемы теории расписаний $1 \parallel \sum T_j$ / А.А. Лазарев, А.Г. Кварацхелия // Материалы Российской конференции «Дискретный анализ и исследование операций», 24–28 июня, 2002 г. – Новосибирск, 2002. – С. 218.
- [5] Лазарев А.А. Теоретическое и экспериментальное исследование NP-трудной проблемы теории расписаний минимизации суммарного запаздывания на одном приборе / А.А. Лазарев, А.Г. Кварацхелия // Исследования по прикладной математике и информатике. – Казань: Изд-во Казан. гос. ун-та, 2003. – Вып. 24. – С. 90–106.
- [6] Лазарев А.А. Алгоритм $O(n^2 \sum p_j)$ решения NP-трудной проблемы теории расписаний $1 \parallel \sum T_j$ / А.А. Лазарев, А.Г. Кварацхелия // Материалы VIII Международного семинара «Дискретная математика и ее приложения» (2–6 февраля 2004 г.). – Изд-во механико-математического факультета МГУ, 2004. – С. 211–213.
- [7] Лазарев А.А. Алгоритм решения проблем $1 \parallel \sum T_j$ и четно-нечетного разбиения / А.А. Лазарев, А.Г. Кварацхелия // Труды VI Международной конференции «Дискретные модели в теории управляющих систем» (7–11 декабря 2004 г.). М.: Изд-во факультета вычислительной математики и кибернетики МГУ, 2004 – С. 184–187.
- [8] Лазарев А.А. Алгоритмы решения NP-трудной проблемы минимизации суммарного запаздывания для одного прибора / А.А. Лазарев, А.Г. Кварацхелия, Е.Р. Гафаров // Доклады Академии Наук. – 2007. – Том 412, № 6. – С. 739–742.
- [9] Lazarev A. Methods to research problem of scheduling theory minimizing total tardiness on a single machine / A. Lazarev, A. Kvaratskhelia // Abstracts of VI Workshop «Models and Algorithms for Planning and Scheduling Problems» (Aussois, France, 30.III - 4.IV.2003). – P. 143–144.
- [10] Lazarev A. Solution Algorithms for the Total Tardiness Scheduling Problem on a Single Machine / A. Lazarev, A. Kvaratskhelia, A. Tchernykh // Proceedings of XV International Conference ENC'04 (Colima, Mexico, 20 - 24.X.2004). – P. 474–480.
- [11] Lazarev A. Algorithms for solving problems $1 \parallel \sum T_j$ and Even-Odd Partition / A. Lazarev, A. Kvaratskhelia // Abstracts of XVIII International Conference «European Chapters on Combinatorial Optimization» (26–28 May, Minsk, Belarus). – Minsk: United Institute of Informatics Problems on the National Academy of Sciences of Belarus, 2005. – P. 32–33.