

УДК 519.72

МЕТРИЧЕСКИЙ ПОДХОД НАХОЖДЕНИЯ ПРИБЛИЖЕННЫХ РЕШЕНИЙ ЗАДАЧ ТЕОРИИ РАСПИСАНИЙ¹⁾

© 2021 г. А. А. Лазарев^{1,*}, Д. В. Лемтюжникова¹, Н. А. Правдивец¹¹ 117997 Москва, ул. Профсоюзная, 65, ИПУ РАН, Россия

*e-mail: jobmath@mail.ru

Поступила в редакцию 26.11.2020 г.
Переработанный вариант 26.11.2020 г.
Принята к публикации 11.03.2021 г.

Вводятся функции метрики для разных классов задач теории расписаний для одного прибора. Показано, как с помощью введенных функций находятся приближенные решения NP-трудных задач. Величина метрики находится в результате решения задачи линейного программирования, ограничениями которой являются системы линейных неравенств полиномиальных или псевдополиномиальных разрешимых случаев исследуемых задач. Фактически находится проекция во введенной метрике решаемого примера на разрешимые подслучаи задачи. Библ. 23. Фиг. 1. Табл. 3.

Ключевые слова: теория расписаний, метрика, аппроксимация, методы оптимизации.

DOI: 10.31857/S0044466921070127

1. ВВЕДЕНИЕ

подавляющее большинство задач теории расписаний являются NP-трудными, поэтому поиск приближенных полиномиальных алгоритмов является актуальной задачей.

Существуют два типа методов решения таких задач: точные и приближенные (см. [1]). К первой группе относятся целочисленное линейное программирование (см. [2]), динамическое программирование (см. [3]), метод ветвей и границ (см. [4]), локальный элиминационный алгоритм (см. [5]) и т.д. В этом случае оптимальное значение целевой функции вычисляется без каких-либо ошибок, но вычисления требуют больших затрат времени и памяти. Приближенные методы, такие как генетические алгоритмы (см. [6]), алгоритмы муравьиной колонии (см. [7]), алгоритмы пчелиного роя (см. [8]), табу-поиск (см. [9]) и многие другие, гораздо быстрее получают приближенное решение, но, как правило, не имеют оценок отклонения значения целевой функции от оптимального (см. [10]).

В настоящей работе мы описываем общий приближенный подход, который называется *метрическим* (см. [11]). Данный подход позволяет находить приближенное решение с гарантированной погрешностью целевой функции, не превышающей значения функции метрики.

Идея метрического подхода заключается в следующем. Пример задачи A характеризуется точкой, координаты которой соответствуют входным данным задачи. Для этой задачи рассмотрим все известные полиномиальные и псевдополиномиальные алгоритмы. Затем вводим некоторую метрику, значение которой фактически ограничивает сверху оценку абсолютной погрешности целевой функции. С помощью этой метрики находим полиномиально или псевдополиномиально разрешимый пример задачи B с наименьшим расстоянием от данного примера A во введенной метрике. Для этого мы формулируем и решаем задачу линейного программирования. Другими словами, строим проекцию начальной точки A на подпространство задач.

На данный момент для этого подхода получены следующие результаты. В [12] проводится сравнение допустимых областей и дается понятие полиномиальной меры неразрешимости для задачи теории расписаний с одним прибором $1|r_j|L_{\max}$. Возможно, его можно уменьшить еще больше, найдя новые подпространства полиномиально разрешимых примеров. В [13], [14] получены соответствующие метрики для задач теории расписаний с функцией суммарного запаздывания.

¹⁾ Работа выполнена при частичной финансовой поддержке РФФИ (коды проектов № 18-31-00458, № 20-58-S52006).

2. ПОСТАНОВКА ЗАДАЧИ СУММАРНОГО ЗАПАЗДЫВАНИЯ ДЛЯ ОДНОГО ПРИБОРА

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

Для требования $j \in N$ заданы следующие параметры: продолжительность обслуживания $p_j > 0$ 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 $s: \mathbb{R} \rightarrow \{0, 1, \dots, n\}$. 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 на приборе без прерываний и искусственных простоев прибора, то расписание однозначно задается перестановкой π элементов множества N . Далее в работе понятие расписания и перестановки множества требований будем отождествлять и называть *расписанием* перестановку π .

Индивидуальный пример исследуемой задачи (далее *пример*) с заданным множеством требований N , продолжительностями обслуживания p_j , директивными сроками d_j и моментом начала обслуживания t_0 будем обозначать через $I = \langle \{p_j, d_j\}_{j \in N}, t_0 \rangle$. В случае, если параметры требований фиксированы (однозначно определены), для обозначения примера I будем использовать запись $\{N, t_0\}$. Множество всех $n!$ расписаний для примера I будем обозначать через $\Pi(I)$.

Через $C_j(\pi)$ будем обозначать момент окончания обслуживания требования j при расписании π . Моменты окончания обслуживания требований при расписании $\pi = (j_1, \dots, j_n)$ вычисляются следующим образом:

$$C_{j_1}(\pi) = t_0 + p_{j_1},$$

$$C_{j_k}(\pi) = C_{j_{k-1}}(\pi) + p_{j_k}, \quad k = 2, 3, \dots, n.$$

Если обслуживание требования i предшествует обслуживанию требования j при расписании π (т.е. выполняется $C_i(\pi) < C_j(\pi)$), то будем использовать запись $(i \rightarrow j)_\pi$. Используя это обозначение, момент окончания обслуживания требования $j \in N$ при расписании π можно записать как $C_j(\pi) = t_0 + \sum_{i \in N: (i \rightarrow j)_\pi} p_i + p_j$. Обслуживание всех требований примера I завершается в момент времени $t_0 + \sum_{j \in N} p_j$ при любом расписании $\pi \in \Pi(I)$.

Под *запаздыванием* требования $j \in N$ при расписании π будем понимать величину

$$T_j(\pi) = \max\{0, C_j(\pi) - d_j\}.$$

Суммарное запаздывание требований при расписании π определяется как

$$F(\pi) = \sum_{j=1}^n T_j(\pi).$$

Задача минимизации суммарного запаздывания заключается в построении оптимального расписания $\pi^* \in \Pi(I)$, при котором выполняется неравенство $F(\pi^*) \leq F(\pi)$ для всех расписаний $\pi \in \Pi(I)$. Отметим, что данная задача является *NP*-трудной (см. [15]). Известен псевдополиномиальный алгоритм ее решения (см. [16]) трудоемкости $O\left(n^4 \sum_{j \in N} p_j\right)$ операций, в случае когда $p_j \in \mathbb{Z}^+ \forall j \in N$, основанный на методе динамического программирования.

Через $\Pi^*(I)$ будем обозначать множество всех оптимальных расписаний для примера I . Для обозначения величины оптимального суммарного запаздывания примера I будем использовать запись $F^*(I)$.

Покажем, что без ограничения общности рассуждений можно предполагать, что рассматриваются только те примеры I , для которых выполняется

$$d_j \in \left[t_0; t_0 + \sum_{i \in N} p_i \right] \quad \forall j \in N. \tag{2.1}$$

В случае, если для примера I условия (2.1) не выполняются, построим пример $I' = \langle \{p_j, d'_j\}_{j \in N}, t_0 \rangle$ такой, что

$$d'_j = \min \left\{ \max\{t_0, d_j\}, t_0 + \sum_{i \in N} p_i \right\} \quad \forall j \in N.$$

Если для некоторого требования $j \in N$ выполняется $d_j > t_0 + \sum_{i \in N} p_i$, то имеем $d'_j = t_0 + \sum_{i \in N} p_i$. Это означает, что требование j не запаздывает при любом расписании π для обоих примеров I и I' . Таким образом, можно исключить такое требование j из рассмотрения и после построения оптимального расписания для редуцированного примера добавить требование j на последнюю позицию в построенное расписание.

Если для некоторого требования $j \in N$ выполняется $d_j < t_0$, то имеем $d'_j = t_0$, и требование j запаздывает при любом расписании π . Лоулером в [16] была доказана следующая

Теорема 1 (см. [16]). Пусть π^* – оптимальное расписание примера $I_1 = \langle \{p_j, d_j\}_{j \in N}, t_0 \rangle$. Выберем величины d'_j так, что

$$\min\{d_j, C_j(\pi^*)\} \leq d'_j \leq \max\{d_j, C_j(\pi^*)\} \quad \forall j \in N. \tag{2.2}$$

Тогда любое оптимальное расписание примера $I_2 = \langle \{p_j, d'_j\}_{j \in N}, t_0 \rangle$ будет оптимальным и для примера I_1 .

В нашем случае условия (2.2) для примеров I и I' выполняются, поскольку при любом расписании π справедливо неравенство $t_0 < C_j(\pi) \leq t_0 + \sum_{i \in N} p_i$, $j \in N$. Следовательно, любое оптимальное расписание для примера I' будет оптимальным и для примера I . Таким образом, параметры требований примера I' удовлетворяют условиям (2.1) и выполняется $\Pi^*(I') \subseteq \Pi^*(I)$.

Необходимо заметить, что не любое оптимальное расписание примера I будет оптимальным и для примера I' .

Как следствие теоремы 1, может быть сформулирована

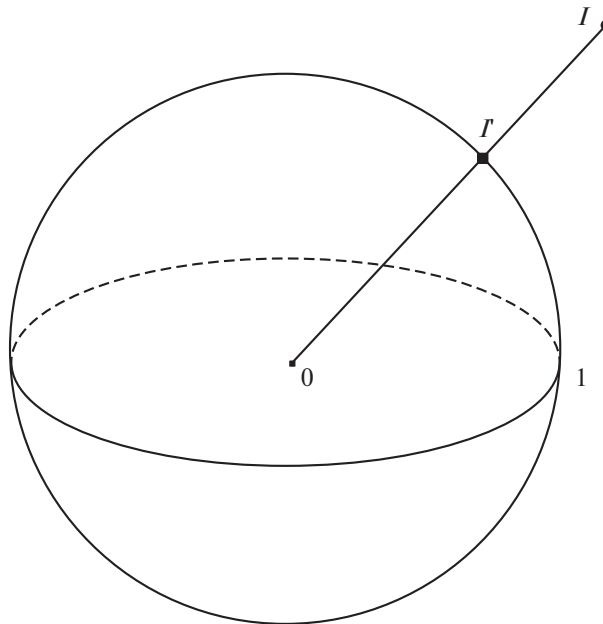
Лемма 1. Для любого оптимального расписания π и любого незапаздывающего требования j верно: все требования, которые обслуживаются в интервале $[C_j(\pi), d_j]$ не запаздывают.

Данная лемма легко доказывается от противного.

Два примера $I = \langle \{p_j, d_j\}_{j \in N}, t_0 \rangle$ и $I' = \langle \{p'_j, d'_j\}_{j \in N}, t'_0 \rangle$ будем называть *равными*, если множества оптимальных расписаний для обоих примеров совпадают, т.е. $\Pi^*(I) = \Pi^*(I')$. Через $T'_j(\pi)$ будем обозначать значение запаздывания требования j при некотором расписании π , вычисленное для примера I' . В рамках данного исследования будут полезны следующие случаи равенства примеров.

(1) Примеры I и I' равны, если $p'_j = p_j$, $d'_j = d_j + C$, $j \in N$, и $t'_0 = t_0 + C$, где C – произвольная константа. Действительно, в этом случае для любого расписания π и любого требования $j \in N$, обслуживаемого при расписании π , выполняется

$$\begin{aligned} T'_j(\pi) &= \max \left\{ 0, t_0 + C + \sum_{i: (i \rightarrow j)_\pi} p_i + p_j - (d_j + C) \right\} = \\ &= \max \left\{ 0, t_0 + \sum_{i: (i \rightarrow j)_\pi} p_i + p_j - d_j \right\} = \max\{0, C_j(\pi) - d_j\} = T_j(\pi). \end{aligned}$$



Фиг. 1. Схематическое изображение проецирования примера I на единичную сферу в $(2n + 1)$ -мерном евклидовом пространстве. Примеры I и I' равны.

Следовательно, $F^*(I) = F^*(I')$ и $\Pi^*(I) = \Pi^*(I')$. В этом смысле без потери общности момент начала обслуживания требований можно считать равным нулю: $t_0 = 0$. Если требуется выполнение некоторых условий относительно директивных сроков, например $d_j \in \mathbb{Z}^+$, то это обычно достигается путем изменения момента начала обслуживания t_0 . Поэтому в работе мы будем предполагать, что t_0 – произвольная величина.

(2) Примеры I и I' равны, если $p'_j = \alpha p_j$, $d'_j = \alpha d_j$, $j \in N$, и $t'_0 = \alpha t_0$, где $\alpha > 0$ – произвольная положительная константа. Действительно, в этом случае для любого расписания π и любого требования $j \in N$, обслуживаемого при расписании π , выполняется

$$T'_j(\pi) = \max \left\{ 0, \alpha t_0 + \sum_{i: (i \rightarrow j)_\pi} \alpha p_i + \alpha p_j - \alpha d_j \right\} = \alpha \max \left\{ 0, t_0 + \sum_{i: (i \rightarrow j)_\pi} p_i + p_j - d_j \right\} = \alpha T_j(\pi).$$

Следовательно, $F^*(I) = \alpha F^*(I')$ и $\Pi^*(I) = \Pi^*(I')$. Рассмотрим любой пример I как точку в $(2n + 1)$ -мерном евклидовом пространстве с координатами $(t_0, p_1, \dots, p_n, d_1, \dots, d_n)$. Тогда все примеры (точки), расположенные на луче, исходящем из нуля пространства, равны между собой (фиг. 1). Поэтому можно предполагать, что рассматриваются только те примеры (точки), которые лежат на поверхности единичной сферы в рассматриваемом пространстве. Однако в общем случае данный подход неприемлем при построении оптимального расписания некоторым алгоритмом, для работы которого необходимо, чтобы продолжительности обслуживания требований были целочисленными величинами. Поэтому ограничимся выбором константы $\alpha = 1/\text{НОД}(p_1, \dots, p_n)$, где НОД – наибольший общий делитель.

Необходимо также отметить, что для любых двух равных примеров I и I' выполняется свойство: если требование при оптимальном расписании π примера I запаздывает (не запаздывает), то и в примере I' при расписании π требование запаздывает (не запаздывает). Доказательство этого можно провести от противного.

Введем некоторые дополнительные обозначения и понятия.

Расписание π будем называть *SPT*-расписанием (Shortest Processing Time) и обозначать через π_{spt} , если требования упорядочены при данном расписании в порядке неубывания продолжительностей обслуживания, т.е. для любых двух требований i и j таких, что $p_i < p_j$, выполняется $(i \rightarrow j)_{\pi_{spt}}$.

Расписание π будем называть *EDD*-расписанием (Earliest Due Date) и обозначать через π_{edd} , если требования упорядочены при данном расписании в порядке неубывания директивных сроков, т.е. для любых двух требований i и j таких, что $d_i < d_j$, выполняется $(i \rightarrow j)_{\pi_{edd}}$.

Через $\{\pi\}$ будем обозначать множество требований, упорядоченных при расписании π . В случае $\{\pi\} \neq N$ ($\{\pi\} \subset N$) расписание π будем называть *частичным* расписанием.

Запись $(i \rightarrow j)_{\pi}$ при необходимости будет расширена до $(i \rightarrow j \rightarrow k)_{\pi}$ для обозначения порядка обслуживания более, чем двух требований при некотором расписании π . Также будем использовать записи $(j \rightarrow N')_{\pi}$ или $(N' \rightarrow N'')_{\pi}$ для обозначения порядка обслуживания между некоторыми подмножествами требований.

Для обозначения структуры расписания π будем использовать запись $\pi = (\pi_1, \dots, \pi_m)$, где расписание π_i , $i = 1, 2, \dots, m$, является частичным расписанием (подрасписанием расписания π). При этом для частичных расписаний π_i выполняется $\{\pi\} = \bigcup_{i=1}^m \{\pi_i\}$, $\{\pi_i\} \cap \{\pi_j\} = \emptyset$ для $i \neq j$, $(\{\pi_1\} \rightarrow \dots \rightarrow \{\pi_m\})_{\pi}$, и требования каждого частичного расписания π_i , $i = 1, 2, \dots, m$, упорядочены в том же порядке, что и при расписании π . Для обозначения структуры расписания π относительно некоторого требования $j \in \{\pi\}$ будем использовать запись $\pi = (\pi_1, j, \pi_2)$.

Пользуясь нотацией, предложенной в [17], для обозначения исследуемой задачи будем использовать запись $1 \parallel \sum T_j$. Рассматриваемая задача $1 \parallel \sum T_j$ является *NP*-трудной в обычном смысле (см. [15]).

В заключение данного раздела приведем пример связи рассматриваемой задачи для одного прибора $1 \parallel \sum T_j$ с некоторой задачей оптимального упорядочения набора требований, возникающей на практике. Рассмотрим некоторый обслуживающий центр (ОЦ), процесс работы которого разбит на циклы, состоящие из двух этапов. На первом этапе центр набирает заявки и на втором этапе осуществляет их исполнение. В ОЦ имеется m однородных исполнителей, каждая заявка может быть исполнена любым из них. Исполнители обладают в общем случае различной производительностью и выполняют в любой момент времени не более одной заявки. Выполнение заявки не может быть прервано, т.е., если исполнитель начинает выполнение некоторой заявки, то он выполняет ее до конца. При приеме заявки ОЦ и клиент договариваются о сумме, которую ОЦ получит за выполнение заявки, а также о моменте времени, к которому заявка должна быть исполнена. При выполнении некоторой заявки позднее обговоренного момента времени ОЦ выплачивает клиенту фиксированный для любой заявки штраф за каждую единицу времени, прошедшую после этого момента. Целью составления расписания работы ОЦ является минимизация суммы штрафных выплат по всем принятым к исполнению заявкам.

Данная задача может быть сформулирована в терминах теории расписаний следующим образом. Задано множество требований $N = \{1, 2, \dots, n\}$ и множество параллельных приборов $M = \{1, 2, \dots, m\}$. Каждое требование $j \in N$ имеет директивный срок d_j , к которому желательно завершить обслуживание данного требования, и продолжительность p_{ji} обслуживания требования j на приборе $i \in M$. Требования обслуживаются на приборах без прерываний, в любой момент времени любой прибор обслуживает не более одного требования. Расписание обслуживания требований π строится с момента времени $t_0 = 0$ и для данной задачи может быть задано в виде набора из m перестановок π_1, \dots, π_m , где π_i , $i \in M$, задает порядок обслуживания требований множества $\{\pi_i\}$ на приборе i . При этом $\{\pi_1\}, \dots, \{\pi_m\}$ задает разбиение множества требований N (т.е. $\bigcup_{i \in M} \{\pi_i\} = N$ и $\{\pi_i\} \cap \{\pi_j\} = \emptyset$ для $i, j \in M$, $i \neq j$). Момент окончания $C_j(\pi)$ требования $j \in N$ при расписании π вычисляется описанным выше образом. Необходимо построить такое расписание обслуживания требований множества N на приборах из множества M , при котором минимизируется суммарное запаздывание требований, т.е. $\sum_{j \in N} \max\{0, C_j(\pi) - d_j\} \rightarrow \min$.

Имея некоторый алгоритм, основанный, например, на идее метода ветвей и границ, для задачи суммарного запаздывания для параллельных приборов было бы желательно иметь некоторый способ вычисления нижних оценок оптимального значения целевой функции. Это может быть сделано с помощью использования методов решения и построения оценок для примеров задачи $1 \parallel \sum T_j$. Однако алгоритм решения задачи суммарного запаздывания для параллельных приборов может быть организован так, что сначала строится разбиение множества требований N на m подмножеств, которые будут обслуживаться на соответствующих приборах, а затем решается m независимых примеров задачи $1 \parallel \sum T_j$. В этом случае исходная задача для многих приборов допускает разбиение на несколько независимых подзадач для одного прибора.

3. МЕТРИКА ДЛЯ ЗАДАЧИ $1|r_j|\sum T_j$

3.1. Постановка задачи

Имеется множество N , состоящее из n требований, которые необходимо обслужить на одном приборе. Прибор готов начать обслуживание в момент времени $t_0 = 0$ и не может обслуживать более одного требования одновременно. Прерывания при обслуживании запрещены. Для каждого требования $j \in N$ заданы: момент поступления r_j , продолжительность обслуживания p_j и директивный срок d_j . Расписание $\pi = \{j_1, \dots, j_n\}$ определяет порядок, в котором обслуживаются требования. Естественно рассматривать ранние расписания, при которых

$$C_{j_1}(\pi) = \max\{t_0, r_{j_1}\} + p_{j_1},$$

$$C_{j_k}(\pi) = \max\{r_{j_k}, C_{j_{k-1}}(\pi)\} + p_{j_k}, \quad k = 2, 3, \dots, n,$$

где $C_{j_k}(\pi)$ — момент окончания обслуживания требования j при расписании π . Необходимо построить оптимальное расписание π^* , минимизирующее целевую функцию — суммарное запаздывание $\sum_{j \in N} T_j(\pi)$, где $T_j(\pi) = \max\{0, C_{j_k}(\pi) - d_j\}$ — запаздывание требования j при расписании π . В дальнейшем, если из контекста ясно, о каком расписании идет речь, то зависимость от π может опускаться. Данная задача является NP -трудной (см. [15]) и обозначается $1|r_j|\sum T_j$ (см. [17]).

Задача $1|r_j|\sum T_j$ полностью характеризуется $3n$ параметрами — моментами поступления, продолжительностями обслуживания и директивными сроками каждого из n требований. Будем говорить, что задан пример A задачи, если задано $3n$ параметров $\{r_j^A, p_j^A, d_j^A, j = 1, 2, \dots, n\}$, характеризующих задачу.

Для частного случая $r_j = 0, j \in N$, задачи минимизации суммарного запаздывания ранее был предложен полиномиальный алгоритм нахождения приближенного решения с относительной погрешностью $\varepsilon > 0$ сложности $O\left(\frac{n^7}{\varepsilon}\right)$ операций (см. [18]). Позднее М.Я. Ковалев усилил оценку до $O(n^6 \log n + n^6/\varepsilon)$ операций (см. [19]). Также для этого случая известен псевдополиномиальный алгоритм сложности $O\left(n^4 \sum p_j\right)$ (см. [16]). В случае, если

$$\begin{aligned} p_1 &\geq \dots \geq p_n, \\ d_1 &\leq \dots \leq d_n, \end{aligned}$$

сложность псевдополиномиального алгоритма может быть уменьшена до $O\left(n^2 \sum p_j\right)$ операций (см. [20]). Для случая $1|r_j, p_j = p|\sum T_j$ известен полиномиальный алгоритм сложности $O(n^7)$ операций, предложенный Ф. Баптисте (см. [21]), на основе метода динамического программирования.

Предлагается подход приближенного решения задачи $1|r_j|\sum T_j$ с гарантированной погрешностью, основанный на введении метрики для пространства параметров задачи, и рассматриваются возможности применения данного подхода к другим задачам теории расписаний. Затем описываются численные эксперименты, проведенные для проверки предложенного метода.

3.2. Метрика для пространства параметров

Задача $1|r_j|\sum T_j$ полностью характеризуется $3n$ параметрами, что позволяет нам рассматривать примеры задачи, как точки в $3n$ -мерном пространстве параметров $\Omega = \{r_1, \dots, r_n, p_1, \dots, p_n, d_1, \dots, d_n\}$.

Лемма 2. Пусть примеры A и B имеют одинаковые продолжительности обслуживания и директивные сроки:

$$p_j^A = p_j^B, \quad d_j^A = d_j^B, \quad j \in N.$$

Тогда для любого расписания π

$$\left| \sum_{j \in N} T_j^A(\pi) - \sum_{j \in N} T_j^B(\pi) \right| \leq n \max_{j \in N} |r_j^A - r_j^B|.$$

Доказательство. Используя определение запаздывания и известное неравенство

$$|\max\{a, b\} - \max\{c, d\}| \leq \max\{|a - c|, |b - d|\} \quad \forall a, b, c, d \in \mathbb{R},$$

получаем

$$\left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| \leq \sum_{j \in N} |C_j^A - C_j^B + d_j^B - d_j^A| \leq \sum_{j \in N} |C_j^A - C_j^B| + \sum_{j \in N} |d_j^A - d_j^B|. \quad (3.3)$$

Или, учитывая равенство директивных сроков,

$$\left| \sum_{j \in N} T_j^A(\pi) - \sum_{j \in N} T_j^B(\pi) \right| \leq \sum_{j \in N} |C_j^A - C_j^B|. \quad (3.4)$$

Учитывая свойства раннего расписания, заметим, что

$$\begin{aligned} |C_{j_1}^A - C_{j_1}^B| &= |r_{j_1}^A - r_{j_1}^B| \leq \max_{j \in N} |r_j^A - r_j^B| \\ |C_{j_k}^A - C_{j_k}^B| &\leq \max\{|r_{j_k}^A - r_{j_k}^B|, |C_{j_{k-1}}^A - C_{j_{k-1}}^B|\} \leq \max_{j \in N} |r_j^A - r_j^B|, \\ &k = 2, 3, \dots, n. \end{aligned}$$

Отсюда и из неравенства (3.4) получаем утверждение леммы.

Лемма 3. Пусть примеры A и B имеют одинаковые моменты поступления и директивные сроки:

$$r_j^A = r_j^B, \quad d_j^A = d_j^B, \quad j \in N.$$

Тогда для любого расписания π

$$\left| \sum_{j \in N} T_j^A(\pi) - \sum_{j \in N} T_j^B(\pi) \right| \leq n \sum_{j \in N} |p_j^A - p_j^B|.$$

Доказательство. При условиях леммы также выполняется неравенство (3.4):

$$\left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| \leq \sum_{j \in N} |C_j^A - C_j^B|.$$

Учитывая свойства раннего расписания и равенство моментов поступления, получаем

$$\begin{aligned} |C_{j_1}^A - C_{j_1}^B| &= |p_{j_1}^A - p_{j_1}^B| \leq \sum_{j \in N} |p_j^A - p_j^B| \\ |C_{j_k}^A - C_{j_k}^B| &\leq |p_{j_k}^A - p_{j_k}^B| + |C_{j_{k-1}}^A - C_{j_{k-1}}^B| \leq \sum_{j \in N} |p_j^A - p_j^B|, \quad k = 2, 3, \dots, n. \end{aligned}$$

Отсюда и из неравенства (3.4) получаем утверждение леммы.

Лемма 4. Пусть примеры A и B имеют одинаковые моменты поступления и продолжительности обслуживания:

$$r_j^A = r_j^B, \quad p_j^A = p_j^B, \quad j \in N.$$

Тогда для любого расписания π

$$\left| \sum_{j \in N} T_j^A(\pi) - \sum_{j \in N} T_j^B(\pi) \right| \leq \sum_{j \in N} |d_j^A - d_j^B|.$$

Доказательство. При условиях леммы $C_{j_k}^A = C_{j_k}^B$, $k \in N$, и неравенство (3.3) имеет вид

$$\left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| \leq \sum_{j \in N} |d_j^A - d_j^B|$$

т.е. утверждение леммы выполняется.

Теорема 2. Функция, определенная на пространстве примеров $\Omega \times \Omega$,

$$\rho(A, B) = n \max_{j \in N} |r_j^A - R_j^B| + n \sum_{j \in N} |p_j^A - p_j^B| + \sum_{j \in N} |d_j^A - d_j^B|$$

удовлетворяет аксиомам метрики.

Доказательство. Очевидно, что функция $\rho(A, B)$ симметрична и неотрицательна, причем $\rho(A, B) = 0$ тогда и только тогда, когда $A = B$. Неравенство треугольника выполняется в силу свойств модуля суммы.

Лемма 5. Для любых примеров A и B и любого расписания π справедливо неравенство

$$\left| \sum_{j \in N} T_j^A(\pi) - \sum_{j \in N} T_j^B(\pi) \right| \leq \rho(A, B). \quad (3.5)$$

Доказательство. Пусть пример C имеет такие же моменты поступления и продолжительности обслуживания, как и пример A , а директивные сроки, как у примера B . Пусть далее пример D имеет моменты поступления, как у примера A , а директивные сроки и продолжительности обслуживания, как у примера B , тогда, используя леммы 2–4, получаем

$$\begin{aligned} \left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B \right| &\leq \left| \sum_{j \in N} T_j^B - \sum_{j \in N} T_j^D \right| + \left| \sum_{j \in N} T_j^D - \sum_{j \in N} T_j^C \right| + \left| \sum_{j \in N} T_j^A - \sum_{j \in N} T_j^C \right| \leq \\ &\leq n \max_{j \in N} |r_j^A - r_j^B| + n \sum_{j \in N} |p_j^A - p_j^B| + \sum_{j \in N} |d_j^A - d_j^B| = \rho(A, B). \end{aligned}$$

3.3. Метод изменения параметров

Теорема 3. Пусть π^A и π^B – оптимальные расписания для примеров A и B , тогда

$$\sum_{j \in N} T_j^A(\pi^B) - \sum_{j \in N} T_j^A(\pi^A) \leq 2\rho(A, B). \quad (3.6)$$

Доказательство. Используя лемму 4, получаем

$$\begin{aligned} \sum_{j \in N} T_j^A(\pi^B) - \sum_{j \in N} T_j^A(\pi^A) &= \left[\sum_{j \in N} T_j^A(\pi^B) - \sum_{j \in N} T_j^B(\pi^B) \right] + \left[\sum_{j \in N} T_j^B(\pi^B) - \sum_{j \in N} T_j^B(\pi^A) \right] + \\ &+ \left[\sum_{j \in N} T_j^B(\pi^A) - \sum_{j \in N} T_j^A(\pi^A) \right] \leq 2\rho(A, B). \end{aligned}$$

Доказанная теорема позволяет использовать новый метод решения задачи $1|r_j| \sum T_j$, названный методом изменения параметров. Метод состоит в том, чтобы использовать оптимальное расписание некоторого полиномиально или псевдополиномиально решаемого примера B в качестве расписания для исходного примера A . Теорема 3 позволяет оценить сверху абсолютную погрешность значения целевой функции такого решения с помощью функции $\rho(A, B)$. Естественно конструировать пример B так, чтобы минимизировать функцию $\rho(A, B)$. Таким образом, задача $1|r_j| \sum T_j$ заменяется задачей на минимизацию функции метрики.

Рассмотрим случай, когда искомый пример B должен принадлежать некоторому полиномиально или псевдополиномиально разрешимому классу примеров, задаваемому системой неравенств

$$\mathcal{A} \cdot R^B + \mathcal{B} \cdot P^B + \mathcal{C} \cdot D^B \leq H, \tag{3.7}$$

где $R^B = (r_1^B, \dots, r_n^B)^T$, $P^B = (p_1^B, \dots, p_n^B)^T$, $D^B = (d_1^B, \dots, d_n^B)^T$, $p_j^B \geq 0$, $r_j^B \geq 0$, $j \in N$, T – символ транспонирования, \mathcal{A} , \mathcal{B} , \mathcal{C} – матрицы $m \times n$, H – столбец из m элементов.

В этом случае задача минимизации функции метрики может быть поставлена как задача линейного программирования:

$$\min n(y^r - x^r) + n \sum_{j \in N} (y_j^p - x_j^p) + \sum_{j \in N} (y_j^d - x_j^d),$$

при условиях

$$\begin{aligned} x^r &\leq r_j^A - r_j^B \leq y^r, \\ x_j^p &\leq p_j^A - p_j^B \leq y_j^p, \\ x_j^d &\leq d_j^A - d_j^B \leq y_j^d, \\ r_j^B &\geq 0, \quad p_j^B \geq 0, \quad j \in N, \\ \mathcal{A} \cdot R^B + \mathcal{B} \cdot P^B + \mathcal{C} \cdot D^B &\leq H. \end{aligned}$$

Неизвестными в данной задаче являются $7n + 2$ переменных:

$$r_j^B, p_j^B, d_j^B, x_j^p, y_j^p, x_j^d, y_j^d, x^r, y^r, j \in N.$$

Тем не менее сепарабельность функции $\rho(A, B)$ во многих случаях позволяет находить ее минимум гораздо проще, без использования методов линейного программирования. Необходимо заметить, что для всех известных полиномиальных и псевдополиномиальных разрешимых случаев задачи выполняется система линейных неравенств типа (3.7).

3.4. Применение метода изменения параметров для других задач теории расписаний

Описанный метод не является жестко привязанным к виду целевой функции, что позволяет использовать его для решения других задач теории расписаний. Теорему 3 можно обобщить на случай общего вида целевой функции $F(\pi)$.

Теорема 4. Пусть $F(\pi)$ – некоторая регулярная целевая функция, а $\rho(A, B)$ – функция метрики для любых A, B, π , удовлетворяющая неравенству

$$|F^A(\pi) - F^B(\pi)| \leq \rho(A, B). \tag{3.8}$$

Пусть далее π^A и π^B – оптимальные расписания для примеров A и B , тогда

$$F^A(\pi^B) - F^A(\pi^A) \leq 2\rho(A, B).$$

Доказательство. Доказательство теоремы 4 повторяет доказательство теоремы (3.3) с заменой $\sum_{j \in N} T_j$ на F .

Таким образом, для применения метода изменения параметров достаточно построить функцию $\rho(A, B)$, удовлетворяющую неравенству (3.8). Такие функции были построены ранее для задач $1 \parallel \sum T_j$ [14] и $1|r_j|L_{\max}$ (см. [22]). Здесь мы приведем вариант построения таких функций для общих случаев аддитивной и “минимаксной” целевой функции.

Лемма 6 (см. [13]). В случае аддитивной целевой функции вида

$$F(\pi) = \sum_{j \in N} \phi_j(\pi, r_1, \dots, r_n, p_1, \dots, p_n, d_j)$$

Таблица 1. Классы примеров, использованные в численных экспериментах

Класс примеров	Метрика между примером B класса и произвольным примером A
$\{\mathcal{PR} : p_j = p, r_j = r, j \in N\}$	$\rho(A, B) = n \sum_{j=1}^n p_j^A - p + n \max_{j \in N} r_j^A - r $
$\{\mathcal{PD} : p_j = p, d_j = d, j \in N\}$	$\rho(A, B) = n \sum_{j \in N} p_j^A - p + \sum_{j \in N} d_j^A - d $
$\{\mathcal{RD} : r_j = r, d_j = d, j \in N\}$	$\rho(A, B) = n \max_{j \in N} r_j^A - r + \sum_{j \in N} d_j^A - d $
$\{\mathcal{P} : p_j = p, j \in N\}$	$\rho(A, B) = n \sum_{j \in N} p_j^A - p $
$\{\mathcal{R0} : r_j = 0, j \in N\}$	$\rho(A, B) = n \max_{j \in N} r_j^A - r $

функция

$$\rho(A, B) = \sum_{j \in N} \sum_{i \in N} (R_{ji}|r_j^A - r_j^B| + P_{ji}|p_j^A - p_j^B|) + \sum_{j \in N} D_j |d_j^A - d_j^B| \tag{3.9}$$

удовлетворяет неравенству (3.8). Здесь R_{ji}, P_{ji} – константы Липшица для функции ϕ_i по переменным r_j и p_j , D_j – константа Липшица для функции ϕ_j по переменной $d_j, i, j \in N$.

Лемма 7 (см. [13]). В случае “минимаксной” целевой функции вида

$$F(\pi) = \max_{j \in N} \phi_j(\pi, r_1, \dots, r_n, p_1, \dots, p_n, d_j)$$

функция

$$\rho(A, B) = \sum_{j \in N} (R_j|r_j^A - r_j^B| + P_j|p_j^A - p_j^B|) + D \max_{j \in N} |d_j^A - d_j^B| \tag{3.10}$$

удовлетворяет неравенству (3.8). Здесь R_j, P_j – наибольшие константы Липшица из констант для функций ϕ_i по переменным r_j и p_j , D – наибольшая из констант Липшица для функций ϕ_j по переменным $d_j, i, j \in N$.

Заметим, что функции (3.9) и (3.10) сепарабельны, что значительно облегчает нахождение их минимумов.

3.5. Численные эксперименты

Для определения эффективности предложенной схемы была проведена серия численных экспериментов. Классы, в которых проводился поиск полиномиально разрешимых примеров, представлены в табл. 1.

В первых трех классах решением является расписание, упорядоченное по неубыванию свободного параметра. Алгоритмы решения задач последних двух классов представлены в [21] и [23] и имеют сложности $O(n^7)$ и $O(n^4 \sum p_j)$ операций соответственно.

Для нахождения в указанных классах полиномиально разрешимого примера B , ближайшего к заданному примеру, необходимо найти минимум функций:

$$f(r) = n \max_{j \in N} |r_j^A - r|, \tag{3.11}$$

$$g(p) = n \sum_{j=1}^n |p_j^A - p|, \tag{3.12}$$

$$h(d) = \sum_{j \in N} |d_j^A - d|. \tag{3.13}$$

Лемма 8. 1. Минимум функции (3.11) достигается в точке $r = (r_{\max}^A + r_{\min}^A)/2$, где $r_{\max}^A = \max_{j \in N} r_j^A$, $r_{\min}^A = \min_{j \in N} r_j^A$.

2. Минимум функции (3.12) достигается в некоторой точке $p \in \{p_1^A, \dots, p_n^A\}$.

3. Минимум функции (3.13) достигается в некоторой точке $d \in \{d_1^A, \dots, d_n^A\}$.

Доказательство. Функция $f(r)$ представима в виде

$$n \max_{j \in N} |r_j^A - r| = n \max\{r - r_{\min}^A, r_{\max}^A - r\} = n \left(\frac{r_{\max}^A - r_{\min}^A}{2} + \left| r - \frac{r_{\max}^A + r_{\min}^A}{2} \right| \right)$$

и, очевидно, имеет минимум в точке $\frac{r_{\max}^A + r_{\min}^A}{2}$.

Пусть функция $g(p)$ имеет минимум в точке p_0 , тогда либо $f'(p_0) = 0$, либо $p_0 \in \{p_1^A, \dots, p_n^A\}$. Поскольку $g(p)$ – кусочно-линейная функция, обращение ее производной в нуль означает, что функция является константой на некотором интервале $[p_k^A, p_{k+1}^A]$, $k = 1, 2, \dots, n - 1$, а значит, граничные точки p_k^A и p_{k+1}^A также являются точками минимума.

Последнее утверждение леммы о минимуме функции $h(d)$ доказывается аналогично.

Было проведено несколько серий экспериментов. Во всех сериях использовались примеры с параметрами, распределенными равномерно на интервалах $[1, 100]$ для p_j^A , $[p_j, \sum_{j \in N} p_j]$ для d_j^A и $[0, d_j - p_j]$ для r_j^A , $j \in N$. В первой серии экспериментов оценивалась величина различия между правой и левой частями неравенства (3.5). Данная величина позволяет оценить погрешность метода. Для каждого $n = 10, 20, \dots, 100$ генерировалось 10 000 пар примеров. Используемые в экспериментах расписания генерировались случайным образом. Для каждой пары вычислялась величина $\frac{|\sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B|}{\rho(A, B)}$. Также для определения параметров, имеющих наибольшее влияние на функцию метрики, вычислялись процентные величины вкладов частей метрики, зависящих от продолжительностей обслуживания, директивных сроков и моментов поступления.

Результаты представлены в табл. 2. Среднее значение $\frac{|\sum_{j \in N} T_j^A - \sum_{j \in N} T_j^B|}{\rho(A, B)}$ меняется от 5 до 10% при росте n , а части функции метрики, зависящие от продолжительностей обслуживания, директивных сроков и моментов поступления дают вклады приблизительно 35, 20 и 25% в общую величину функции соответственно.

Вторая серия экспериментов проводилась для проверки метода изменения параметров по следующей схеме. Рассматривались значения $n = 4, 5, \dots, 10$, для каждого n генерировались по 10000 примеров. К каждому примеру применялась вышеописанная схема для нахождения приближенного решения со значением целевой функции F_e , затем с помощью алгоритма ветвей и границ искалось точное решение с оптимальным значением целевой функции F^* . Далее вычислялось Δ – отношение реальной погрешности схемы $\delta = F_e - F^*$ к ее верхней оценке, определяемой неравенством (3.6):

$$\Delta = \frac{F_e - F^*}{2\rho(A, B)}.$$

Было обнаружено, что если поиск полиномиально разрешимого примера ведется в классе RD , то средняя погрешность решения растет от 20 до 30% от верхней оценки (3.6) при увеличении n . Это показывает, что расписание по возрастанию продолжительностей обслуживания плохо применимо для примеров с выбранным распределением параметров. Для остальных классов сред-

Таблица 2. Средняя разница между целевыми функциями и доли составных частей метрики

n	$\frac{ \sum T_j^A - T_j^B }{\rho}$	$\frac{\rho_r}{\rho}$	$\frac{\rho_p}{\rho}$	$\frac{\rho_d}{\rho}$
10	11.7%	35.6%	42.3%	20.6%
20	10.4%	39.7%	39.4%	19.4%
40	8.9%	42.4%	37.4%	18.6%
60	7.8%	43.6%	36.6%	18.3%
80	7.3%	44.4%	34.4%	18%
100	6.7%	44.9%	35.7%	17.9%

Таблица 3. Средняя экспериментальная погрешность в процентах от теоретической

n	\mathcal{PR}	\mathcal{PD}	\mathcal{RD}	\mathcal{P}	$\mathcal{R0}$
4	2.5%	4.6%	20.8%	1.8%	2.9%
5	2.6%	4.8%	23.1%	1.9%	2.8%
6	2.6%	4.6%	24.6%	1.9%	2.7%
7	2.6%	4.7%	26%	1.9%	2.5%
8	2.5%	4.6%	27%	2%	2.3%
9	2.4%	4.7%	27.9%	2%	2.2%
10	2.4%	4.6%	28.6%	1.9%	2.1%

няя погрешность решения не зависит от n и составляет несколько процентов от максимальной теоретической. Столь малая погрешность обусловлена тем, что примерно в 20% случаев метод изменения параметров давал точное решение задачи. Зависимость средней ошибки Δ от n представлена в табл. 3 (см. [13]).

СПИСОК ЛИТЕРАТУРЫ

1. *Brucker P., Knust S.* Complex scheduling. Berlin Heidelberg: Springer-Verlag, 2011.
2. *Wagner H.M.* An integer linear-programming model for machine scheduling // *Naval Res. Logist. Quart.* 1959. V. 6. № 2. P. 131–140.
3. *Rothkopf M.H.* Scheduling independent tasks on parallel processors // *Management Sci.* 1966. V. 12. № 5. P. 437–447.
4. *Ignall E., Schrage L.* Application of the branch and bound technique to some flow-shop scheduling problems // *Operat. Res.* 1965. V. 13. № 3. P. 400–412.
5. *Lemtyuzhnikova D., Leonov V.* Large-scale problems with quasi-block matrices // *J. of Comp. and Syst. Sci. Int.* 2019. V. 58. № 4. P. 571–578.
6. *Werner F.* A Survey of Genetic Algorithms for Shop Scheduling Problems // P. Siarry: *Heuristics: Theory and Applications*, Nova Sci. Publ., 2013. P. 161–222.
7. *Zuo L., Shu L., Dong S., Zhu C., Hara T.* A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing // *IEEE Access.* 2015. V. 3. P. 2687–2699.
8. *Pan Q.* An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling // *Europ. J. Operat. Res.* 2016. V. 250. № 3. 702–714.
9. *Sels V., Coelho J., Dias A., Vanhoucke M.* Hybrid tabu search and a truncated branch-and-bound for the unrelated parallel machine scheduling problem // *Comput. Operat. Res.* 2015. V. 53. P. 107–117.
10. *Lazarev A.* Estimation of absolute error in scheduling problems of minimizing the maximum lateness // *Dokl. Math.* 2007. V. 76. P. 572–574.
11. *Лазарев А.А.* Теория расписаний: методы и алгоритмы. М.: ИПУ РАН, 2019. 408 с.
12. *Лазарев А.А., Архипов Д.И.* Оценка абсолютной погрешности и полиномиальной разрешимости для классической NP-трудной задачи теории расписаний // *Докл. АН.* 2018. Т. 480. № 5. С. 523–527.
13. *Лазарев А.А., Корнев П.С., Сологуб А.А.* Метрика для задачи минимизации суммарного запаздывания // *Управление большими системами.* 2015. Вып. 57. С. 123–137.

14. Лазарев А.А., Кварацхелия А.Г. Метрики в задачах теории расписаний // Докл. АН. 2010. Т. 432. № 6. С. 746–749.
15. Du J., Leung J.Y.-T. Minimizing total tardiness on one processor is *NP*-hard // Math. Operat. Res. 1990. V. 15. P. 483–495.
16. Lawler E.L. A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness // Ann. Discrete Math. 1977. V. 1. P. 331–342.
17. Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G. Optimization and approximation in deterministic sequencing and scheduling: a survey // Ann. Discrete Math. 1979. V. 5. P. 287–326.
18. Lawler E.L. A fully polynomial approximation scheme for the total tardiness problem // Operat. Res. Lett. 1982. V. 1. № 6. P. 207–208.
19. Kovalyov M.Y. Improving the complexities of approximation algorithms for optimization problems // Operat. Res. Lett. 1995. V. 17. P. 85–87.
20. Lazarev A.A., Werner F. Algorithms for special single machine total tardiness problem and an application to the even-odd partition problem // Math. and Comp. Model. 2009. № 49. P. 2078–2089.
21. Baptiste Ph. Scheduling equal-length jobs on identical parallel machines // Discret. Appl. Math. 2000. № 103. P. 21–32.
22. Лазарев А.А., Садыков Р.Р., Севастьянов С.В. Схема приближенного решения проблемы $1|r_j|L_{\max}$ // Дискретный анализ и исслед. операций. 2006. Сер. 2. Т. 13. № 1. С. 57–76.
23. Лазарев А.А., Гафаров Е.Р. Теория расписаний. Минимизация суммарного запаздывания для одного прибора // М.: Научное издание. ВЦ им. А.А. Дородницына РАН, 2006. 134 с.