ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Январь—июнь 2006. Серия 2. Том 13, № 1. 57–76

УДК 519.854.2

СХЕМА ПРИБЛИЖЁННОГО РЕШЕНИЯ ЗАДАЧИ $1 \mid r_j \mid L_{\max}$ [*)]

А. А. Лазарев, Р. Р. Садыков, С. В. Севастьянов

Рассматривается NP-трудная в сильном смысле задача теории расписаний о минимизации максимального временно́го смещения на одном приборе при неодновременном поступлении работ. Представлена схема приближённого решения, основанная на отыскании по заданному примеру другого (наиболее близкого в некоторой метрике) примера, принадлежащего к известному полиномиально разрешимому классу примеров. Для нескольких конкретных вариантов схемы (с использованием различных полиномиально разрешимых классов примеров) найдены аналитические формулы, позволяющие по любому заданному примеру легко вычислить оценку абсолютной погрешности его приближённого решения.

Введение

На одном приборе необходимо обслужить требования множества $N = \{1, 2, \dots, n\}$. Запрещается одновременное обслуживание и прерывания при обслуживании требований. Для требования $j \in N$ имеем: r_j — минимально возможный момент начала обслуживания, $p_j \geqslant 0$ — продолжительность обслуживания, d_j — директивный срок завершения обслуживания.

Расписание задается совокупностью $S = \{s_j \mid j \in N\}$ моментов начала обслуживания требований. Расписание S называется *допустимым*, если $s_j(S) \geqslant r_j$ при любом $j \in N$. Момент завершения обслуживания требования $j \in N$ в расписании S обозначим через $c_j(S)$. Тогда $L_j(S) = c_j(S) - d_j$, $j \in N$, обозначает временно́е смещение требования j в расписании S и $L(S) = \max\limits_{j \in N} L_j(S)$ — максимальное временно́е смещение расписания S. Задача состоит в нахождении расписания S^* с наименьшим значением максимального временно́го смещения.

Такая задача обычно обозначается через $1 \mid r_j \mid L_{\max}$ [7]. Алгоритмы её решения используются при решении других задач теории расписаний,

---

[*)]Исследование третьего автора выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 05–01–00960).

© **2006 Лазарев А. А., Садыков Р. Р., Севастьянов С. В.**

например, задачи job shop [3], задачи минимизации (взвешенного) числа запаздывающих требований [16].

Интенсивные работы по разработке методов решения этой задачи ведутся с начала 50-х годов прошлого века. В [13] показано, что общий случай задачи $1 \mid r_j \mid L_{\max}$ является NP-трудным в сильном смысле. Был выделен ряд полиномиально разрешимых случаев задачи, начиная с раннего результата из [10] для случая $r_j = 0,\ j \in N$, когда решением является расписание, построенное по правилу EDD, т. е. расписание, в котором требования упорядочены по неубыванию директивных сроков. Задачи $1 \mid\prec; r_j \mid L_{\max}$, $1 \mid\prec; p_j = p; r_j \mid L_{\max}$ и $1 \mid\prec; r_j; \mathrm{pmtn} \mid L_{\max}$ с ограничениями предшествования для требований были рассмотрены в [4, 12, 18]. В [9] предложен полиномиальный алгоритм сложности $O(n^2 \log n)$ для специального случая, когда для каждого требования $j \in N$ заданного примера выполняется соотношение $d_j - r_j - \min\limits_i(d_i - r_i) \leqslant p_j$. Псевдополиномиальный алгоритм для NP-трудного случая, когда времена поступления и директивные сроки упорядочены в обратном порядке ($d_1 \leqslant \cdots \leqslant d_n$ и $r_1 \geqslant \cdots \geqslant r_n$), был разработан А. А. Лазаревым и О. М. Шульгиной [2].

Наиболее часто используемым точным методом типа ветвей и границ является алгоритм Карлье [6]. Этот метод неплохо показал себя на примерах большой размерности. Другие точные методы решения задачи рассматривались в [5, 11, 15].

В опубликованных работах имеется несколько подходов к решению задачи, гарантирующих относительную погрешность решения. К. Поттс [17] представил итерационную версию расширенного правила Джексона (IJ) и доказал, что $L_{\max}(\mathrm{IJ})/L^*_{\max} \leqslant \frac{3}{2}$. А. Холл и Д. Шмойс [8] модифицировали итерационную версию и предложили алгоритм (MIJ) с гарантированной оценкой $L_{\max}(\mathrm{MIJ})/L^*_{\max} \leqslant \frac{4}{3}$. Они также представили две аппроксимационные схемы, гарантирующие нахождение ε-приближённого решения за $O(n \log n + n(1/\varepsilon)^{O(1/\varepsilon^2)})$ и $O((n/\varepsilon)^{O(1/\varepsilon)})$ операций. М. Мастролилли [14] представил улучшенную аппроксимационную схему, выполняющуюся за время $O(n + (1/\varepsilon)^{O(1/\varepsilon)})$. Также известно несколько полиномиальных алгоритмов, гарантирующих абсолютную погрешность решения, которая не превосходит $\max\limits_{j \in N} p_j$.

В данной статье описывается другой подход для нахождения решения задачи с гарантированной абсолютной погрешностью в худшем случае. Идея подхода состоит в построении по заданному примеру A такого примера C (с тем же числом требований), который, во-первых, принадлежит некоторому известному полиномиально разрешимому классу примеров, и, во-вторых, отличается от примера A минимальным образом (в неко-

торой метрике $\rho(A, C)$). Применив полиномиальный алгоритм для решения примера C, мы найдем оптимальную перестановку его требований и применим её в качестве приближённого решения примера A. Как будет доказано, абсолютная погрешность полученного решения не будет превосходить $\rho(A, C)$.

Таким образом, в статье предложена эффективная комбинация двух классических подходов к решению NP-трудной задачи: 1) построения приближённых алгоритмов и 2) отыскания специальных случаев задачи, допускающих эффективное точное решение. Насколько нам известно, подобная схема комбинации этих подходов представлена впервые.

Структура статьи следующая. В разделе 1 приводятся основные обозначения и определения. В разделе 2 выводится формула, оценивающая абсолютное изменение оптимума заданного примера при изменении времён поступления и директивных сроков (но при фиксированных длительностях) требований. В разделе 3 описывается схема нахождения приближённого решения произвольного примера задачи. В подразделах 3.1 и 3.2 рассматриваются варианты схемы, основанные на двух полиномиально разрешимых случаях задачи. Для данных вариантов оценивается абсолютная погрешность получаемого решения через значения параметров требований исходного примера. Наконец, в заключении формулируются основные результаты, полученные в этой статье, и намечаются пути дальнейшего исследования.

1. Обозначения и определения основных понятий

В этом разделе приводятся основные обозначения и определения, которые будут использоваться в дальнейшем.

Через $L_j^A(S)$ и $c_j^A(S)$ будем обозначать временно́е смещение и время окончания обслуживания требования $j \in N$ в расписании S для примера A с параметрами требований $\{r_j^A, p_j^A, d_j^A\}$, $j \in N$. Соответственно, $L^A(S) = \max\limits_{j \in N} L_j^A(S)$ — максимальное временно́е смещение расписания S для примера A.

Определение 1. Для примера A каждая перестановка π требований множества N однозначно определяет *раннее расписание* S_π^A. В раннем расписании каждое требование $j \in N$ начинает обслуживаться в наиболее ранний допустимый момент времени: либо в момент его поступления r_j^A, либо сразу после окончания обслуживания предыдущего требования в соответствующей перестановке, т. е. если $\pi = (j_1, j_2, \dots, j_n)$ и $S_\pi^A = \{s_j \mid j \in N\}$, то

$$s_{j_1} = r_{j_1}^A,\ s_{j_k} = \max\{s_{j_{k-1}} + p_{j_{k-1}}^A, r_{j_k}^A\},\ k = 2, \dots, n.$$

В описываемых построениях ранние расписания играют исключительную роль, поскольку оптимальное расписание любого примера содержится в множестве ранних расписаний.

Через π^A и S^A обозначим оптимальную перестановку и оптимальное расписание примера A. В качестве оптимальных расписаний примера A будут рассматриваться только ранние, т. е. такие, что $S^A = S^A_{\pi^A}$.

Через $\Pi(N)$ будем обозначать множество перестановок требований множества N.

Определение 2. Пусть задан пример A на множестве требований N. Будем говорить, что пример B на том же множестве требований *наследует* у примера A параметр x, если $x_j^B = x_j^A$ при любом $j \in N$.

Определение 3. *Пример* $Q = \{(r_j^Q, p_j^Q, d_j^Q) \mid j \in N\}$ называется *инверсным* к примеру $P = \{(r_j^P, p_j^P, d_j^P) \mid j \in N\}$, если при любом $j \in N$ выполнены соотношения

$$r_j^Q = -d_j^P,\ p_j^Q = p_j^P,\ d_j^Q = -r_j^P.$$

Перестановка $\pi' = (i_n, i_{n-1}, \ldots, i_1)$ называется *инверсной* к перестановке $\pi = (i_1, \ldots, i_n)$.*Расписание* $S' = \{s'_j \mid j \in N\}$ называется *инверсным* к расписанию $S = \{s_j \mid j \in N\}$, если $s'_j = -s_j - p_j$ при любом $j \in N$.

Нетрудно видеть, что понятие инверсии является симметричным, т. е. если объект X является инверсным к Y, то Y инверсен к X. В частности, интервалы I_j и I'_j выполнения требования j во взаимно инверсных расписаниях S и S' симметричны относительно нуля.

Определение 4. Расписание S, допустимое для заданного примера $V = \{(r_j^V, p_j^V, d_j^V) \mid\ j \in N\}$, называется *вполне допустимым*, если каждое требование $j \in N$ выполняется в своём директивном интервале $[r_j^V, d_j^V]$.

Кроме того, через $\Delta = \Delta(V, S)$ будем обозначать минимальную величину (возможно, отрицательную), которую следует прибавить ко всем директивным срокам требований примера V, чтобы допустимое расписание S стало вполне допустимым для полученного примера $V(\Delta)$. Очевидно, что $\Delta(V, S) = L^V(S)$.

Определение 5. Для двух примеров A и B определим следующие функции:

$$\rho_d(A, B) = \max_{j \in N}\{d_j^A - d_j^B\} + \max_{j \in N}\{d_j^B - d_j^A\}$$

$$\rho_r(A, B) = \max_{j \in N}\{r_j^A - r_j^B\} + \max_{j \in N}\{r_j^B - r_j^A\}$$

$$\rho(A,B) = \rho_d(A,B) + \rho_r(A,B).$$

Нетрудно убедиться, что функция $\rho(A,B)$ удовлетворяет свойствам метрики и её можно рассматривать как *расстояние* между примерами A и B.

2. Абсолютная погрешность приближённого решения

Лемма 1. *Пусть пример B наследует у примера A моменты поступления и длительности обслуживания требований. Тогда для любого допустимого (для обоих примеров) расписания S верно соотношение*

$$L^B(S) - L^A(S) \leqslant \max_{j\in N}\{d_j^A - d_j^B\}. \tag{1}$$

Доказательство. Для любого $i \in N$ имеем

$$L^A(S) + \max_{j\in N}\{d_j^A - d_j^B\} \geqslant c_i(S) - d_i^A + d_i^A - d_i^B = c_i(S) - d_i^B.$$

Следовательно, $L^A(S) + \max\limits_{j\in N}\{d_j^A - d_j^B\} \geqslant \max\limits_{i\in N}(c_i(S) - d_i^B) = L^B(S)$. Лемма 1 доказана.

Лемма 2. *Пусть пример B наследует у примера A моменты поступления и длительности обслуживания требований. Тогда*

$$0 \leqslant L^A(S^A_{\pi^B}) - L^A(S^A) \leqslant \rho_d(A,B).$$

Доказательство. Заменив в (1) S на S^A, получим

$$L^A(S^A) + \max_{j\in N}\{d_j^A - d_j^B\} \geqslant L^B(S^A) = L^B(S^B_{\pi^A}). \tag{2}$$

Поменяв местами обозначения примеров A и B, из (1) получим

$$L^B(S^B) + \max_{j\in N}\{d_j^B - d_j^A\} \geqslant L^A(S^A_{\pi^B}). \tag{3}$$

По определению S^B имеем

$$L^B(S^B_{\pi^A}) \geqslant L^B(S^B). \tag{4}$$

Из (2)–(4) следует, что

$$L^A(S^A) + \max_{j\in N}\{d_j^A - d_j^B\} \geqslant L^A(S^A_{\pi^B}) - \max_{j\in N}\{d_j^B - d_j^A\},$$

или $L^A(S^A) + \rho_d(A,B) \geqslant L^A(S^A_{\pi^B}) \geqslant L^A(S^A)$. Лемма 2 доказана.

Поскольку $\rho_d(A,B) = \rho_d(B,A)$, поменяв местами A и B, из леммы 2 получаем

Следствие 1. $\rho_d(A,B) \geqslant L^B(S^B_{\pi^A}) - L^B(S^B) \geqslant 0$.

Лемма 3. *Пусть V и W — взаимно инверсные примеры с множеством требований N, а π и π' — взаимно инверсные перестановки из $\Pi(N)$. Тогда $L^V(S^V_\pi) = L^W(S^W_{\pi'})$.*

Доказательство. Пусть $\Delta = \Delta(V, S^V_\pi)$. Так как расписание S^V_π вполне допустимо для примера $V(\Delta)$, то расписание S', инверсное к S^V_π, вполне допустимо для примера W', инверсного к примеру $V(\Delta)$. Это означает, что $L^{W'}(S') \leqslant 0$. Замечаем, что пример W' отличается от примера W (инверсного к V) тем, что все r_j уменьшены на величину Δ. Если расписание S' сдвинуть вправо на Δ, то полученное расписание S'' будет допустимым для примера W, причем $L^W(S'') \leqslant \Delta = L^V(S^V_\pi)$. Так как требования в расписании S'' выполняются в порядке π', то $L^W(S^W_{\pi'}) \leqslant L^W(S'') \leqslant L^V(S^V_\pi)$.

Поменяв местами V с W, а π с π', получим обратное неравенство $L^V(S^V_\pi) \leqslant L^W(S^W_{\pi'})$. Отсюда вытекает равенство $L^V(S^V_\pi) = L^W(S^W_{\pi'})$. Лемма 3 доказана.

Следствие 2. *Если перестановка π является оптимальной для примера V, то инверсная перестановка π' будет оптимальной для примера, инверсного к V.*

Лемма 4. *Пусть пример C наследует у примера B продолжительности обслуживания и директивные сроки. Тогда*

$$0 \leqslant L^B(S^B_{\pi^C}) - L^B(S^B) \leqslant \rho_r(B,C).$$

Доказательство. Рассмотрим два инверсных к B и C примера E и F с параметрами требований $r^E_j = -d^B_j$, $p^E_j = p^B_j$, $d^E_j = -r^B_j$ и $r^F_j = -d^C_j$, $p^F_j = p^C_j$, $d^F_j = -r^C_j$. Пусть π^E и π^F — перестановки, инверсные к π^B и π^C соответственно. Согласно следствию 2 перестановки π^E и π^F оптимальны для примеров E и F соответственно. По лемме 2

$$\rho_d(E,F) \geqslant L^E(S^E_{\pi^F}) - L^E(S^E) \geqslant 0.$$

Согласно лемме 3 имеем $L^B(S^B) = L^E(S^E)$ и $L^B(S^B_{\pi^C}) = L^E(S^E_{\pi^F})$. Следовательно,

$$\rho_d(E,F) \geqslant L^B(S^B_{\pi^C}) - L^B(S^B) \geqslant 0. \tag{5}$$

Имеем $\rho_d(E,F) = \max\limits_{j\in N}\{d_j^E - d_j^F\} + \max\limits_{j\in N}\{d_j^F - d_j^E\} = \max\limits_{j\in N}\{r_j^C - r_j^B\} + \max\limits_{j\in N}\{r_j^B - r_j^C\} = \rho_r(B,C)$. Отсюда и из (5) следует утверждение леммы. Лемма 4 доказана.

Теорема 1. *Пусть пример C наследует у примера A длительности обслуживания требований. Тогда* $0 \leqslant L^A(S^A_{\pi^C}) - L^A(S^A) \leqslant \rho(A,C)$.

Доказательство. Неравенство $0 \leqslant L^A(S^A_{\pi^C}) - L^A(S^A)$ вытекает из оптимальности расписания S^A для примера A. Докажем неравенство $L^A(S^A_{\pi^C}) - L^A(S^A) \leqslant \rho(A,C)$.

Рассмотрим пример B с параметрами требований $r_j^B = r_j^A$, $p_j^B = p_j^A = p_j^C$, $d_j^B = d_j^C$ и его оптимальное расписание S^B. По лемме 4 имеем

$$L^B(S^B_{\pi^C}) - L^B(S^B) \leqslant \rho_r(B,C) = \rho_r(A,C). \tag{6}$$

Из оптимальности π^B для примера B следует, что

$$L^B(S^B) \leqslant L^B(S^B_{\pi^A}). \tag{7}$$

Из леммы 1 для примеров A и B и расписаний $S^A_{\pi^C} = S^B_{\pi^C}$ и $S^B_{\pi^A} = S^A_{\pi^A} = S^A$ получаем

$$L^A(S^A_{\pi^C}) - L^B(S^B_{\pi^C}) \leqslant \max_{j\in N}\{d_j^B - d_j^A\}, \tag{8}$$

$$L^B(S^B_{\pi^A}) - L^A(S^A) \leqslant \max_{j\in N}\{d_j^A - d_j^B\}. \tag{9}$$

Сложив неравенства (6)–(9), получим

$$L^A(S^A_{\pi^C}) - L^A(S^A) \leqslant \rho_r(A,C) + \rho_d(A,B) = \rho_r(A,C) + \rho_d(A,C) = \rho(A,C).$$

Теорема 1 доказана.

Следствие 3. $0 \leqslant L^C(S^C_{\pi^A}) - L^C(S^C) \leqslant \rho(A,C)$.

Данный результат может быть применён к решению задачи $1 \mid r_j \mid L_{\max}$ в условиях неопределённости времён поступления и/или директивных сроков требований. Пусть для заданного примера A каким-то образом найдено оптимальное расписание и оптимальный порядок π_A выполнения требований, а затем поступила новая информация о значениях параметров $\{r_j, d_j\}$ (длительности обслуживания требований при этом остались неизменными). Новый пример обозначим через C. Возникает вопрос: каково качество решения, получаемого путём применения найденной последовательности требований π_A к примеру C? Следствие 3 позволяет оценить абсолютную погрешность такого решения через расстояние ρ между примерами A и C.

Другое применение следствия 3 связано с разработкой эффективных алгоритмов приближённого решения NP-трудной задачи $1 \mid r_j \mid L_{\max}$. Схема приближённого решения с использованием алгоритмов, разработанных для специальных полиномиально разрешимых случаев задачи, также как и варианты её применения, рассматриваются в следующем разделе.

3. Схема нахождения приближённого решения

Идея общей схемы приближённого решения задачи $1 \mid r_j \mid L_{\max}$ заключается в выполнении следующих двух шагов. На первом шаге по заданному примеру A находится такое изменение его параметров r_j^A, d_j^A, чтобы полученный пример C с параметрами требований $r_j^C, p_j^C = p_j^A, d_j^C$ принадлежал заданному полиномиально разрешимому классу примеров исходной (NP-трудной) задачи. На втором шаге для решения примера C используется уже известный для данного класса примеров полиномиальный алгоритм, а затем найденную им оптимальную перестановку требований применяем к примеру A.

Согласно теореме 1 абсолютная погрешность такого решения не будет превосходить расстояния $\rho(A, C)$ между примерами A и C. Ясно, что минимальная оценка погрешности получится в случае, когда на первом шаге этой схемы будет найден пример C, минимизирующий значение $\rho(A, C)$. Таким образом, решение задачи сводится к отысканию для заданного примера A наиболее близкого примера из заданного полиномиально разрешимого класса.

Рассмотрим случай, когда некоторый полиномиально разрешимый класс примеров рассматриваемой задачи определяется системой линейных неравенств вида:

$$\boldsymbol{A} * R^C + \boldsymbol{B} * P^C + \boldsymbol{C} * D^C \leqslant H \tag{10}$$

(при ограничениях $p_j^C \geqslant 0$, $j \in N$), где $R^C = (r_1^C, \ldots, r_n^C)^T$, $P^C = (p_1^C, \ldots, p_n^C)^T$, $D^C = (d_1^C, \ldots, d_n^C)^T$, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ — матрицы размера $m \times n$ и $H = (h_1, \ldots, h_m)^T$ — m-мерный вектор (верхний индекс ^{T} обозначает транспонирование). Тогда для отыскания в этом классе примера C, минимизирующего расстояние $\rho(A, C)$ (для заданного примера A), достаточно решить следующую задачу линейного программирования. Найти

$$\min\,(x_d - y_d + x_r - y_r) \tag{11}$$

при условиях:

$$
\begin{array}{l}
y_d \leqslant d_j^A - d_j^C \leqslant x_d, \\
y_r \leqslant r_j^A - r_j^C \leqslant x_r, \\
p_j^A = p_j^C, \qquad j \in N, \\
\boldsymbol{A} * R^C + \boldsymbol{B} * P^C + \boldsymbol{C} * D^C \leqslant H.
\end{array} \tag{11a}
$$

Конечно, при использовании в общей “схеме сведения” конкретных полиномиально разрешимых классов следует применять наименее трудоёмкие алгоритмы минимизации $\rho(A, C)$. В частности, с этой целью при использовании системы (11a) следует учитывать её специфику, проистекающую из особенностей матриц $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ и H.

Например, для специального случая [1], когда $d_j - r_j - p_j = 0,\ j \in N$, в системе линейных неравенств (10) матрицы $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ и H задаются следующим образом:

$$
\boldsymbol{A} = \boldsymbol{B} = (\boldsymbol{I} \oplus (-\boldsymbol{I}))^T, \quad \boldsymbol{C} = ((-\boldsymbol{I}) \oplus \boldsymbol{I})^T, \quad H = (h)^T,
$$

где $\boldsymbol{I}$ — единичная матрица размера n, h — $2n$-мерный нулевой вектор, $\boldsymbol{A} \oplus \boldsymbol{B}$ обозначает конкатенацию матрицы $\boldsymbol{A}$ размера $l \times p$ и матрицы $\boldsymbol{B}$ размера $l \times q$.

В качестве следующего примера возьмём класс примеров задачи $1 \mid r_j \mid L_{\max}$, когда $d_j = \delta,\ j \in N$, где δ — константа. Оптимальным для любого заданного примера из данного класса является такое расписание S', в котором требования упорядочены по неубыванию моментов поступления. Расписание S' находится за $O(n \log n)$ операций. Чтобы свести произвольный пример A задачи к примеру C из рассматриваемого класса, необходимо приравнять все директивные сроки к какой-либо константе δ. Сформулируем задачу линейного программирования, аналогичную (11)–(11a), для нахождения примера C с минимальным расстоянием $\rho(A, C)$: найти

$$
\min\ (x_d - y_d) \tag{12}
$$

при ограничении

$$
y_d \leqslant d_j^A - \delta \leqslant x_d, \quad j \in N. \tag{12a}
$$

Решением задачи (12)–(12a) будет произвольное значение δ и x_d, y_d такие, что $x_d - y_d = \max\limits_{j \in N} d_j^A - \min\limits_{j \in N} d_j^A$, т. е. расстояние $\rho(A, C)$ в данном случае не зависит от того, к какой константе δ были сведены директивные сроки требований примера A. В любом случае абсолютная погрешность расписания S' будет не больше разницы между максимальным и минимальным директивными сроками требований примера A.

Третьим примером полиномиально разрешимого класса, записываемого с помощью линейной системы вида (10), является *класс Хогевена* [9]. Пример $C = \{(r_j^C, p_j^C, d_j^C) \mid j \in N\}$ принадлежит классу Хогевена, если существует такая константа β, что для каждого требования $j \in N$ выполняются соотношения

$$d_j^C - r_j^C - p_j^C \leqslant \beta \leqslant d_j^C - r_j^C. \tag{13}$$

Заметим, что достаточно рассматривать лишь класс, определяемый данным свойством при $\beta = 0$. Действительно, если по примеру A построен пример C, принадлежащий классу Хогевена с $\beta \neq 0$, то вместо него можно взять пример C', полученный из C уменьшением всех d_j на β. При этом C' будет принадлежать классу Хогевена с $\beta = 0$ и $\rho(A, C) = \rho(A, C')$.

Класс Хогевена с $\beta = 0$ определяется неравенствами

$$d_j^C - r_j^C - p_j^C \leqslant 0, \quad -d_j^C + r_j^C \leqslant 0, \quad j \in N, \tag{14}$$

которые, очевидно, легко записываются в виде (10).

В следующих подразделах рассмотрены другие варианты применения предложенной общей схемы, которые не основаны на решении задачи (11)–(11a).

3.1. Вариант схемы для класса L примеров

Рассмотрим полиномиально разрешимый класс L примеров задачи $1 \mid r_j \mid L_{\max}$, предложенный Лазаревым [1]. Пример принадлежит данному классу примеров, если он удовлетворяет

Свойству P_L. Будем говорить, что пример $C = \{(r_j^C, p_j^C, d_j^C) \mid j \in N\}$ удовлетворяет *свойству* P_L, если существует такая нумерация требований $\{1, 2, \ldots, n\}$, что

$$d_1^C \leqslant \cdots \leqslant d_n^C; \quad \Delta_1^C \geqslant \cdots \geqslant \Delta_n^C, \tag{15}$$

где $\Delta_j^C = d_j^C - r_j^C - p_j^C$ обозначает *временно́й запас* требования j.

Далее для заданного примера A определим функцию

$$\rho^L(A) = \max_{i,j \in N} \rho_{ij}^L(A), \tag{16}$$

где

$$\rho_{ij}^L(A) = \min\{d_j^A - d_i^A, \Delta_j^A - \Delta_i^A\}. \tag{17}$$

Нетрудно видеть, что $\rho^L(A) \geqslant 0$ для любого примера A (например, $\rho_{ij}^L(A) = 0$ при $i = j$), причём, $\rho^L(A) = 0$, если и только если пример

A удовлетворяет свойству P_L. Таким образом, функцию $\rho^L(A)$ можно считать *мерой невыполнения* свойства P_L.

Для точного решения примеров задачи $1 \mid r_j \mid L_{\max}$ из класса L существует алгоритм сложности $O(n^3 \log n)$ [1].

Пусть задан пример A, не принадлежащий классу L. Применим к нему описанную выше схему приближённого решения в следующем виде. Сведём пример A к примеру C, наследующему у A длительности обслуживания и принадлежащему классу L. Заметим, что класс L не записывается с помощью системы (10), поскольку он не является выпуклым многогранником в $3n$-мерном параметрическом пространстве, а представляет собой семейство $n!$, вообще говоря, несвязных между собой конусов, что следует из (15). Поэтому для применения вышеизложенной схемы нахождения приближённого решения используется описываемый ниже нестандартный подход.

Следующая теорема позволяет найти такой пример C из класса L, для которого расстояние $\rho(A, C)$ будет минимальным.

Теорема 2. *Для любого примера A задачи $1 \mid r_j \mid L_{\max}$, не удовлетворяющего свойству P_L, и для любого примера C, наследующего длительности обслуживания примера A и удовлетворяющего свойству P_L, справедлива оценка расстояния между A и C :*

$$\rho(A, C) \geqslant \rho^L(A). \tag{18}$$

Оценка (18) *достигается на некотором примере C, который может быть найден за время $O(n \log n)$.*

Доказательство. Для любого примера C, удовлетворяющего свойству P_L, параметры любой пары требований $i, j \in N$ должны удовлетворять одному из следующих неравенств:

$$d_i^C - d_j^C \geqslant 0 \tag{19}$$

или

$$\Delta_i^C - \Delta_j^C \geqslant 0. \tag{20}$$

Если для пары (i, j) выполняется (19), то из определения расстояния $\rho(A, C)$ и определения (17) получаем

$$\rho(A, C) \geqslant \rho_d(A, C) \geqslant (d_j^A - d_j^C) + (d_i^C - d_i^A) \geqslant d_j^A - d_i^A \geqslant \rho_{ij}^L(A). \tag{21}$$

В противном случае из (17), (20) и определения функции $\rho(A, C)$ следуют

соотношения

$$\begin{aligned}\rho(A,C) &\geqslant (d_j^A - d_j^C) + (d_i^C - d_i^A) + (r_j^C - r_j^A) + (r_i^A - r_i^C)\\ &= (d_j^A - r_j^A - p_j) - (d_j^C - r_j^C - p_j) + (d_i^C - r_i^C - p_i) - (d_i^A - r_i^A - p_i)\\ &= \Delta_j^A - \Delta_j^C + \Delta_i^C - \Delta_i^A \geqslant \Delta_j^A - \Delta_i^A \geqslant \rho_{ij}^L(A).\end{aligned} \tag{22}$$

Из (21), (22) и (16) вытекает искомая оценка (18):

$$\rho(A,C) \geqslant \max_{i,j\in N} \rho_{ij}^L(A) = \rho^L(A).$$

Для доказательства достижимости оценки (18) на некотором примере A из класса L построим пример C, наследующий у A длительности и моменты поступления требований. В силу того, что параметры p_j и r_j не будут различаться для примеров A и C, мы будем обозначать их без верхнего индекса A и C соответственно.

Пронумеруем требования примера A по неубыванию $r_j + p_j$:

$$r_1 + p_1 \leqslant \cdots \leqslant r_n + p_n. \tag{23}$$

Определим возрастающую последовательность *разделяющих индексов* $0 = j_0 < j_1 < \cdots < j_K = n$:

$$\begin{aligned}&j_0 := 0;\ k := 0;\\ &\textbf{while } j_k < n \textbf{ do}\quad \{k := k+1;\ j_k := \arg\min_{\{j\,|\,j_{k-1}<j\leqslant n\}} d_j^A\}.\end{aligned} \tag{24}$$

Нетрудно видеть, что

$$d_{j_1}^A \leqslant d_{j_2}^A \leqslant \cdots \leqslant d_{j_K}^A. \tag{25}$$

Положим

$$d_j^C = \begin{cases} d_{j_1}^A & \text{при } j \leqslant j_1,\\ \min\{d_{j_k}^A, d_j^A - \Delta_j^A + \Delta_{j_{k-1}}^C\} & \text{при } j_{k-1} < j \leqslant j_k,\ k > 1.\end{cases} \tag{26}$$

Докажем, что так определённый пример C удовлетворяет свойству P_L. Для этого индукцией по $k = 1,\ldots,K$ докажем неравенства:

$$d_j^C \leqslant d_j^A \qquad (j \leqslant j_k); \tag{27}$$

$$d_i^C \leqslant d_j^C \qquad (i < j \leqslant j_k); \tag{28}$$

$$\Delta_i^C \geqslant \Delta_j^C \qquad (i < j \leqslant j_k). \tag{29}$$

$k = 1$. Для любого $j \leqslant j_1$ по определению (24) имеем $d^A_{j_1} \leqslant d^A_j$, а по определению (26) имеем $d^C_j = d^A_{j_1}$, откуда следует (27) при $k = 1$. Вместе с тем при любых i и j, $i < j \leqslant j_1$, неравенство (28) выполняется как равенство (по определению (26)), а (29) вытекает из соотношений

$$\Delta^C_i = d^C_i - r_i - p_i \overset{\text{из } (26)}{=} d^A_{j_1} - r_i - p_i \overset{\text{из } (23)}{\geqslant} d^A_{j_1} - r_j - p_j \overset{\text{из } (26)}{=} d^C_j - r_j - p_j = \Delta^C_j.$$

Индукционный шаг.

Пусть (27)–(29) верны при $k = k' - 1 < K$. Докажем, что они верны и для $k = k'$.

Для любого j $(j_{k'-1} < j \leqslant j_{k'})$ из (26) и (24) имеем $d^C_j \leqslant d^A_{j_{k'}} \leqslant d^A_j$, откуда с учётом индукционного предположения следует (27).

Для доказательства (28) и (29) с учётом индукционного предположения достаточно рассмотреть следующие варианты индексов i и j: а) $i = j_{k'-1} < j \leqslant j_{k'}$; б) $j_{k'-1} < i < j \leqslant j_{k'}$.

Для любого j $(j_{k'-1} < j \leqslant j_{k'})$ из (26) имеем либо c) $d^C_j = d^A_{j_{k'}}$, либо d) $d^C_j = r_j + p_j + d^C_{j_{k'-1}} - r_{j_{k'-1}} - p_{j_{k'-1}}$.

В случае c) получаем

$$d^C_{j_{k'-1}} \overset{\text{из } (27)}{\leqslant} d^A_{j_{k'-1}} \overset{\text{из } (25)}{\leqslant} d^A_{j_{k'}} = d^C_j.$$

В случае d) из (23) следует, что $d^C_j \geqslant d^C_{j_{k'-1}}$. В обоих случаях имеем (28) для варианта индексов a). Покажем, что (28) справедливо и для варианта индексов b).

В случае c) из определения (26) имеем $d^C_i \leqslant d^A_{j_{k'}} = d^C_j$. В случае d)

$$d^C_i \overset{\text{из } (26)}{\leqslant} r_i + p_i + \Delta^C_{j_{k'-1}} \overset{\text{из } (23)}{\leqslant} r_j + p_j + \Delta^C_{j_{k'-1}} = d^C_j.$$

Таким образом, в обоих случаях получаем (28) для варианта индексов b).

Далее, из соотношений

$$\begin{aligned} \Delta^C_{j_{k'-1}} &\geqslant \min\{\Delta^C_{j_{k'-1}}, d^A_{j_{k'}} - d^A_j + \Delta^A_j\} \\ &= \min\{\Delta^C_{j_{k'-1}} + d^A_j - \Delta^A_j, d^A_{j_{k'}}\} - d^A_j + \Delta^A_j = d^C_j - r_j - p_j = \Delta^C_j \end{aligned}$$

вытекает (29) для варианта индексов a). В случае б) неравенство (29) следует из

$$\begin{aligned} \Delta^C_i &= d^C_i - r_i - p_i \overset{\text{из } (26)}{=} \min\{d^A_{j_{k'}} - r_i - p_i, \Delta^C_{j_{k'-1}}\} \\ &\overset{\text{из } (23)}{\geqslant} \min\{d^A_{j_{k'}} - r_j - p_j, \Delta^C_{j_{k'-1}}\} \overset{\text{из } (26)}{=} d^C_j - r_j - p_j = \Delta^C_j. \end{aligned}$$

Таким образом, неравенства (27)–(29) доказаны при всех k. В частности, отсюда следует, что пример C принадлежит классу L.

Для доказательства достижимости (18) на примере C вначале заметим, что если $\rho^L_{ij}(A)$, определяемое согласно (17), записать в виде $\rho^L_{ij}(A) = \min\{d^A_j - d^A_i, d^A_j - d^A_i + (r_i + p_i) - (r_j + p_j)\}$, то ввиду (23) будем иметь

$$\rho^L_{ij}(A) = \begin{cases} d^A_j - d^A_i & \text{при } i \geqslant j, \\ \Delta^A_j - \Delta^A_i & \text{при } i < j. \end{cases} \tag{30}$$

Из (26) при $k \geqslant 2$ и $j = j_k$ имеем $d^C_{j_k} = \min\{d^A_{j_k}, d^A_{j_k} - \Delta^A_{j_k} + \Delta^C_{j_{k-1}}\}$. Отнимая от обеих частей этого равенства по $r_{j_k} + p_{j_k}$, получаем

$$\Delta^C_{j_k} = \min\{\Delta^A_{j_k}, \Delta^C_{j_{k-1}}\}. \tag{31}$$

Из (31) и равенства $\Delta^C_{j_1} = \Delta^A_{j_1}$ для любого $k \geqslant 1$ получаем

$$\Delta^C_{j_k} = \min_{\nu \leqslant k} \Delta^A_{j_\nu}. \tag{32}$$

Далее для любого j докажем неравенство

$$d^A_j - d^C_j \leqslant \rho^L(A). \tag{33}$$

При $j \leqslant j_1$, учитывая (30), имеем $d^A_j - d^C_j = d^A_j - d^A_{j_1} = \rho^L_{j_1 j}(A) \leqslant \rho^L(A)$.

При любом j, $j_{k-1} < j \leqslant j_k$, и $k \geqslant 2$, имеем

$$d^A_j - d^C_j = d^A_j - \min\{d^A_{j_k}, d^A_j - \Delta^A_j + \Delta^C_{j_{k-1}}\} = \max\{d^A_j - d^A_{j_k}, \Delta^A_j - \Delta^C_{j_{k-1}}\} =$$

(с учетом (32) для некоторого $\nu \leqslant k - 1$)

$$= \max\{d^A_j - d^A_{j_k}, \Delta^A_j - \Delta^A_{j_\nu}\} \overset{\text{из } (30)}{=} \max\{\rho^L_{j_k j}(A), \rho^L_{j_\nu j}(A)\} \leqslant \rho^L(A),$$

что доказывает (33). Наконец, поскольку $\rho_r(A, C) = 0$, имеем

$$\rho(A, C) = \rho_d(A, C) = \max_j\{d^A_j - d^C_j\} + \max_j\{d^C_j - d^A_j\} \overset{\text{из (33) и (27)}}{\leqslant} \rho^L(A),$$

откуда с использованием (18) получаем равенство $\rho(A, C) = \rho^L(A)$.

Временна́я сложность алгоритма нахождения искомого примера C складывается из сложности трёх его шагов:
1) нумерации требований согласно (23);
2) нахождения последовательности разделяющих индексов согласно (24);
3) вычисления новых директивных сроков согласно (26).

Первый из шагов выполняется с привлечением стандартной процедуры упорядочения n чисел, которая требует времени $O(n \log n)$. Третий шаг, очевидно, выполняется за линейное время. Что касается второго шага, то при "прямолинейной" реализации алгоритма согласно (24) его временна́я сложность не превосходит $O(n^2)$. Однако можно существенно уменьшить объём вычислений при нахождении разделяющих индексов, если просматривать требования в обратном порядке и определять разделяющие индексы следующим образом.

Предположим (только для удобства обозначений), что число K разделяющих индексов известно. Определим $j_K = n$. Пусть $1 < j_k < \cdots < j_K$ уже определены. Продолжим последовательно просматривать требования $j = j_k - 1, j_k - 2, \ldots$ и в качестве j_{k-1} выберем первый встреченный индекс $j < j_k$ такой, что $d_j^A < d_{j_k}^A$ (если такой встретится). Легко убедиться, что так определённая последовательность индексов $j_1, \ldots, j_K$ удовлетворяет (24), причём все неравенства (25) выполняются как строгие, а временна́я сложность этой процедуры линейна по n. Теорема 2 доказана.

На самом деле в алгоритме нахождения искомого примера C с самого начала целесообразно занумеровать требования примера A в обратном порядке — по невозрастанию $r_j + p_j$. Формальная запись такого алгоритма представлена ниже (предполагается, что требования в примере A уже отсортированы по невозрастанию $r_j^A + p_j^A$).

Алгоритм 1 (нахождения примера из класса L, ближайшего к заданному примеру A)

FOR $j := 1$ TO n $\quad r_j^C := r_j^A$; $p_j^C := p_j^A$ END FOR

$k := 1$; $j_1 = 1$;

FOR $i := 2$ TO n IF $d_i^A < d_{j_k}^A$ THEN $k := k + 1$; $j_k := i$ END IF END FOR

FOR $j := n$ DOWNTO j_k $\quad d_j^C := d_{j_k}^A$ END FOR

$\Delta_{j_k}^C := d_{j_k}^C - r_{j_k}^C - p_{j_k}^C$;

FOR $\nu := k$ DOWNTO 2 FOR $j := j_\nu - 1$ DOWNTO $j_{\nu-1}$ $d_j^C := \min\{d_{j_{\nu-1}}^A, d_j^A - \Delta_j^A + \Delta_{j_\nu}^C\}$; $\Delta_j^C := d_j^C - r_j^C - p_j^C$ END FOR END FOR

Пример. Рассмотрим процесс работы алгоритма 1. Пусть дан пример A задачи $1 \mid r_j \mid L_{\max}$ с параметрами требований, указанными в таблице 1, причём требования отсортированы по невозрастанию $r_j^A + p_j^A$.

Таблица 1

Параметры требований для примера A

j	1	2	3	4	5	6	7	8
r_j^A	7	5	3	5	1	2	3	0
p_j^A	2	4	5	3	5	3	1	4
d_j^A	16	18	13	14	15	11	12	14

Начнём выполнение алгоритма при $N = \{1,\ldots,8\}$. После первого цикла при любом $j \in N$ получаем $r_j^C = r_j^A$ и $p_j^C = p_j^A$. После второго цикла имеем: $j_1 = 1$, $j_2 = 3$, $j_3 = 6$. После третьего цикла получаем $d_6^C = d_7^C = d_8^C = d_6^A = 11$.

Начинается выполнение цикла по ν:

$\nu = 3$.

$d_5^C = \min\{d_3^A, d_6^C - r_6^C - p_6^C + r_5^A + p_5^A\} = 12.$
$d_4^C = \min\{d_3^A, d_6^C - r_6^C - p_6^C + r_4^A + p_4^A\} = 13.$
$d_3^C = \min\{d_3^A, d_6^C - r_6^C - p_6^C + r_3^A + p_3^A\} = 13.$

$\nu = 2$.

$d_2^C = \min\{d_1^A, d_3^C - r_3^C - p_3^C + r_2^A + p_2^A\} = 14.$
$d_1^C = \min\{d_1^A, d_3^C - r_3^C - p_3^C + r_1^A + p_1^A\} = 14.$

Алгоритм заканчивает работу.

Итак, мы получили пример C, удовлетворяющий свойству P_L. Параметры требований примера C указаны в таблице 2. Оценка абсолютной погрешности решения равна

$$\rho(A,C) = \max_{j\in N}\{d_j^A - d_j^C\} + \max_{j\in N}\{d_j^C - d_j^A\} = d_2^A - d_2^C + 0 = 4.$$

Таблица 2

Параметры требований для примера C

j	1	2	3	4	5	6	7	8
r_j^A	7	5	3	5	1	2	3	0
p_j^A	2	4	5	3	5	3	1	4
d_j^C	14	14	13	13	12	11	11	11

3.2. Вариант схемы на основе случая Хогевена

Следующий вариант схемы приближённого решения основан на сведе́нии любого заданного примера к примеру из полиномиально разрешимого класса Хогевена [9] (напомним, что для точного решения примеров из класса Хогевена имеется алгоритм сложности $O(n^2 \log n)$).

Из (13) следует, что пример $A = \{(r_j^A, p_j^A, d_j^A) \mid j \in N\}$ принадлежит классу Хогевена, если и только если

$$\Delta_{\max}^A \doteq \max_i \Delta_i^A \leqslant d_j^A - r_j^A,\ j \in N. \tag{34}$$

Для заданного примера A определим функцию

$$\rho^H(A) = \max_{j \in N}(\Delta_{\max}^A - d_j^A + r_j^A). \tag{35}$$

Ясно, что пример A принадлежит классу Хогевена, если и только если $\rho^H(A) \leqslant 0$.

Теорема 3. *Для любого примера A задачи $1 \mid r_j \mid L_{\max}$, не принадлежащего классу Хогевена, и для любого примера C, наследующего длительности обслуживания примера A и принадлежащего классу Хогевена, справедлива оценка расстояния между A и C :*

$$\rho(A, C) \geqslant \rho^H(A). \tag{36}$$

Оценка (36) достигается на некотором примере C, наследующем у A длительности обслуживания и моменты поступления требований (*этот пример можно найти за время $O(n)$*).

Доказательство. Как было отмечено выше, неравенство (36) достаточно доказать для любого примера C, удовлетворяющего (14). По определению расстояния $\rho(A, C)$ имеем

$$\begin{gathered}\rho(A, C) = \max_i(d_i^C - d_i^A) + \max_j(d_j^A - d_j^C) + \max_j(r_j^C - r_j^A) + \max_i(r_i^A - r_i^C) \\ \geqslant d_i^C - d_i^A + d_j^A - d_j^C + r_j^C - r_j^A + r_i^A - r_i^C = (d_j^A - r_j^A - p_j) \\ + (r_j^C - d_j^C + p_j) + (r_i^A - d_i^A) + (d_i^C - r_i^C) \overset{\text{из (13)}}{\geqslant} (d_j^A - r_j^A - p_j) + (r_i^A - d_i^A).\end{gathered}$$

Так как эти соотношения верны при любых $j, i \in N$, с учётом (35) получаем (36):

$$\rho(A, C) \geqslant \max_j(d_j^A - r_j^A - p_j) + \max_i(r_i^A - d_i^A) = \rho^H(A).$$

Докажем, что оценка (36) достигается на некотором примере C из класса Хогевена (так как параметры p_j и r_j не будут различаться для примеров A и C, мы будем обозначать их без верхнего индекса A и C соответственно).

Поскольку пример A не удовлетворяет (34), найдутся требования j такие, что

$$\Delta^A_{\max} > d^A_j - r_j. \tag{37}$$

Пусть N' — множество таких j. Для каждого требования $j \in N$ определим новое (увеличенное) значение d^C_j по формуле

$$d^C_j := d^A_j + (\Delta^A_{\max} - d^A_j + r_j)^+. \tag{38}$$

Тогда

$$\Delta^C_{\max} = \Delta^A_{\max}. \tag{39}$$

Действительно, так как $d^C_j \geqslant d^A_j$ при любом $j \in N$ (при этом r_j и p_j остались прежними), то $\Delta^C_{\max} \geqslant \Delta^A_{\max}$.

Докажем обратное неравенство. Если $j \notin N'$, то $\Delta^C_j = \Delta^A_j \leqslant \Delta^A_{\max}$. При $j \in N'$ из (37), (38) имеем $d^C_j = \Delta^A_{\max} + r_j$. Следовательно, $\Delta^C_j = \Delta^A_{\max} - p_j \leqslant \Delta^A_{\max}$. Таким образом, $\Delta^C_{\max} \leqslant \Delta^A_{\max}$, откуда следует (39).

Докажем, что пример C удовлетворяет (34).

При любом $j \in N \setminus N'$ неравенства (34) следуют из $\Delta^C_{\max} = \Delta^A_{\max} \leqslant d^A_j - r_j = d^C_j - r_j$.

Если $j \in N'$, то $d^C_j - r_j = \Delta^A_{\max} = \Delta^C_{\max}$. Таким образом, пример C принадлежит классу Хогевена. Его расстояние от примера A вычисляется из соотношений

$$\rho(A, C) = \max_{j \in N}(d^C_j - d^A_j) = \max_{j \in N'}(\Delta^A_{\max} - d^A_j + r_j) = \rho^H(A).$$

Теорема 3 доказана.

Для примера из таблицы 1 оценка абсолютной погрешности равна $\rho(A, C) = 1$.

4. Заключение

В данной статье представлена полиномиальная схема нахождения приближённого решения задачи $1 \mid r_j \mid L_{\max}$. Схема основана на сведéнии заданного примера к другому примеру, принадлежащему к известному полиномиально разрешимому классу примеров. Для двух вариантов схемы (с использованием полиномиально разрешимых классов примеров L и Хогевена) найдены аналитические формулы, позволяющие вычислить априорную оценку абсолютной погрешности приближённого решения для любого заданного примера.

Дальнейшие исследования могут быть направлены на теоретическое и практическое сравнение абсолютной погрешности приближённого решения, получаемого с использованием предложенной схемы, и абсолютной погрешности решений рассматриваемой задачи, найденных другими приближёнными методами.

ЛИТЕРАТУРА

1. **Лазарев А. А.** Эффективные алгоритмы решения некоторых задач теории расписаний для одного прибора с директивными сроками обслуживания требований. Диссертация на соискание учёной степени к. ф.-м. н. Казань: Казанский гос. ун-т, 1989.
2. **Лазарев А. А., Шульгина О. Н.** Псевдополиномиальный алгоритм решения NP-трудной задачи минимизации максимального временного смещения // Труды XI международной Байкальской школы-семинара "Методы оптимизации и их приложения 5–12 июля. Иркутск, Байкал: СЭИ СО РАН, 1998. С. 163–167.
3. **Adams J., Balas E., Zawack D.** The shifting bottleneck procedure for job shop scheduling // J. Manag. Sci. 1988. V. 34, N 3. P. 391–401.
4. **Baker K. R., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G.** Preemptive scheduling of a single machine to minimize maximum cost subject to release dates and precedence constraints // J. Oper. Res. 1983. V. 31, N 2. P. 381–386.
5. **Baker K. R.,Su Z. S.** Sequencing with due dates and early start times to minimize tardiness // J. Naval Res. Logist. Quart. 1974. V. 21, N 1. P. 171–176.
6. **Carlier J.** The one-machine sequencing problem // European J. of Oper. Res. 1982. V. 11, N 1. P. 42–47.
7. **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.
8. **Hall L. A., Shmoys D. B.** Jackson's rule for one-machine scheduling: making a good heuristic better // J. Math. Oper. Res. 1992. V. 17, N 1. P. 22–35.
9. **Hoogeveen J. A.** Minimizing maximum promptness and maximum lateness on a single machine // J. Math. Oper. Res. 1996. V. 21, N 1. P. 100–114.
10. **Jackson J. R.** Scheduling a production line to minimize maximum tardiness. Manag. Sci. Res. Project, N 43. UCLA, 1955.
11. **Lageweg B. J., Lenstra J. K., Rinnooy Kan A. H. G.** Minimizing maximum lateness on one machine: computational experience and some applications // J. Statistica Neerlandica. 1976. V. 30, N 1. P. 25–41.
12. **Lawler E. L.** Optimal sequencing of a single machine subject to precedence constraints // J. Manag. Sci. 1973. V. 19, N 5. P. 544–546.

13. **Lenstra J. K., Rinnooy Kan A. H. G., Brucker P.** Complexity of machine scheduling problems // Annals of Oper. Res. 1975. V. 1. P. 343–362.
14. **Mastrolilli M.** Efficient approximation schemes for scheduling problems with release dates and delivery times // J. of Scheduling. 2003. V. 6, N 6. P. 521–531.
15. **McMahon G., Florian M.** On scheduling with ready times and due dates to minimize maximum lateness // J. Oper. Res. 1975. V. 23, N 3. P. 475–482.
16. **Péridy L., Pinson E., Rivraux D.** Using short-term memory to minimize the weighted number of late jobs on a single machine // European J. of Oper. Res. 2003. V. 148, N 3. P. 591–603.
17. **Potts C. N.** Analysis of a heuristic for one machine sequencing with release dates and delivery times // J. Oper. Res. 1980. V. 28, N 6. P. 1436–1441.
18. **Simons B. B.** A fast algorithm for single processor scheduling // Proc. of the 19th Annual symposium foundftion of computer science. New York: Ann. Arbor, 1978. P. 246–252.

Адреса авторов:

А. А. Лазарев, Р. Р. Садыков

Казанский государственный университет,
кафедра экономической кибернетики,
ул. Кремлёвская, 18,
420008 Казань, Россия.
E-mail: Alexandr.Lazarev@ksu.ru

С. В. Севастьянов

Институт математики
им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4,
630090 Новосибирск, Россия.

Статья поступила
10 апреля 2005 г.

Переработанный вариант —
13 ноября 2005 г.