

УДК 519.2+621.391

О ЦЕХОВОЙ ЗАДАЧЕ ОТКРЫТОГО ТИПА
НА ДВУХ МАШИНАХ С МАРШРУТИЗАЦИЕЙ
В ДВУХВЕРШИННОЙ СЕТИ *)

А. В. Кононов

Аннотация. Рассматривается цеховая задача открытого типа для двух машин с маршрутизацией в двухвершинной сети. Задача является NP-трудной. Для её решения предлагаются точный псевдополиномиальный алгоритм и вполне полиномиальная приближённая схема и выделяются полиномиально разрешимые случаи.

Ключевые слова: цеховая задача открытого типа, маршрутизация, вполне полиномиальная приближённая схема.

Введение

Формулировка задачи. Рассматривается следующая задача теории расписаний. Множество работ N должно быть обслужено двумя машинами A и B . Для каждой работы J_j из N известны длительности её обслуживания a_j и b_j машинами A и B соответственно. Задана сеть, состоящая из двух вершин V_0 и V_1 , соединённых ребром. Множество работ разбито на подмножества N_0 и N_1 . Все работы из N_0 лежат в вершине V_0 , а все работы из N_1 — в V_1 . Для обслуживания каждой работы машина должна переместиться в вершину, где находится эта работа. Каждой машине требуется τ единиц времени для перемещения из одной вершины в другую. В начальный момент времени обе машины находятся в V_0 и должны вернуться туда после выполнения всех работ. Таким образом, каждая машина должна сделать чётное число перемещений. Каждая работа имеет ровно две операции, одна из которых должна быть выполнена на машине A , а другая — на B . Никакие две операции, выполняемые на одной машине или принадлежащие одной работе, не могут выполняться одновременно. Прерывания во время выполнения операции запрещены. Требуется составить расписание выполнения работ и перемещений машин, в котором машины выполняют все работы и вернутся в исходную

*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 08-01-00370).

вершину V_0 за минимальное время. Назовём сформулированную задачу задачей $ROS_{2,2}$.

Обзор результатов. Задача $ROS_{2,2}$ является частным случаем цеховой задачи открытого типа с маршрутизацией (задача ROS), в которой число машин и конфигурация сети произвольны. В свою очередь, цеховая задача открытого типа с маршрутизацией является обобщением двух классических задач дискретной оптимизации: цеховой задачи открытого типа (Open Shop) и метрической задачи коммивояжёра.

Цеховая задача открытого типа с маршрутизацией впервые была рассмотрена Авербахом, Берманом и Черных в [2, 3]. Задача ROS является NP-трудной в сильном смысле, поскольку включает в себя метрическую задачу коммивояжёра как частный случай. Более того, она NP-трудна в обычном смысле даже на двухвершинной сети с двумя машинами [3], т. е. задача $ROS_{2,2}$ также NP-трудна.

Для последнего случая в [2] предложен $6/5$ -приближённый алгоритм. Для двухмашинного случая с работами, расположенными на произвольной сети, в [3] построен приближённый алгоритм с гарантированной оценкой точности $7/4$, а для задачи с произвольным числом машин m предложен $(m + 4)/2$ -приближённый алгоритм. В [4] два последних результата улучшены: для случая с двумя машинами на произвольной сети построен $13/8$ -приближённый алгоритм, а в общем случае — $O(\sqrt{m})$ -приближённый алгоритм.

В нашей статье выделяются новые полиномиально разрешимые классы задачи $ROS_{2,2}$, предлагается точный псевдополиномиальный алгоритм и вполне полиномиальная приближённая схема. В разд. 1 вводятся основные понятия, приводится алгоритм Гонзалеса — Сани для классической цеховой задачи открытого типа на двух машинах и перечисляются его свойства, которые понадобятся для построения новых полиномиальных алгоритмов. В разд. 2 описываются новые нетривиальные полиномиально-разрешимые подклассы задачи $ROS_{2,2}$, а в разд. 3 — псевдополиномиальный алгоритм и вполне полиномиальная приближённая схема для её решения.

1. Основные обозначения и предварительные результаты

Решение в цеховых задачах теории расписаний удобно представлять ориентированным графом $G = (V, E)$, в котором множество вершин V содержит по одной вершине для каждой операции и две специальные вершины — источник 0 и сток C_{\max} . Каждой вершине приписан неотрицательный вес, равный длительности соответствующей операции. Ве-

са стока и источника положим равными нулю. Для двух операций x и y дуга $(x, y) \in E$ указывает на отношение предшествования между этими операциями, т. е. операция x должна завершиться до начала операции y . Так как никакие две операции, выполняемые на одной машине или принадлежащие одной работе, не могут выполняться одновременно, в G должны существовать дуги между любыми двумя операциями, выполняемыми на одной машине, и любыми двумя операциями одной работы. Кроме того, заданы дуги из источника в каждую вершину и из каждой вершины в сток.

Поскольку в задаче $ROS_{2,2}$ машинам потребуется переезд из одной вершины в другую, дополнительно предполагается, что каждая дуга имеет неотрицательный вес δ . Запись $x \xrightarrow{\delta} y$ означает, что операция y начинает выполняться не раньше, чем через время δ после завершения операции x . Запись $x \rightarrow y$ означает, что вес дуги (x, y) равен 0. Длина пути в данном графе определяется как сумма весов вершин и дуг, принадлежащих этому пути.

Через $s(x, \sigma)$ и $C(x, \sigma)$ обозначим моменты начала и завершения обслуживания операции x в расписании σ . Построим по заданному графу G расписание σ по следующему правилу. Для каждой операции x положим $s(x, \sigma)$ равным длине максимального пути из 0 в x . Расписание, построенное по такому правилу, называется *активным*, и каждый такой граф определяет единственное активное расписание [1]. Длина активного расписания равна длине самого длинного пути в графе из источника в сток, который будем называть *критическим*. Дуга $(x, y) \in E$ называется *транзитивной*, если в G существует путь из x в y , проходящий через другую вершину. Легко показать, что в G всегда существует критический путь, не содержащий транзитивных дуг нулевого веса.

При построении расписаний будем использовать понятие «схемы расписания», упрощающее его представление в виде ориентированного графа. В такой схеме каждая вершина x , кроме источника и стока, может быть либо операцией, либо подмножеством операций. При этом предполагается, что все операции в подмножестве, соответствующем вершине, выполняются без задержек и вес вершины равен сумме длин операций. Назовём такое подмножество операций *блоком*. Дополнительно потребуем, чтобы критический путь либо не проходил через вершину, соответствующую блоку, либо содержал его целиком.

Пусть $l_1 = \sum_{j \in N} a_j$ и $l_2 = \sum_{j \in N} b_j$ — загрузки первой и второй машин соответственно и $l_{\max} = \max\{l_1, l_2\}$. Величина $d_j = a_j + b_j$ назы-

вается длиной работы j . Пусть $d_{\max}^0 = \max_{j \in N_0} d_j$ и $d_{\max}^1 = \max_{j \in N_1} d_j$ — длины максимальных работ в вершинах V_0 и V_1 соответственно. Положим $d_{\max} = \max \{d_{\max}^0, d_{\max}^1\}$.

Если все работы сосредоточены в одной вершине, то задача $\text{ROS}_{2,2}$ эквивалентна классической цеховой задаче открытого типа на двух машинах $O2||C_{\max}$. Напомним, что для задачи $O2||C_{\max}$ справедлива следующая оценка длины оптимального расписания:

$$C_{\max}^*(O2||C_{\max}) \geq \lambda_O \doteq \max\{l_{\max}, d_{\max}\}.$$

Для задачи $\text{ROS}_{2,2}$ аналогичная оценка выписывается следующим образом [2]:

$$C_{\max}^*(\text{ROS}_{2,2}) \geq \lambda_R \doteq \max\{l_{\max} + 2\tau, d_{\max}^0, d_{\max}^1 + 2\tau\}.$$

Для задачи $O2||C_{\max}$ Гонзалес и Сани [5] предложили простой полиномиальный алгоритм, строящий расписание длины λ_O . Поскольку λ_O является нижней оценкой длины любого допустимого расписания, алгоритм Гонзалеса — Сани находит оптимальное расписание. Операции работы J_j на машинах A и B будем обозначать через A_j и B_j соответственно.

АЛГОРИТМ ГОНЗАЛЕЗА — САНИ

ШАГ 1. Разделим множество работ N на подмножества

$$N_{\leq} = \{J_j \in N \mid a_j \leq b_j\} \quad \text{и} \quad N_{>} = \{J_j \in N \mid a_j > b_j\}.$$

ШАГ 2. Работу J_0 , на которой достигается

$$\max\{\max\{a_i \mid J_i \in N_{\leq}\}, \max\{b_i \mid J_i \in N_{>}\}\},$$

назовём *диагональной*, $d_0 = a_0 + b_0$. Без ограничения общности предположим, что $J_0 \in N_{\leq}$. В противном случае переименуем машины. Положим $N'_{\leq} \doteq N_{\leq} \setminus \{J_0\}$.

ШАГ 3. Если $d_0 = \lambda_O$,

ШАГ 3.1. то занумеруем оставшиеся работы на каждой машине в произвольном порядке,

ШАГ 3.2. иначе занумеруем оставшиеся работы так, что первыми идут все работы из N'_{\leq} , а затем все работы из $N_{>}$.

ШАГ 4. Построим расписание σ согласно следующему правилу.

ШАГ 4.1. Машина A выполняет операции в порядке $A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_{n-1} \rightarrow A_0$.

ШАГ 4.2. Машина B выполняет операции в порядке $B_0 \rightarrow B_1 \rightarrow B_2 \rightarrow \dots \rightarrow B_{n-1}$.

ШАГ 4.3. Диагональная работа сначала выполняется на машине B , а потом на A , а все остальные работы сначала выполняются на машине A , а потом на B .

Лемма 1 [5]. Алгоритм Гонзалеза — Сани строит за $O(n)$ элементарных операций расписание σ с $C_{\max}(\sigma) = \lambda_O$.

Заметим, что на шаге 3.1 алгоритма Гонзалеза — Сани можно выбрать произвольный порядок работ из множества $N \setminus \{J_0\}$ на каждой из машин, а на шаге 3.2 можно выбрать любой порядок работ из множества N_{\leq} и любой из $N_{>}$. В дальнейшем используем это свойство для построения оптимальных расписаний в задаче $\text{ROS}_{2,2}$, выбирая подходящий порядок работ в зависимости от их расположения в сети.

2. Достаточные условия полиномиальной разрешимости задачи $\text{ROS}_{2,2}$

Установим достаточные условия, когда оптимальное решение задачи $\text{ROS}_{2,2}$ может быть найдено за линейное от числа работ время. Как и в алгоритме Гонзалеза — Сани, обозначим диагональную работу через J_0 . Для любого множества работ X через $\{X\}_A$ ($\{X\}_B$) будем обозначать множество его операций на машине A (B).

СЛУЧАЙ 1. Пусть $J_0 \in N_0$. Рассмотрим следующее расписание σ_1 . Пусть машина A сначала выполняет все работы из $N'_{\leq} \cap N_0$, затем все работы из $N_{\leq} \cap N_1$, потом все работы из $N_{>} \cap N_1$, после этого все работы из $N_{>} \cap N_0$ и последней работу J_0 . Машина B первой выполняет работу J_0 , затем оставшиеся работы в том же порядке, как и машина A . Схема $G(\sigma_1)$ этого расписания представлена на рис. 1.

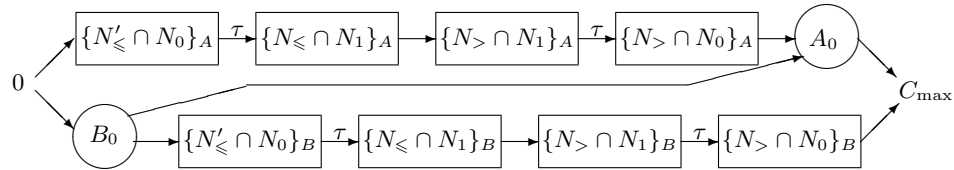


Рис. 1. Схема $G(\sigma_1)$, $J_0 \in N_0$

Покажем, что один из следующих путей является критическим в $G(\sigma_1)$.

- (i) $0 \rightarrow \{N'_\leq \cap N_0\}_A \xrightarrow{\tau} \{N_\leq \cap N_1\}_A \rightarrow \{N_> \cap N_1\}_A \xrightarrow{\tau} \{N_> \cap N_0\}_A \rightarrow A_0 \rightarrow C_{\max}$,
- (ii) $0 \rightarrow B_0 \rightarrow \{N'_\leq \cap N_0\}_B \xrightarrow{\tau} \{N_\leq \cap N_1\}_B \rightarrow \{N_> \cap N_1\}_B \xrightarrow{\tau} \{N_> \cap N_0\}_B \rightarrow C_{\max}$,
- (iii) $0 \rightarrow B_0 \rightarrow A_0 \rightarrow C_{\max}$.

Заметим, что длина пути (i) равна $l_1 + 2\tau$, пути (ii) — $l_2 + 2\tau$ и пути (iii) — d_{\max}^0 . Кроме указанных путей только пути вида $0 \rightarrow A_1 \rightarrow \dots \rightarrow A_i \rightarrow B_i \rightarrow \dots \rightarrow B_{n-1} \rightarrow C_{\max}$ не содержат транзитивных дуг нулевого веса. Покажем, что ни один из этих путей не может быть критическим.

Пусть $J_i \in N'_\leq$. Учитывая, что $a_j \leq b_j$ для всех $j \in N_\leq$ и $a_i \leq a_0 \leq b_0$, получаем

$$\sum_{j=1}^i a_j + \sum_{j=i}^{n-1} b_j + 2\tau \leq \sum_{j=1}^{n-1} b_j + a_i + 2\tau \leq \sum_{j=1}^{n-1} b_j + b_0 + 2\tau = l_2 + 2\tau.$$

Пусть $J_i \in N_>$. Учитывая, что $a_j \geq b_j$ для всех $j \in N_>$ и $b_i \leq a_0$, имеем

$$\sum_{j=1}^i a_j + \sum_{j=i}^{n-1} b_j + 2\tau \leq \sum_{j=1}^{n-1} a_j + b_i + 2\tau \leq \sum_{j=1}^{n-1} a_j + a_0 + 2\tau = l_1 + 2\tau.$$

Следовательно, $C_{\max}(\sigma_1) = \lambda_R$ и σ_1 — оптимальное расписание.

СЛУЧАЙ 2. Пусть $J_0 \in N_1$ и $d_0 \geq l_{\max}$. Рассмотрим следующее расписание σ_2 . Пусть машина A сначала выполняет все работы из N_0 , затем все работы из $N_1 \setminus \{J_0\}$ и последней работу J_0 , а машина B первой выполняет работу J_0 , затем все работы из $N_1 \setminus \{J_0\}$ и последними работы из N_0 . Схема $G(\sigma_2)$ этого расписания представлена на рис. 2.

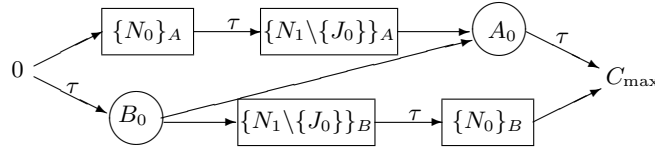


Рис. 2. Схема $G(\sigma_2)$, $J_0 \in N_1$ и $d_0 \geq l_{\max}$

Заметим, что $d_0 \geq l_{\max}$ влечёт $\sum_{i=1}^{n-1} a_i \leq b_0$ и $\sum_{i=1}^{n-1} b_i \leq a_0$. Значит, путь $0 \xrightarrow{\tau} B_0 \rightarrow A_0 \xrightarrow{\tau} C_{\max}$ является критическим в $G(\sigma_2)$. Длина этого

пути равна $d_{\max}^1 + 2\tau$. Следовательно, $C_{\max}(\sigma) = \lambda_R$ и σ_2 — оптимальное расписание.

Так как расписания σ_1 и σ_2 могут быть построены за линейное от числа работ время, получаем следующее утверждение.

Теорема 1. *Если в примере I задачи $\text{ROS}_{2,2}$ выполнено одно из двух условий*

- (i) $J_0 \in N_0$,
- (ii) $J_0 \in N_1$ и $d_0 \geq l_{\max}$,

то оптимальное расписание имеет длину λ_R и может быть построено за время $O(n)$.

3. Точный алгоритм и приближённая схема для задачи $\text{ROS}_{2,2}$

Теорема 1 определяет конструктивные и легко проверяемые условия, гарантирующие полиномиальную разрешимость задачи $\text{ROS}_{2,2}$. Таким образом, только в случае, когда диагональная работа лежит в удаленной вершине и её длина меньше l_{\max} , нам не удаётся построить для решения задачи $\text{ROS}_{2,2}$ точного полиномиального алгоритма. Именно для этого случая в [3] доказана NP-трудность задачи $\text{ROS}_{2,2}$.

3.1. Конфигурации оптимальных расписаний в трудных примерах. В оставшейся части статьи будем предполагать, что $J_0 \in N_1$ и $d_0 < l_{\max}$. Следующая теорема устанавливает верхнюю оценку на длину оптимального расписания.

Теорема 2. *Пусть $J_0 \in N_1$ и $d_0 < l_{\max}$. Тогда*

$$C_{\max}^*(\text{ROS}_{2,2}) \leq l_{\max} + 4\tau.$$

ДОКАЗАТЕЛЬСТВО. Пусть σ_3 — расписание, в котором машины выполняют работы в том же порядке, что и в расписании σ_1 . Схема $G(\sigma_3)$ этого расписания представлена на рис. 3.

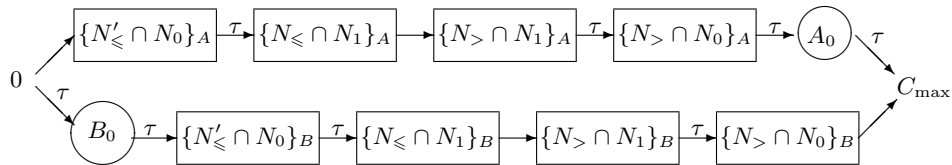


Рис. 3. Схема $G(\sigma_3)$, $J_0 \in N_1$ и $d_0 < l_{\max}$.

Повторяя рассуждения случая 1 из разд. 2, получаем, что один из двух путей

$$\begin{aligned} 0 \rightarrow \{N'_{\leq} \cap N_0\}_A \xrightarrow{\tau} \{N_{\leq} \cap N_1\}_A \rightarrow \{N_{>} \cap N_1\}_A \\ \xrightarrow{\tau} \{N_{>} \cap N_0\}_A \xrightarrow{\tau} A_0 \xrightarrow{\tau} C_{\max}, \end{aligned}$$

$$\begin{aligned} 0 \xrightarrow{\tau} B_0 \xrightarrow{\tau} \{N'_{\leq} \cap N_0\}_B \xrightarrow{\tau} \{N_{\leq} \cap N_1\}_B \rightarrow \{N_{>} \cap N_1\}_B \\ \xrightarrow{\tau} \{N_{>} \cap N_0\}_B \rightarrow C_{\max} \end{aligned}$$

критический. Следовательно, $C_{\max}^*(\text{ROS}_{2,2}) \leq l_{\max} + 4\tau$. Теорема 2 доказана.

Обозначим через $\psi(M, \sigma)$ количество перемещений из одной вершины в другую машины M в расписании σ . Далее без ограничения общности будем считать, что $l_1 \geq l_2$. Тогда непосредственно из теоремы 2 вытекает

Следствие 1. *Если существует оптимальное расписание σ^* такое, что $\psi(A, \sigma^*) \geq 4$, то расписание σ_3 также оптимально.*

Пусть σ — некоторое расписание, в котором машина A совершает ровно одну поездку в вершину V_1 . Тем самым машина A выполняет какие-то работы из N_0 , затем перемещается в вершину V_1 , выполняет все работы из множества N_1 , возвращается в V_0 и выполняет оставшиеся работы, т. е. $\psi(A, \sigma) = 2$.

Покажем, что для любого допустимого расписания σ с $\psi(A, \sigma) = 2$ множество работ N можно разбить на конечный (не зависящий от n) набор подмножеств \mathcal{N} так, что существует расписание σ^* , удовлетворяющее следующим условиям:

- (C1) $C_{\max}(\sigma^*) \leq C_{\max}(\sigma)$;
- (C2) $\psi(A, \sigma^*) = 2$;
- (C3) для каждого подмножества X из \mathcal{N} операции работ из X на каждой из машин образуют блок в σ^* .

Для этого построим цепочку преобразований расписания σ в σ^* . Будем говорить, что *операция сдвинута влево (вправо)* относительно её позиции в расписании σ , если в новом расписании она начинает выполняться не позднее (не раньше) момента начала её выполнения в σ . Блоки операций одного подмножества работ будем называть *парными*.

Пусть $t_1 = \min\{s(A_i, \sigma) \mid J_i \in N_1\}$, $t_2 = \max\{C(A_i, \sigma) \mid J_i \in N_1\}$. Тогда все работы из множества N_1 выполняются на первой машине на отрезке $[t_1, t_2]$. Обозначим через N'_0 множество работ из N_0 , выполняющихся на машине A на отрезке $[0, t_1]$, и через N''_0 — множество работ из N_0 , выполняющихся на машине A на отрезке $[t_2, C_{\max}]$. Для введённых обозначений порядок выполнения работ на машине A в σ может быть записан схемой $0 \rightarrow \{N'_0\}_A \xrightarrow{\tau} \{N_1\} \xrightarrow{\tau} \{N''_0\}_A \rightarrow C_{\max}$.

Пусть L — множество работ с первой операцией на машине A и R — множество работ с первой операцией на машине B в σ . Тогда следующие преобразования допустимы относительно операций работ этих множеств:

сдвиг влево операции работы из L на машине A или сдвиг вправо операции работы из L на машине B ;

сдвиг вправо операции работы из R на машине A или сдвиг влево операции работы из R на машине B .

Заметим, что множества L и R не будут изменяться в последующих преобразованиях.

С учётом введённых обозначений можно переупорядочить работы на A . Рассмотрим расписание $\bar{\sigma}$, в котором на машине A операции выполняются в следующем порядке:

$$0 \rightarrow \{N'_0 \cap L\}_A \rightarrow \{N'_0 \cap R\}_A \xrightarrow{\tau} \{N_1 \cap L\}_A \rightarrow \{N_1 \cap R\}_A \xrightarrow{\tau} \{N''_0 \cap L\}_A \rightarrow \{N''_0 \cap R\}_A \rightarrow C_{\max},$$

при этом внутри каждого из блоков операции идут в том же порядке, что и в σ . Нетрудно видеть, что переупорядочение работ эквивалентно сдвигу работ из L влево и работ из R вправо, и в $\bar{\sigma}$ каждое из множеств операций $\{N'_0\}_A$, $\{N_1\}_A$, $\{N''_0\}_A$ может быть выполнено блоком внутри того же интервала времени, в котором оно выполняется в σ .

Дополнительно разобьём каждое из множеств $N'_0 \cap L$ и $N''_0 \cap R$ на два подмножества. Пусть

$$t_3 = \begin{cases} 0, & \text{если } N_1 \cap R = \emptyset, \\ \min\{s(B_i, \sigma) \mid J_i \in N_1 \cap R\} & \text{в противном случае;} \end{cases}$$

$$t_4 = \begin{cases} C_{\max}, & \text{если } N_1 \cap L = \emptyset, \\ \max\{C(B_i, \sigma) \mid J_i \in N_1 \cap L\} & \text{в противном случае.} \end{cases}$$

Положим $\bar{N}'_{0L} = \{J_i \in N'_0 \cap L \mid s(B_i, \bar{\sigma}) \geq t_4\}$ и $N'_{0L} = N'_0 \cap L \setminus \bar{N}'_{0L}$. Без ограничения общности можно считать, что все работы из N'_{0L} предшествуют работам из \bar{N}'_{0L} на обеих машинах. Действительно, по определению все работы из N'_{0L} заканчиваются на машине B до момента t_4 .

Так как первая операция этих работ выполняется на машине A , на ней они заканчиваются до момента t_4 . Пусть некоторая работа $J_i \in \bar{N}'_{0L}$ выполняется на машине A перед работой $J_k \in N'_{0L}$. Поставим её на машине A после последней работы из N'_{0L} и сдвинем все работы с J_k по J_i влево к моменту $s(A_i, \bar{\sigma})$. Полученное расписание допустимо, так как операция A_i , которая сдвигается вправо, завершится до момента t_4 , а все остальные работы сдвигались влево, что не создаёт конфликтов со вторыми операциями тех же работ.

Положим $\bar{N}''_{0R} = \{J_i \in N''_0 \cap R \mid C(B_i, \bar{\sigma}) \leq t_3\}$ и $N''_{0R} = N''_0 \cap R \setminus \bar{N}''_{0R}$. Аналогично предыдущему случаю можно считать, что работы из N''_{0R} выполняются после всех работ из \bar{N}''_{0R} . Пусть $N'_{0R} = N'_0 \cap R$, $N_{1L} = N_1 \cap L$, $N_{1R} = N_1 \cap R$, $N''_{0L} = N''_0 \cap L$. Тогда в расписании $\bar{\sigma}$ на A операции выполняются в следующем порядке π_0 :

$$\begin{aligned} 0 \rightarrow \{N'_{0L}\}_A \rightarrow \{\bar{N}'_{0L}\}_A \rightarrow \{N'_{0R}\}_A \xrightarrow{\tau} \{N_{1L}\}_A \\ \rightarrow \{N_{1R}\}_A \xrightarrow{\tau} \{N''_{0L}\}_A \rightarrow \{\bar{N}''_{0R}\}_A \rightarrow \{N''_{0R}\}_A \rightarrow C_{\max}. \end{aligned}$$

Положим $\mathcal{N}_1 = \{N'_{0R}, \bar{N}''_{0R}, N_{1R}, N''_{0R}, N'_{0L}, N_{1L}, \bar{N}'_{0L}, N''_{0L}\}$. В расписании $\bar{\sigma}$ для каждого подмножества $X \in \mathcal{N}_1$ операции работ из X образуют блок на машине A и $C_{\max}(\bar{\sigma}) \leq C_{\max}(\sigma)$. Заметим, что некоторые из подмножеств в наборе \mathcal{N}_1 могут быть пустыми.

Покажем, что операции на машине B можно переупорядочить так, что в полученном расписании будут выполнены условия (C1) и (C3). Переупорядочение работ на машине B зависит от количества поездок машины B и от распределения по подмножествам N_{1R} и N_{1L} работ из N_1 в расписании $\bar{\sigma}$. Поэтому далее мы рассмотрим три случая.

СЛУЧАЙ 1: $\psi(B, \bar{\sigma}) \geq 4$, $N_{1L} \neq \emptyset$ и $N_{1R} \neq \emptyset$.

Преобразуем $\bar{\sigma}$ в расписание σ^* , для которого выполнены условия (C1)–(C3) и (C4) $\psi(B, \sigma^*) \leq 4$.

Рассмотрим следующую вспомогательную задачу I построения допустимого расписания работ на одной машине с заданными директивными сроками. Пусть $|R| = k$. Пример I содержит $k + 2$ работы, занумерованные от 1 до $k + 2$. Длительности работ p_i и директивные сроки D_i , $i = 1, \dots, k$, заданы в табл. 1.

Первые k работ соответствуют k операциям из $\{R\}_B$. Две последние работы соответствуют интервалам переезда машины B из одной вершины в другую. Покажем, что для данного примера существует допустимое расписание σ_R .

Пусть первые k работ в σ_R выполняются в те же интервалы времени, что и соответствующие им операции на машине B в расписании $\bar{\sigma}$. Если $J_i \in \bar{N}_{0R}''$, то по определению множества \bar{N}_{0R}'' работа J_i заканчивает выполнение до момента t_3 , и $C(i, \sigma_R) \leq t_3 = D_i$. Если $J_i \in R \setminus \bar{N}_{0R}''$, то J_i заканчивает выполнение до того, как она начнёт выполняться на A . Имеем $C(i, \sigma_R) \leq s(A_i, \bar{\sigma}) = D_i$.

Т а б л и ц а 1

Исходные данные примера I

i	p_i	D_i
$i : J_i \in R \setminus \bar{N}_{0R}''$	$p_i = b_i$	$D_i = s(A_i, \sigma_2)$
$i : J_i \in \bar{N}_{0R}''$	$p_i = b_i$	$D_i = t_3$
$k + 1$	$p_{k+1} = \tau$	$D_{k+1} = \min\{s(A_i, \sigma_2) \mid J_i \in N_{1R}\}$
$k + 2$	$p_{k+2} = \tau$	$D_{k+2} = C_{\max}$, если $N_{0R}'' = \emptyset$, или $D_{k+2} = \min\{s(A_i, \sigma_2) \mid J_i \in N_{0R}''\}$

Пусть в σ_R работа $k + 1$ выполняется в интервале времени длины τ , соответствующем первому переезду машины B из V_0 в V_1 . Обозначим время прибытия B в V_1 через t . Покажем, что $t \leq D_{k+1}$.

Действительно, так как $N_{1R} \neq \emptyset$, до момента $\min\{s(A_i, \bar{\sigma}) \mid J_i \in N_{1R}\}$ на B должна быть закончена по крайней мере одна операция работы из N_{1R} . Следовательно, $t \leq \min\{s(A_i, \bar{\sigma}) \mid J_i \in N_{1R}\} = D_{k+1}$.

Пусть в σ_R работа $k + 2$ выполняется в интервале времени длины τ , соответствующем первому переезду машины B из V_1 в V_0 . Обозначим время прибытия машины B в V_2 через t' . Очевидно, что $t' \leq C_{\max}$. Пусть $N_{0R}'' \neq \emptyset$. Тогда до момента $\min\{s(A_i, \bar{\sigma}) \mid J_i \in N_{0R}''\}$ на B должна быть закончена по крайней мере одна операция работы из N_{0R}'' . Следовательно, $t' \leq \min\{s(A_i, \bar{\sigma}) \mid J_i \in N_{0R}''\} = D_{k+2}$ и расписание σ_R допустимо относительно заданных директивных сроков.

Упорядочим в примере I все работы по неубыванию директивных сроков. Такая перестановка работ также определяет расписание, допустимое относительно директивных сроков. В исходной задаче такому расписанию соответствует следующий порядок выполнения операций на машине B :

$$0 \rightarrow \{N_{0R}'\}_B \rightarrow \{\bar{N}_{0R}''\}_B \xrightarrow{\tau} \{N_{1R}\}_B \xrightarrow{\tau} \{N_{0R}''\}_B.$$

Согласно этому порядку множество операций $\{R\}_B$ может быть выполнено без простоев на машине B в интервале $\left[0, \sum_{J_i \in R} b_i + 2\tau\right]$.

Рассмотрев течение времени в $\bar{\sigma}$ в обратном направлении от C_{\max} к 0 и проведя аналогичные рассуждения для множества L , получим, что операции множества $\{L\}_B$ можно выполнить на машине B в интервале $\left[C_{\max} - \sum_{J_i \in L} b_i - 2\tau, C_{\max}\right]$ без конфликта с операциями этих работ на A в следующем порядке:

$$\{N'_{0L}\}_B \xrightarrow{\tau} \{N_{1L}\}_B \xrightarrow{\tau} \{N'_{0L}\}_B \rightarrow \{N''_{0L}\}_B \rightarrow C_{\max}.$$

Так как $C_{\max}(\bar{\sigma}) \geq l_2 + 4\tau \geq \sum_{J_i \in N} b_i + 4\tau$, совместная длина интервалов $\left[0, \sum_{J_i \in R} b_i + 2\tau\right]$ и $\left[C_{\max} - \sum_{J_i \in L} b_i - 2\tau, C_{\max}\right]$ не превосходит C_{\max} .

Суммируя вышеизложенное, получаем расписание σ_1^* , в котором на машине A операции выполняются в порядке π_0 , а на машине B — в порядке π_1 :

$$\begin{aligned} 0 \rightarrow \{N'_{0R}\}_B \rightarrow \{\bar{N}''_{0R}\}_B \xrightarrow{\tau} \{N_{1R}\}_B \xrightarrow{\tau} \{N''_{0R}\}_B \\ \rightarrow \{N'_{0L}\}_B \xrightarrow{\tau} \{N_{1L}\}_B \xrightarrow{\tau} \{\bar{N}'_{0L}\}_B \rightarrow \{N''_{0L}\}_B \rightarrow C_{\max}. \end{aligned}$$

Лемма 2. Пусть σ — допустимое расписание, для которого выполнены следующие условия: $\psi(A, \sigma) = 2$, $\psi(B, \sigma) \geq 4$, $N_{1L}, N_{1R} \neq \emptyset$. Тогда существует расписание σ^* , в котором A обслуживает работы в порядке π_0 , B — в порядке π_1 и выполнены условия (C1)–(C4).

Отметим, что разбиение работ на множества L и R задаёт порядок выполнения операций для каждой работы, а перестановки π_0 и π_1 задают порядок между блоками операций одной машины. Будем говорить, что набор подмножеств \mathcal{N}_1 и перестановки π_0 и π_1 определяют конфигурацию K_1 . Допустимое расписание σ лежит в K_1 (будем писать $\sigma \in K_1$), если σ удовлетворяет следующим условиям:

- (i) множество работ N может быть разбито на набор \mathcal{N}_1 попарно не пересекающихся подмножеств, покрывающих все множество;
- (ii) все операции работ одного подмножества образуют блок на каждой машине в σ ;
- (iii) на машине A блоки выполняются в порядке π_0 , а на машине B — в порядке π_1 .

Как и расписания, конфигурации удобно представлять в виде ориентированного графа, где каждая вершина является блоком. Два парных блока будут связаны дугой, указывающей операции какого блока должны быть выполнены первыми. Вес дуги будет переменной, значение которой зависит от конкретного расписания.

Имеем $\sigma_1^* \in K_1$. Схема конфигурации K_1 представлена на рис. 4, где в верхних блоках опущен нижний индекс A , а в нижних — нижний индекс B .

Отметим, что в конфигурации K_1 представлено восемь парных блоков. Величины δ_i , $i = 1, \dots, 6$, указывают на минимальное время задержки между завершением выполнения первого блока и началом выполнения второго. Дуги между парными блоками подмножеств \bar{N}'_{0L} и \bar{N}''_{0R} опущены, так как интервалы, в которых они выполняются, не пересекаются. Нетрудно понять, что $\delta_i \geq 0$, и их конкретное значение зависит от перестановки операций в парных блоках. Вопрос определения оптимальной перестановки операций в парных блоках рассмотрен в следующем разделе.

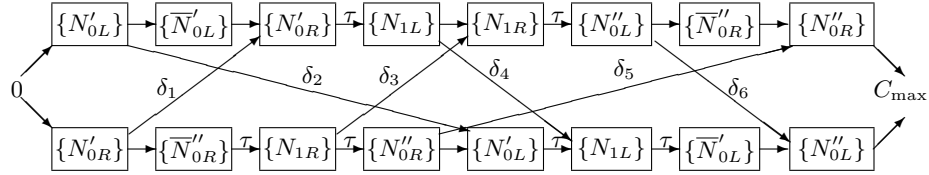


Рис. 4. Конфигурация K_1

СЛУЧАЙ 2: $\psi(B, \bar{\sigma}) \geq 2$, $N_{1L} \neq \emptyset$ и $N_{1R} = \emptyset$.

Последнее равенство влечёт $t_3 = 0$ и $\bar{N}''_{0R} = \emptyset$. Следовательно, перестановку π_0 выполнения работ на машине A можно переписать как π_2 :

$$0 \rightarrow \{N'_{0L}\}_A \rightarrow \{\bar{N}'_{0L}\}_A \rightarrow \{N'_{0R}\}_A \xrightarrow{\tau} \{N_{1L}\}_A \xrightarrow{\tau} \{N''_{0L}\}_A \rightarrow \{N''_{0R}\}_A \rightarrow C_{\max}.$$

Пусть $|R| = k$. Рассмотрим вспомогательную задачу I построения допустимого расписания k работ на одной машине с длительностями $p_i = b_i$ и заданными директивными сроками $D_i = s(A_i, \bar{\sigma})$.

Упорядочим в примере I все работы по неубыванию директивных сроков. Такая перестановка работ определяет расписание, допустимое относительно директивных сроков. В исходной задаче этому расписанию соответствует следующий порядок выполнения блоков операций из $\{R\}_B$ на машине B : $0 \rightarrow \{N'_{0R}\}_B \rightarrow \{N''_{0R}\}_B$. Согласно этому порядку множество операций $\{R\}_B$ может быть выполнено без простоев на машине B в интервале $\left[0, \sum_{J_i \in R} b_i\right]$.

Из анализа для случая 1 мы знаем, что операции множества $\{L\}_B$ можно выполнить на машине B в интервале $\left[C_{\max} - \sum_{J_i \in L} b_i - 2\tau, C_{\max}\right]$ без конфликта с операциями этих работ на A в следующем порядке:

$$\{N'_{0L}\}_B \xrightarrow{\tau} \{N_{1L}\}_B \xrightarrow{\tau} \{\bar{N}'_{0L}\}_B \rightarrow \{N''_{0L}\}_B \rightarrow C_{\max}.$$

В итоге получаем расписание σ_2^* , в котором на машине A операции выполняются в порядке π_2 , а на машине B — в порядке π_3 :

$$0 \rightarrow \{N'_{0R}\}_B \rightarrow \{N''_{0R}\}_B \rightarrow \{N'_{0L}\}_B \xrightarrow{\tau} \{N_{1L}\}_B \xrightarrow{\tau} \{\bar{N}'_{0L}\}_B \rightarrow \{N''_{0L}\}_B \rightarrow C_{\max}.$$

Лемма 3. Пусть σ — допустимое расписание такое, что $\psi(A, \sigma) = 2$, $\psi(B, \sigma) \geq 2$, $N_{1L} \neq \emptyset$ и $N_{1R} = \emptyset$. Тогда существует расписание σ^* , в котором A обслуживает работы в порядке π_2 , B — в порядке π_3 и выполнены условия (C1)–(C3) и (C5) $\psi(B, \sigma^*) \leq 2$.

Набор подмножеств $\mathcal{N}_2 = \{N'_{0R}, N''_{0R}, N'_{0L}, N_{1L}, \bar{N}'_{0L}, N''_{0L}\}$ и перестановки π_2 и π_3 определяют конфигурацию K_2 . Конфигурация K_2 , содержащая расписание σ_2^* , представлена на рис. 5, где в верхних блоках опущен нижний индекс A , а в нижних — нижний индекс B .

Случай, когда $N_{1L} = \emptyset$ и $N_{1R} \neq \emptyset$, симметричен рассмотренному и может быть сведён к нему заменой порядка выполнения операций на каждой из машин на обратный в исходном расписании σ .

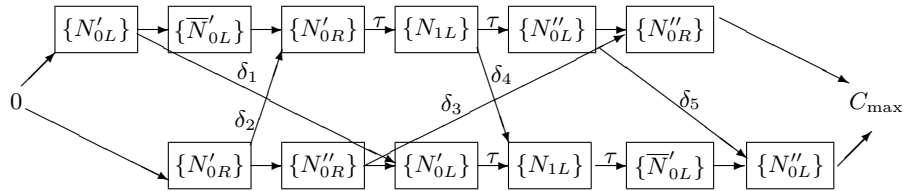


Рис. 5. Конфигурация K_2

СЛУЧАЙ 3: $\psi(A, \bar{\sigma}) = 2$, $\psi(B, \bar{\sigma}) = 2$, $N_{1L} \neq \emptyset$ и $N_{1R} \neq \emptyset$.

Пусть $t_5 = \min\{s(B_i, \bar{\sigma}) \mid J_i \in N_1\}$ и $t_6 = \max\{C(B_i, \bar{\sigma}) \mid J_i \in N_1\}$. Тогда $N_{1L} \neq \emptyset$ влечёт $t_1 \leq t_6$ и из $N_{1R} \neq \emptyset$ следует, что $t_5 \leq t_2$. Напомним, что на машине A в интервале $[0, t_1 - \tau]$ выполняются операции из $\{N'_{0L}\}_A \cup \{\bar{N}'_{0L}\}_A \cup \{N'_{0R}\}_A$, в $[t_1, t_2]$ — из $\{N_{1L}\}_A \cup \{N_{1R}\}_A$ и в $[t_1 + \tau, C_{\max}]$ — из $\{N'_{0L}\}_A \cup \{\bar{N}'_{0R}\}_A \cup \{N''_{0R}\}_A$. В свою очередь, на машине B операции из множеств $\{N'_{0L}\}_B$, $\{N'_{0R}\}_B$ и $\{\bar{N}'_{0R}\}_B$ выполняются

в интервале $[0, t_5 - \tau]$, операции из множеств $\{N_{1L}\}_B$ и $\{N_{1R}\}_B$ — в $[t_5, t_6]$, а остальные операции — в $[t_6 + \tau, C_{\max}]$. Так как $[0, t_1 - \tau] \cap [t_6 + \tau, C_{\max}] = \emptyset$ и $[0, t_5 - \tau] \cap [t_2 + \tau, C_{\max}] = \emptyset$, используя перестановочный приём, получим порядок π_4 выполнения операций на машине B :

$$\begin{aligned} 0 \rightarrow \{N'_{0R}\}_B \rightarrow \{\overline{N''}_{0R}\}_B \rightarrow \{N'_{0L}\}_B \xrightarrow{\tau} \{N_{1R}\}_B \\ \rightarrow \{N_{1L}\}_B \xrightarrow{\tau} \{N''_{0R}\}_B \rightarrow \{\overline{N'}_{0L}\}_B \rightarrow \{N''_{0L}\}_B \rightarrow C_{\max}. \end{aligned}$$

Лемма 4. Пусть σ — допустимое расписание такое, что $\psi(A, \sigma) = 2$, $\psi(B, \sigma) \geq 2$, $N_{1L} \neq \emptyset$ и $N_{1R} \neq \emptyset$. Тогда существует расписание σ^* , в котором A обслуживает работы в порядке π_0 , B — в порядке π_4 и выполнены условия (C1)–(C3) и (C5).

Набор подмножеств \mathcal{N}_A и перестановки π_0 и π_4 определяют конфигурацию K_3 , соответствующую расписанию σ_3^* , которая представлена на рис. 6, где в верхних блоках опущен нижний индекс A , а в нижних — нижний индекс B .

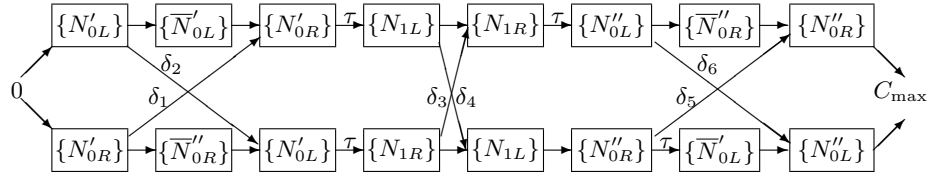


Рис. 6. Конфигурация K_3

Рассмотрим произвольное оптимальное расписание. Согласно леммам 2–4 оно может быть преобразовано в расписание такой же длины, принадлежащее одной из трёх описанных конфигураций. Как следствие, получим следующий результат.

Теорема 3. Пусть $J_0 \in N_1$ и $d_0 < l_{\max}$. Тогда существует оптимальное расписание σ^* такое, что $\sigma^* \in K_1 \cup K_2 \cup K_3$.

3.2. Построение оптимальной перестановки внутри идентичных блоков. В разд. 2 установлено, что при нахождении оптимального расписания достаточно ограничиться принадлежащими одной из трёх описанных конфигураций. Таким образом, достаточно найти оптимальное расписание в каждой из конфигураций и выбрать среди них наилучшее. Для нахождения оптимального расписания в данной конфигурации требуется распределить множество работ по заданным подмножествам

и определить внутри каждого подмножества порядок выполнения операций на каждой машине. В этом пункте предположим, что распределение работ по подмножествам задано и требуется определить оптимальную перестановку операций внутри каждого блока.

Пусть p_{ij} — время выполнения работы j на машине i . Перестановкой Джонсона π_{JR} называют последовательность, в которой сначала идут работы множества N_{\leq} в порядке неубывания значений a_j , а затем работы $N_{>}$ в порядке неубывания значений b_j . Через $\bar{\pi}_{JR}$ обозначим последовательность, в которой работы идут в порядке, обратном к перестановке Джонсона.

Поскольку первые операции работ в каждом блоке могут выполняться одна за другой без задержек, время работы первой машины, требуемое для выполнения всех операций из одного блока, равно сумме длительностей операций и не зависит от выбранной перестановки. Следовательно, требуется найти перестановку, которая минимизирует время завершения работы второй машины. Эта задача эквивалентна задаче построения минимального по длине расписания в системе поточного типа на двух машинах ($F2||C_{\max}$) [7]. Согласно введённым выше обозначениям перестановка π_{JR} оптимальна для работ из множеств N'_{0L} , \bar{N}'_{0L} , N_{1L} и N''_{0L} , а перестановка $\bar{\pi}_{JR}$ — для работ из множеств N'_{0R} , N_{1R} , N''_{0R} и \bar{N}''_{0R} .

Пусть X — одно из перечисленных выше множеств и $h(X)$ — длина расписания в соответствующей задаче $F2||C_{\max}$ для этого множества работ. Тогда вес дуги δ между парными блоками в конфигурациях вычисляется по формуле

$$\delta = h(X) - \sum_{J_j \in X} (a_j + b_j),$$

что соответствует времени простоя второй машины в оптимальном расписании для задачи $F2||C_{\max}$. Более того, если дуга между парными блоками X_A и X_B входит в критический путь, то общее время выполнения работ из X равно $h(X)$. Отсюда следует, что по назначению работ по множествам N'_{0L} , \bar{N}'_{0L} , N_{1L} , N''_{0L} , N'_{0R} , N_{1R} , N''_{0R} и \bar{N}''_{0R} можно определить длину расписания, оптимальную для данного назначения, в каждой конфигурации.

3.3. Алгоритм динамического программирования. В этом пункте опишем общую схему алгоритма динамического программирования, строящего оптимальное решение для каждой из трёх конфигураций, рассмотренных выше. Занумеруем работы из N согласно их порядку в π_{JR} . Для каждого $X \in \mathcal{N}_A$ обозначим через $[A, B, C]_X$ целочисленный вектор,

определяемый по назначению работ по множествам из \mathcal{N}_A , где $A = \sum_{i \in X} a_i$, $B = \sum_{i \in X} b_i$ и $C = h(X)$. Тогда каждое расписание $\sigma \in K_i$, $i = 1, 2, 3$, может быть закодировано упорядоченным набором W из не более чем восьми таких векторов. Пусть $\Delta = \sum_{j=1}^n d_j$. Очевидно, что значения координат векторов $[A, B, C]_X$ не превосходят Δ .

Пусть \mathcal{W}_k обозначает множество наборов, соответствующих допустимым расписаниям работ J_1, \dots, J_k . Тогда \mathcal{W}_0 содержит единственный набор с нулевыми векторами.

Пусть $W \in \mathcal{W}_j$, $j = 0, \dots, n-1$. Если работа J_{j+1} назначается в X , где X — одно из множеств N'_{0L} , \bar{N}'_{0L} , N_{1L} , N''_{0L} , то вектор $[A, B, C]_X$ преобразуется в вектор $[A + a_{j+1}, B + b_{j+1}, \max\{A + a_{j+1}, C\}]_X$, а все остальные векторы в наборе не меняются. Если работа J_{j+1} назначается в X , где X — одно из множеств N'_{0R} , N_{1R} , N''_{0R} , \bar{N}''_{0R} , то вектор $[A, B, C]_X$ заменяется вектором $[A + a_{j+1}, B + b_{j+1}, \max\{B + b_{j+1}, C\}]_X$, а все остальные векторы в наборе не меняются. В конце алгоритма среди наборов из \mathcal{W}_n выбирается тот, у которого длина полученного расписания наименьшая.

Дадим формальное описание алгоритма. Для удобства изложения положим $X_1 \doteq N_{1L}$, $X_2 \doteq N'_{0L}$, $X_3 \doteq N''_{0L}$, $X_4 \doteq \bar{N}'_{0L}$, $X_5 \doteq N_{1R}$, $X_6 \doteq N'_{0R}$, $X_7 \doteq N''_{0R}$ и $X_8 \doteq \bar{N}''_{0R}$.

АЛГОРИТМ DP

ШАГ 1. Положим $\mathcal{W}_0 = \{[0, 0, 0]_{X_k}\}$, $k = 1, \dots, 8$.

ШАГ 2. Для всех $j = 1, \dots, n$:

ШАГ 3. Если $J_j \in N_1$, то для каждого $W \in \mathcal{W}_{j-1}$:

ШАГ 3.1. Положив $[A, B, C]_{X_1} := [A + a_j, B + b_j, \max\{A + a_j, C\}]_{X_1}$ и $[A, B, C]_Y = [A, B, C]_Y$ для $Y \in \mathcal{N}_A \setminus X_1$, получим новый набор W' и добавим его в \mathcal{W}_j .

ШАГ 3.2. Положив $[A, B, C]_{X_5} := [A + a_j, B + b_j, \max\{B + b_j, C\}]_{X_5}$ и $[A, B, C]_Y = [A, B, C]_Y$ для $Y \in \mathcal{N}_A \setminus X_5$, получим новый набор W' и добавим его в \mathcal{W}_j .

ШАГ 4. Иначе для каждого $W \in \mathcal{W}_{j-1}$:

ШАГ 4.1. Для всех $k = 2, \dots, 4$:

ШАГ 4.2. Положив $[A, B, C]_{X_k} := [A + a_j, B + b_j, \max\{A + a_j, C\}]_{X_k}$ и $[A, B, C]_Y = [A, B, C]_Y$ для $Y \in \mathcal{N}_A \setminus X_k$, получим новый набор W' и добавим его в \mathcal{W}_j .

ШАГ 4.3. Для всех $k = 6, \dots, 8$:

ШАГ 4.4. Положив $[A, B, C]_{X_k} := [A + a_j, B + b_j, \max\{B + b_j, C\}]_X$ и $[A, B, C]_Y = [A, B, C]_Y$ для $Y \in \mathcal{N}_A \setminus X_k$, получим новый набор W' и добавим его в W_j .

ШАГ 5. Для каждого $W \in \mathcal{W}_n$ и $i = 1, \dots, 3$ вычислим длину расписания $C_{\max}(W, K_i)$ согласно конфигурации K_i . Для конфигурации K_2 рассматриваются только решения, в которых $[A, B, C]_{X_4} = [0, 0, 0]$ и $[A, B, C]_{X_8} = [0, 0, 0]$.

ШАГ 6. Выберем W^* и i^* такие, что

$$C_{\max}(W^*, K_{i^*}) = \min_{W \in \mathcal{W}_n, i \in \{1, 2, 3\}} C_{\max}(W, K_i).$$

Так как алгоритм DP перебирает все значения векторов, соответствующих допустимым расписаниям в искомым конфигурациях, он найдёт и наилучшие решения, принадлежащие этим конфигурациям. Вычислительная сложность алгоритма определяется общим количеством вычисляемых векторов и оценивается как $O(n\Delta^{24})$ элементарных операций.

Теорема 4. Алгоритм DP строит оптимальное расписание для трудного примера задачи $\text{ROS}_{2,2}$ не более чем за $O(n\Delta^{24})$ элементарных операций.

3.4. Вполне приближённая полиномиальная схема. Алгоритм решения задачи на минимум называется ρ -приближённым, если он за полиномиальное от размера входа задачи время находит решение, отличающееся от оптимального не более чем в ρ раз.

Семейство алгоритмов H_ε называется *вполне приближённой полиномиальной схемой*, если для любого $\varepsilon > 0$ любой алгоритм из H_ε является $(1 + \varepsilon)$ -приближённым алгоритмом и время его работы ограничено полиномом от размера входа и величины $1/\varepsilon$.

Вычислительная сложность алгоритма DP полиномиально зависит от Δ , и, следовательно, алгоритм DP псевдополиномиальный. Существует несколько различных способов преобразования алгоритма динамического программирования во вполне приближённую полиномиальную схему. Воспользуемся техникой сокращения числа рассматриваемых векторов [6]. Её основная идея — итеративно уменьшать число рассматриваемых векторов в ходе выполнения алгоритма динамического программирования, заменяя группу «близких» друг к другу векторов одним из представителей этой группы. Для полноты изложения опишем, как применить эту технику к решению задачи $\text{ROS}_{2,2}$.

$$\text{Пусть } \psi(x) = \begin{cases} 0 & \text{при } x = 0, \\ \lfloor \log_{(1+\varepsilon/2n)} x \rfloor & \text{при } x > 0. \end{cases}$$

Для набора векторов $W = \{[A, B, C]_{X_k} \mid k = 1, \dots, 8\}$ определим отображение $\Psi(W) = \{[\psi(A), \psi(B), \psi(C)]_{X_k} \mid k = 1, \dots, 8\}$. Будем говорить, что наборы W_1 и W_2 *близки друг к другу*, если $\Psi(W_1) = \Psi(W_2)$. Используя эти обозначения, опишем $(1+\varepsilon)$ -приближённый алгоритм для фиксированного $\varepsilon > 0$. На каждой итерации этого алгоритма, кроме «сокращённого» множества набора векторов \mathcal{W}'_i , соответствующих допустимым расписаниям, будем хранить «полное» множество их отображений \mathcal{V}_j , $j = 1, \dots, n$.

АЛГОРИТМ FPTAS(ε)

ШАГ 1. Положим $\mathcal{W}'_0 = \{[0, 0, 0]_{X_k}\}$, $k = 1, \dots, 8$.

ШАГ 2. Для всех $j = 1, \dots, n$:

ШАГ 3. Если $J_j \in N_1$, то для каждого $W \in \mathcal{W}_{j-1}$:

ШАГ 3.1. Получим новый набор W' , положив

$$[A, B, C]_{X_1} := [A + a_j, B + b_j, \max\{A + a_j, C\}]_{X_1} \text{ и } [A, B, C]_Y = [A, B, C]_Y$$

для $Y \in \mathcal{N}_A \setminus X_1$. Если $\Psi(W') \notin \mathcal{V}_j$, то добавим W' в \mathcal{W}'_j и $\Psi(W')$ в \mathcal{V}_j .

ШАГ 3.2. Получим новый набор W' , положив

$$[A, B, C]_{X_5} = [A + a_j, B + b_j, \max\{B + b_j, C\}]_{X_5} \text{ и } [A, B, C]_Y = [A, B, C]_Y$$

для $Y \in \mathcal{N}_A \setminus X_5$. Если $\Psi(W') \notin \mathcal{V}_j$, то добавим W' в \mathcal{W}'_j и $\Psi(W')$ в \mathcal{V}_j .

ШАГ 4. Иначе для каждого $W \in \mathcal{W}_{j-1}$:

ШАГ 4.1. Для всех $k = 2, 3, 4$:

ШАГ 4.2. Получим новый набор W' , положив

$$[A, B, C]_{X_k} := [A + a_j, B + b_j, \max\{A + a_j, C\}]_{X_k} \text{ и } [A, B, C]_Y = [A, B, C]_Y$$

для $Y \in \mathcal{N}_A \setminus X_k$. Если $\Psi(W') \notin \mathcal{V}_j$, то добавим W' в \mathcal{W}'_j и $\Psi(W')$ в \mathcal{V}_j .

ШАГ 4.3. Для всех $k = 6, 7, 8$:

ШАГ 4.4. Получим новый набор W' , положив

$$[A, B, C]_{X_k} = [A + a_j, B + b_j, \max\{B + b_j, C\}]_{X_k} \text{ и } [A, B, C]_Y = [A, B, C]_Y$$

для $Y \in \mathcal{N}_A \setminus X_k$. Если $\Psi(W') \notin \mathcal{V}_j$, то добавим W' в \mathcal{W}'_j и $\Psi(W')$ в \mathcal{V}_j .

ШАГ 5. Для каждого $W \in \mathcal{W}'_n$ и $i = 1, 2, 3$ вычислим длину расписания $C_{\max}(W, K_i)$ согласно конфигурации K_i . Для конфигурации K_2 рассматриваются только решения, в которых $[A, B, C]_{X_4} = [0, 0, 0]$ и $[A, B, C]_{X_8} = [0, 0, 0]$.

ШАГ 6. Выберем \bar{W} и \bar{i} такие, что

$$C_{\max}(\bar{W}, K_{\bar{i}}) = \min_{W \in \mathcal{W}_n, i \in \{1,2,3\}} C_{\max}(W, K_i).$$

Вычислительная сложность алгоритма определяется общим числом вычисляемых векторов. В начале итерации j число векторов ограничено размером множества \mathcal{V}_{j-1} и оценивается величиной

$$O(\log_{(1+\varepsilon/2n)} \Delta) \leq O((1 + 2n/\varepsilon) \ln \Delta).$$

Отсюда следует, что вычислительная сложность алгоритма оценивается через $O(n((1 + 2n/\varepsilon) \ln \Delta)^{24})$ и ограничена полиномом от размера входа и величины $1/\varepsilon$.

Осталось оценить качество полученного решения. Заметим, что на итерации j отбрасывается набор W , если существует другой набор W' такой, что $\Psi(W) = \Psi(W')$. Следовательно, для векторов $[A, B, C]_{X_i} \in W$ и $[A', B', C']_{X_i} \in W'$ имеют место неравенства

$$A' \leq A(1 + \varepsilon/2n), \quad B' \leq B(1 + \varepsilon/2n), \quad C' \leq C(1 + \varepsilon/2n), \quad (1)$$

откуда вытекает

Утверждение 1. Для каждого набора $W \in \mathcal{W}_n$, полученного алгоритмом DP, существует набор $W' \in \mathcal{W}'_n$ такой, что имеют место неравенства (1) для векторов $[A, B, C]_{X_i} \in W$ и $[A', B', C']_{X_i} \in W'$.

Утверждение 1 легко доказать по индукции, и его доказательство приведено в [8]. На шаге 5 алгоритмы DP и FPTAS(ε) вычисляют длину расписания для каждого полученного набора векторов и соответствующей конфигурации. Легко проверить, что длина любого пути из 0 в C_{\max} в расписании, соответствующем одной из трёх конфигураций, является линейной функцией от значений координат векторов $[A, B, C]_{X_i} \in W$, $i = 1, \dots, 8$.

Пусть W^* и i^* — набор векторов и номер конфигурации, соответствующие оптимальному решению, полученному алгоритмом DP. Тогда из утверждения 1 следует, что существует набор $W' \in \mathcal{W}'_n$, найденный алгоритмом FPTAS(ε), такой, что

$$C_{\max}(W', i^*) \leq (1 + \varepsilon/2n)^n C_{\max}(W^*, i^*) \leq (1 + \varepsilon) C_{\max}(W^*, i^*).$$

Согласно шагу 6 алгоритма FPTAS(ε) получаем

$$C_{\max}(\bar{W}, K_{\bar{i}}) \leq C_{\max}(W', i^*) \leq (1 + \varepsilon) C_{\max}(W^*, i^*),$$

откуда следует

Теорема 5. Семейство алгоритмов FPTAS(ε) является вполне полиномиальной приближённой схемой для задачи ROS_{2,2}.

ЛИТЕРАТУРА

1. Танаев В. С., Сотсков Ю. Н., Струсевич В. А. Теория расписаний. Многостадийные системы. — М.: Наука, 1989. — 329 с.
2. Averbakh I., Berman O., Chernykh I. A $\frac{6}{5}$ -approximation algorithm for the two-machine routing open shop problem on a 2-node network // Eur. J. Oper. Res. — 2005. — Vol. 166, N 1. — P. 3–24.
3. Averbakh I., Berman O., Chernykh I. The routing open-shop problem on a network: complexity and approximation // Eur. J. Oper. Res. — 2006. — Vol. 173, N 2. — P. 521–539.
4. Chernykh I., Dryuck N., Kononov A., Sevastyanov S. The routing open shop problem: new approximation algorithms // Approximation and Online algorithms. 7th Int. Workshop WAOA 2009 (Copenhagen, Denmark, September 10–11, 2009). Revised Papers. — Heidelberg: Springer-Verl., 2010. — P. 75–85. (Lect. Notes Comp. Sci.; Vol. 5893.)
5. Gonzalez T., Sahni S. Open shop scheduling to minimize finish time // J. ACM. — 1976. — Vol. 23 — P. 665–679.
6. Ibarra O., Kim C. E. Fast approximation algorithms for the knapsack and sum of subset problems // J. ACM. — 1975. — Vol. 22. — P. 463–468.
7. Johnson S. M. Optimal two- and three stage production schedules with set-up times included // Naval Res. Logistic Quarterly. — 1954. — Vol. 1. — P. 61–68.
8. Schuurman P., Woeginger G. Approximation schemes — a tutorial // <http://www.win.tue.nl/~gwoegi/papers/ptas.pdf>

Кононов Александр Вениаминович,
e-mail: alvenko@math.nsc.ru

Статья поступила
9 июня 2011 г.

Переработанный вариант —
24 ноября 2011 г.