

О МИНИМИЗАЦИИ ВРЕМЕНИ ПЕРЕДАЧИ ПАКЕТОВ
В АЦИКЛИЧЕСКОЙ СЕТИ С ЧЕТЫРЬМЯ СЕРВЕРАМИ *)

А. В. Кононов^{1,2}, П. А. Кононова^{1,2}

¹Институт математики им. С. Л. Соболева СО РАН,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия;

²Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия

e-mail: alvenko@math.nsc.ru, polik83@ngs.ru

Аннотация. Рассматривается задача оптимизации, которая возникает при передаче пакетов в иерархической ациклической сети. Данная задача является специальным случаем задачи построения кратчайшего расписания многопроцессорных работ на специализированных машинах. Изучается комбинаторная сложность её подзадач, характеризующихся фиксированным набором типов работ, где типом работы называется подмножество машин, на котором она выполняется. Ил. 17, библиогр. 14.

Ключевые слова: многопроцессорное расписание, полиномиальный алгоритм, NP-трудность.

Введение

Рассматривается задача оптимизации передачи данных в компьютерной сети, состоящей из множества компьютеров, каждый из которых напрямую соединён ровно с одним сервером. Серверы связаны между собой ациклической сетью. При передаче каждого пакета данных одновременно используются сервер передающего компьютера, сервер принимающего компьютера и все серверы, расположенные между ними. Для некоторых пакетов передающий и принимающий серверы совпадают. Считается, что одновременно сервер может обрабатывать только один пакет данных. Множество пакетов и их размеры заданы и известны заранее. Скорость передачи постоянная, поэтому для каждого пакета известно время передачи, пропорциональное его размеру. Прерывания в процессе

*) Исследование выполнено при финансовой поддержке Российского научного фонда (проект 15-11-10009).

передачи пакета не разрешены. Требуется найти порядок передачи пакетов, при котором все данные будут переданы за минимальное время. В случае, когда сеть представляет собой цепь и число серверов не фиксировано, эта задача также известна как задача о динамическом распределении памяти. Она присутствует в списке NP-трудных задач в монографии Гэри и Джонсона [1] о труднорешаемых комбинаторных задачах, в которой указано, что задача о динамическом распределении памяти NP-трудна в сильном смысле со ссылкой на неопубликованную работу Кьерстада.

Описанную выше задачу принято формулировать в терминах теории расписаний. Каждому серверу соответствует машина, и каждому пакету данных — работа, которая должна быть выполнена одновременно на множестве машин, соответствующих тем серверам, которые участвуют в передаче этого пакета. *Типом работы* назовём подмножество машин, на котором она выполняется. Длительность каждой работы равна времени передачи соответствующего пакета. Требуется найти расписание работ минимальной длины.

Более формально, рассматривается следующая задача. Заданы множество $\mathcal{M} = \{1, \dots, m\}$ параллельных машин и n многопроцессорных работ $\mathcal{J} = \{J_1, \dots, J_n\}$. На множестве машин задан связный ациклический граф смежности $G = (\mathcal{M}, \mathcal{E})$. Для каждой работы $J_j \in \mathcal{J}$ известна её длительность p_j и подмножество машин $\mu_j \subseteq \mathcal{M}$, на котором она должна быть выполнена, при этом вершины множества μ_j индуцируют цепь в графе G . Будем считать, что длительности p_j являются рациональными числами. *Расписанием* называется назначение для каждой работы $J_j \in \mathcal{J}$ полуоткрытого слева интервала длины p_j , так что для любых двух работ J_i и J_j с $\mu_i \cap \mu_j \neq \emptyset$ их интервалы выполнения не пересекаются. Требуется найти расписание работ, в котором время завершения последней работы минимально. Назовём сформулированную задачу задачей П.

Заметим, что задача П — специальный случай задачи $P|\text{fix}_j|C_{\max}$ минимизации длины расписания многопроцессорных работ на специализированных машинах [4, 9]. Основное отличие между задачами в том, что в задаче $P|\text{fix}_j|C_{\max}$ все типы работ допустимы. Ограничения на допустимые типы работ, как правило, связаны с пространственным расположением ресурсов. В англоязычной литературе такие типы ресурсов часто называют *смежными* [10]. Кроме многочисленных приложений, относящихся к передаче информации в сетях, постановки со смежными ресурсами возникают в оптимизации параллельных вычислений [9],

при планировании разгрузки судов [5, 13, 14], при назначении рейсов по терминалам аэропорта для регистрации пассажиров [8].

Задача $P|\text{fix}_j|C_{\max}$ достаточно хорошо изучена, ей посвящены десятки статей и даже отдельная глава в монографии [9] о задачах с многопроцессорными работами. В отличие от $P|\text{fix}_j|C_{\max}$ задаче Π , насколько нам известно, уделено гораздо меньше внимания. В [10] изучалась ситуация, в которой машины выстроены в линию и для каждой работы задано подмножество последовательно расположенных машин, т. е. граф G является цепью. Авторы показали, что задача полиномиально разрешима на трёх машинах, NP-трудна в случае четырёх и более машин и NP-трудна в сильном смысле, если число машин является частью входа рассматриваемой задачи. Отметим, что результаты статьи [10] показывают, что для малого числа машин задача Π проще задачи $P|\text{fix}_j|C_{\max}$, поскольку $P|\text{fix}_j|C_{\max}$ NP-трудна в сильном смысле уже в случае трёх машин [6, 12].

В свою очередь, задача $P|\text{fix}_j|C_{\max}$ является частным случаем задачи построения бесконфликтного расписания на неограниченном множестве параллельных машин. В этой задаче на множестве работ \mathcal{J} задан граф конфликтов $H = (\mathcal{J}, E)$. Каждое ребро $e \in E$ определяет пару работ, интервалы выполнения которых не могут пересекаться. В англоязычной литературе эта задача получила название задача о расписаниях с конфликтами (задача SWC) [11]. Нетрудно видеть, что любой пример задачи $P|\text{fix}_j|C_{\max}$ может быть сформулирован как пример задачи SWC, в котором для любой пары работ в графе H есть ребро тогда и только тогда, когда подмножества машин, необходимых для выполнения этих работ, пересекаются. Задача SWC с вычислительной точки зрения значительно сложнее задачи $P|\text{fix}_j|C_{\max}$. В случае, если все работы имеют единичную длительность, задача SWC эквивалентна классической задаче о минимальной раскраске графа, для которой построение приближённого алгоритма с относительной погрешностью меньше $n^{1-\varepsilon}$ для любого $\varepsilon > 0$ влечёт $P=NP$, тогда как для задачи $Pm|\text{fix}_j|C_{\max}$ с фиксированным числом машин m известна полиномиальная приближённая схема [3], а для случая, когда число машин не фиксировано, в [7] предложен $(2+\varepsilon)$ -приближённый алгоритм. Такой скачок сложности связан с тем, что для задачи $P|\text{fix}_j|C_{\max}$ граф H имеет хорошую структуру, все его вершины разбиты на кластеры (типы работ), число которых зависит от числа машин и ограничено константой, когда число машин фиксированно. При этом две вершины из одного кластера связаны ребром друг с другом и любая вершина, отличная от них, либо смежна с каждой из них, либо не имеет общих рёбер ни с одной из них. Это наблюдение позволяет

рассмотреть редуцированный граф конфликтов $\bar{H} = (T, E)$, где T — заданное множество типов работ, и ребро $e \in E$ между двумя типами работ указывает на то, что работы этих двух типов не могут выполняться одновременно. Без ограничения общности можно считать, что множество T не содержит пустого множества и множества, содержащего все машины.

Обозначим через Ω_m множество всех нетривиальных подмножеств множества \mathcal{M} . Рассмотрим семейство подзадач задачи $P|\text{fix}_j|C_{\max}$, в которых множество $T \subseteq \Omega_m$ типов работ задано. Заметим, что каждую подзадачу (класс примеров) можно определить либо явным перечислением всех заданных типов работ, т. е. указанием, на каком множестве машин работы этого типа выполняются, либо перечислением типов работ и редуцированным графом конфликтов на вершинах из T . Например, задача $P3|\text{fix}_j|C_{\max}$ определяется графом, изображённым на рис. 1. Очевидно, что задачи, в которых их редуцированные графы конфликтов совпадают, эквивалентны.

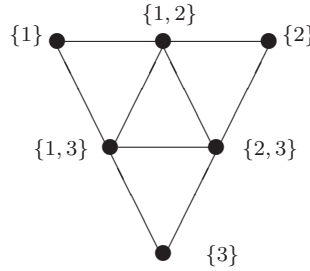


Рис. 1. Редуцированный граф конфликтов для задачи $P3|\text{fix}_j|C_{\max}$

Обозначим через $\Pi(T)$ класс примеров (подзадачу) задачи Π , в которой набор типов работ определяется множеством $T \subseteq \Omega_m$. Будем говорить, что задача $\Pi(T_1)$ является *подзадачей* задачи $\Pi(T_2)$, если $T_1 \subset T_2$. Данное определение согласуется с определением подзадачи, данным в [1]. Предположим, что число машин в задачах $\Pi(T_1)$ и $\Pi(T_2)$ ограничено константой. Тогда полиномиальная разрешимость задачи $\Pi(T_2)$ влечёт полиномиальную разрешимость задачи $\Pi(T_1)$ и, в обратную сторону, NP-трудность задачи $\Pi(T_1)$ влечёт NP-трудность задачи $\Pi(T_2)$.

Будем говорить, что класс примеров задачи Π , определённый множеством T , является *минимальным NP-трудным*, если подзадача $\Pi(T)$ NP-трудна и для любого нетривиального подмножества $T_1 \subset T$ подзадача $\Pi(T_1)$ полиномиально разрешима. Будем говорить, что класс примеров задачи Π , определённый множеством $T \subseteq \Omega_m$, является *максимально полиномиально разрешимым*, если подзадача $\Pi(T)$ полиномиально

разрешима и для любого его надмножества $T_1 \supset T$ подзадача $\Pi(T_1)$ NP-трудна.

В статье рассматривается задача Π на четырёх машинах (задача Π_4). У задачи Π_4 имеется конечное число подзадач, которые характеризуются множеством T — набором присутствующих типов работ. Напомним, что в рассматриваемой задаче на множестве машин задан связный ациклический граф смежности G и каждый тип — это простая цепь в графе G . Следовательно, набор возможных типов зависит от структуры графа G . Нетрудно увидеть, что есть только два топологически разных варианта связного ациклического графа на четырёх вершинах — это цепь и звезда. Цепи на четырёх вершинах соответствуют 9 типов работ и $2^9 - 1 = 511$ возможных подзадач задачи Π_4 . Звезде с четырьмя вершинами соответствуют 10 типов работ и $2^{10} - 1 = 1023$ возможных подзадач задачи Π_4 . Целью нашей работы является полная классификация сложности всех возможных подзадач задачи Π_4 по всем возможным значениям параметра T .

Для этого в статье описаны все максимальные полиномиально разрешимые классы примеров задачи Π_4 относительно параметра T . Доказано, что в примерах из этих классов длина оптимального расписания совпадает с нижней оценкой оптимума. Дополнительно выделены все минимальные NP-трудные классы примеров задачи Π_4 относительно параметра T . В разд. 1 рассматриваются задачи, в которых машины соединены последовательно, т. е. граф смежности является цепью. В разд. 2 рассмотрены задачи, в которых одна машина соединена со всеми остальными машинами, т. е. граф смежности — звезда.

1. Последовательное соединение

В этом разделе рассмотрим задачу Π на четырёх машинах, которые соединены в одну линию, как показано на рис. 2.

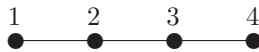


Рис. 2. Граф смежности в задаче Π_4^L

Обозначим полученную задачу через Π_4^L . Нетрудно проверить, что в задаче Π_4^L возможны 9 нетривиальных типов работ:

$$\mathcal{T}_4 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{2, 3\}, \{3, 4\}, \{1, 2, 3\}, \{2, 3, 4\}\}.$$

Обозначим через λ_μ сумму длительностей всех работ типа $\mu \subseteq M$. Отметим, что запись $\lambda_\mu = 0$ эквивалентна тому, что в задаче нет работ типа μ .

В [10] показано, что задача Π_4^L NP-трудна. В этом разделе выделим все полиномиально разрешимые случаи задачи Π_4^L для различных множеств $T \subseteq \mathcal{T}_4$. Более точно, для каждого множества $T \subseteq \mathcal{T}_4$ либо установим, что задача Π_4^L на этом множестве NP-трудна, либо покажем существование линейного от числа работ алгоритма её решения.

Теорема 1. *Если $T = \mathcal{T}_4 \setminus \{2\}$ или $T = \mathcal{T}_4 \setminus \{3\}$, то задача $\Pi_4^L(T)$ NP-трудна.*

ДОКАЗАТЕЛЬСТВО. В силу симметрии докажем NP-трудность только для одного из этих двух случаев, например, когда $T = \mathcal{T}_4 \setminus \{3\}$.

Рассмотрим пример задачи Π_4^L с девятью работами, по одной работе каждого типа. Для удобства в качестве индексов работ будем использовать множество машин, на которых они выполняются.

Длительности работ зададим следующим образом:

$$p_{\{1\}} = \frac{4}{7}, \quad p_{\{2\}} = p_{\{3\}} = 0, \quad p_{\{4\}} = \frac{3}{7},$$

$$p_{\{1,2\}} = \frac{2}{7}, \quad p_{\{2,3\}} = \frac{1}{7}, \quad p_{\{3,4\}} = \frac{3}{7}, \quad p_{\{1,2,3\}} = \frac{1}{7}, \quad p_{\{2,3,4\}} = \frac{1}{7}.$$

Обозначим это множество работ через I . Покажем, что для данного примера существует единственное с точностью до симметрии расписание длины 1. Заметим, что в таком расписании в интервале $[0, 1]$ машины 1 и 4 должны работать без простоев, суммарный простой на машине 2 равен $\frac{2}{7}$, а суммарный простой на машине 3 равен $\frac{1}{7}$. Переберём все возможные расписания, фиксируя порядок выполнения работ на машине 4. Заметим, что порядок, обратный к заданному, определяет расписание той же длины. Рассмотрим три случая, определяемых порядком выполнения работ на четвёртой машине.

СЛУЧАЙ 1. Пусть работы выполняются в порядке $J_{\{2,3,4\}}, J_{\{4\}}, J_{\{3,4\}}$. Заметим, что только работа $J_{\{1\}}$ может выполняться параллельно с работой $J_{\{2,3,4\}}$. Тогда для того чтобы не было простоя на машине 1, работа $J_{\{1\}}$ должна выполняться первой. Следовательно, работа $J_{\{1\}}$ выполняется в интервале $[0, \frac{4}{7}]$. Так как работа $J_{\{3,4\}}$ выполняется в интервале $[\frac{4}{7}, 1]$, работа $J_{\{1,2,3\}}$ не может быть выполнена в интервале $[0, 1]$ без пересечения с конфликтующими с ней работами (рис. 3). Следовательно, расписания длины 1 в этом случае не существует.

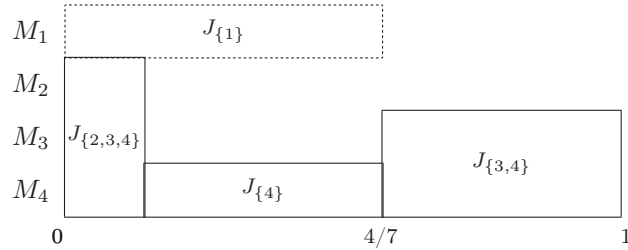


Рис. 3. Иллюстрация к случаю 1 теоремы 1

СЛУЧАЙ 2. Пусть работы выполняются в порядке $J_{\{2,3,4\}}, J_{\{3,4\}}, J_{\{4\}}$ (рис. 4). Поскольку работа $J_{\{1\}}$ должна выполняться параллельно с работой $J_{\{2,3,4\}}$, она занимает интервал $[0, \frac{4}{7}]$. Учитывая, что работа $J_{\{3,4\}}$ выполняется в интервале $[\frac{1}{7}, \frac{4}{7}]$, получим, что на машине 2 в интервале $[\frac{1}{7}, \frac{4}{7}]$ не может выполняться ни одной работы. Отсюда следует, что простой на машине 2 не меньше $\frac{3}{7}$. Это противоречит выводу о величине простоя на этой машине в расписании длины 1, а значит, расписания длины 1 в этом случае не существует.

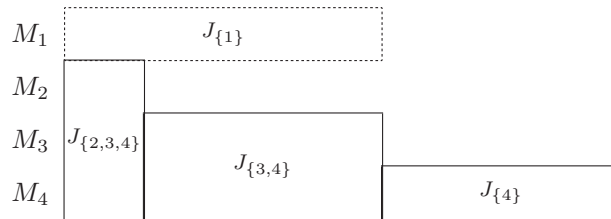


Рис. 4. Иллюстрация к случаю 2 теоремы 1

СЛУЧАЙ 3. Пусть работы выполняются в порядке $J_{\{3,4\}}, J_{\{2,3,4\}}, J_{\{4\}}$. Если работа $J_{\{1\}}$ выполняется первой на машине 1 в интервале $[0, \frac{4}{7}]$, то её выполнение параллельно с работой $J_{\{3,4\}}$ в интервале $[0, \frac{3}{7}]$ и создаёт простой длины $\frac{3}{7}$ на машине 2, что противоречит выводу о величине простоя на этой машине в расписании длины 1 (рис. 5). Так как работа $J_{\{1,2,3\}}$ не может стартовать раньше момента $\frac{4}{7}$, первой на машине 1 может выполняться только работа $J_{\{1,2\}}$, а затем работы $J_{\{1\}}$ и $J_{\{1,2,3\}}$. Тогда работа $J_{\{2,3\}}$ выполняется в интервале $[\frac{4}{7}, \frac{6}{7}]$. Полученное расписание является единственным расписанием длины 1 с точностью до мест-

расположения работы $J_{\{2,3\}}$ (рис. 6). В любом случае машина 2 простаивает в интервале $[\frac{2}{7}, \frac{3}{7}]$ и выполняет работы в интервалах $[0, \frac{2}{7}]$ и $[\frac{3}{7}, \frac{4}{7}]$. Кроме того, в интервале $[\frac{4}{7}, \frac{6}{7}]$ машина 2 выполняет работу $J_{\{2,3\}}$ в течение $\frac{1}{7}$ единицы времени и простаивает в течение $\frac{1}{7}$ единицы времени.

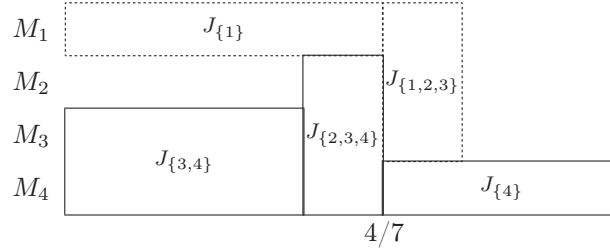


Рис. 5. Иллюстрация к случаю 3 теоремы 1

Рассмотрим известную NP-трудную задачу о разбиении. Заданы конечное множество элементов X и их веса $s(x) \in \mathbb{Z}^+$, $x \in X$. Требуется найти подмножество $X' \subseteq X$ такое, что

$$\sum_{x \in X'} s(x) = \sum_{x \in X \setminus X'} s(x),$$

либо показать, что такого подмножества не существует.

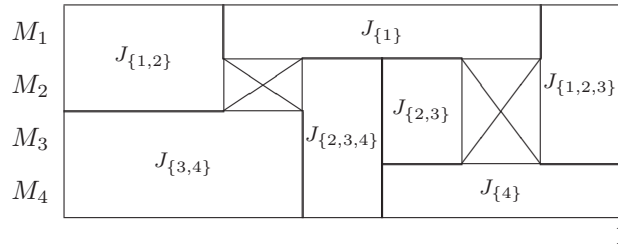


Рис. 6. Оптимальное расписание в примере I

Рассмотрим произвольный пример задачи о разбиении. Пусть множество X состоит из n элементов и суммарная стоимость всех элементов равна Y . Построим по этому примеру следующий пример задачи Π_4^L . К множеству работ I из описанного выше примера из девяти работ добавим множество работ J . Множество J состоит из n работ, которые выполняются на машине 2. Каждая работа J_x соответствует элементу x

множества X . Длительность каждой работы p_x равна $\frac{2x}{7Y}$, $x \in X$. Покажем, что расписание длины 1 существует тогда и только тогда, когда задача о разбиении имеет решение.

Действительно, если существует расписание длины 1, то машина 2 должна работать без простоев. Следовательно, существует подмножество работ J' , которые целиком занимают интервал $[\frac{2}{7}, \frac{3}{7}]$ и их суммарная длительность равна $\frac{1}{7}$. Тогда определим подмножество X' как множество элементов, соответствующих работам из J' . Их суммарный вес равен $\frac{Y}{2}$. Следовательно, задача о разбиении имеет решение тогда и только тогда, когда для построенного примера существует расписание длины 1. Значит, задача $\Pi_4^L(\mathcal{T}_4 \setminus \{\{3\}\})$ NP-трудна.

По аналогии с вышеизложенным можно доказать NP-трудность задачи $\Pi_4^L(\mathcal{T}_4 \setminus \{\{2\}\})$. Теорема 1 доказана.

Итак, из теоремы 1 следует, что задача $\Pi_4^L(T)$ NP-трудна, когда параметр T принимает одно из значений \mathcal{T}_4 , $\mathcal{T}_4 \setminus \{\{2\}\}$ или $\mathcal{T}_4 \setminus \{\{3\}\}$. Покажем, что остальные случаи задачи Π_4^L полиномиально разрешимы. Для этого рассмотрим расписания, в которых работы одного типа выполняются последовательно без простоев. Докажем, что среди таких расписаний всегда существует расписание, длина которого равна максимальной машинной нагрузке.

Введём соответствующие обозначения. Обозначим через l_i нагрузку машины M_i . Нагрузки машин определяют четыре тривиальные нижние оценки оптимума рассматриваемой задачи:

$$\begin{aligned} l_1 &= \lambda_{\{1\}} + \lambda_{\{1,2\}} + \lambda_{\{1,2,3\}}, \\ l_2 &= \lambda_{\{2\}} + \lambda_{\{1,2\}} + \lambda_{\{2,3\}} + \lambda_{\{1,2,3\}} + \lambda_{\{2,3,4\}}, \\ l_3 &= \lambda_{\{3\}} + \lambda_{\{2,3\}} + \lambda_{\{3,4\}} + \lambda_{\{1,2,3\}} + \lambda_{\{2,3,4\}}, \\ l_4 &= \lambda_{\{4\}} + \lambda_{\{3,4\}} + \lambda_{\{2,3,4\}}. \end{aligned}$$

Тогда $l_{\max} = \max_i l_i$ будет нижней оценкой длины оптимального расписания C_{\max}^* . Величина l_{\max} называется *максимальной машинной нагрузкой*.

Решения в задачах теории расписаний удобно представлять ориентированным ациклическим графом (сетью) $\overline{G} = (V, E)$, в котором множество вершин V содержит по одной вершине для каждой работы и две специальные вершины — источник 0 и сток C_{\max} . Каждой вершине приписан неотрицательный вес, равный длительности соответствующей работы. Поскольку будем рассматривать расписания, в которых работы одного типа выполняются последовательно, заменим все работы одного типа одной работой того же типа, длительность которой равна сум-

марной длительности всех работ этого типа. Таким образом, в нашем случае множество вершин V будет включать источник, сток и по одной вершине J_μ для каждого типа $\mu \in T$, а вес этой вершины будет равен λ_μ . Веса стока и источника положим равными 0. Так как никакие две работы J_μ и $J_{\mu'}$ с $\mu \cap \mu' \neq \emptyset$ не могут выполняться одновременно, в сети \bar{G} должна существовать дуга между вершинами J_μ и $J_{\mu'}$. Кроме того, в графе \bar{G} заданы дуги из источника в каждую вершину и из каждой вершины в сток. Длина пути в данном графе определяется как сумма весов его вершин.

Построим по заданному графу \bar{G} расписание σ по следующему правилу. Для каждой работы J_μ положим момент начала её обслуживания равным длине максимального пути из 0 в J_μ . Расписание, построенное по такому правилу, называется *активным*, и каждый такой граф определяет единственное активное расписание [2]. Длина активного расписания равна длине максимального пути из источника в сток. Дуга $(x, y) \in E$ называется *транзитивной*, если в \bar{G} существует путь из x в y , проходящий через другую вершину. Путь из источника в сток назовём *полным*, если он не содержит транзитивных дуг. Максимальный по длине полный путь называется *критическим*. Легко показать, что в графе \bar{G} всегда существует максимальный путь из источника в сток, который является критическим.

Заметим, что если в сети все полные пути совпадают с путями, определяющими нагрузку машин, то такая сеть определяет оптимальное расписание.

Лемма 1. *Задача Π_4^L , в которой нет работ типа $\{1\}$, полиномиально разрешима.*

ДОКАЗАТЕЛЬСТВО. Рассмотрим произвольный пример задачи Π_4^L , в котором нет работ типа $\{1\}$. Заменяем все работы каждого типа μ одной работой того же типа J_μ и положим $p_\mu = \lambda_\mu$. Рассмотрим сеть \bar{G}_1 , изображённую на рис. 7. В этой сети есть три полных пути:

$$0 \rightarrow J_{\{2,3,4\}} \rightarrow J_{\{2\}} \rightarrow J_{\{1,2\}} \rightarrow J_{\{1,2,3\}} \rightarrow J_{\{2,3\}} \rightarrow C_{\max},$$

$$0 \rightarrow J_{\{2,3,4\}} \rightarrow J_{\{3,4\}} \rightarrow J_{\{3\}} \rightarrow J_{\{1,2,3\}} \rightarrow J_{\{2,3\}} \rightarrow C_{\max},$$

$$0 \rightarrow J_{\{2,3,4\}} \rightarrow J_{\{3,4\}} \rightarrow J_{\{4\}} \rightarrow C_{\max},$$

каждый из которых задаёт нагрузки машин 2, 3 и 4 соответственно. Следовательно активное расписание, определяемое сетью \bar{G}_1 , оптимально. Лемма 1 доказана.

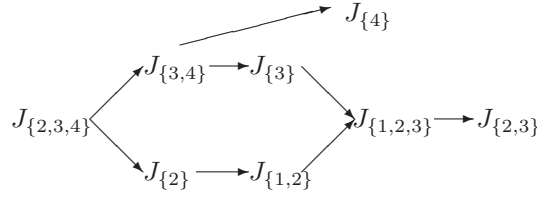


Рис. 7. Сеть \bar{G}_1 для задачи Π_4^L , в которой нет работ типа $\{1\}$

Лемма 2. *Задача Π_4^L , в которой нет работ типа $\{1, 2\}$ (типа $\{3, 4\}$), полиномиально разрешима.*

ДОКАЗАТЕЛЬСТВО. Рассмотрим произвольный пример задачи Π_4^L , в котором нет работ типа $\{1, 2\}$. Заменяем все работы каждого типа μ одной работой того же типа J_μ и положим $p_\mu = \lambda_\mu$. Рассмотрим сеть \bar{G}_2 , изображённую на рис. 8. В этой сети есть четыре полных пути:

$$\begin{aligned} 0 &\rightarrow J_{\{1\}} \rightarrow J_{\{1,2,3\}} \rightarrow C_{\max}, \\ 0 &\rightarrow J_{\{2,3,4\}} \rightarrow J_{\{2\}} \rightarrow J_{\{2,3\}} \rightarrow J_{\{1,2,3\}} \rightarrow C_{\max}, \\ 0 &\rightarrow J_{\{2,3,4\}} \rightarrow J_{\{3,4\}} \rightarrow J_{\{3\}} \rightarrow J_{\{2,3\}} \rightarrow J_{\{1,2,3\}} \rightarrow C_{\max}, \\ 0 &\rightarrow J_{\{2,3,4\}} \rightarrow J_{\{3,4\}} \rightarrow J_{\{4\}} \rightarrow C_{\max}, \end{aligned}$$

каждый из которых задаёт нагрузки машин 1, 2, 3 и 4 соответственно. Следовательно, активное расписание, определяемое сетью \bar{G}_2 , оптимально.

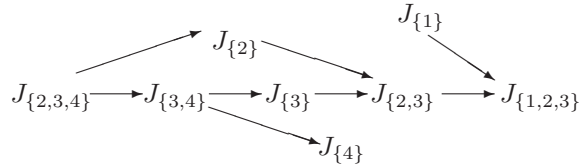


Рис. 8. Сеть \bar{G}_2 для задачи Π_4^L , в которой нет работ типа $\{1, 2\}$

В силу симметрии задачи оптимальная сеть для примера задачи, в котором нет работ типа $\{3, 4\}$, строится заменой в индексах машины i на машину $5 - i$ для $i = 1, \dots, 4$. Лемма 2 доказана.

Лемма 3. *Задача Π_4^L , в которой нет работ типа $\{2, 3\}$, полиномиально разрешима.*

ДОКАЗАТЕЛЬСТВО. Рассмотрим произвольный пример задачи Π_4^L , в котором нет работ типа $\{2, 3\}$. Заменяем все работы каждого типа μ одной работой того же типа J_μ и положим $p_\mu = \lambda_\mu$. Рассмотрим сеть \bar{G}_3 ,

изображённую на рис. 9. Аналогично доказательствам лемм 1 и 2 легко проверить, что все полные пути определяют нагрузку одной из машин, следовательно, активное расписание, определяемое сетью \overline{G}_3 , оптимально. Лемма 3 доказана.

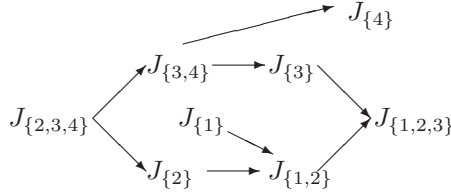


Рис. 9. Сеть \overline{G}_3 для задачи Π_4^L , в которой нет работ типа $\{2, 3\}$

Лемма 4. Задача Π_4^L , в которой нет работ типа $\{1, 2, 3\}$ ($\{2, 3, 4\}$), полиномиально разрешима.

ДОКАЗАТЕЛЬСТВО. Рассмотрим произвольный пример задачи Π_4^L , в котором нет работ типа $\{1, 2, 3\}$. Заменяем все работы каждого типа μ одной работой того же типа J_μ и положим $p_\mu = \lambda_\mu$. Рассмотрим сеть \overline{G}_4 , изображённую на рис. 10. Аналогично доказательствам лемм 1 и 2 легко проверить, что все полные пути определяют нагрузку одной из машин, следовательно, активное расписание, определяемое сетью \overline{G}_4 , оптимально.

В силу симметрии задачи оптимальная сеть для примеров, в которых нет работ типа $\{2, 3, 4\}$, строится заменой в индексах машины i на машину $5 - i$ для $i = 1, \dots, 4$. Лемма 4 доказана.

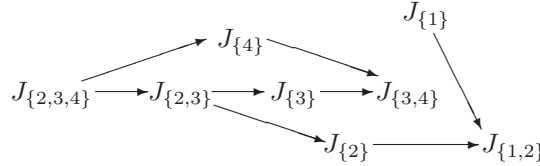


Рис. 10. Сеть \overline{G}_4 для задачи Π_4^L , в которой нет работ типа $\{1, 2, 3\}$

Лемма 5. Задача Π_4^L , в которой нет работ типов $\{2\}$ и $\{3\}$, полиномиально разрешима.

ДОКАЗАТЕЛЬСТВО. Рассмотрим произвольный пример задачи Π_4^L , в котором нет работ типов $\{2\}$ и $\{3\}$. Заменяем все работы каждого типа μ одной работой того же типа J_μ и положим $p_\mu = \lambda_\mu$.

Рассмотрим две сети \overline{G}_5 и \overline{G}_6 , изображённые на рис. 11. Каждая из них имеет пять полных путей. Четыре пути в каждой сети задают нагрузку одной из машин. Сеть \overline{G}_5 содержит один нетривиальный полный путь

$0 \rightarrow J_{\{4\}} \rightarrow J_{\{2,3,4\}} \rightarrow J_{\{1,2\}} \rightarrow C_{\max}$, и сеть \bar{G}_6 содержит один нетривиальный полный путь $0 \rightarrow J_{\{1\}} \rightarrow J_{\{1,2,3\}} \rightarrow J_{\{3,4\}} \rightarrow C_{\max}$. Покажем, что оба пути не могут быть одновременно критическими путями из источника в сток. Для того чтобы первый из указанных путей был критическим в сети \bar{G}_5 , необходимо, чтобы выполнялось неравенство $\lambda_{\{1,2\}} > \lambda_{\{3,4\}}$. Для того чтобы второй путь был критическим путём в сети \bar{G}_6 , необходимо, чтобы выполнялось неравенство $\lambda_{\{3,4\}} > \lambda_{\{1,2\}}$; противоречие. Следовательно, кратчайшее из двух активных расписаний, определяемых сетями \bar{G}_5 и \bar{G}_6 , оптимально. Лемма 5 доказана.

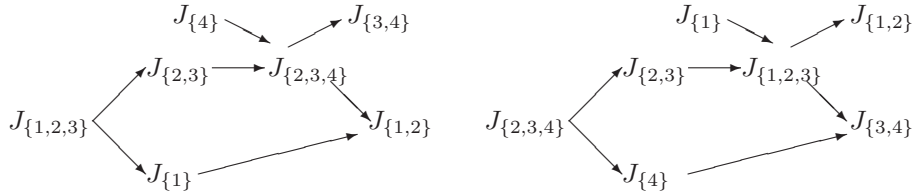


Рис. 11. Сети \bar{G}_5 и \bar{G}_6 для задачи Π_4^L , в которой нет работ типов $\{2\}$ и $\{3\}$

В леммах 1–5 рассмотрены все максимальные по включению подмножества \mathcal{T}_4 , отличные от $\mathcal{T}_4 \setminus \{\{2\}\}$ и $\mathcal{T}_4 \setminus \{\{3\}\}$. Для каждого из этих подмножеств существует сеть, определяющая расписание, длина которого совпадает с нижней оценкой. Заметим, что вычисление величин λ_μ требует $O(n)$ элементарных операций, а построение ациклического графа \bar{G} и оптимального расписания можно осуществить за константное число шагов.

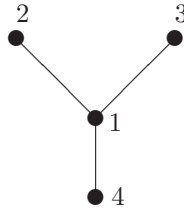


Рис. 12. Граф смежности в задаче Π_4^S

Теорема 2. Пусть множество типов работ в примере задачи Π_4^L не совпадает с \mathcal{T}_4 , $\mathcal{T}_4 \setminus \{\{2\}\}$ и $\mathcal{T}_4 \setminus \{\{3\}\}$. Тогда существует линейный алгоритм построения оптимального расписания, при этом длина оптимального расписания совпадает с максимальной машинной нагрузкой.

2. Соединение — звезда

В этом разделе рассмотрим задачу Π на четырёх машинах, в которой одна машина соединена со всеми остальными, а они соединены только с ней, как показано на рис. 12.

Обозначим новую задачу через Π_4^S . Заметим, что в такой конфигурации возможны 10 нетривиальных типов работ:

$$\mathcal{T}_4^* = \{\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}\}.$$

Теорема 3. Пусть $T = \mathcal{T}_4^* \setminus \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}\}$. Тогда задача $\Pi_4^S(T)$ является NP-трудной.

Доказательство. Результат теоремы следует из того факта, что редуцированный граф конфликтов (рис. 13) для рассматриваемой задачи совпадает с редуцированным графом конфликтов задачи $P3|fix_j|C_{\max}$. Теорема 3 доказана.

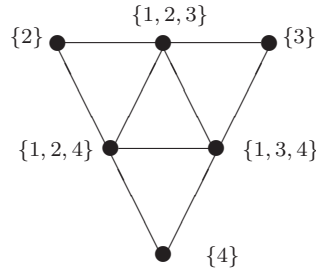


Рис. 13. Редуцированный граф конфликтов в задаче Π_4^S , в которой нет работ типов $\{1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}$

Теорема 4. Пусть $T = \mathcal{T}_4^* \setminus \{\{1\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}\}$. Тогда задача $\Pi_4^S(T)$ является NP-трудной.

Доказательство. Рассмотрим пример задачи PS с пятью работами, по одной работе каждого из пяти следующих типов: $\{2\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}$, и длительностями работ $p_{\{2\}} = p_{\{4\}} = \frac{2}{3}$, $p_{\{1, 2\}} = p_{\{1, 3\}} = p_{\{1, 4\}} = \frac{1}{3}$. Обозначим это множество работ через I .

Заметим, что в расписании длины 1 машины 1, 2 и 4 должны работать без простоев, а машина 3 должна простаивать $\frac{2}{3}$ единицы времени в интервале $[0, 1]$. Отсюда легко получить, что для данного примера расписание длины 1 существует тогда и только тогда, когда машина 3 выполняет работу $J_{\{1, 3\}}$ в интервале $[\frac{1}{3}, \frac{2}{3}]$ и простаивает в интервалах $[0, \frac{1}{3}]$ и $[\frac{2}{3}, 1]$. Пример одного из таких расписаний изображён на рис. 14.

Рассмотрим произвольный пример NP-трудной задачи о разбиении. Пусть множество X состоит из n элементов и суммарная стоимость всех

элементов равна Y . Построим по этому примеру пример задачи Π_4^S . К множеству работ I добавим множество работ J . Множество J состоит из n работ, которые выполняются на машине 3. Каждая работа $J_x \in J$ соответствует элементу x множества X . Длительность каждой работы $p_x = \frac{2x}{3Y}$, $x \in X$. Очевидно, что расписание длины 1 существует тогда и только тогда, когда задача о разбиении имеет решение. Теорема 4 доказана.

M_1	$J_{\{1,2\}}$	$J_{\{1,3\}}$	$J_{\{1,4\}}$
M_2		$J_{\{2\}}$	
M_3		$J_{\{1,3\}}$	
M_4	$J_{\{4\}}$		$J_{\{1,4\}}$

Рис. 14. Расписание длины 1 для работ из множества I

Теоремы 3 и 4 описывают все минимальные NP-трудные случаи задачи Π_4^S . В следующей теореме перечислим все максимальные полиномиально разрешимые классы примеров этой задачи.

Теорема 5. Пусть в примере I задачи Π_4^S выполнено одно из следующих условий:

- (i) нет работ по крайней мере одного из типов $\{2\}, \{3\}, \{4\}$;
- (ii) нет работ по крайней мере одного из типов $\{1, 2\}, \{1, 3\}, \{1, 4\}$ и по крайней мере одного из типов $\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}$.

Тогда для примера I существует линейный алгоритм построения оптимального расписания, длина которого совпадает с максимальной машинной нагрузкой.

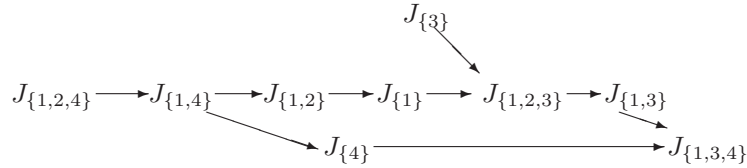


Рис. 15. Сеть \overline{G}_7

ДОКАЗАТЕЛЬСТВО. Рассмотрим произвольный пример задачи Π_4^S , в котором нет работ типа $\{2\}$. Заменяем все работы каждого типа μ одной работой того же типа J_μ и положим $p_\mu = \lambda_\mu$. Рассмотрим сеть \overline{G}_7 , изображённую на рис. 15. В ней три полных пути, которые задают нагрузку

машин 1, 3 и 4. Следовательно, расписание, определяемое сетью \bar{G}_7 , оптимально. Оптимальные сети для задач, в которых нет работ типа $\{3\}$ или $\{4\}$, строятся аналогично.

Пусть в примере задачи Π_4^S нет работ типов $\{1, 2\}$ и $\{1, 2, 3\}$. Аналогично рассуждению в предыдущем случае склеим все работы одного типа в одну и получим пример с восемью работами. Рассмотрим сеть \bar{G}_8 , изображённую на рис. 16. В ней четыре полных пути, длина каждого из которых совпадает с нагрузкой соответствующей машины. Следовательно, активное расписание, определяемое сетью \bar{G}_8 , оптимально.

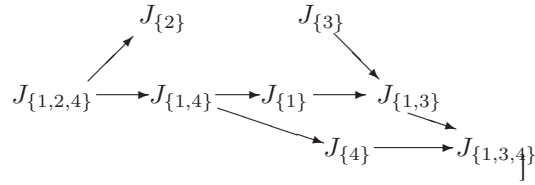


Рис. 16. Сеть \bar{G}_8

Наконец, предположим, что в примере задачи Π_4^S нет работ типов $\{1, 2\}$ и $\{1, 3, 4\}$. Заменяем все работы одного типа одной работой и рассмотрим для полученного примера сеть \bar{G}_9 , представленную на рис. 17. В ней четыре полных пути, длина каждого из которых совпадает с нагрузкой соответствующей машины. Значит, сеть \bar{G}_9 задаёт оптимальное активное расписание.

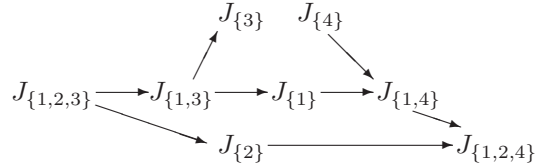


Рис. 17. Сеть \bar{G}_9

Оптимальные сети для задач, в которых отсутствуют другие пары типов работ, определяемых условием 2, строятся аналогично. Теорема 5 доказана.

В заключение покажем, что в статье предложена полная классификация сложности всех подзадач задачи Π_4 по всем возможным значениям параметра T .

Утверждение 1. Для любого значения параметра T результаты, полученные в теоремах 1–5, позволяют определить, что задача $\Pi(T)$ является либо NP -трудной, либо полиномиально разрешимой.

ДОКАЗАТЕЛЬСТВО. Напомним, что значение параметра T зависит от структуры ациклического графа G . Утверждение, очевидно, верно, когда граф G — цепь с четырьмя вершинами. Действительно, из теоремы 1 следует, что задача $\Pi_4^L(T)$ NP-трудна, когда параметр T принимает три следующих значения: \mathcal{T}_4 , $\mathcal{T}_4 \setminus \{\{2\}\}$ и $\mathcal{T}_4 \setminus \{\{3\}\}$. Теорема 2 гласит, что для всех остальных значений параметра T задача полиномиально разрешима.

Рассмотрим случай, когда граф G — звезда с четырьмя вершинами. Предположим, что в задаче $\Pi_4^S(T)$ нет работ по крайней мере одного из типов $\{2\}$, $\{3\}$, $\{4\}$. Тогда согласно утверждению (i) теоремы 5 задача $\Pi_4^S(T)$ полиномиально разрешима.

Если в множестве T нет по крайней мере одного из типов $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$ и по крайней мере одного из типов $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, то согласно утверждению (ii) теоремы 5 задача $\Pi_4^S(T)$ полиномиально разрешима.

Предположим, что множество T содержит подмножества $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, $\{2\}$, $\{3\}$, и $\{4\}$. Тогда по теореме 4 задача $\Pi_4^S(T)$ NP-трудна. Наконец, если множество T содержит подмножества $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, $\{2\}$, $\{3\}$ и $\{4\}$, то по теореме 3 задача $\Pi_4^S(T)$ также NP-трудна. Утверждение 1 доказано.

3. Заключение

В статье рассмотрена задача Π_4 построения оптимального по быстродействию расписания многопроцессорных работ на четырёх специализированных машинах. Предполагается, что машины связаны в сеть ациклическим графом G и каждая работа должна быть выполнена на заданном множестве смежных машин. В статье показано, что сложность задачи зависит от разнообразия типов работ, которые присутствуют в задаче, где типом работы называется множество обслуживающих её машин. Таким образом, множество типов работ является параметром задачи Π_4 . В работе представлены все её максимальные полиномиально разрешимые и минимальные NP-трудные подзадачи относительно выбранного параметра.

Результаты статьи являются первым шагом в описании полиномиально разрешимых классов для задачи $P|\text{fix}_j|C_{\max}$ в зависимости от множества всех типов работ, участвующих в ней. Представляет интерес нахождение полного описания всех полиномиально разрешимых классов задачи $P|\text{fix}_j|C_{\max}$ независимо от числа машин, вида сети или условия смежности подмножества машин. Другая интересная проблема — най-

ти зависимость комбинаторной сложности задачи $P|fix_j|C_{\max}$ от структуры редуцированного графа конфликтов, описанного во введении нашей статьи. Третьей открытой проблемой является либо установление NP-трудности в сильном смысле задач, рассматриваемых в теоремах 1 и 4, либо построение для этих задач псевдополиномиального алгоритма.

Авторы благодарят С. В. Севастьянова и анонимного рецензента за ряд ценных советов, позволивших улучшить текст этой статьи.

ЛИТЕРАТУРА

1. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
2. **Танаев В. С., Сотсков Ю. Н., Струевич В. А.** Теория расписаний. Многостадийные системы. М.: Наука, 1989. 329 с.
3. **Amoura A. K., Vampis E., Kenyon C., Manoussakis Y.** Scheduling independent multiprocessor tasks // *Algorithmica*. 2002. Vol. 32, No. 2. P. 247–261.
4. **Bo Chen, Potts C. N., Woeginger G. J.** A review of machine scheduling: complexity, algorithms and approximability // *Handbook of combinatorial optimization* (D.-Z. Du and P. M. Pardalos, Eds.). Amsterdam: Kluwer Acad. Publ., 1998. Vol. 3. P. 21–169.
5. **Bierwirth C., Meisel F.** A survey of berth allocation and quay crane scheduling problems in container terminals // *Eur. J. Oper. Res.* 2015. Vol. 244, No. 3. P. 675–689.
6. **Blazewicz J., Dell’Olmo P., Drozdowski M., Speranza M. G.** Scheduling multiprocessor tasks on three dedicated processors // *Inf. Process. Lett.* 1992. Vol. 41. P. 257–280.
7. **Buchsbaum A. L., Karloff H., Kenyon C., Reingold N., Thorup M.** OPT versus LOAD in dynamic storage allocation // *SIAM J. Comput.* 2012. Vol. 33, No. 3. P. 632–646.
8. **Chun H. N.** Scheduling as a multidimensional placement problem // *Engineering Appl. Artificial Intelligence*. 1996. Vol. 9. P. 261–273.
9. **Drozdowski M.** Scheduling for parallel processing. London: Springer-Verl., 2009. 386 p. (Comput. Commun. Netw).
10. **Duin C. W., Sluis E. V.** On the complexity of adjacent resource scheduling // *J. Scheduling*. 2006. Vol. 9, No. 1. P. 49–62.
11. **Even G., Hallderson M. M., Kaplan L., Ron D.** Scheduling with conflicts: online and offline algorithms // *J. Scheduling*. 2009. Vol. 12, No. 2. P. 199–224.
12. **Hoogeveen J. A., van de Velde S. L., Veltman B.** Complexity of scheduling multiprocessor task with prespecified processor allocations // *Discrete Appl. Math.* 1994. Vol. 55, No. 3. P. 259–272.

13. **Lim A.** The berth planning problem // Oper. Res. Lett. 1998. Vol. 22, No. 2–3. P. 105–110.
14. **Paulus J. J., Hurink J.** Adjacent-resource scheduling. Why spatial resources are so hard to incorporate // Electron. Notes Discrete Math. 2006. Vol. 25. P. 113–116.

*Кононов Александр Вениаминович,
Кононова Полина Александровна*

Статья поступила
4 февраля 2016 г.

Исправленный вариант —
24 июня 2016 г.

DISKRETNYYI ANALIZ I ISSLEDOVANIE OPERATSII
October–December 2016. Volume 23, No. 4. P. 5–25

UDC 519.8

DOI: 10.17377/dai0.2016.23.525

ON MINIMIZING DATASET TRANSFER TIME
IN AN ACYCLIC NETWORK WITH FOUR SERVERS

A. V. Kononov^{1,2}, P. A. Kononova^{1,2}

¹Sobolev Institute of Mathematics,
4 Acad. Koptuyug Ave., 630090 Novosibirsk, Russia

²Novosibirsk State University,
2 Pirogov St., 630090 Novosibirsk, Russia

e-mail: alvenko@math.nsc.ru, polik83@ngs.ru

Abstract. Under consideration is some optimization problem of data transmission in a hierarchical acyclic network. This problem is a special case of the makespan minimization problem with multiprocessor jobs on dedicated machines. We study computational complexity of the subproblems with a specific set of job types, where the type of a job is a subset of the machines required by the job. Ill. 17, bibliogr. 14.

Keywords: multiprocessor scheduling, polynomial time algorithm, NP-hardness.

REFERENCES

1. **M. R. Garey** and **D. S. Johnson**, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979. Translated under the title *Vychislitel'nye mashiny i trudnoreshaemye zadachi*, Mir, Moscow, 1982.
2. **V. S. Tanaev**, **Yu. N. Sotskov**, and **V. A. Strusevich**, *Teoriya raspisaniy: Mnogostadiynye sistemy*, Nauka, Moscow, 1989. Translated under the title *Scheduling Theory. Multi-Stage Systems*, Kluwer Acad. Publ., Dordrecht, 1994 (Math. Its Appl., Vol. 285).
3. **A. K. Amoura**, **E. Bampis**, **C. Kenyon**, and **Y. Manoussakis**, Scheduling independent multiprocessor tasks, *Algorithmica*, **32**, No. 2, 247–261, 2002.
4. **B. Chen**, **C. N. Potts**, and **G. J. Woeginger**, A review of machine scheduling: complexity, algorithms and approximability, in D.-Z. Du and P. M. Pardalos (eds.), *Handbook of Combinatorial Optimization*, Vol. 3, pp. 21–169, Kluwer Acad. Publ., Dordrecht, 1998.
5. **C. Bierwirth** and **F. Meisel**, A survey of berth allocation and quay crane scheduling problems in container terminals, *Eur. J. Oper. Res.*, **244**, No. 3, 675–689, 2015.

6. **J. Błażewicz, P. Dell’Olmo, M. Drozdowski, and M. G. Speranza**, Scheduling multiprocessor task on three dedicated processors, *Inf. Process. Lett.*, **41**, No. 5, 275–280, 1992.
7. **A. L. Buchsbaum, H. Karloff, C. Kenyon, N. Reingold, and M. Thorup**, OPT versus LOAD in dynamic storage allocation, *SIAM J. Comput.*, **33**, No. 3, 632–646, 2012.
8. **H. W. Chun**, Scheduling as a multi-dimensional placement problem, *Eng. Appl. Artif. Intell.*, **9**, No. 3, 261–273, 1996.
9. **M. Drozdowski**, *Scheduling for Parallel Processing*, Springer, London, 2009 (Comput. Commun. Netw.).
10. **C. W. Duin and E. V. Sluis**, On the complexity of adjacent resource scheduling, *J. Sched.*, **9**, No. 1, 49–62, 2006.
11. **G. Even, M. M. Halldersson, L. Kaplan, and D. Ron**, Scheduling with conflicts: Online and offline algorithms, *J. Sched.*, **12**, No. 2, 199–224, 2009.
12. **J. A. Hoogeveen, S. L. van de Velde, and B. Veltman**, Complexity of scheduling multiprocessor task with prespecified processor allocations, *Discrete Appl. Math.*, **55**, No. 3, 259–272, 1994.
13. **A. Lim**, The berth planning problem, *Oper. Res. Lett.*, **22**, No. 2–3, 105–110, 1998.
14. **J. J. Paulus and J. Hurink**, Adjacent-resource scheduling: Why spatial resources are so hard to incorporate, *Electron. Notes Discrete Math.*, **25**, 113–116, 2006.

Alexander V. Kononov,
Polina A. Kononova

Received
4 February 2016

Revised
24 June 2016