ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь—декабрь 2000. Серия 1. Том 7, № 4, 59–77

УДК 519.854

ЧЕТЫРЕХПАРАМЕТРИЧЕСКИЙ АНАЛИЗ СЛОЖНОСТИ ЗАДАЧИ OPEN SHOP*)

К. Н. Каширских, С. В. Севастьянов, И. Д. Черных

Рассматривается задача open shop на минимум длины расписания. Предлагается сложностной анализ этой задачи в зависимости от значений четырех параметров (числа работ, максимального числа операций работы, числа машин и максимального числа операций на машине), объединяющихся в четырехмерный характеристический вектор x_I заданного входа I. Для различных значений четырехмерного вектора x определяются классы $\mathscr{I}(x)$ индивидуальных задач open shop, характеристический вектор которых не превосходит вектора x. Показывается, что в бесконечном множестве нетривиальных классов $\mathscr{I}(x)$ (допускающих неограниченное общее число операций) существует конечная система так называемых базисных классов, позволяющая определить сложность задачи open shop на классе входов $\mathscr{I}(x)$ для любого допустимого значения вектора x.

Введение

В статье проводится анализ сложности известной задачи теории расписаний — open shop. Как известно, анализ сложности является одной из основных проблем, волнующих исследователя той или иной задачи дискретной оптимизации. Исходно под анализом сложности какой-либо оптимизационной задачи понималось такое ее исследование, которое позволяло определить одну из двух альтернатив: существует ли полиномиальный алгоритм точного решения задачи или же задача относится к классу NP-трудных. Позднее, когда почти для всех известных задач такой первоначальный анализ был проделан и выяснилось, что бо́льшая часть задач является NP-трудной, стали исследовать сложность их приближенного решения. Возможные результаты в этом направлении: построение эффективных C-приближающих алгоритмов (либо доказательство невозможности их построения — в предположении $\mathrm{P} \neq \mathrm{NP}$)

*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 99–01–00601, 99–01–00581) и Федеральной целевой программы «Интеграция» (проект 274).

для какой-либо константы C, а также построение *полиномиальных аппроксимационных схем*, т. е. семейств полиномиальных по сложности $(1+\varepsilon)$-приближающих алгоритмов для всевозможных $\varepsilon > 0$. К этому же направлению относятся результаты по построению аппроксимационных схем, состоящих из $(C+\varepsilon)$-приближающих алгоритмов при $C \neq 1$, и результаты по построению f-приближающих алгоритмов, где f — уже не константа, а некоторая функция от каких-то входных параметров задачи. Однако в данной статье мы не будем касаться анализа сложности приближенного решения задачи open shop (отсылая интересующегося читателя к обзору [2]), а вернемся к исходному этапу анализа этой задачи — к анализу сложности ее точного решения.

В пионерной работе Т. Гонзалеза и С. Сани [3] при анализе сложности задачи open shop было проведено четкое разграничение между полиномиально разрешимыми случаями, когда число машин m не превосходит 2, и NP-трудными случаями ($m \geqslant 3$). Правда, это разграничение пролегало лишь по значениям одного из возможных параметров, а именно по числу машин. Такова же зависимость от другого основного параметра — числа работ (в силу симметричности понятий «машина» и «работа» в модели open shop). Позднее появились результаты по анализу сложности цеховых (многостадийных) моделей, где существенное значение приобретали другие параметры, такие как максимальное число операций работы. Так, например, в [4] строится аппроксимационная схема для задачи job shop при условии, что число машин и максимальное число операций одной работы ограничены произвольными константами, а в [6] показано, что если в задаче job shop и число машин, и число работ не превосходит 3, но число операций каждой работы является переменной, то задача NP-трудна. Эти результаты навели нас на мысль о том, что, во-первых, для задачи open shop такой параметр, как максимальное число операций работы, может оказаться существенным в анализе ее сложности; во-вторых, что назрела необходимость осуществления более полного анализа одной из наиболее известных задач теории расписаний многостадийных систем, какой является задача open shop. Такой более полный анализ должен основываться на анализе значений не одного, а совокупности нескольких ключевых параметров задачи.

В настоящей статье делается попытка описания общего подхода к многопараметрическому анализу сложности дискретных задач. Этот подход иллюстрируется на примере 4-параметрического анализа задачи open shop, в которой в качестве ключевых выбраны следующие четыре параметра: число работ, максимальное число операций работы, число машин и максимальное число операций на машине. Ясно, что последние два параметра являются симметричным отражением двух первых,

если поменять местами понятия «работа» и «машина».[1] В то же время нельзя ограничиться только двумя первыми параметрами или двумя последними, поскольку в некоторых случаях оказывается существенной комбинация значений всех четырех параметров.

Рассматриваются классы $\mathscr{I}(x)$ индивидуальных задач, каждый из которых определяется некоторым четырехмерным вектором x: i-я компонента вектора x задает ограничение на допустимые значения i-го ключевого параметра задачи. Показывается, что в бесконечном множестве нетривиальных классов[2] $\mathscr{I}(x)$ существует конечная базисная система классов, позволяющая определить сложность любого класса $\mathscr{I}(x)$. Более точно: устанавливается, что эта базисная система состоит из 10 либо 14 классов. Окончательный ответ на вопрос о мощности базисной системы зависит от ответа на открытый вопрос (поставленный Т. Гонзалезом и С. Сани в 1976 г.) о сложности трехмашинной задачи open shop, в которой каждая работа имеет не более двух операций.

1. Дискретные оптимизационные задачи и их многопараметрический анализ сложности

При анализе сложности задачи open shop будем действовать по схеме, аналогичной той, которая использовалась А. Кононовым и С. Севастьяновым в работе [1] при 4-параметрическом анализе сложности задачи о связной предписанной вершинной раскраске графа. Начнем с уточнения используемой терминологии.

Прежде всего, разграничим понятия «массовой задачи», «индивидуальной задачи» и «входа» индивидуальной задачи. *Массовой оптимизационной задачей* A будем называть семейство индивидуальных задач $A(I)$, определенных для всевозможных *входов* $I \in I_A$. (Множество I_A будем называть *множеством входов* задачи A.)

Индивидуальной задачей $A(I)$, соответствующей заданному входу I, будем называть тройку $(\mathrm{sol}_A(I), \theta_A, \mathrm{type}_A)$, где

- $\mathrm{sol}_A(I)$ — *множество допустимых решений задачи* $A(I)$;
- $\theta_A : I_A \times S_A \to \mathbb{R}^+$ — неотрицательная *целевая функция*, определенная на множестве входов $I \in I_A$ и их решений $S \in S_A$, где $S_A = \bigcup\limits_{I \in I_A} \mathrm{sol}_A(I)$;

---

[1] С точки зрения календарного планирования понятия «работа» и «машина» замещают обычные ресурсные ограничения возобновимого типа, а потому при отсутствии ограничений предшествования (как в случае задачи open shop) эти понятия совершенно равноправны.

[2] Класс индивидуальных задач open shop считаем *нетривиальным*, если число операций в задачах из этого класса может быть сколь угодно большим.

- $\text{type}_A \in \{\max, \min\}$ — тип задачи (на максимум или минимум целевой функции).

Решением задачи A называется функция $h: I_A \to S_A$, устанавливающая такое соответствие между входами задачи A и ее решениями, что для любого входа $I \in I_A$ выполнено $h(I) \in \text{sol}_A(I)$. Произвольное решение задачи A будем называть также *допустимым,* или *приближенным.* Множество (допустимых) решений задачи A будем обозначать через H_A. Решение $h^* \in H_A$ называется *точным* (или *оптимальным*), если для каждого $I \in I_A$

$$\theta_A(I, h^*(I)) = \text{type}_A\ \theta_A(I, h(I)),$$

где min или max (в зависимости от значения type_A) берется по всем $h \in H_A$. Значение $h^*(I)$ для заданного входа I будем называть *оптимальным решением* индивидуальной задачи, однозначно определяемой по входу I.

Предполагаем, что каким-то образом определен способ записи любого входа $I \in I_A$ и любого решения $S \in S_A$. Таким образом, более точно под массовой задачей будем понимать семейство определенных выше индивидуальных задач, для которых определены функции $\rho_A^+ : I_A \to \mathbb{R}^+$ и $\rho_A^- : S_A \to \mathbb{R}^+$; при этом значение $\rho_A^+(I)$ будем называть *длиной записи входа* $I \in I_A$, а значение $\rho_A^-(S)$ — *длиной записи решения* $S \in S_A$.

Будем рассматривать также произвольные семейства индивидуальных задач исходной массовой задачи A. Всякое такое семейство B будем называть *классом* индивидуальных задач, или *подзадачей* задачи A. Целью нашего исследования является классификация подзадач по их временно́й сложности. Будем говорить, что подмножество входов $I_B \subseteq I_A$ определяет *полиномиально разрешимый* класс B индивидуальных задач $\{A(I) \mid I \in I_B\}$, если существуют алгоритм $\mathscr{A}$ и полином $P(x, y)$ такие, что для любого входа $I \in I_B$ алгоритм $\mathscr{A}$ находит точное решение $h^*(I)$ задачи $A(I)$ за время, ограниченное сверху полиномом $P(\rho_A^+(I), \rho_A^-(h^*(I)))$. Если же из предположения существования такого алгоритма вытекает $\text{P} = \text{NP}$, то класс B является NP-*трудным.*

Из приведенного выше определения полиномиально разрешимых и NP-трудных классов непосредственно следует, что любой подкласс полиномиально разрешимого класса является полиномиально разрешимым, и наоборот: любой надкласс NP-трудного класса является NP-трудным. Таким образом, наше определение устраняет известное противоречие, когда возможно существование NP-трудных подзадач полиномиально разрешимой задачи [5].

Очевидно, что не имеет смысла говорить о сложности решения индивидуальной задачи, как и о сложности класса, состоящего из константного числа индивидуальных задач, — в обоих случаях временна́я

сложность простейшего переборного алгоритма ограничена константой (т. е. полиномом нулевой степени от длины входа). Поэтому в дальнейшем *классами* будем называть бесконечные по мощности семейства индивидуальных задач. Поскольку класс индивидуальных задач однозначно определяется по множеству его входов, то для удобства изложения (там, где это не вызывает разночтений) то или иное подмножество входов будем отождествлять с определяемым им классом индивидуальных задач.

Сформулируем понятие n-параметрического анализа сложности массовой задачи A для заданного натурального n. Пусть каким-то образом выбраны n числовых параметров (будем называть их *ключевыми*), задающих некоторую частичную информацию о входных данных задачи A. Поскольку мы рассматриваем лишь дискретные задачи, то без ограничения общности все параметры задачи можем считать целочисленными. Для каждого входа $I \in I_A$ определяем n-мерный характеристический вектор x_I, компонентами которого являются значения выбранных ключевых параметров. Заданному набору значений этих параметров может отвечать бесконечное множество индивидуальных задач $A(I)$. Таким образом, вектор $x \in \mathbb{R}^n$ может определять класс индивидуальных задач, соответствующих множеству входов $\{I \in I_A \mid x_I = x\}$.

Мы, однако, будем исследовать сложность не этих, а более широких классов, определяемых для заданного вектора x по множеству входов $\mathscr{I}(x) \doteq \{I \in I_A \mid x_I \leqslant x\}$, где неравенство выполняется покомпонентно. При этом вектор x будем называть *определяющим* для класса $\mathscr{I}(x)$. Целью n-параметрического анализа является выяснение сложности классов $\mathscr{I}(x)$ для **всех допустимых** значений вектора x. Значение определяющего вектора $x = (x(1), \ldots, x(n))$ считается допустимым, если каждая его компонента $x(i)$ $(i = 1, \ldots, n)$ принимает некоторое допустимое значение; в свою очередь, значение компоненты $x(i)$ определяющего вектора считаем допустимым в двух случаях: 1) если существует хотя бы одна индивидуальная задача I, для которой $x_I(i) = x(i)$; 2) если $x(i) = \infty$. (В последнем случае неравенство $x_I(i) \leqslant x(i)$ означает, что класс индивидуальных задач $\mathscr{I}(x)$ никак не ограничен по i-му параметру.)

Через X_A будем обозначать множество определяющих векторов в задаче A. Для каждого вектора $x' \in X_A$ введем в рассмотрение области $D^-(x') = \{x \in X_A \mid x \leqslant x'\}$ и $D^+(x') = \{x \in X_A \mid x \geqslant x'\}$. Из определения классов $\mathscr{I}(x)$ непосредственно следует, что если класс $\mathscr{I}(x')$ полиномиально разрешим, то $\mathscr{I}(x)$ полиномиально разрешим для любого вектора $x \in D^-(x')$, а если $\mathscr{I}(x')$ является NP-трудным, то $\mathscr{I}(x)$ — NP-трудный класс при любом $x \in D^+(x')$.

Пусть задано подмножество векторов $X_B \subseteq X_A$. Будем говорить, что подмножество векторов $\widetilde{X}_B \subseteq X_B$ задает *базисную систему классов* $\{\mathscr{I}(x) \mid x \in \widetilde{X}_B\}$ в семействе классов $\{\mathscr{I}(x) \mid x \in X_B\}$, если выполняются следующие четыре свойства:

1) свойство *определенности*: задано разбиение $\widetilde{X}_B = (Y_P; Y_{NP})$ множества $\widetilde{X}_B$ на два подмножества такое, что все классы $\mathscr{I}(y)$ $(y \in Y_P)$ полиномиально разрешимы, а все классы $\mathscr{I}(y)$ $(y \in Y_{NP})$ NP-трудны;

2) свойство *полноты*: $X_B \subseteq \Big(\bigcup\limits_{y \in Y_P} D^-(y)\Big) \bigcup \Big(\bigcup\limits_{y \in Y_{NP}} D^+(y)\Big)$;

3) свойство *независимости*: любые два вектора $y_1, y_2 \in Y_P$ несравнимы; любые два вектора $y_1, y_2 \in Y_{NP}$ несравнимы;

4) свойство *непротиворечивости*: ни для каких двух векторов $y_1 \in Y_P, y_2 \in Y_{NP}$ не выполняется неравенство $y_2 \leqslant y_1$.

Из свойства 2 следует, что для любого вектора $x \in X_B$ найдется либо вектор $y' \in Y_P$ такой, что $x \in D^-(y')$ (откуда вытекает полиномиальная разрешимость класса $\mathscr{I}(x)$), либо вектор $y'' \in Y_{NP}$ такой, что $x \in D^+(y'')$ (откуда вытекает NP-трудность класса $\mathscr{I}(x)$). Из свойства 4 следует, что не может одновременно существовать двух векторов $y' \in Y_P$ и $y'' \in Y_{NP}$ таких, что $(x \in D^-(y'))\&(x \in D^+(y''))$. Таким образом, для определения сложности класса $\mathscr{I}(x)$ для произвольного вектора $x \in X_B$ достаточно сравнить вектор x со всеми базисными векторами $y \in \widetilde{X}_B$.

Лемма 1. *Если существует базисная система для семейства классов $\{\mathscr{I}(x) \mid x \in X_B\}$, то она единственна (в предположении* P $\neq$ NP*).*

Доказательство. Предположим, что $\widetilde{X}'_B = (Y'_P; Y'_{NP})$ и $\widetilde{X}''_B = (Y''_P; Y''_{NP})$ — две базисные системы семейства $\{\mathscr{I}(x) \mid x \in X_B\}$, и пусть $y_1 \in Y'_P$. Докажем, что $y_1 \in Y''_P$. Из полноты базисной системы $\widetilde{X}''_B$ следует, что либо найдется такой вектор $y_2 \in Y''_P$, что $y_1 \in D^-(y_2)$, либо такой вектор $y_3 \in Y''_{NP}$, что $y_1 \in D^+(y_3)$. Так как класс $\mathscr{I}(y_1)$ полиномиально разрешим, то вторая альтернатива невозможна (в предположении P $\neq$ NP). Таким образом, $y_1 \in D^-(y_2)$. Симметрично из полноты базисной системы $\widetilde{X}'_B$ следует, что найдется такой вектор $y_4 \in Y'_P$, что $y_2 \in D^-(y_4)$. Следовательно, $y_1 \leqslant y_2 \leqslant y_4$ и в силу свойства 3 имеем $y_1 = y_4$. Поэтому $y_1 = y_2 \in Y''_P$. Так как y_1 — произвольный вектор из Y'_P, то $Y'_P \subseteq Y''_P$. Из симметричного включения $Y''_P \subseteq Y'_P$ следует, что $Y'_P = Y''_P$. Аналогично доказывается, что $Y'_{NP} = Y''_{NP}$. Таким образом, базисные системы $\widetilde{X}'_B$ и $\widetilde{X}''_B$ совпадают. Лемма 1 доказана.

Лемма 2. *Если допустимые значения ключевых параметров ограничены снизу и существует базисная система для семейства классов $\{\mathscr{I}(x) \mid x \in X_B\}$, то она конечна.*

Доказательство. Пусть существует базисная система $\widetilde{X}_B = (Y_P; Y_{NP})$ для семейства классов $\{\mathscr{I}(x) \mid x \in X_B\}$; $\bar{c}$ — нижняя граница значений ключевых параметров. Фиксируем набор компонент $\Phi \subset \{1, \dots, n\}$ и обозначим $\widetilde{X}_B^\Phi = \{x \in \widetilde{X}_B \mid x(i) = \infty,\ \forall\ i \in \Phi\}$; $X_P^\Phi = \widetilde{X}_B^\Phi \cap Y_P$. Предположим, что множество X_P^Φ имеет бесконечную мощность. Выберем произвольный вектор $x' \in X_P^\Phi$. Поскольку любые два вектора из Y_P несравнимы, каждый из оставшихся векторов $x \in X_P^\Phi$ имеет хотя бы одну такую компоненту $x(i)$, что $x(i) < x'(i)$. Так как X_P^Φ имеет бесконечную мощность, то найдется такая координата i, что множество $X_i = \{x \in X_P^\Phi \mid x(i) < x'(i)\}$ бесконечно. Поскольку при этом набор допустимых значений $\{x(i) \mid x \in X_i\} \subseteq \{\bar{c}, \bar{c}+1, \dots, x'(i)-1\} \doteq C_i$ конечен, то существует константа $c \in C_i$ такая, что множество векторов $X_{i,c} \doteq \{x \in X_P^\Phi \mid x(i) = c\}$ имеет бесконечную мощность. Таким образом, в подмножестве базисных векторов X_P^Φ мы выделили бесконечное подмножество $X_{i,c} \subseteq X_P^\Phi$ векторов с фиксированным значением одной из компонент.

Индукцией по числу координат нетрудно прийти к заключению, что существует бесконечное число базисных векторов, координаты которых полностью совпадают, а это противоречит свойству независимости. Таким образом, предположение о бесконечности множества X_P^Φ привело к противоречию. Аналогично доказывается конечность множества $X_{NP}^\Phi \doteq \widetilde{X}_B^\Phi \cap Y_{NP}$. (При этом используется несравнимость любых двух векторов из Y_{NP}.) Поскольку $\widetilde{X}_B = \bigcup\limits_{\{\Phi\}} \widetilde{X}_B^\Phi$, а число различных наборов Φ конечно, то мощность базиса $\widetilde{X}_B$ конечна. Лемма 2 доказана.

Нетрудно понять, что хотя мощность базиса конечна, уже при $n \geqslant 2$ ее нельзя априори ограничить какой-либо функцией, зависящей лишь от n. (То есть при фиксированном n могут существовать массовые задачи со сколь угодно большой мощностью базиса.) Также ясно, что при неограниченных снизу допустимых значениях ключевых параметров может не существовать конечной базисной системы классов.

2. Обобщенная задача open shop: 4-параметрический анализ сложности

Рассматривается задача open shop в следующей постановке. Заданы множества $\mathscr{J} = \{J_1, \dots, J_n\}$, $\mathscr{M} = \{M_1, \dots, M_m\}$ и $\mathscr{O} = \{o_1, \dots, o_K\}$, элементы которых называются соответственно *работами, машинами* и *операциями.* Каждая операция $o_i \in \mathscr{O}$ принадлежит определенной работе $J(o_i) \in \mathscr{J}$. Кроме того, для выполнения операции $o_i \in \mathscr{O}$ требуется выделить некоторый связный интервал времени длины $p(o_i)$ на определенной машине $M(o_i) \in \mathscr{M}$. Таким образом, M — функция из $\mathscr{O}$ в $\mathscr{M}$,

J — функция из $\mathscr{O}$ в $\mathscr{J}$, а p — функция из $\mathscr{O}$ в $\mathbf{Z}^+$. *Допустимым расписанием* S называется такая совокупность $\{s(o_i) \mid o_i \in \mathscr{O}\}$ неотрицательных моментов начала выполнения каждой операции, что для любых двух операций $o_i, o_j \in \mathscr{O}$, требующих одинакового ресурса ($M(o_i) = M(o_j)$ либо $J(o_i) = J(o_j)$), интервалы $(s(o_i), s(o_i) + p(o_i))$ и $(s(o_j), s(o_j) + p(o_j))$ выполнения этих операций не пересекаются. Требуется назначить момент $s(o_i)$ начала выполнения каждой операции $o_i \in \mathscr{O}$ так, чтобы расписание $S = \{s(o_i) \mid o_i \in \mathscr{O}\}$ было допустимым, а момент завершения последней операции $C_{\max}(S) \doteq \max\limits_{o_i \in \mathscr{O}}(s(o_i) + p(o_i))$ был минимальным.

Задача open shop в приведенной постановке (будем использовать для нее аббревиатуру OS) является обобщением классической задачи open shop, сформулированной Т. Гонзалезом и С. Сани в [3]. В классической постановке дополнительно предполагается, что в множестве $\mathscr{O}$ не существует двух операций o_i и o_j, имеющих полностью идентичные векторы потребления ресурсов ($M(o_i) = M(o_j)$ и $J(o_i) = J(o_j)$). Другими словами, любая работа $J_j \in \mathscr{J}$ имеет на каждой машине M_i не более одной операции. В рассматриваемой нами задаче OS множество $\mathscr{O}$ не обязано удовлетворять этому требованию.

Как видно из дальнейшего, существенное влияние на сложность задачи OS оказывают параметры $\nu = \max\limits_{J_j \in \mathscr{J}} |J^{-1}(J_j)|$ и $\mu = \max\limits_{M_i \in \mathscr{M}} |M^{-1}(M_i)|$, которые будем называть *максимальным числом операций одной работы* и *максимальным числом операций одной машины*. В классической постановке эти параметры привязаны к параметрам n и m (конкретно, $\nu = m$ и $\mu = n$). В задаче OS параметры ν и μ являются независимыми, что позволяет проводить нетривиальный анализ сложности задачи, когда n и m одновременно фиксированы, а ν и μ произвольны, либо наоборот: отыскивать полиномиально разрешимые случаи, когда обе переменные n и m не ограничены константами. Таким образом, каждый вход I задачи OS характеризуется четырехмерным характеристическим вектором $x_I = (n, \nu, m, \mu)$.

Обозначим $\mathbb{N} = \{1, 2, \ldots\}$, $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$. Ясно, что характеристический вектор x_I любого входа I задачи OS лежит в множестве $\mathbb{N}^4$ и что множество X_{OS} допустимых значений определяющего вектора x в задаче OS содержится в множестве $\overline{\mathbb{N}}^4$.

Заметим, что если первые две (либо последние две) компоненты вектора x ограничены константами c_1 и c_2, то общее число операций любой индивидуальной задачи из класса $\mathscr{I}(x)$ не превосходит константы $c = c_1 c_2$. Хотя класс $\mathscr{I}(x)$ при этом имеет бесконечную мощность (индивидуальные задачи различаются длительностями операций), нетрудно показать, что каждый такой класс, определяемый константами c_1 и c_2,

полиномиально разрешим. Действительно, задавая различными способами порядок выполнения операций каждой работы и порядок выполнения операций на каждой машине (число различных комбинаций таких порядков не превосходит константы $(c!)^{2c}$), для каждого полученного графа G отношений предшествования на множестве операций за константное время проверяем бесконтурность графа G. В случае положительного результата оптимальным (для заданного графа G) расписанием является наиболее раннее расписание. Для его нахождения достаточно вычислить наиболее ранние моменты начала всех операций, для чего, в свою очередь, достаточно выполнить не более $O(u)$ операций сложения и не более $O(u)$ операций сравнения над числами, длина записи каждого из которых не превышает длины записи входа, где u — число дуг в графе G. В случае задачи OS число дуг в графе G не превосходит удвоенного числа операций (в нашем случае — константы $2c$), поэтому можем заключить, что класс $\mathscr{I}(x)$ разрешим за линейное время.

В дальнейшем такие классы, в которых число операций любой индивидуальной задачи не превосходит константы c, будем называть *тривиальными*. Ввиду сказанного выше мы можем исключить тривиальные классы из дальнейшего рассмотрения, поскольку вопрос об их сложности тривиален. Оставшиеся классы (содержащие индивидуальные задачи со сколь угодно большим числом операций) будем называть *нетривиальными*. Множество векторов x, определяющих нетривиальные классы $\mathscr{I}(x)$, будем обозначать через $\overline{X}_{\mathrm{OS}}$. Учитывая сделанное выше замечание, имеем

$$\overline{X}_{\mathrm{OS}} = \{x \in X_{\mathrm{OS}} \mid (x(1) = \infty \vee x(2) = \infty) \& (x(3) = \infty \vee x(4) = \infty)\}.$$

Лемма 3. *В предположении* $\mathrm{P} \neq \mathrm{NP}$ *для семейства классов* $\{\mathscr{I}(x) \mid x \in X_{\mathrm{OS}}\}$ *не существует базисной системы.*

Доказательство. Предположим противное, т. е. что существует базисная система $\widetilde{X}_{\mathrm{OS}} = (Y_P; Y_{NP})$ семейства классов $\{\mathscr{I}(x) \mid x \in X_{\mathrm{OS}}\}$. Тогда по лемме 1 она единственна, а по лемме 2 — конечна (поскольку все допустимые значения ключевых параметров положительны, а следовательно, ограничены снизу). Так как вектор x, все компоненты которого равны ∞, не может определять полиномиально разрешимый класс (ввиду NP-трудности общей задачи OS отсюда следовало бы $\mathrm{P} = \mathrm{NP}$), то каждый вектор $x \in Y_P$ имеет хотя бы одну конечную компоненту.

Так как множество Y_P конечно, то существует константа c, строго бóльшая каждой конечной компоненты любого вектора $x \in Y_P$. Но тогда вектор $x' = (c, c, c, c)$ не лежит ни в одной из областей $D^-(x)$ $(x \in Y_P)$ и по свойству полноты базисной системы $\widetilde{X}_{\mathrm{OS}}$ должен лежать в одной из областей $D^+(x)$ $(x \in Y_{NP})$. Таким образом, класс $\mathscr{I}(x')$ является

NP-трудным и одновременно (как всякий тривиальный класс) полиномиально разрешимым, что невозможно в предположении $\mathrm{P} \neq \mathrm{NP}$. Лемма 3 доказана.

Оставшаяся часть статьи посвящена доказательству следующего результата.

Теорема 4. *Для семейства* $\{\mathscr{I}(x) \mid x \in \overline{X}_{\mathrm{OS}}\}$ *нетривиальных классов задачи* OS *существует базисная система классов, включающая либо* 10, *либо* 14 *базисных классов.*

Сложность базисных классов задачи OS

Класс $\mathscr{I}(x_i)$, $i = \ldots$	n	ν	m	μ	Сложность
1	∞	1	∞	∞	P
2	∞	∞	∞	1	P
3	2	∞	∞	∞	P
4	∞	∞	2	∞	P
5	∞	2	∞	2	P
6	∞	2	3	∞	??
7	3	∞	∞	2	??
8	∞	2	∞	3	NPH
9	∞	3	∞	2	NPH
10	3	∞	3	∞	NPH
11	3	∞	∞	3	NPH
12	∞	3	3	∞	NPH
13	4	∞	∞	2	NPH
14	∞	2	4	∞	NPH

Результаты по сложности классов $\mathscr{I}(x_i)$ для 14 различных значений вектора x_i приведены в таблице. При этом для первых пяти классов установлена их полиномиальная разрешимость, для последних семи — NP-трудность, а вопрос о сложности двух симметричных друг другу классов $\mathscr{I}(x_6)$ и $\mathscr{I}(x_7)$ остался открытым. (Классы $\mathscr{I}(x_6)$ и $\mathscr{I}(x_7)$ либо одновременно NP-трудны, либо полиномиально разрешимы.) Будет показано, что если два указанных класса полиномиально разрешимы (это предположение будем обозначать через $P_{6,7}$), то 14 векторов x_i, представленных в таблице, определяют базисную систему для семейства нетривиальных классов $\mathscr{I}(x)$ задачи OS. Если же классы $\mathscr{I}(x_6)$ и $\mathscr{I}(x_7)$ NP-трудны (предположение $NP_{6,7}$), то базисная система определяется первыми десятью векторами из таблицы.

Приведем доказательство теоремы 4 в предположении, что сложность каждого класса (см. таблицу) уже установлена. Доказательство результатов о сложности отдельных базисных классов приводится в следующем разделе.

Доказательство теоремы 4. Докажем, что приведенная в таблице система из 14 классов обладает свойством полноты. Для этого достаточно доказать, что

$$\overline{X}_{\mathrm{OS}} \subseteq \left(\bigcup_{i=1}^{5} D^-(x_i)\right) \bigcup \left(\bigcup_{i=8}^{14} D^+(x_i)\right) \bigcup \{x_6, x_7\}. \tag{1}$$

Нетрудно видеть, что

$$\overline{X}_{\mathrm{OS}} \setminus \left(\bigcup_{i=1}^{4} D^-(x_i)\right) \subseteq \overline{X}_{\mathrm{OS}} \cap \{x \in X_{\mathrm{OS}} \mid x(1) \geqslant 3,\ x(2) \geqslant 2,$$
$$x(3) \geqslant 3,\ x(4) \geqslant 2\} = X_1 \cup X_2 \cup X_3 \cup X_4,$$

где

$$X_1 = \{x \in \overline{X}_{\mathrm{OS}} \mid x(1) = \infty,\ x(2) \geqslant 2,\ x(3) = \infty,\ x(4) \geqslant 2\},$$
$$X_2 = \{x \in \overline{X}_{\mathrm{OS}} \mid x(1) \geqslant 3,\ x(2) = \infty,\ x(3) \geqslant 3,\ x(4) = \infty\},$$
$$X_3 = \{x \in \overline{X}_{\mathrm{OS}} \mid x(1) = \infty,\ x(2) \geqslant 2,\ x(3) \geqslant 3,\ x(4) = \infty\},$$
$$X_4 = \{x \in \overline{X}_{\mathrm{OS}} \mid x(1) \geqslant 3,\ x(2) = \infty,\ x(3) = \infty,\ x(4) \geqslant 2\}.$$

Множество X_1 покрывается областью $D^+(x_8) \cup D^+(x_9) \cup D^-(x_5)$.

Множество X_2 есть в точности $D^+(x_{10})$.

Множество X_3 за вычетом областей $D^+(x_{12})$ и $D^+(x_{14})$ дает единственный вектор x_6, а $X_4 \setminus (D^+(x_{11}) \cup D^+(x_{13})) = \{x_7\}$, что завершает доказательство (1). Из (1) видно, что при любом из двух возможных предположений о сложности классов $\mathscr{I}(x_6)$ и $\mathscr{I}(x_7)$ система из 14 классов $\mathscr{I}(x_i)$ $(i = 1, \ldots, 14)$ удовлетворяет свойству полноты.

Для доказательства непротиворечивости системы достаточно убедиться, что неравенство $x_j \leqslant x_i$ не выполняется ни для какой пары векторов x_i $(i \in \{1, \ldots, 7\})$, x_j $(j \in \{8, \ldots, 14\})$ (когда по предположению $P_{6,7}$ векторы x_6 и x_7 определяют полиномиально разрешимые классы) и ни для какой пары векторов x_i $(i \in \{1, \ldots, 5\})$, x_j $(j \in \{6, 7\})$ (когда по предположению $NP_{6,7}$ векторы x_6 и x_7 определяют NP-трудные классы), это проверяется перебором.

Нетрудно убедиться, что 5 полиномиально разрешимых классов $\mathscr{I}(x_i)$ $(i \in \{1, \ldots, 5\})$ независимы друг от друга (для этого достаточно проверить, что векторы x_i, $i \in \{1, \ldots, 5\}$, попарно несравнимы), так же как независимы друг от друга NP-трудные классы $\mathscr{I}(x_j)$ $(j \in \{8, \ldots, 14\})$.

Что касается двух оставшихся классов $\mathscr{I}(x_6)$ и $\mathscr{I}(x_7)$ из таблицы, то в предположении $P_{6,7}$ система из 14 классов остается независимой, поскольку векторы x_6 и x_7 несравнимы ни друг с другом, ни с одним из векторов x_i, $i \in \{1, \dots, 5\}$. В предположении $NP_{6,7}$ классы $\mathscr{I}(x_j)$ ($j \in \{11, \dots, 14\}$) становятся зависимыми от классов $\mathscr{I}(x_6)$ и $\mathscr{I}(x_7)$. Эта зависимость устраняется после удаления классов $\mathscr{I}(x_j)$ ($j \in \{11, \dots, 14\}$) из базисной системы. Свойство полноты системы при этом сохраняется. Таким образом, в предположении $NP_{6,7}$ семейство классов $\mathscr{I}(x_j)$ ($j \in \{1, \dots, 10\}$) образует базисную систему. Теорема 4 доказана.

3. Установление сложности базисных классов

Начнем с более простого доказательства полиномиальной разрешимости классов $\mathscr{I}(x_i)$ ($i \in \{1, \dots, 5\}$).

3.1. Полиномиально разрешимые базисные классы

Величину $L_i = \sum_{o_k \in M^{-1}(M_i)} p(o_k)$ будем называть *нагрузкой машины* M_i, а $l_j = \sum_{o_k \in J^{-1}(J_j)} p(o_k)$ — *длиной* работы J_j. Обозначим $L_{\max} = \max_{M_i \in \mathscr{M}} L_i$ и $l_{\max} = \max_{J_j \in \mathscr{J}} l_j$. Очевидно, что величина $H = \max\{L_{\max}, l_{\max}\}$ является нижней оценкой длины любого допустимого расписания.

Если каждая работа имеет единственную операцию (класс $\mathscr{I}(x_1)$), то выполнение операций на каждой машине в любом порядке без простоев дает оптимальное расписание, и алгоритм построения такого расписания, очевидно, полиномиален. Симметричный класс $\mathscr{I}(x_2)$ также полиномиально разрешим.

Если имеется только две машины (класс $\mathscr{I}(x_4)$), то оптимальное расписание строится за два этапа: 1) от исходной задачи OS переходим к «агрегированной» задаче A путем склеивания всех операций каждой работы, выполняемых на одной машине; 2) применяем алгоритм Гонзалеза и Сани [3] к полученной в результате агрегирования классической двух-машинной задаче open shop.

Оптимальность полученного расписания вытекает из следующего:

1) расписание является допустимым для исходной задачи OS;

2) длина расписания равна величине $H_A = \max\{L_{\max}, l_{\max}\}$, определенной для агрегированной задачи и совпадающей с аналогичной величиной H для исходной задачи, поскольку склейка операций не влияет на величины $L_{\max}$ и $l_{\max}$;

3) величина H_A является нижней оценкой оптимума исходной задачи.

Класс $\mathscr{I}(x_3)$, симметричный классу $\mathscr{I}(x_4)$, также полиномиально разрешим.

Для доказательства полиномиальной разрешимости класса $\mathscr{I}(x_5)$ построим граф G, вершинами которого являются операции из множества $\mathscr{O}$. Две операции соединяем ребром, если они потребляют одинаковый ресурс (машину либо работу) и, следовательно, не могут выполняться одновременно. Соответственно будем различать *машина-причинные* и *работа-причинные* ребра. Поскольку при любом входе $I \in \mathscr{I}(x_5)$ каждая операция $o \in \mathscr{O}$ может быть инцидентна не более чем одному машина-причинному ребру и не более чем одному работа-причинному ребру, степень каждой вершины o в графе G не превосходит двух. Таким образом, G есть семейство цепей и циклов четной длины. (Четность любого цикла следует из того, что машина-причинные и работа-причинные ребра должны в нем чередоваться.) Вершины-операции каждой цепи или цикла занумеруем по порядку числами $1, 2, \dots$, после чего ориентируем каждое ребро, поставив на каждой машине и для каждой работы операцию с нечетным номером первой, а операцию с четным номером — второй. Полученный орграф обозначим через $\overline{G}$. Для заданной последовательности выполнения операций построим наиболее раннее допустимое расписание. Длина этого расписания равна длине критического пути в графе $\overline{G}$. Так как в каждом пути графа $\overline{G}$ содержится не более одной дуги (и соответственно не более двух вершин-операций, относящихся либо к одной и той же работе, либо к одной и той же машине), то длина критического пути совпадает с нижней оценкой оптимума H. Следовательно, построенное расписание оптимально. Нетрудно видеть, что временная сложность алгоритма линейна от числа операций.

3.2. NP-трудность классов $\mathscr{I}(x_j)$ ($j \in \{10, \dots, 14\}$)

Т. Гонзалез и С. Сани в [3] доказали NP-трудность задачи open shop для классов входов $\mathscr{I}'$ и $\mathscr{I}''$, являющихся соответственно подклассами для $\mathscr{I}(x_{12})$ и $\mathscr{I}(x_{14})$. (В отличие от $\mathscr{I}(x_{12})$ и $\mathscr{I}(x_{14})$ классы $\mathscr{I}'$ и $\mathscr{I}''$ включают только такие входы, в которых все операции каждой работы выполняются на разных машинах.) Таким образом, классы $\mathscr{I}(x_{12})$ и $\mathscr{I}(x_{14})$ и симметричные им классы $\mathscr{I}(x_{11})$ и $\mathscr{I}(x_{13})$ являются NP-трудными.

Для доказательства NP-трудности задачи open shop на классах входов $\mathscr{I}'$ и $\mathscr{I}''$ в [3] доказывалась NP-полнота задачи распознавания существования расписания длины H для любого входа $I \in \mathscr{I}'$ ($I \in \mathscr{I}''$). К последней сводилась известная NP-полная задача

Разбиение.

Вход: Семейство натуральных чисел $\{b_1, \dots, b_n\}$.

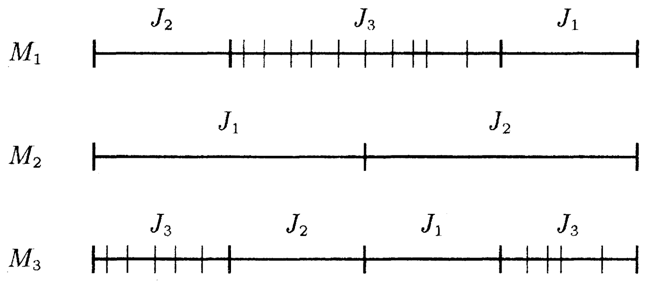

Рис. 1. Расписание длины H для входа $I \in \mathscr{I}(x_{10})$

Вопрос: Существует ли подмножество индексов $\Lambda' \subset \Lambda \doteq \{1, \dots, n\}$ такое, что $\sum\limits_{i \in \Lambda'} b_i = \sum\limits_{i \in \Lambda \setminus \Lambda'} b_i$?

Для доказательства NP-трудности класса $\mathscr{I}(x_{10})$ мы также воспользуемся сведе́нием задачи Разбиение к соответствующей задаче распознавания существования расписания длины H для входа $I \in \mathscr{I}(x_{10})$. Пусть задано семейство чисел $\{b_1, \dots, b_n\} \subset \mathrm{N}^+$ с четной суммой, т. е. $\sum b_j = 2B$, $B \in \mathrm{N}^+$. Определим вход $I \in \mathscr{I}(x_{10})$ с тремя работами J_1, J_2, J_3, выполняемыми на трех машинах M_1, M_2, M_3. Каждая из работ J_1, J_2 состоит из трех операций длины $B, 2B$ и B, выполняемых соответственно на машинах M_1, M_2 и M_3. Работа J_3 состоит из $2n$ операций, причем n операций имеют длительности $b_1, \dots, b_n$ и выполняются на машине M_1, а n аналогичных операций должны быть выполнены на машине M_3. Для заданного таким образом входа I имеем $H = L_{\max} = l_{\max} = 4B$.

Нетрудно видеть (рис. 1), что для входа I существует расписание длины H, если и только если существует разбиение семейства чисел $\{b_1, \dots, b_n\}$ на два подмножества с равными суммами.

3.3. NP-трудность классов $\mathscr{I}(x_8)$ и $\mathscr{I}(x_9)$

В статье [7] фактически была доказана NP-трудность задачи open shop на классе входов $\mathscr{I}(x_8')$ для $x_8' = (\infty, 3, \infty, 3)$. Искомая NP-трудность вытекала из NP-полноты задачи распознавания существования расписания длины H для входа $I \in \mathscr{I}(x_8')$. К последней сводилась известная NP-полная проблема

Monotone-Not-All-Equal-3Sat [7].

Вход: Множество булевых переменных U и набор C дизъюнкций над U такой, что каждая дизъюнкция $c \in C$ состоит не более чем из трех литералов, а каждый литерал является переменной из U без отрицания.

Вопрос: Существует ли такое назначение переменных, доставляющее значение **истина** всем дизъюнкциям $c \in C$, что в каждой дизъюнкции найдется по крайней мере один истинный и один ложный литерал?

Для доказательства NP-трудности класса $\mathscr{I}(x_8)$ (и симметричного ему класса $\mathscr{I}(x_9)$) воспользуемся той же схемой: искомая NP-трудность будет следовать из NP-полноты задачи распознавания существования расписания длины 6 для входов $I \in \mathscr{I}(x_8)$, а для доказательства NP-полноты последней будет использована та же эталонная NP-полная проблема. Единственное отличие от схемы из [7], которое мы должны обеспечить при построении входа $I \in \mathscr{I}(x_8)$ по заданному входу эталонной проблемы, состоит в том, чтобы каждая работа имела не более двух операций. То, как реализуется данное требование, будет видно из доказательства следующей леммы.

Лемма 5. *Проблема распознавания существования расписания длины 6 для задачи OS со входом из $\mathscr{I}(x_8)$ является NP-полной.*

Доказательство. Пусть задан вход задачи Monotone-Not-All-Equal-3Sat. Последнее означает, что на множестве логических переменных $U = \{x_1, \dots, x_u\}$ задано семейство дизъюнкций $C = \{c_1, \dots, c_v\}$, где каждая дизъюнкция $c_i \in C$ состоит не более чем из трех литералов. Без ограничения общности можем считать, что каждая дизъюнкция c_i состоит в точности из трех литералов: $c_i = l_{i1} \vee l_{i2} \vee l_{i3}$. (В противном случае добавляем в дизъюнкцию недостающие литералы, совпадающие с одним из уже имеющихся в данной дизъюнкции литералов.) Приставка Monotone- в названии задачи означает, что каждый литерал l_{ij} является вхождением некоторой переменной $x_p \in U$ без отрицания.

По заданному семейству C построим вход $I \in \mathscr{I}(x_8)$ задачи OS с $9v$ работами и $7v$ машинами следующим образом. Пусть $\mathscr{L} = \{l_{ij} \mid i = 1, \dots, v;\ j = 1, 2, 3\}$ — семейство литералов, входящих в дизъюнкции $c_i \in C$; функции $c : \mathscr{L} \to C$ и $x : \mathscr{L} \to U$ задают принадлежность каждого литерала $l \in \mathscr{L}$ к какой-то дизъюнкции $c(l) \in C$ и переменной $x(l) \in U$.

Будем определять работы трех типов: α, β и γ. Для каждого литерала $l \in \mathscr{L}$ определим две машины $A(l)$ и $B(l)$, а также работу $\alpha(l)$, состоящую из двух операций $\alpha_A(l)$ и $\alpha_B(l)$; операции выполняются соответственно на машинах $A(l)$ и $B(l)$ за 3 единицы времени. Очевидно, что внутри интервала $[0, 6]$ каждая α-работа $\alpha(l)$ может выполняться одним из двух способов: либо $\alpha_A(l)$ выполняется в интервале $[0, 3]$, а $\alpha_B(l)$ — в интервале $[3, 6]$, либо наоборот. Эти два способа сопоставим со значениями литерала l:

$$l := \text{"операция } \alpha_A(l) \text{ предшествует операции } \alpha_B(l)\text{"}. \qquad (2)$$

Пусть $p_j = |x^{-1}(x_j)|$ — число литералов, соответствующих вхождениям переменной x_j в дизъюнкции $c_i \in C$. Занумеруем элементы из $x^{-1}(x_j)$ произвольным образом числами $1, \dots, p_j$ и будем обозначать их через $x_{j1}, \dots, x_{jp_j}$. Для переменной x_j определим p_j так называемых β-работ $\beta(x_{j1}), \dots, \beta(x_{j,p_j})$. Каждая β-работа $\beta(x_{ji})$ состоит из двух операций: операции $\beta_B(x_{ji})$ длины 3, выполняемой на машине $B(x_{ji})$, и операции $\beta_A(x_{ji})$ длины 1, выполняемой на машине $A(x_{j,(i \bmod p_j)+1})$.

Наконец, для каждой дизъюнкции $c_i = l_{i1} \vee l_{i2} \vee l_{i3}$ определим машину $D(c_i)$, а для каждого литерала $l \in \mathscr{L}$ — работу $\gamma(l)$, состоящую из двух операций длины 2: одна операция $(\gamma_A(l))$ выполняется на машине $A(l)$, а другая $(\gamma_B(l))$ — на машине $D(c(l))$.

Нетрудно убедиться, что так определенный вход I задачи OS принадлежит классу $\mathscr{I}(x_8)$, поскольку каждая работа состоит в точности из двух операций, а на каждой машине выполняется не более трех операций. Докажем, что для входа I существует расписание длины 6, если и только если существует требуемое назначение переменных в задаче Monotone-Not-All-Equal-3Sat.

Предположим, что для входа I существует допустимое расписание длины 6. Определим значение каждого литерала $l \in \mathscr{L}$ согласно (2). Тогда легко убедиться, что синхронность значений литералов, относящихся к одной и той же переменной $x_j \in U$, обеспечивается допустимостью расписания для α- и β-работ (рис. 2).

Рассмотрим произвольную дизъюнкцию $c_k \in C$. Из трех операций γ-работ, выполняемых на машине $D(c_k)$, одна обязательно выполняется в интервале $[0, 2]$. Пусть эта операция относится к литералу l_{ks}, являющемуся i-м вхождением переменной x_j. Тогда вторая операция работы $\gamma(l_{ks})$ выполняется на машине $A(l_{ks})$ во второй половине расписания. Отсюда следует, что операция $\alpha_A(l_{ks})$ выполняется в первой половине расписания и предшествует операции $\alpha_B(l_{ks})$. Следовательно, литерал l_{ks} принимает значение **истина**. Другая из трех операций машины $D(c_k)$ выполняется в интервале $[4, 6]$. Отсюда аналогично заключаем, что соответствующий ей литерал принимает значение **ложь**. Таким образом, в каждой дизъюнкции c_k хотя бы один литерал имеет значение **истина** и хотя бы один — значение **ложь**. Импликация доказана.

Докажем обратную импликацию, т. е. по требуемому в задаче Monotone-Not-All-Equal-3Sat назначению переменных построим допустимое расписание длины 6 в задаче OS. Взаимное расположение операций $\alpha_A(l)$ и $\alpha_B(l)$ определим по заданному значению литерала l из (2). Тогда расписание выполнения этих операций определяется однозначно, как и расписание выполнения B-операций β-работ. Вторую операцию $(\beta_A(x_{ji}))$ работы $\beta(x_{ji})$ «распишем» в интервале $[3, 4]$, если $\beta_B(x_{ji})$

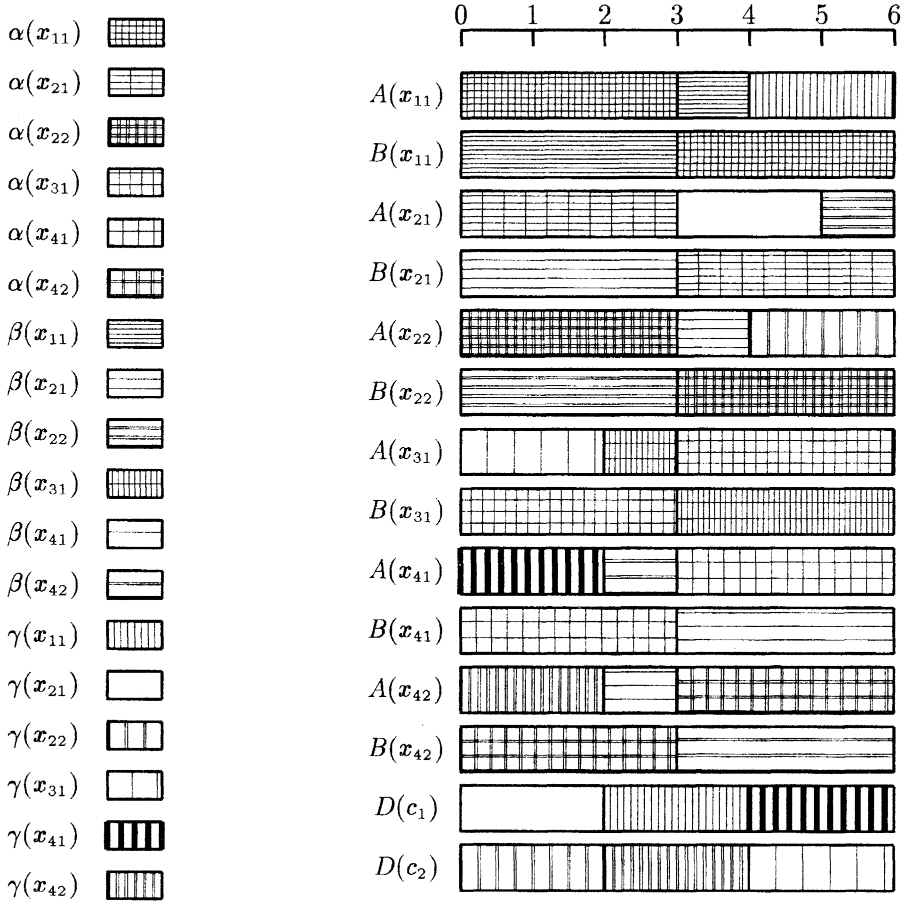

Рис. 2. Расписание длины 6 для входа $I \in \mathscr{I}(x_8)$, соответствующего формуле $(x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee x_4)$

выполняется в первой половине расписания, и в интервале [2, 3], если $\beta_B(x_{ji})$ выполняется во второй половине расписания. Таким образом, интервал выполнения каждой операции $\gamma_A(l)$ на соответствующей машине определяется однозначно: $\gamma_A(l)$ выполняется в интервале [4, 6], если литерал l имеет значение **истина**, и в интервале [0, 2], если l имеет значение **ложь**. Покажем, что оставшиеся D-операции γ-работ можно допустимым образом расписать в интервале [0, 6]. Это следует из того, что для любой дизъюнкции $c_k = l_{k1} \vee l_{k2} \vee l_{k3}$ имеет место один из двух случаев: либо два ее литерала имеют значения **истина**, а один — значение **ложь**, либо наоборот. Таким образом, либо две из трех операций $\gamma_A(l_{ki})$ $(i = 1, 2, 3)$ выполняются в интервале [0, 2], а одна — в интервале [4, 6], либо наоборот. В обоих случаях допустимое расписание для

операций $\gamma_D(l_{ki})$ $(i = 1, 2, 3)$ на машине $D(c_k)$ строится без труда.

Для завершения доказательства леммы 5 остается заметить, что сведе́ние NP-полной задачи Monotone-Not-All-Equal-3Sat к задаче OS на классе входов $\mathscr{I}(x_8)$ выполняется за время, полиномиальное от длины записи эталонной задачи. Отсюда следует NP-полнота рассматриваемой задачи OS. Лемма 5 доказана.

Из леммы 5 помимо NP-трудности класса $\mathscr{I}(x_8)$ вытекает значительно более сильный результат. А именно: для задачи OS на классе входов $\mathscr{I}(x_8)$ доказано несуществование (в предположении $\mathrm{P} \neq \mathrm{NP}$) полиномиального C-приближающего алгоритма для любой константы $C < 7/6$, а следовательно, несуществование полиномиальной аппроксимационной схемы.

Полученные выводы в той же мере применимы и к задаче OS на классе входов $\mathscr{I}(x_9)$.

4. Заключительные замечания и открытые вопросы

Хотя данная работа содержит несколько новых результатов по сложности задачи OS на различных классах входов, основное ее предназначение, по замыслу авторов, — методологическое. Две основные идеи, которые, как предполагается, должен вынести читатель из прочтения данной статьи, должны быть следующие.

Во-первых, назрела потребность в осуществлении более комплексного (многопараметрического) сложностного анализа классических задач дискретной оптимизации, поскольку на самом деле бо́льшая часть задач имеет, как правило, несколько существенных параметров.

Во-вторых, по-видимому, уже созрели достаточные предпосылки для того, чтобы начать рассматривать задачи теории расписаний в более широком контексте, а именно в рамках календарного планирования, где нет «машин» и «работ», но есть всевозможные ресурсные ограничения и ограничения предшествования на множестве операций. Есть ощущение того, что многие современные результаты, полученные в теории расписаний (в частности, результаты по построению полиномиальных аппроксимационных схем или доказательства их несуществования), могут быть перенесены на более широкие модели календарного планирования.

Из открытых проблем мы, безусловно, хотели бы выделить вопрос о сложности задачи OS на классе входов $\mathscr{I}(x_6)$. В зависимости от того, будет ли ответ на этот вопрос положительным (полиномиальная разрешимость) или отрицательным (NP-трудность), мы сможем заключить, состоит ли базисная система для семейства нетривиальных классов задачи OS из 14 или же только из 10 классов.

Другой вопрос, который остается открытым после доказательства леммы 5: сложность распознавания существования расписания длины 4 и 5 для задачи OS со входами из $\mathscr{I}(x_8)$. (Полиномиальность проверки существования расписания длины 3 была доказана в [7].) Если бы NP-полноту проверки удалось доказать для расписаний длины 5 или 4, то это позволило бы поднять барьер «неприближаемости» задачи OS на классе входов $\mathscr{I}(x_8)$ до 6/5 или 5/4 соответственно.

ЛИТЕРАТУРА

1. **Кононов А. В., Севастьянов С. В.** О сложности нахождения связной предписанной раскраски вершин графа // Дискрет. анализ и исслед. операций. Сер. 1. 2000. Т. 7, № 2. С. 21–46.
2. **Chen B., Potts C. N., Woeginger G. J.** A review of machine scheduling: complexity, algorithms and approximability // Handbook of Combinatorial Optimization. Boston, MA: Kluwer Acad. Publ., 1998. V. 3. P. 21–169.
3. **Gonzalez T., Sahni S.** Open shop scheduling to minimize finish time // J. Assoc. Comput. Mach. 1976. V. 23, N 4. P. 665–679.
4. **Jansen K., Solis-Oba R., Sviridenko M.** Makespan minimization in job shops: a polynomial time approximation scheme // Proc. of the thirty first annual ACM symposium on theory of computing. (Atlanta, May 1–4, 1999). New York: ACM Press, 1999. P. 394–399.
5. **Shafransky Y. M.** On a revision of some notions of the theory of computational complexity. Минск, 1997. (Препринт / Академия наук Беларуси. Ин-т техн. кибернетики; № 1).
6. **Sotskov Y. N., Shakhlevich N. V.** NP-hardness of shop scheduling problems with three jobs // Discrete Appl. Math. 1995. V. 59, N 3. P. 237–266.
7. **Williamson D. P., Hall L. A., Hoogeveen J. A., Hurkens C. A. J., Lenstra J. K., Sevastianov S. V., Shmoys D. B.** Short shop schedules // Oper. Res. 1997. V. 45, N 2. P. 288–294.

Адрес авторов:

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

Статья поступила
27 апреля 2000 г.