Д. И. Коган, Ю. С. Федосенко, Задача диспетчеризации: анализ вычислительной сложности и полиномиально разрешимые подклассы, *Дискрет. матем.*, 1996, том 8, выпуск 3, 135–147

Дискретная математика

том 8 выпуск 3 * 1996

УДК 519.854

Задача диспетчеризации: анализ вычислительной сложности и полиномиально разрешимые подклассы

© 1996 г. Д. И. Коган, Ю. С. Федосенко

Рассматривается задача составления оптимального расписания обслуживания конечного детерминированного потока заявок одним прибором по критерию минимума суммы линейных функций индивидуальных штрафов по заявкам. Для вводимых иерархий частных классов рассматриваемой массовой задачи устанавливаются границы возникновения NP-трудности. Показано, что наложение некоторых естественных с точки зрения приложений ограничений на класс моделей или на класс управлений позволяет построить основанные на рекуррентных соотношениях динамического программирования полиномиальные алгоритмы синтеза оптимальных расписаний.

Работа выполнена при поддержке Российского фонда фундаментальных исследований, грант 93–013–16253.

1. Введение

Рассматривается задача составления оптимального расписания обслуживания конечного детерминированного потока заявок одним прибором при линейных функциях индивидуального штрафа по заявкам; минимизируемый критерий выражает суммарный штраф. Данная задача возникает в процессах диспетчеризации производственных и транспортных систем; время ее решения должно быть малым, обеспечивающим возможность оперативного управления. Вместе с тем по своей вычислительной сложности рассматриваемая задача относится к числу NP-трудных [1], расход машинного времени на синтез оптимального расписания общим решающим алгоритмом может оказаться недопустимо большим уже для 17–20 заявок. Актуальными оказываются проблема выделения полиномиально разрешимых подклассов и разработка для общей задачи приближенных решающих методов.

В качестве базисного для проблем диспетчеризации естественен выбор метода динамического программирования или метода ветвей и границ. Сравнительный анализ эффективности соответствующих процедур для задач типа рассматриваемой проведен в [2]. Алгоритмы приближенного решения некоторых задач диспетчеризации и оценки их эффективности приведены в [3].

Данная статья включает два основных раздела. В первом разделе для вводимых иерархий частных классов задачи диспетчеризации устанавливаются границы возникновения NP-трудности (доказательства соответствующих теорем приводятся в §4). Во втором разделе (§3) показывается, что наложение некоторых естественных с точки зрения приложений ограничений на класс моделей или на класс управлений позволяет построить основанные на рекуррентных соотношениях динамического программирования полиномиальные алгоритмы синтеза оптимальных расписаний. Результаты этого раздела открывают возможность эффективного применения метода динамического программирования для решения реально возникающих задач диспетчеризации.

2. Модель задачи, частные классы и оценки вычислительной сложности

Считается заданным поток Z, представляющий собой упорядоченное множество заявок $\{z_1, \dots, z_n\}$, которые должны быть обслужены прибором П. Каждая заявка z_i, $i = 1, \dots, n$, характеризуется тремя целочисленными неотрицательными параметрами, моментом поступления (готовности к обслуживанию) t_i, продолжительностью обслуживания τ_i и коэффициентом линейной функции штрафа a_i (если обслуживание заявки завершается в момент времени $\bar{t}_i$, то $\varphi_i = a_i(\bar{t}_i - t_i)$ — величина индивидуального штрафа по данной заявке).

Полагаем, что $0 \leqslant t_1 \leqslant 1 \ldots \leqslant t_n$ и прибор П готов к обслуживанию заявок потока Z с момента $t = 0$, обслуживание каждой заявки выполняется без прерываний, одновременное обслуживание прибором более одной заявки недопустимо.

Расписание обслуживания отождествляем с перестановкой $\rho = (i_1, \dots, i_n)$ множества индексов заявок; заявка z_{i_k} в расписании ρ обслуживается k-ой по очереди, $k = 1, \dots, n$. Обозначим через t'_α и $\bar{t}_\alpha$ моменты соответственно начала и завершения обслуживания заявки z_α по расписанию ρ. Считаем, что реализация расписания компактна, т.е. имеют место соотношения

$$\begin{aligned} &t'_{i_1}(\rho) = t_{i_1}, \\ &t'_{i_k}(\rho) = \max\{\bar{t}_{i_{k-1}}(\rho), t_{i_k}\}, \qquad && k = 2, \dots, n, \\ &\bar{t}_\alpha(\rho) = t'_\alpha(\rho) + \tau_\alpha, && \alpha = 1, \dots, n. \end{aligned}$$

Суммарный штраф по всем заявкам потока Z, обслуживаемых по расписанию ρ, есть

$$W(\rho) = \sum_{\alpha=1}^{n} a_\alpha(\bar{t}_\alpha(\rho) - t_\alpha).$$

Проблема заключается в минимизации суммарного штрафа, т.е. в нахождении

$$\min_\rho W(\rho). \tag{1}$$

Теорема 1. *Задача* (1) *является NP-трудной в сильном смысле.*

Рассмотрим данную задачу при дополнительных условиях. Введем следующие частные классы рассматриваемой массовой задачи. Задачу (1) отнесем к классу $\mathcal{K}_p^t$, если среди величин t_α, $\alpha = 1, \dots, n$, имеется не более, чем p различных. Задачу (1) отнесем к классу $\mathcal{K}_q^\mu$, если среди величин $\mu_\alpha = a_\alpha / \tau_\alpha$, $\alpha = 1, \dots, n$, имеется не более, чем q различных.

В задачах класса $\mathcal{K}_1^t$ все заявки поступают одновременно, в момент $t = 0$. Для них известен (см., например, [4]) простой алгоритм синтеза оптимального расписания: вычислить величины

$$\mu_\alpha = a_\alpha \tau_\alpha, \qquad \alpha = 1, \dots, n,$$

и упорядочить индексы заявок потока по убыванию значений μ_α; полученная перестановка является искомым решением.

Для задач класса $\mathcal{K}_1^\mu$ выполняется условие

$$\frac{a_1}{\tau_1} = \ldots = \frac{a_n}{\tau_n}. \tag{2}$$

Соотношение (2) может иметь, например, такую интерпретацию: штраф за единицу времени простоя транспортного средства пропорционален его грузоподъемности, а, следовательно, и времени обслуживания (погрузки или выгрузки).

Теорема 2. *Для задач класса $\mathcal{K}_1^\mu$ перестановка $\rho^0 = (1, 2, \dots, n)$ является оптимальным расписанием.*

Приведенные результаты показывают, что задачи классов $\mathcal{K}_1^t$ и $\mathcal{K}_1^\mu$ полиномиально разрешимы, их вычислительная сложность не превышает сложности сортировки n-элементного массива натуральных чисел, т.е. $C_1 n \ln n$, где C_1 — не зависящая от n константа.

Вычислительную сложность задач классов $\mathcal{K}_h^t$ и $\mathcal{K}_q^\mu$ при $p \geqslant 2$, $q \geqslant 2$ характеризуют следующие теоремы 3 и 4.

Теорема 3. *Задача* (1) *для класса $\mathcal{K}_2^t$ является NP-трудной.*

Теорема 4. *Задача* (1) *для класса $\mathcal{K}_2^\mu$ является NP-трудной.*

Легко видеть, что наложение на задачу (1) в общей постановке дополнительного условия $t_n \leqslant c$, где c — некоторая константа, обеспечивает ее полиномиальную разрешимость. Возникающие на практике типовые задачи диспетчеризации удовлетворяют более слабому условию

$$t_n \leqslant (1 + c^*) n, \tag{3}$$

где c^* — положительная константа. Класс задач (1), удовлетворяющий дополнительному условию (3), обозначим через $\mathcal{K}[c^*]$.

Теорема 5. *Задача* (1) *для класса $\mathcal{K}[c^*]$, где c^* — произвольная положительная константа, является NP-трудной в сильном смысле.*

3. Полиномиальные алгоритмы решения частных подклассов задачи диспетчеризации

В качестве базисной будем использовать процедуру динамического программирования. Начнем с рассмотрения общей задачи (1). Очевидно, что в процессе синтеза оптимального расписания управленческие решения принимаются в те моменты дискретного времени, когда прибор свободен. В каждый такой момент при заданных исходных данных ситуация вполне определяется парой (t, S), где t— момент принятия решения, а S — множество заявок, в момент t ожидающих обслуживания.

Пусть $\Sigma(t, S)$ обозначает минимальную величину суммарного штрафа за период времени от момента t до завершения обслуживания всех заявок потока для определяемой парой (t, S) ситуации. Множество S считаем непустым, что обеспечивается обязательным включением в него фиктивной заявки z_0 с параметрами $a_0 = 0$ и $\tau_0 = 1$. Через $G(t, S, \tau)$ обозначим величину суммарного штрафа с момента t до момента $t + \tau$ по заявкам множества S и заявкам, пополняющим его во временном интервале $(t, t + \tau)$. Пусть V_t — совокупность заявок, прибывающих в момент t. Тогда

$$G(t, S, \tau) = \sum_{\alpha \in \tilde{S}} a_\alpha \tau + \sum_{\gamma=1}^{\tau-1} \sum_{\beta \in \tilde{V}_{t+\gamma}} a_\beta(\tau - \gamma)$$

(для произвольного множества заявок $\mathcal{M}$ через $\tilde{\mathcal{M}}$ обозначаем совокупность их индексов). Через $D(t, \tau)$ обозначим совокупность заявок, поступающих на временном отрезке $[t + 1, t + \tau]$, ясно, что

$$D(t, \tau) = \bigcup_{\gamma=1}^{\tau} V_{t+\tau}.$$

В принятых обозначениях рекуррентное соотношение динамического программирования записывается в виде

$$\Sigma(t, S) = \min_{\alpha \in \tilde{S}}(G(t, S, \tau_\alpha) + \Sigma(t + \tau_\alpha, \{S \setminus z_\alpha\} \cup D(t, \tau_\alpha))). \tag{4}$$

Если минимум реализуется при $\alpha = \alpha^*$, то в момент t на прибор следует направить заявку z_{α^*} из совокупности ожидающих обслуживания. Направление на обслуживание заявки z_0 означает простой прибора в течение одного такта; очевидно, что в таком случае он будет простаивать вплоть до момента пополнения множества S по крайней мере одной новой заявкой потока.

Реализации рекуррентной процедуры (4) предшествует вычисление величин $\Sigma(t_n, S)$, где S — произвольное подмножество заявок. Построение оптимального расписания от момента t_n, т.е. в ситуации, когда все заявки потока уже поступили, выполняется очевидным образом — упорядочением необслуженных заявок z_α по убыванию величин $\mu_\alpha = a_\alpha/\tau_\alpha$. При известном расписании значение $\Sigma(t_n, S)$ подсчитывается без затруднений.

Отметим справедливость равенства

$$\Sigma(t_n + \theta, S) = \Sigma(t_n, S)$$

при любых $\theta > 0$ и $S \subseteq \{z_1, \dots, z_n\}$. .

При реализации процедуры (4) последовательно для всех подмножеств заявок S вычисляются значения $\Sigma(t_{n-1}, S)$, $\Sigma(t_{n-2}, S)$, и т.д. Процесс счета по соотношению (4) заканчивается отысканием значения $\Sigma(0, V_0)$. Очевидно, что

$$\min_{\rho} W(\rho) = \Sigma(0, V_0).$$

Вычислительная сложность изложенного алгоритма определяется количеством наборов (t, S), на которых подсчитываются значения функции $\Sigma(t, S)$, и сложностью вычисления каждого ее значения. При подсчетах по соотношению (4) в рассматриваемых парах (t, S) первый аргумент принимает t_n различных значений, так как S является произвольным множеством из числа прибывших на момент t заявок. Таким образом, количество подсчитываемых значений $\Sigma(t, S)$ не превышает $t_n 2^n$. Число элементарных операций, выполняемых при подсчете каждого отдельного значения функции $\Sigma(t, S)$ мажорируется линейной функцией параметра n. Поэтому верхняя оценка числа реализуемых алгоритмом операций есть

$$C_2 n 2^n t_n, \tag{5}$$

где C_2 — не зависящая от n константа. Очевидно, оценка (5) соответствует NP-трудности общей задачи (см. теорему 1).

Далее в данном параграфе будем рассматривать только задачи, принадлежащие классу $\mathcal{K}[c^*]$. Как следует из теоремы 5, NP-трудность проблемы минимизации суммарного штрафа при этом сохраняется. Из оценки (5) следует, что ее экспоненциальный характер определяется для класса $\mathcal{K}[c^*]$ только числом подмножеств S. Вводимые ниже дополнительные, естественные с точки зрения приложений ограничения обеспечивают возможность построения полиномиальных по временной вычислительной сложности процедур динамического программирования. При этом первое ограничение является требованием к исходным данным, остальные требованиями к расписаниям.

Определение 1. Заявки z_α и z_β однотипны, если $a_\alpha = a_\beta$ и $\tau_\alpha = \tau_\beta$.

Ограничение 1. В потоке Z присутствуют заявки не более, чем l типов.

Задачу класса $\mathcal{K}[c^*]$ считаем входящей в подкласс $\mathcal{K}^l$, если для нее выполняется ограничение 1. В подклассе $\mathcal{K}^l$ любое множество заявок S может быть представлено l-мерным вектором $N(S) = (n_1, \dots, n_l)$, где n_i — число заявок i-го типа в множестве S, $i = 1, \dots, l$.

Вместо функции $\Sigma(t, S)$ введем соответствующую функцию $\Sigma'(t, N(S))$, полагая для всех S

$$\Sigma(t, S) = \Sigma'(t, N(S)).$$

Рекуррентное соотношение (4) для задач подкласса $\mathcal{K}^l$ модифицируется очевидным образом, а общее количество рассматриваемых векторов (аргументов функции $\Sigma'(t, N(S))$) оценивается сверху величиной n^l. С учетом этого обстоятельства и соотношения (3) верхняя оценка (5) числа реализуемых алгоритмом элементарных операций принимает вид $C_3 n^l$, где C_3 — константа, не зависящая от l и n. Таким образом, вычислительная сложность задач подкласса $\mathcal{K}^l$ оказывается полиномиальной.

Ограничение 2. Количество ожидающих обслуживания заявок потока в любой момент времени не превышает константы k.

Множество расписаний, удовлетворяющих этому ограничению, обозначим через $R(k)$ и рассмотрим задачу нахождения

$$\min_{\rho\in R(k)} W(\rho). \tag{6}$$

Пусть $\Sigma^k(t,S)$ обозначает минимальную величину суммарного штрафа за период времени от момента t до момента завершения обслуживания всех заявок потока при условиях, что прибор в момент времени t свободен, ожидающие обслуживания заявки в момент t образуют не более, чем $(k+1)$-элементное множество S, включающее, в том числе, фиктивную заявку, и реализуемое расписание принадлежит классу $R(k)$.

Если в момент t начинается обслуживание заявки z_α, то через временной промежуток τ_α количество $r(t,\alpha)$ ожидающих обслуживания заявок будет равно

$$|S| + \left|\bigcup_{\theta=t+1}^{t+\tau_\alpha} V_\theta\right| - \operatorname{sign}\alpha.$$

Обозначим через $\tilde{S}^k$ совокупность тех индексов α из S, для которых выполняется неравенство $r(t,\alpha) \leqslant k+1$. С учетом введенных выше обозначений рекуррентное соотношение динамического программирования для задачи (6) записывается следующим образом:

$$\Sigma^k(t,S) = \min_{\alpha\in\tilde{S}^k}(G(t,S,\tau_\alpha) + \Sigma^k(t+\tau_\alpha, \{S\setminus z_\alpha\}\cup D(t,\tau_\alpha))). \tag{7}$$

Заметим, что для некоторых пар (t,S) множество индексов $\tilde{S}^k$ может оказаться пустым. Такие пары считаем запрещенными, а значения соответствующих величин $\Sigma^k(t,S)$ полагаем равными ∞. Если при подсчете оказалось, что $\Sigma^k(0,V_0)=\infty$, то в рассматриваемой задаче множество расписаний $R(k)$ пусто.

Верхняя оценка количества наборов аргументов, на которых определяется значение функции Σ^k при подсчетах по рекуррентному соотношению (7), есть $\binom{n+k}{k}$. С учетом условия (3) верхней оценкой временной вычислительной сложности изложенного алгоритма решения задачи (6) является полином от n степени $k+2$.

Определение 2. В расписании ρ заявка z_β обходит в обслуживании заявкуz_α, если $\alpha<\beta$, но заявка z_β обслуживается раньше, чем z_α. Разность $\beta-\alpha$ называем величинойобхода.

Ограничение 3. Величина обхода не может превышать заданной константы d.

Обозначим через $R^0(d)$ множество расписаний, удовлетворяющих ограничению 3. Очевидно, что при реализации расписания из $R^0(d)$ любая заявка z_α может быть обойдена в обслуживании только заявками из совокупности $M_d(\alpha)=\{z_{\alpha+1},\dots,z_{\alpha+d}\}$.

Рассмотрим задачу нахождения

$$\min_{\rho \in R^0(d)} W(\rho). \tag{8}$$

Пусть $\Sigma_d(t, j, V_j^t)$ есть минимально возможная величина суммарного штрафа за период времени с момента t и до окончания процесса обслуживания всех заявок потока при условиях, что прибор в момент времени t свободен, наименьшим из индексов множества необслуженных заявок является j, V_j^t — множество ожидающих обслуживания заявок из совокупности $M_d(j)$, включающее, кроме того, фиктивную заявку, и реализуемое расписание принадлежит классу $R^0(d)$.

Пусть Q_t — совокупность всех необслуженных к моменту t заявок, т.е.

$$Q_t = V_j^t \cup \{z_{j+d+1}, z_{j+d+2}, \dots, z_n\},$$

$\xi[\mathcal{M}]$ — минимальный из положительных индексов заявок множества $\mathcal{M}$, $[\mathcal{M}]_d$ — пересечение множества $\mathcal{M}$ с множеством $\{z_{\xi[\mathcal{M}]}, \dots, z_{\xi[\mathcal{M}]}\}$, пополненное фиктивной заявкой z_0.

С учетом выше введенных обозначений рекуррентное соотношение динамического программирования для задачи (8) запишется в виде

$$\Sigma_d(t, j, V_j^t) = \min_{\alpha \in \tilde{V}_j^t} (G(t, S, \tau_\alpha) + \Sigma_d(t + \tau_\alpha, \xi[Q_t \setminus z_\alpha], [(S \setminus z_\alpha) \cup D(t, \tau_\alpha)]_d)) \tag{9}$$

(полная совокупность заявок S, в момент времени t ожидающих обслуживания, по известным аргументам j и V_j^t определяется однозначно).

При вычислениях по соотношению (9) аргумент t функции Σ_d следует считать принимающим последовательно значения $t_n, t_n - 1, \dots, 0$, а аргумент j принимающим значения из множества $\{1, 2, \dots, n\}$. Множество V_j^t образуется заявками из совокупности $\{z_j, \dots, z_{j+d}\}$, причем $z_j \in V_j^t$. Таким образом, при фиксированном значении j число различных множеств V_j^t не может быть больше 2^d. С учетом условия (3) верхней оценкой временной вычислительной сложности изложенного алгоритма решения задачи (8) является полином от n третьей степени.

Ограничение 4. Каждую заявку могут обойти в обслуживании не более q других заявок (q — заданная константа).

Определение 3. Расписание ρ назовем q-расписанием, если не существует заявки, которая при его реализации допускает пропуск вперед на обслуживание более q других заявок.

Подкласс q-расписаний обозначим через $R^1(q)$. Рассмотрим задачу нахождения

$$\min_{\rho \in R^1(q)} W(d). \tag{10}$$

Пусть $\Sigma_q(t, j, M_j^t)$ обозначает минимальную величину суммарного штрафа за период времени с момента t до момента завершения обслуживания всех заявок потока при условиях, что прибор в момент времени t свободен, наименьшим из индексов множества необслуженных заявок является j, M_j^t — множество обслуженных заявок, обошедших на момент времени t заявку z_j, и реализуемое расписание принадлежит классу $R^1(q)$. Очевидно, $|M_j^t| \leqslant q$. Для произвольного множества $\mathcal{M}$ заявок потока и натуральной константы r через $\{\mathcal{M}\}_r$ обозначим совокупность заявок из $\mathcal{M}$ с индексами, большими r. Тогда с учетом выше введенных обозначений рекуррентные соотношения динамического программирования для задачи (10) запишутся в виде

$$\begin{aligned}\Sigma_q(t,j,M_j^t) = \min(&\min_{\alpha\in\tilde{S}\setminus j}(G(S,t,\tau_\alpha)+\Sigma_q(t+\tau_\alpha,j,M_j^t\cup z_\alpha),\\ &= G(S,t,\tau_j)+\Sigma_q(t+\tau_j,\xi[Q_t\setminus z_j],\{M_j^t\}_{\xi[Q_t\setminus z_j]}))),\end{aligned} \tag{11}$$

если $|M_j^t| < q$, и

$$\Sigma_q(t,j,M_j^t) = G(S,t,\tau_j)+\Sigma_q(t+\tau_j,\xi[Q_t\setminus z_j],\{M_j^t\}_{\xi[Q_t\setminus z_j]}), \tag{12}$$

если $|M_j^t| = q$.

По аргументам j и M_j^t значение S определяется однозначным образом.

При вычислениях по соотношениям (11)–(12) аргумент t функции $\Sigma_q(t,j,M_j^t)$ следует считать принимающим последовательно значения $t_n, t_n-1,\dots,0$, а аргумент j принимающим значения $1,\dots,n$. Множество заявок M_j^t не более, чем q-элементное множество. Следовательно, общее количество вычисляемых значений функции Σ_q не превышает $\binom{n+q}{q}nt_n$. С учетом условия (3) верхней оценкой временной вычислительной сложности изложенного алгоритма решения задачи (10) является полином от n степени $q+2$.

4. Доказательства теорем

В этом параграфе приводятся доказательства теорем 1–5.

Доказательство теоремы 1. Покажем, что к задаче (1) полиномиально сводится NP-полная в сильном смысле задача "3-разбиение" [1].

Задача "3-разбиение" определяется множеством натуральных чисел $M = \{b_1,b_2,\dots,b_{3m}\}$ таким, что

$$\sum_{i-1}^{3m} b_i = mB,$$

где B — натуральное число, связанное с элементами множества M неравенствами

$$\frac{B}{4} \leqslant b_i \leqslant \frac{B}{2}, \qquad i=1,\dots,3m.$$

Требуется определить, существует ли разбиение множества индексов $\tilde{M} = \{1,\dots,3m\}$ на m непересекающихся подмножеств $S_1,\dots,S_m$ таких, что

$$\sum_{i\in S_\alpha} b_i = B, \qquad \alpha=1,\dots,m.$$

Указанные для значений $b_1,\dots,b_{3m}$ ограничения снизу и сверху таковы, что каждое из искомых подмножеств S_α должно быть трехэлементным.

По исходным данным задачи “3-разбиение” строим следующую задачу синтеза оптимального расписания (задачу A1). Совокупность заявок на обслуживание считаем состоящей из двух множеств

$$\Omega_1(z_1, z_2, \dots, z_{3m}), \qquad \Omega_2(Z_{3m+1}, z_{3m+2} \dots, z_{4m}).$$

Заявки множества Ω_1 поступают в очередь на обслуживание одновременно в момент $t = 0$, все коэффициенты их функций штрафа равны единице, а продолжительность обслуживания τ_i равна b_i, $i = 1, \dots, 3m$. Заявка z_{3m+j} поступает в очередь на обслуживание в момент $t_{3m+j} = jB + j - 1$, $j = 1, \dots, m$, продолжительность обслуживания каждой заявки из $\Omega_2 \setminus z_{4m}$ полагаем равной единице; продолжительностью обслуживания заявки z_{4m} считаем достаточно большое число p (можно положить $p = 3m^2B$). Коэффициенты линейных функций штрафа для заявок множества Ω_2 считаем одинаковыми и равными достаточно большому числу $T = 3m(2Bm + m + B)$. В оптимальном расписании выбранное значение T гарантирует каждой заявке из Ω_2 обслуживание без промедлений, т.е. с момента времени $jB + j - 1$; при этом суммарный штраф по этим заявкам равен $T(m - 1 + p)$. Суммарный же штраф по заявкам множества Ω_1 в оптимальном расписании, очевидно, не превышает величины p в том и только в том случае, если заявки этого множества можно обслужить полностью до момента прибытия заявки z_{4m}. В свою очередь, это возможно, лишь когда исходная задача “3-разбиение” имеет положительное решение. Получаем, что определяемая множеством задача “3-разбиение” имеет положительное решение в том и только в том случае, если в построенной задаче $A1$ существует решение $\tilde{\rho}$, для которого

$$W(\tilde{\rho}) < T(m + p - 1) + p.$$

Кроме того, если максимальное из чисел определяющего исходную задачу множества M не превышает значение некоторого полинома $A(m)$, то значение максимального из числовых параметров в конструируемой задаче $A1$ не превышает значения некоторого полинома $A'(m)$. Таким образом, из сильной NP-полноты задачи “3-разбиение” следует сильная NP-полнота задачи (1).

Доказательство теоремы 2. Пусть для перестановки $\rho^0 = (1, 2, \dots, n)$ при некотором индексе q имеет место условие $t_q < t_{q+1}$. Через $\tilde{\rho}$ обозначим перестановку, получаемую из ρ^0 переменой местами индексов q и $q + 1$. Покажем, что

$$W(\rho^0) \leqslant W(\tilde{\rho}).$$

При исследовании знака разности $W(\rho^0) - W(\tilde{\rho})$ удобно пользоваться обозначением $W[\rho, i, j]$ для суммы индивидуальных штрафов по заявкам, входящим в расписание ρ, начиная с i-ой и кончая j-ой по порядку обслуживания, $i < j$. Сумму $W(\rho^0)$ представим в виде

$$W(\rho^0) = W[\rho^0, 1, q - 1] + W[\rho^0, q, q + 1] + W[\rho^0, q + 2, n].$$

Аналогично запишем

$$W(\tilde{\rho}) = W[\tilde{\rho}, 1, q - 1] + W[\tilde{\rho}, q, q + 1] + W[\tilde{\rho}, q + 2, n].$$

Ясно, что

$$W(\rho^0) - W(\tilde{\rho}) = k_1 + k_2 + k_3,$$

где

$$k_1 = W[\rho^0, 1, q-1] - W[\tilde{\rho}, 1, q-1],$$
$$k_2 = W[\rho^0, q, q+1] - W[\tilde{\rho}, q, q+1],$$
$$k_3 = W[\rho^0, q+2, n] - W[\tilde{\rho}, q+2, n].$$

Очевидно, что $k_1 = 0$. Поскольку для $\alpha = q+2, q+3, \ldots, n$ имеют место неравенства $t'(\tilde{\rho}) \geqslant t'(\rho^0)$, справедлива оценка $k_3 \leqslant 0$. Докажем, что $k_2 \leqslant 0$. Как легко подсчитать,

$$k_2 = a_{q+1} \max(\Delta, \tau_q) - a_{q+1}\Delta - a_q(\Delta + \tau_{q+1}),$$

где $\Delta = \max(t_{q+1} - t'_q(\rho), 0)$.

При $\max(\Delta, \tau_q) = \Delta$ получаем, что

$$k_2 = -a_q(\Delta + \tau_{q+1} \leqslant 0.$$

В случае $\max(\Delta, \tau_q) = \tau_q$

$$k_2 = -\Delta(a_{q+1} + a_q) \leqslant 0.$$

Таким образом, $W(\rho^0) - W(\tilde{\rho}) \leqslant 0$. Отсюда следует, что перестановка $\rho^0 = (1, 2, \ldots, n)$ представляет собой оптимальное расписание для любой задачи класса $\mathcal{K}_1^{\mu}$. Теорема доказана.

Доказательство теоремы 3. Доказательство проведем путем полиномиального сведения к рассматриваемой проблеме NP-трудной задачи о ранце: максимизировать

$$\sum_{i=1}^{m} s_i x_i \tag{13}$$

при условиях

$$\sum_{i=1}^{m} v_i x_i \leqslant k, \tag{14}$$

$$x_i \in \{0, 1\}, \qquad i = 1, \ldots, m, \tag{15}$$

где k, s_i, v_i, $i = 1, \ldots, m$, — натуральные числа.

В конструируемой по условиям (13)–(15) задаче синтеза оптимального расписания класса $\mathcal{K}_2^t$ (задача А2) поток образуют заявки $z_1, \ldots, z_m$, поступающие в очередь на обслуживание в момент $t = 0$, и привилегированная заявка z^*, поступающая в очередь на обслуживание в момент $t^* = k$. Для заявки z_i полагается $a_i = s_i$, $\tau_i = v_i$, $i = 1, \ldots, m$. Коэффициент a^* линейной функции штрафа по заявке z^* назначается столь большим, чтобы в оптимальном расписании ее обслуживание начиналось с момента t^*. Как легко показать, для этого достаточно выбрать

$$a^* = \sum_{i=1}^{m} s_i \left(\tau^* + \sum_{i=1}^{m} v_i\right) + 1,$$

где τ^* — определяемая ниже продолжительность обслуживания заявки z^*.

Минимизируемая величина $W(\rho)$ может быть представлена в виде

$$W(\rho) = W^1(\rho) + W^2(\rho),$$

где $W^1(\rho)$ — суммарный штраф по имеющимся заявкам за период $[k, k+\tau^*]$, а $W^2(\rho)$ — суммарный штраф за период $[0, k]$ и за период от момента $k+\tau^*$ до окончания процесса обслуживания потока заявок. Как легко видеть,

$$W^2(\rho) \leqslant \sum_{i=1}^{m} s_i \sum_{j=1}^{m} v_j.$$

Введем булев вектор $X_\rho = (x_1, \ldots, x_m)$, где $x_i = 1$, если обслуживание заявки z_i по расписанию ρ завершается до момента k, и $x_i = 0$ в противном случае, $i = 1, \ldots, m$.

Ясно, что

$$W^1(\rho) = \left(\sum_{i=1}^{m} s_i - \sum_{i=1}^{m} s_i x_i + a^*\right).$$

Положим

$$\tau^* = \sum_{i=1}^{m} s_i \sum_{j=1}^{m} v_l + 1.$$

Тогда каждая единица разности

$$\sum_{i=1}^{m} s_i - \sum_{i=1}^{m} s_i x_i$$

вносит в суммарный штраф $W(\rho)$ больший вклад, чем его компонента $W^2(\rho)$. Построение задачи А2, принадлежащей классу $\mathcal{K}_2^t$, закончено.

Очевидно, что перестановке ρ^*, являющейся оптимальным решением задачи А2, соответствует булев вектор $X_{\rho^*} = (x_1^*, \ldots, x_m^*)$, представляющий собой оптимальное решение задачи (13)–(15). Выполненное сведение реализуемо за полиномиальное время. Теорема доказана.

Доказательство теоремы 4. Доказательство проводится путем полиномиального сведения к рассматриваемой проблеме NP-полной задачи "разбиение" (см., например, [1]), формулируемой следующим образом: имеются натуральные числа $p_1, \ldots, p_n$; требуется определить, существует ли подмножество M множества $N = \{1, \ldots, n\}$ такое, что

$$\sum_{i \in M} p_i = \sum_{i \in N \setminus M} p_i.$$

В конструируемой задаче оптимизации подкласса $\mathcal{K}_2^\mu$ (задача $A3$) поток образуют заявки $z_1, z_2, \ldots, z_n, z^0$ и z^{00}. Заявки $z_1, \ldots, z_n$ поступают в очередь на обслуживание в момент времени $t = 0$; для каждой заявки z_i полагаем $a_i = \tau_i = p_i$. Таким образом, $\mu_i = 1$, $i = 1, \ldots, n$. В момент времени

$$t^0 = \frac{1}{2} \sum_{i=1}^{n} p_i$$

поступает заявка z^0. Коэффициент ее функции штрафа полагаем столь большим, что в оптимальном расписании обслуживание z^0 начинается непосредственно в момент ее поступления. Для этого достаточно положить

$$a^0 = \sum_{i=1}^{n} a_i \left(\sum_{i=1}^{n} \tau_i + \tau^0 + \tau^{00} \right) + 1,$$

где τ^0 и τ^{00} — продолжительности обслуживания заявок z^0 и z^{00} соответственно. Считаем $\tau^0 = 1$, а значение τ^{00} установим ниже.

Заявка z^{00} поступает на обслуживание в момент

$$t^{00} = t^0 + \tau^0 + \frac{1}{2} \sum_{i=1}^{n} p_i,$$

соответствующий коэффициент функции штрафа a^{00} полагаем равным $a^0 \tau^{00}$; обслуживание данной заявки в оптимальном расписании начинается с момента времени t^{00}.

Построенная задача А3 принадлежит классу $\mathcal{K}_2^{\mu}$. Действительно, $\mu_1 = \ldots = \mu_n = 1$; для заявок z^0 и z^{00} имеем $\mu^0 = \mu^{00} = a^0$. Оптимальное расписание определяет разбиение множества индексов заявок $z_1, \ldots, z_n$ на три группы: пусть G_1 — индексы заявок, обслуживаемых на временном отрезке $\theta_1 = [0, t^0]$, G_2 — индексы заявок, обслуживаемых на временном отрезке $\theta_2 = [t^0 + \tau^0, t^{00}]$, G_3 — индексы заявок, обслуживание которых начинается не ранее момента времени $t^0 + \tau^{00}$. Отметим, что протяженность отрезков θ_1 и θ_2 равна $(p_1 + \ldots + p_n)/2$.

Суммарный штраф по заявкам с индексами, принадлежащими подмножеству $G_1 \cup G_2$, не превышает

$$\sum_{i=1}^{n} a_i \left(\sum_{i=1}^{n} \tau_i + 1 \right),$$

а суммарный штраф по заявкам с индексами из непустого подмножества G_3 превосходит τ^{00}.

Положим

$$\tau^{00} = \sum_{i=1}^{n} a_i \left(\sum_{i=1}^{n} \tau_i + 1 \right)$$

и заметим, что если в исходной задаче "разбиение" существует искомое подмножество M, то для оптимального расписания ρ^* подмножество G_3 должно оказаться пустым. Таким образом, наличие полиномиального по временной сложности алгоритма синтеза оптимального расписания для задач класса $\mathcal{K}_2^{\mu}$ влекло бы за собой существование полиномиального алгоритма решения задачи "разбиение". Описанное сведение реализуется в полиномиальном времени. Теорема доказана.

Доказательство теоремы 5. Как следует из доказательства теоремы 4.4 в [1], задача "3-разбиение" сохраняет NP-полноту и в случае, когда на величины натуральных чисел $b_1, \ldots, b_{3m}$, определяющих эту задачу, наложено ограничение

$$\max_{1 \leqslant i \leqslant 3m} b_i \leqslant P(m), \tag{16}$$

где $P(m)$ — фиксированный полином одной переменной.

В построенной при доказательстве теоремы 1 задаче A1 последняя заявка z_{4m} поступает на обслуживание в момент $t_{4m} = mB + m - 1$; при этом

$$mB = \sum_{i=1}^{3m} b_i.$$

Считая, что исходная задача “3-разбиение” удовлетворяет условию (16), получаем, что в задаче $A1$ момент прибытия последней заявки t_{4m} можно считать удовлетворяющим условию $t_{4m} \leqslant 3mP(m) + m - 1$.

По NP-трудной задаче A1 построим новую задачу синтеза оптимального расписания A4 следующим образом:

каждый из заданных моментов времени поступления заявок множеств Ω_1 и Ω_2 в очередь на обслуживание сместим по оси времени вправо на l единиц;

введем в поток множество заявок $M^0 = \{z^1, \dots, z^l\}$; продолжительность обслуживания и коэффициент функции штрафа каждой заявки z^i равны единице, а момент поступления в очередь на обслуживание определяется по формуле $t^i = i - 1$, $i = 1, \dots, l$.

Значение l определим из условия

$$l + 3mP(m) + m \leqslant (1 + c^*)(4m + l).$$

Таким образом, можно считать, что

$$l = [(3mP(m) + m - 1 - 4m(1 + c^*)/c^*] + 1.$$

NP-трудная задача A1 сведена к задаче A4, принадлежащей, благодаря сделанному выбору значения l, классу $\mathcal{K}[c^*]$. Сведение реализуется в полиномиальном времени. Теорема доказана.

Список литературы

1. Гэри М., Джонсон Д. *Вычислительные машины и труднорешаемые задачи.* Мир, Москва, 1982.

2. Abdul-Razaq T. S., Potts C. N., Van Wassenhove L. N. A survey of algorithms for the single machine total weighted tardiness scheduling problem. *Discrete Appl. Math.* (1990) **26**, №2/3, 235–253.

3. Chiwei C., Hirofumi M., Guochun T. Worst-case analysis of local search heuristics for the one-machine total tardiness problem. *Naval Res. Logist.* (1990) **37**, №1, 111–121.

4. Танаев В. С., Шкурба В. В. *Введение в теорию расписаний.* Наука, Москва, 1975.

Статья поступила 01.07.94.