ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Январь – март 1995. Том 2, № 1, 21–35

УДК 519.8

О РАСПИСАНИЯХ РАБОТ НА ОДНОЙ МАШИНЕ С ДЛИТЕЛЬНОСТЯМИ, НЕЛИНЕЙНО ЗАВИСЯЩИМИ ОТ ВРЕМЕНИ

А. В. Кононов

Изучается комбинаторная сложность задачи построения расписания работ на одной машине, а также ее частных случаев. Устанавливается, что задача NP-трудна в сильном смысле. Задача остается NP-трудной также в предположении, что директивные сроки для всех работ одинаковы, но в этом случае для ее решения удается построить псевдополиномиальный алгоритм.

При рассмотрении задачи построения расписания работ на одной машине предполагаем, что каждая работа имеет директивный срок и длительность работы определяется суммой фиксированной части и некоторой штрафной добавки за нарушение директивного срока. В качестве критерия оптимальности принимаем длину полученного расписания, точнее, время завершения всех работ. Точная формулировка задачи содержится в [1] и будет приведена ниже. Отметим, что в [1] построены точные алгоритмы, основанные на методе ветвей и границ и динамическом программировании, а также ряд эвристических полиномиальных алгоритмов.

§ 1. Постановка задачи

Пусть $I = \{1, 2, \ldots, n\}$ — множество работ и для каждой работы $i \in I$ заданы следующие величины:

$p_i \geqslant 0$ — фиксированная длительность выполнения,
d_i — директивный срок,
$v_i \geqslant 0$ — штрафной вес.

Длительность выполнения работы i определяется функцией

$$y_i = p_i + \max\{0, v_i(s_i - d_i)\},$$

где s_i — момент начала выполнения работы i. Все работы поступают на выполнение в момент времени $t = 0$. Прерывания в выполнении работ

недопустимы. Одновременно на машине может выполняться не более одной работы. Время окончания выполнения всех работ обозначается через $C = \max(s_i + y_i)$, где максимум берется по $i \in I$. Расписание, минимизирующее величину C, называется *оптимальным.*

Задача GP (нахождение оптимального расписания). *При заданных $p_i \geqslant 0$, $v_i \geqslant 0$, $d_i \in R$ $(i \in I)$ найти значения s_i, удовлетворяющие условиям*

$$y_i = p_i + \max\{0, v_i(s_i - d_i)\},$$

$$y_i + s_i \leqslant s_j, \text{ если } s_i < s_j, \; j \in I,$$

$$y_i y_j = 0, \text{ если } s_i = s_j, \; j \in I, \; i \neq j,$$

$$s_i \geqslant 0$$

и минимизирующие время C окончания всех работ.

Заметим, что при оптимальном расписании машина будет работать без простоев. С учетом сказанного целевая функция может быть записана в виде $C = \sum\limits_{i \in I} y_i$, а расписание полностью определяется последовательностью π запуска работ на обслуживание.

Расписанием для задачи GP называется перестановка π, указывающая порядок запуска работ на выполнение.

Через $C(\pi)$ обозначим длину расписания, заданного перестановкой π. Перестановка π, минимизирующая величину $C(\pi)$, называется *оптимальной.*

Лемма 1 [1]. *Пусть имеется расписание для первых k работ из множества I, $\sum\limits_{i=1}^{k} y_i = t$, такое, что $d_j \leqslant t$ для всех $j > k$. Тогда оптимальная последовательность оставшихся работ получается упорядочением их по возрастанию величин $(p_i - d_i v_i)/v_i$.*

§ 2. Сложность задачи GP

Рассмотрим следующую известную задачу.

Задача SL (минимизация суммарного взвешенного запаздывания на одной машине). *При заданных целочисленных параметрах $\lambda_i \geqslant 0$, $\nu_i \geqslant 0$, $\delta_i \geqslant 0$ $(i \in I)$ найти величины β_i, удовлетворяющие условиям*

$$\xi_i = \max\{0, \beta_i + \lambda_i - \delta_i\},$$

$$\beta_i + \lambda_i \leqslant \beta_j, \text{ если } \beta_i < \beta_j, \; j \in I,$$

$$\lambda_i \lambda_j = 0, \text{ если } \beta_i = \beta_j, \; j \in I, \; i \neq j,$$

$$\beta_i \geqslant 0$$

и минимизирующие величину $L = \sum_{i \in I} \nu_i \xi_i$.

Параметры λ_i, δ_i и ν_i характеризуют длительность выполнения, директивный срок и штрафной вес работы i соответственно, β_i — переменная, обозначающая время начала выполнения работы i, а ξ_i — величина запаздывания работы i.

В [2] показано, что задача SL является NP-трудной в сильном смысле. Заметим, что для задачи SL расписание определяется перестановкой π, указывающей порядок запуска работ на выполнение.

По входу задачи SL при некотором $\varepsilon > 0$ получим вход задачи GP_ε. Введем обозначения $\nu = \max_{i \in I} \nu_i$, $\lambda = \max_{i \in I} \lambda_i$ и положим

$$p_i = \lambda_i, \quad d_i = \delta_i - \lambda_i, \quad v_i = \nu_i \varepsilon, \quad i \in I, \qquad \varepsilon > 0.$$

Пусть $\Lambda = \sum_{i \in I} \lambda_i$ — длина расписания в задаче SL, $r_i = s_i - d_i$. Так как $\delta_i \geqslant 0$, имеем $\xi_i \leqslant \Lambda$ для всех $i \in I$.

Пусть π — некоторая перестановка работ множества I, а $S_\pi(\mathrm{GP}_\varepsilon) = \{s_i |\ i \in I\}$ и $S_\pi(\mathrm{SL}) = \{\beta_i |\ i \in I\}$ — расписания, определяемые перестановкой π в задачах GP_ε и SL. Множества запаздывающих работ в расписаниях $S_\pi(\mathrm{GP}_\varepsilon)$ и $S_\pi(\mathrm{SL})$ обозначим через $J_\varepsilon(\pi) = \{i \in I |\ s_i \geqslant d_i\}$ и $J(\pi) = \{i \in I |\ \beta_i + \lambda_i \geqslant \delta_i\}$.

Пусть $\{j_1, \dots, j_k\}$ — работы множества $J_\varepsilon(\pi)$, перечисленные по порядку их выполнения. Тогда величина $\rho_{j_q} = \sum_{i=1}^{q} r_{j_i} v_{j_i}$ задает увеличение времени окончания работы j_q в расписании $S_\pi(\mathrm{GP}_\varepsilon)$ по сравнению с расписанием $S_\pi(\mathrm{SL})$.

Лемма 2. *Для любой работы* $j_q \in J_\varepsilon(\pi)$ *справедлива оценка*

$$\rho_{j_q} \leqslant \Lambda((1 + \nu\varepsilon)^q - 1). \tag{1}$$

Доказательство. По определению r_j, d_j, ρ_j, ξ_j для каждого j $(1 \leqslant j \leqslant k)$

$$r_{j_i} = s_{j_i} - d_{j_i} = \beta_{j_i} + \rho_{j_{i-1}} + \lambda_{j_i} - \delta_{j_i} = \xi_{j_i} + \rho_{j_{i-1}}, \tag{2}$$

где $\rho_{j_0} = 0$. Оценку (1) для работ $j_q \in J_\varepsilon(\pi)$ докажем индукцией по q $(0 \leqslant q \leqslant k)$. При $q = 0$ оценка (1) справедлива, так как $\rho_{j_0} = 0$. Пусть оценка (1) справедлива для $q = i - 1$. Тогда при $q = i$ из (2) получаем

$$\rho_{j_i} = \rho_{j_{i-1}} + r_{j_i} v_{j_i} \leqslant \rho_{j_{i-1}} + (\Lambda + \rho_{j_{i-1}})\nu\varepsilon$$

$$= \rho_{j_{i-1}}(1 + \nu\varepsilon) + \Lambda\nu\varepsilon \leqslant \Lambda((1 + \nu\varepsilon)^i - (1 + \nu\varepsilon) + \nu\varepsilon) = \Lambda((1 + \nu\varepsilon)^i - 1).$$

Лемма 2 доказана.

Лемма 3. *Существует* $\varepsilon_0 > 0$ *такое, что при любом* $\varepsilon < \varepsilon_0$ *для любой перестановки* π *имеет место равенство* $J(\pi) = J_\varepsilon(\pi)$.

Доказательство. Поскольку длительности работ в задаче GP_ε не меньше длительностей соответствующих работ в задаче SL, всякая работа, запаздывающая в расписании $S_\pi(\mathrm{SL})$, является запаздывающей и для расписания $S_\pi(\mathrm{GP}_\varepsilon)$. Поэтому имеет место включение $J(\pi) \subseteq J_\varepsilon(\pi)$. Докажем, что при достаточно малом ε имеет место обратное включение. Действительно, $j_q \in J(\pi)$ для любой работы $j_q \in J_\varepsilon(\pi)$ в силу (2), неравенства $r_{j_i} \geqslant 0$ при $j_i \in J_\varepsilon(\pi)$, целочисленности величин ξ_i и неравенств

$$\rho_{j_{i-1}} < 1, \quad 1 \leqslant i \leqslant k, \tag{3}$$

вытекающих из (1) при малых ε, например при

$$\varepsilon < \varepsilon_0 = 1/\nu((1/\Lambda + 1)^{1/n-1} - 1).$$

Лемма 3 доказана.

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

Доказательство. Достаточно показать, что к задаче GP псевдополиномиально сводится какая-либо эталонная NP-трудная в сильном смысле задача. В качестве «эталона» мы выбрали задачу SL. Покажем, что решение задачи SL сводится к решению задачи GP_ε. Более того, оптимальная перестановка π^* задачи GP_ε является оптимальной перестановкой работ и для задачи SL, если ε достаточно мало.

Пусть π^* — оптимальная перестановка работ в задаче GP_ε при некотором $\varepsilon \geqslant 0$. Предположим, что она не оптимальна для задачи SL. Так как функционал $L(\pi)$ задачи SL принимает целые значения, согласно предположению

$$L(\pi^*) - L(\overline{\pi}) \geqslant 1, \tag{4}$$

где $\overline{\pi}$ — оптимальная перестановка для задачи SL. Покажем, что из (4) следует неравенство

$$C(\pi^*) - C(\overline{\pi}) > 0, \tag{5}$$

которое противоречит оптимальности π^* для задачи GP_ε и доказывает оптимальность π^* для задачи SL. Поскольку лемма 3 справедлива при $\varepsilon < \varepsilon_0$, пользуясь леммой 3 и соотношением (2), для любых перестановки π и работы $j_i \in J_\varepsilon(\pi)$ получаем

$$r_{j_i} = \xi_{j_i} + \rho_{j_{i-1}}.$$

Согласно определению функционала C для любой перестановки π

$$C(\pi) = \sum_{i \in I} y_i = \sum_{i \in I} (p_i + v_i(s_i - d_i)^+)$$

$$= \sum_{i \in I} p_i + \sum_{i \in J_\varepsilon(\pi)} v_i(s_i - d_i) = \sum_{i \in I} p_i + \sum_{j_i \in J_\varepsilon(\pi)} v_{j_i} r_{j_i}$$

$$= \sum_{i \in I} p_i + \sum_{j_i \in J_\varepsilon(\pi)} \nu_{j_i} \varepsilon(\xi_{j_i} + \rho_{j_{i-1}}) = \sum_{i \in I} p_i + \varepsilon \sum_{j_i \in J_\varepsilon(\pi)} \nu_{j_i}(\xi_{j_i} + \rho_{j_{i-1}}).$$

Учитывая неравенство (4) и определение функционала L, получаем

$$C(\pi^*) - C(\overline{\pi})$$

$$= \varepsilon \Bigg[\sum_{j \in J(\pi^*)} \nu_j \xi_j - \sum_{j \in J(\overline{\pi})} \nu_j \xi_j \Bigg] + \varepsilon \Bigg[\sum_{j_i \in J(\pi^*)} \nu_{j_i} \rho_{j_{i-1}} - \sum_{j_i \in J(\overline{\pi})} \nu_{j_i} \rho_{j_{i-1}} \Bigg]$$

$$\geqslant \varepsilon \Bigg[L(\pi^*) - L(\overline{\pi}) - \sum_{j_i \in J(\overline{\pi})} \nu_{j_i} \rho_{j_{i-1}} \Bigg] \geqslant \varepsilon - \varepsilon\nu \sum_{j_i \in J(\overline{\pi})} \rho_{j_{i-1}}. \quad (6)$$

Оценим сверху величину $\varepsilon\nu \sum_{j_i \in J(\overline{\pi})} \rho_{j_{i-1}}$. Из (1) для любой перестановки π имеем

$$\varepsilon\nu \sum_{j_i \in J(\overline{\pi})} \rho_{j_{i-1}} \leqslant \varepsilon\nu\Lambda \Bigg[\sum_{q=1}^{n} (1+\nu\varepsilon)^q - n \Bigg] \leqslant \varepsilon\nu\Lambda \Bigg[\frac{(1+\nu\varepsilon)^{n+1} - 1 - \nu\varepsilon}{\nu\varepsilon} - n \Bigg]$$

$$= \Lambda((1+\nu\varepsilon)^{n+1} - 1 - \nu\varepsilon(n+1)) = \Lambda((1+\nu\varepsilon)^{n+1} - 1 - \nu\varepsilon - n\nu\varepsilon)$$

$$= \Lambda \Bigg[\sum_{i=0}^{n+1} \binom{n+1}{i} (\nu\varepsilon)^i - 1 - \nu\varepsilon - n\nu\varepsilon \Bigg] = \Lambda \sum_{i=2}^{n+1} \binom{n+1}{i} (\nu\varepsilon)^i$$

$$< \Lambda \sum_{i=2}^{n+1} \frac{(n+1)^i (\nu\varepsilon)^i}{i!} = \Lambda(\nu\varepsilon(n+1))^2 \sum_{i=2}^{n+1} \frac{(n+1)^{i-2}(\nu\varepsilon)^{i-2}}{i!}.$$

Возьмем рациональное $\varepsilon = 1/(\Lambda\nu^2(n+1)^2)$. Заметим, что $\varepsilon < \varepsilon_0$, а длина записи ε не превосходит $\log_2(\Lambda\nu^2(n+1)^2)$ и ограничена полиномом от длины исходной информации задачи SL. Имеем

$$\Lambda(\nu\varepsilon(n+1))^2 \sum_{i=2}^{n+1} \frac{(n+1)^{i-2}(\nu\varepsilon)^{i-2}}{i!} \leqslant \varepsilon \sum_{i=2}^{n+1} \frac{1}{i!} \leqslant (e-2)\varepsilon < 0.8\varepsilon,$$

откуда в силу (6) следует (5). Теорема 1 доказана.

Задача PV. *Решить задачу* GP *с постоянными* v_i.

Теорема 2. *Задача* PV *является* NP-*трудной.*

Задача SL с $\nu_i = 1$ известна как задача минимизации суммарного времени запаздывания. В [3] доказана NP-трудность этой задачи. Повторяя доказательство теоремы 1, убеждаемся в том, что задача минимизации суммарного времени запаздывания сводится к задаче PV.

§ 3. Задачи с общим директивным сроком

Ниже предполагаем, что директивные сроки для всех работ одинаковы.

Задача PD. *Решить задачу GP при условии, что имеет место равенство $d_i = D$ для всех $i \in I$.*

Рассмотрим также несколько частных случаев задачи PD.

Задача (PD$\geqslant$). *Решить задачу PD при условии, что $p_i/v_i \geqslant p_j/v_j$ при всех i, j $(i, j \in I)$ таких, что $p_i > p_j$ и $v_i > 0, v_j > 0$.*

Задача (PD$\leqslant$). *Решить задачу PD при условии, что $p_i/v_i \leqslant p_j/v_j$ при всех i, j $(i, j \in I)$ таких, что $p_i > p_j$ и $v_i > 0, v_j > 0$.*

Задача PDP. *Решить задачу PD при условии, что существует промежуток времени $[t_0, t_1]$ $(t_0 < t_1)$, когда машина находится на профилактике, т. е. не обслуживает ни одной работы, и $D \in [t_0, t_1]$.*

В этом параграфе мы установим полиномиальную разрешимость задачи (PD$\geqslant$) и построим «псевдополиномиальные» алгоритмы для остальных задач. Отметим, что согласно лемме 1 справедливо следующее утверждение: если $D \leqslant 0$, то оптимальная последовательность работ в задаче PD (следовательно, во всех ее частных случаях) получается упорядочением работ по возрастанию величин p_i/v_i.

Пусть множество работ I занумеровано следующим образом:

$$\begin{gathered}\text{если } p_i < p_j, \text{ то } i < j,\\ \text{если } p_i = p_j \text{ и } p_i/v_i < p_j/v_j, \text{ то } i < j.\end{gathered} \tag{7}$$

Лемма 4. *Перестановка π, указывающая порядок запуска работ на выполнение в соответствии с порядком из* (7), *задает оптимальное расписание.*

Доказательство проведем от противного. Пусть π — оптимальное расписание, в котором существуют работы i и j такие, что $j < i$, и работа i выполняется раньше работы j. Без ограничения общности можно считать, что работы i и j выполняются одна за другой. Если это не так, то всегда можно найти работы i_1 и j_1, которые выполняются аналогичным образом и $j_1 < i_1$.

Если $D \leqslant s_i < s_j$, то работы i и j принадлежат множеству Q работ, не упорядоченных к моменту времени D. Работы множества Q упорядочим согласно порядку из (7). Тогда по определению задачи (PD$\geqslant$) эти работы будут упорядочены по возрастанию величин p_i/v_i, что согласно лемме 1 влечет оптимальность расписания.

Пусть $t_0 = s_i < D < s_j$. Обозначим через $\bar{\pi}$ расписание, полученное из π перестановкой работ i и j. Пусть $t_{ij} = y_i(\pi) + y_j(\pi)$ и $t_{ji} = y_j(\bar{\pi}) + y_i(\bar{\pi})$. Тогда

$$t_{ij} = p_i + p_j + (t_0 + p_i - D)v_j, \quad t_{ji} = p_i + p_j + (t_0 + p_j - D)^+ v_i.$$

Покажем, что $t_{ij} \geqslant t_{ji}$. Действительно, последнее неравенство выполнено при $(t_0 + p_j - D)^+ = 0$. Если $(t_0 + p_j - D)^+ > 0$, то

$$\begin{aligned} t_{ij} - t_{ji} &= (t_0 + p_i - D)v_j - (t_0 + p_j - D)v_i \\ &= v_i v_j (p_i/v_i - (D - t_0)/v_i - p_j/v_j + (D - t_0)/v_j) \\ &= v_i v_j (p_i/v_i(1 - (D - t_0)/p_i) - p_j/v_j(1 - (D - t_0)/p_j)) \geqslant 0. \end{aligned}$$

Следовательно, при перестановке работ i и j в π местами длина расписания не увеличится. Наконец, если $s_i < s_j \leqslant D$, то от перемены последовательности выполнения работ i и j в перестановке π длина расписания не изменится.

В каждом из разобранных случаев перестановка работ i и j не ухудшает значения функционала C. Поэтому существует оптимальное расписание π, в котором работы выполняются по возрастанию величин p_i. Лемма 4 доказана.

Теорема 3. *Для решения задачи* (PD $\geqslant$) *существует алгоритм трудоемкости* $O(n \log n)$.

Доказательство непосредственно вытекает из леммы 4.

Лемма 5. *Пусть π — оптимальное расписание в задаче* PD, *в котором выполнение работы i заканчивается до наступления директивного срока D. Тогда существует оптимальное расписание π^*, в котором работа i выполняется первой.*

Лемма 6. *Пусть π — оптимальное расписание в задаче* PD, *для работы i выполнено неравенство $s_i(\pi) \geqslant D$ и $p_i/v_i \geqslant p_j/v_j$ для любой работы $j \in I$ такой, что $s_j(\pi) \geqslant D$. Тогда существует оптимальное расписание π^*, в котором работа i выполняется последней.*

Лемма 7. *Пусть в задаче* PD *в оптимальном расписании π для работы i справедливо неравенство $s_i \leqslant D$ и $p_i \leqslant p_j$ для всех j таких, что $s_j < s_i$. Тогда существует оптимальное расписание π^*, в котором работа i выполняется первой.*

Утверждения лемм 5 и 7 очевидны, а лемма 6 непосредственно вытекает из леммы 1.

Перейдем к построению алгоритмов. До конца параграфа предполагаем, что множество работ I занумеровано следующим образом:

$$\begin{gathered} \text{если } p_i/v_i < p_j/v_j, \text{ то } i < j, \\ \text{если } p_i/v_i = p_j/v_j \text{ и } p_i > p_j, \text{ то } i < j. \end{gathered} \tag{8}$$

Через $\langle k, d, l_0, l_1 \rangle$ обозначим частную задачу PDP, в которой множество работ состоит из первых k работ множества I, общий директивный срок

равен d, а $[l_0, l_1]$ — интервал профилактики. Пусть $\langle n, D, t_0, t_1\rangle$ — исходная задача PDP, $0 \leqslant t_0 \leqslant D \leqslant t_1$. Рассмотрим двухпараметрическое семейство частных задач PDP: $\{\langle k, \Delta\rangle \mid k = 1, \dots, n; \Delta = 0, \dots, t_0\}$, где $\langle k, \Delta\rangle = \langle k, D-\Delta, t_0-\Delta, t_1-\Delta\rangle$. Через $C^*(k, \Delta)$ обозначим оптимальное значение целевой функции C в задаче $\langle k, \Delta\rangle$. Для упрощения выкладок предположим, что функция $C^*(k, \Delta)$ определена для всех значений параметров k и Δ, причем $C^*(k, \Delta) = \infty$ для $k \notin [1, n]$ или $\Delta \notin [0, t_0]$.

Лемма 8. *Для задачи PDP справедливы следующие рекуррентные соотношения динамического программирования:*

$$C^*(1, \Delta) = \begin{cases} p_1 & \text{при } p_1 \leqslant t_0 - \Delta, \\ t_1 - \Delta + p_1 + (t_1 - D)v_1 & \text{при } p_1 > t_0 - \Delta, \end{cases} \tag{9}$$

$$C^*(k, \Delta) = \min\{C_1, C_2\}, \tag{10}$$

где

$$\begin{aligned} C_1 &= p_k + C^*(k-1, \Delta + p_k), \\ C_2 &= C' + p_k + (C' - D + \Delta)v_k, \\ C' &= \max\{t_1 - \Delta, C^*(k-1, \Delta)\}. \end{aligned}$$

Доказательство. Соотношение (9) позволяет вычислить длину оптимального расписания, когда требуется выполнить одну работу.

Для доказательства (10) предположим, что значения всех величин из правой части выражения (10) уже известны. Тогда по леммам 5 и 6 для работы k существует оптимальное расписание, в котором эта работа выполняется либо первой до интервала профилактики, либо последней после этого интервала.

В первом случае оптимальное расписание остальных работ получается решением задачи $\langle k-1, \Delta + p_k\rangle$. Следовательно, $C^*(k, \Delta) = C_1$.

Во втором случае работа k начинает выполняться в момент времени $t_1 - \Delta$, если предшествующие ей работы завершились до момента времени $t_0 - \Delta$, или в момент времени $C^*(k-1, \Delta)$ в противном случае. Поэтому $C^*(k, \Delta) = C_2$. Лемма 8 доказана.

Введем обозначение $D^+ = \max\{0, D\}$.

Теорема 4. *Задача PDP с целочисленными исходными данными, n работами и директивным сроком D разрешима с трудоемкостью $O(nD^+ + n\log n)$.*

Доказательство. Действительно, в силу (9), (10) для получения оптимального расписания задачи $\langle n, 0\rangle$ достаточно рассмотреть $n(t_0+1)$

задач $\langle k, \Delta \rangle$. Кроме того, для получения исходной нумерации работ требуется $O(n \log n)$ операций. Следовательно, решение задачи PDP находится за $O(nD^+ + n \log n)$ операций. Теорема 4 доказана.

Через $\langle k, d \rangle'$ обозначим частную задачу PD, в которой множество работ состоит из первых k работ множества I и общий директивный срок равен d. Пусть $\langle n, D \rangle'$ — исходная задача PD. Через $C^*(k, d)$ обозначим оптимальное значение целевой функции C в задаче $\langle k, d \rangle$. Для простоты выкладок предположим, что функция $C^*(k, d)$ определена для всех значений параметров k и d, причем $C^*(k, d) = \infty$, если $k \notin [1, n]$ или $d \notin [0, D]$.

Лемма 9. *Для задачи* PD *справедливы следующие рекуррентные соотношения динамического программирования:*

$$C^*(1, d) = p_1, \tag{11}$$

$$C^*(k, d) = \min \begin{cases} C^*(k-1, d-p_k) + p_k, \\ C^*(k-1, d) + p_k + v_k(C^*(k-1, d) - d)^+, \\ \min\limits_{i \in (0, p_k)} \max\{d + i, C^*(k-1, d, d - p_k + i, d + i)\}, \end{cases} \tag{12}$$

где $C^*(k-1, d, d-p_k+i, d+i)$ *является оптимальным решением частной задачи* PDP $\langle k-1, d, d-p_k+i, d+i \rangle$.

Доказательство. Соотношение (11) позволяет вычислить длину оптимального расписания, когда требуется выполнить одну работу.

Для доказательства (12) предположим, что значения всех величин из правой части выражения (12) уже известны. Тогда если работа k в оптимальном расписании выполняется до директивного срока, то по лемме 5 она выполняется первой; оптимальное расписание остальных работ получается решением задачи $\langle k-1, d-p_k \rangle'$. Следовательно, $C^*(k, d) = C^*(k-1, d-p_k) + p_k$. Если работа k в оптимальном расписании выполняется после директивного срока, то по лемме 6 она выполняется последней. Тогда

$$C^*(k, d) = C^*(k-1, d) + p_k + v_k(C^*(k-1, d) - d)^+.$$

Наконец, если работа k в оптимальном расписании начинает выполняться до директивного срока, а заканчивает выполняться после него, то оптимальное расписание остальных работ получается решением задачи $\langle k-1, d, d-p_k+i, d+i \rangle$ для некоторого $i \in (0, p_k)$. Следовательно,

$$C^*(k, d) = \min_{i \in (0, p_k)} \max\{d + i, C^*(k-1, d, d-p_k+i, d+i)\}.$$

Лемма 9 доказана.

Теорема 5. *Задача PD с целочисленными исходными данными, n работами и директивным сроком D разрешима с трудоемкостью порядка $O(nD^+p^2 + n\log n)$, где $p = \max\limits_{i\in I} p_i$.*

Доказательство. Заметим, что при фиксированных значениях d и k имеется не более p^2 различных задач $\langle k-1, d, d-p_k+i, d+i\rangle$. Поэтому в силу (11), (12) для получения оптимального расписания задачи $\langle n, D\rangle'$ достаточно решить не более nD^+p^2 задач $\langle k-1, d, d-p_k+i, d+i\rangle$ и не более nD^+ задач $\langle k, d\rangle'$. Кроме того, для получения исходной нумерации работ надо выполнить $O(n\log n)$ операций. Следовательно, решение задачи PD находится за $O(nD^+p^2 + n\log n)$ операций. Теорема 5 доказана.

Лемма 10. *Для задачи* (PD $\leqslant$) *справедливы следующие рекуррентные соотношения динамического программирования:*

$$C^*(1,d) = p_1, \tag{13}$$

$$C^*(k,d) = \min \begin{cases} C^*(k-1, d-p_k) + p_k, \\ C^*(k-1,d) + p_k + v_k(C^*(k-1,d) - d)^+. \end{cases} \tag{14}$$

Доказательство. Соотношение (13) позволяет вычислить длину оптимального расписания, когда требуется выполнить одну работу.

Для доказательства (14) предположим, что значения всех величин из правой части выражения (14) уже известны. Заметим, что так как работы занумерованы в порядке, указанном в (8), для задачи (PD$\leqslant$) эти работы занумерованы по убыванию p_i. Тогда если работа k в оптимальном расписании начинает выполняться до директивного срока, то по лемме 7 она выполняется первой; оптимальное расписание остальных работ получается решением задачи $\langle k-1, d-p_k\rangle'$. Следовательно, $C^*(k,d) = C^*(k-1, d-p_k) + p_k$. Если работа k в оптимальном расписании выполняется после директивного срока, то по лемме 6 она выполняется последней. Тогда $C^*(k,d) = C^*(k-1,d) + p_k + v_k(C^*(k-1,d) - d)^+$. Лемма 10 доказана.

Теорема 6. *Задача* (PD $\leqslant$) *с целочисленными исходными данными, n работами и директивным сроком D разрешима с трудоемкостью $O(nD^+ + n\log n)$.*

Доказательство. Действительно, из (13), (14) следует, что для получения оптимального расписания задачи $\langle n, D\rangle'$ достаточно решить nD^+ задач $\langle k, d\rangle'$. Кроме того, для получения исходной нумерации работ требуется выполнить $O(n\log n)$ операций. Следовательно, решение задачи (PD$\leqslant$) находится за $O(nD^+ + n\log n)$ операций. Теорема 6 доказана.

§ 4. Сложность задач с общим директивным сроком

В этом параграфе установим NP-трудность задач PD, PDP, (PD$\leqslant$). Сформулируем следующую известную задачу.

Задача ZR. *При заданных целых $a_i \geqslant 1$ $(i \in I)$ и целом $B \geqslant 1$ найти подмножество $J \subseteq I$ такое, что $\sum\limits_{i\in J} a_i = B$.*

NP-полнота задачи о сумме размеров показана в [4]. Отметим, что эта задача приведена в списке NP-полных задач в [5].

По входу задачи ZR получим вход задачи PD. Введем обозначения

$$a = \max_{i\in I} a_i, \quad A = \sum_{i\in I} a_i$$

и положим

$$p_0 = 1, \quad v_0 = \mu^{-1},$$

$$p_i = \mu a_i,\ v_i = \varepsilon a_i, \quad i \in I,$$

$$D = \mu B,$$

где $\varepsilon = a^{-2}n^{-2}$, $\mu = \varepsilon(28An)^{-1}$.

Обозначим через $\overline{t_i}$ время окончания выполнения работы i, т. е. $\overline{t_i} = s_i + y_i$. Справедлива

Лемма 11. *Пусть $K = \{1, \dots, k\}$ — список нерасписанных работ к моменту времени $T > D$. Предположим, что, начиная с момента T, работы выполняются последовательно по порядку номеров в списке K. Тогда при любом $l = 1, \dots, k$*

$$\overline{t_l} = D + (T - D)\prod_{i=1}^{l}(1 + v_i) + \sum_{i=1}^{l} p_i \prod_{j=i+1}^{l}(1 + v_j). \tag{15}$$

Доказательство. Из определения величин $\overline{t_i}, y_i$ и условия леммы для каждого $i = 1, \dots, k$ имеем

$$\overline{t_i} = \overline{t_{i-1}} + p_i + (\overline{t_{i-1}} - D)v_i. \tag{16}$$

Доказательство проведем индукцией по q $(1 \leqslant q \leqslant k)$. При $q = 1$

$$\overline{t_1} = D + (T - D)(1 + v_1) + p_1 = T + p_1 + (T - D)v_1,$$

и равенство (15) выполняется.

Пусть (15) верно при $q = l - 1$. Тогда из (16) при $q = l$ получаем

$$\overline{t_l} = \overline{t_{l-1}} + p_l + (\overline{t_{l-1}} - D)v_l$$

$$= D + (T-D)\prod_{i=1}^{l-1}(1+v_i) + \sum_{i=1}^{l-1} p_i \prod_{j=i+1}^{l-1}(1+v_j) + p_l$$

$$+\left[D + (T-D)\prod_{i=1}^{l-1}(1+v_i) + \sum_{i=1}^{l-1} p_i \prod_{j=i+1}^{l-1}(1+v_j) - D\right] v_l$$

$$= D + (T-D)\prod_{i=1}^{l-1}(1+v_i) + v_l(T-D)\prod_{i=1}^{l-1}(1+v_i)$$

$$+\sum_{i=1}^{l-1} p_i \prod_{j=i+1}^{l-1}(1+v_j) + v_l \sum_{i=1}^{l-1} p_i \prod_{j=i+1}^{l-1}(1+v_j) + p_l$$

$$= D + (T-D)\left[\prod_{i=1}^{l-1}(1+v_i)\right](1+v_l) + \left[\sum_{i=1}^{l-1} p_i \prod_{j=i+1}^{l-1}(1+v_j)\right](1+v_l) + p_l$$

$$= D + (T-D)\prod_{i=1}^{l}(1+v_i) + \sum_{i=1}^{l} p_i \prod_{j=i+1}^{l}(1+v_j).$$

Лемма 11 доказана.

Теорема 7. *Задача* PD *является* NP*-трудной.*

Доказательство. Достаточно показать, что к задаче PD полиномиально сводится какая-либо эталонная NP-трудная задача. В качестве последней рассмотрим задачу ZR. Преобразование входа задачи ZR ко входу задачи PD показано в начале этого параграфа. Покажем, что решение задачи ZR сводится к решению задачи PD.

Пусть π — оптимальная перестановка в задаче PD и Q — множество работ, выполняющихся без нарушения директивного срока. Рассмотрим два случая.

Случай 1: работа 0 принадлежит Q. Так как $p_0 \geqslant p_i$ при любом $i \in I$, можно считать, что среди работ из Q работа 0 выполняется последней. Если в расписании, заданном перестановкой π, работа 0 начинает обслуживаться в момент времени μB, то множество работ $Q \setminus \{0\}$ является решением задачи ZR. Пусть работа 0 в оптимальной перестановке π начинает обслуживаться в момент времени T, $T < \mu B$. Покажем, что не существует $J \subseteq I$ такого, что $\sum\limits_{i \in J} a_i = B$. Предположим противное, т. е. пусть такое подмножество существует. Построим другую перестановку π^* следующим образом. Пусть первыми выполняются работы из множества J, затем — работа 0 и, наконец, оставшиеся

работы в произвольном порядке. Будем считать, что работы в перестановке π занумерованы по порядку $j_0, j_1, \ldots, j_n$, а в перестановке π^* — по порядку $k_0, k_1, \ldots, k_n$. Работы множества J будут завершены в момент времени μB, так как по предположению $\sum\limits_{i \in J} p_i = \sum\limits_{i \in J} \mu a_i = \mu B$. По лемме 11 получаем

$$C(\pi) = D + (T + 1 - D) \prod_{j_i \in I \setminus Q} (1 + v_{j_i}) + \sum_{j_i \in I \setminus Q} p_{j_i} \prod_{l=i}^{n} (1 + v_{j_l}),$$

$$C(\pi^*) = D + \prod_{k_i \in I \setminus J} (1 + v_{k_i}) + \sum_{k_i \in I \setminus J} p_{k_i} \prod_{l=i}^{n} (1 + v_{k_l}).$$

Следовательно,

$$\begin{aligned} C(\pi) - C(\pi^*) = (T + 1 - D) \prod_{j_i \in I \setminus Q} (1 + v_{j_i}) + \sum_{j_i \in I \setminus Q} p_{j_i} \prod_{i}^{n} (1 + v_{j_l}) \\ - \prod_{k_i \in I \setminus J} (1 + v_{k_i}) - \sum_{k_i \in I \setminus J} p_{k_i} \prod_{l=i}^{n} (1 + v_{k_l}) \end{aligned}$$

$$\geqslant (1 - (D - T)) \prod_{j_i \in I \setminus Q} (1 + v_{j_i}) - \prod_{k_i \in I \setminus J} (1 + v_{k_i}) - \sum_{k_i \in I \setminus J} p_{k_i} \prod_{l=i}^{n} (1 + v_{k_l})$$

$$\geqslant (1 - (D - T)) \prod_{j_i \in I \setminus Q} (1 + \varepsilon a_{j_i}) - \prod_{k_i \in I \setminus J} (1 + \varepsilon a_{k_i}) - \sum_{k_i \in I \setminus J} \mu a_{k_i} \prod_{l=i}^{n} (1 + \varepsilon a_{k_l})$$

$$\begin{aligned} \geqslant 1 - (D - T) + (1 - (D - T)) \sum_{j_i \in I \setminus Q} \varepsilon a_{j_i} - 1 - \sum_{k_i \in I \setminus J} \varepsilon a_{j_i} \\ - \sum_{i=2}^{n} \binom{n}{i} (\varepsilon a)^i - \sum_{k_i \in I \setminus J} \mu a_{k_i} \prod_{l=i}^{n} (1 + \varepsilon a_{k_l}). \end{aligned} \tag{17}$$

Далее имеем

$$\sum_{i=2}^{n} \binom{n}{i} (\varepsilon a_i)^i \leqslant \varepsilon \sum_{i=2}^{n} \frac{n^i a^i \varepsilon^{i-1}}{i!} \leqslant \varepsilon \sum_{i=2}^{n} \frac{n^{1-i} a^{1-i}}{i!} < 0.8\varepsilon, \tag{18}$$

Поэтому

$$\sum_{k_i \in I \setminus J} \mu a_{k_i} \prod_{l=i}^{n} (1 + \varepsilon a_{k_l}) \leqslant \varepsilon \sum_{k_i \in I \setminus J} \frac{1}{28n} \left(1 + \sum_{i=1}^{n} \varepsilon a_i + 0.8\varepsilon\right)$$

$$\leqslant \frac{1}{28n}\varepsilon\left(1+\frac{1}{an^2}+\frac{0.8}{a^2n^2}\right)\leqslant 0.1\varepsilon. \qquad (19)$$

Используя (18) и (19), из (17) получаем

$$C(\pi)-C(\pi^*)\geqslant 1-(D-T)+(1-(D-T))\sum_{j_i\in I\setminus Q}\varepsilon a_{j_i}-1-\sum_{k_i\in I\setminus J}\varepsilon a_{j_i}-0.9\varepsilon.$$

Так как $D-T\leqslant D=\mu B=\frac{\varepsilon}{28n}$, а $\sum\limits_{j_i\in I\setminus Q} a_{j_i}-\sum\limits_{k_i\in I\setminus J} a_{j_i}\geqslant 1$, имеем

$$C(\pi)-C(\pi^*)\geqslant 0.1\varepsilon-\frac{\varepsilon}{28n}-\frac{\varepsilon}{28an^2}>0,$$

что противоречит оптимальности π.

Случай 2: работа 0 не принадлежит Q. Так как a_i целые, работа 0 начинает выполняться не раньше момента времени $\mu B+\mu$ и $C(\pi)\geqslant \mu B+\mu+1+\mu v_0>2$. Пусть π^* — перестановка, в которой работа 0 выполняется первой, а затем — оставшиеся работы в произвольном порядке. В силу леммы 11 и оценок (18), (19) получаем

$$C(\pi^*)=D+(1-D)\prod_{i=1}^{n}(1+v_i)+\sum_{i=1}^{n}\left(p_i\prod_{j=i+1}^{n}(1+v_j)\right)\leqslant D+1 + \sum_{i=1}^{n}\varepsilon a_i+0.9\varepsilon<2,$$

что противоречит оптимальности π. Теорема 7 доказана.

Теорема 8. *Задача* (PD $\leqslant$) *является* NP-*трудной.*

Доказательство вытекает из того факта, что параметры задачи PD, построенной в ходе доказательства теоремы 7, удовлетворяют следующему условию: если $p_i>p_j$, то $p_i/v_i\leqslant p_j/v_j$ для любых $i,j\in I$.

Теорема 9. *Задача* PDP *является* NP-*трудной.*

Доказательство. Положив в задаче PD профилактику машины в промежутке $[1+\mu B, 1+\mu B+\mu]$ и повторяя рассуждения из доказательства теоремы 7, получаем требуемое утверждение.

ЛИТЕРАТУРА

1. **Kunnathur A. S., Gupta S. K.** Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem // European J. Oper. Res. 1990. V. 47, N 1. P. 56–64.
2. **Lawler E. L.** A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness // Ann. Discrete Math. Amsterdam: North-Holland, 1977. V. 1. P. 331–342.
3. **Du J., Leung J. Y.-T.** Minimizing total tardiness on one machine is NP-hard // Math. Oper. Res. 1990. V. 15, N 3. P. 483–495.
4. **Karp R. M.** Reducibility among combinatorial problems // Complexity of Computer Computations. New York: Plenum Press, 1972. P. 85–103.
5. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

Адрес автора:

РОССИЯ,
630090, Новосибирск,
Университетский пр., 4,
Институт математики СО РАН

Статья поступила

13 января 1994 г.