Исследование комбинаторной структуры оптимальных решений задачи одного станка

Ксения Черных, Владимир Сервах

Институт математики им. С.Л.Соболева СО РАН
Омский филиал

Новосибирск, август 2019 года

Постановка задачи

n – число работ;

p_i – длительность работы i, $i = 1, 2, \ldots, n$;

w_i – "вес" работы i.

Необходимо составить расписание выполнения работ с учетом возможности их прерываний, при котором суммарное взвешенное время завершения работ $\sum_{i=1}^{n} w_i C_i$ минимально. Здесь C_i момент завершения работы i.

Задача $1||\sum w_i C_i$ полиномиально разрешима. Оптимальная перестановка $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ удовлетворяет условию

$$\frac{p_{\alpha_1}}{w_{\alpha_1}} \leq \frac{p_{\alpha_2}}{w_{\alpha_2}} \leq \ldots \leq \frac{p_{\alpha_n}}{w_{\alpha_n}}.$$

r_i – момент поступления работы i.

Сложность различных постановок задач

$1|r_i|\sum w_iC_i$ является NP-трудной в сильном смысле.

(**1977**) J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Brucker. *Complexity of machine scheduling problems.* Ann. of Discrete Math., 1. 343-362.

$1|r_i, p_i = p|\sum w_iC_i$ полиномиально разрешима.

(**2000**) P. Baptiste. *Scheduling equal-length jobs on identical parallel machines.* Discrete Appl. Math., 103(1). 21-32.

$1|r_i, pmtn|\sum w_iC_i$ NP-трудна в сильном смысле. К ней сводится задача три разбиения.

(**1984**) J. Labetoulle, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. *Preemptive scheduling of uniform machines subject to release dates.* In Progress in combinatorial optimization.Academic Press, Toronto. 245-261.

Вопрос о вычислительной сложности задачи $1|r_i, p_i = p, pmtn|\sum w_iC_i$ открыт.

Пример входных данных задачи

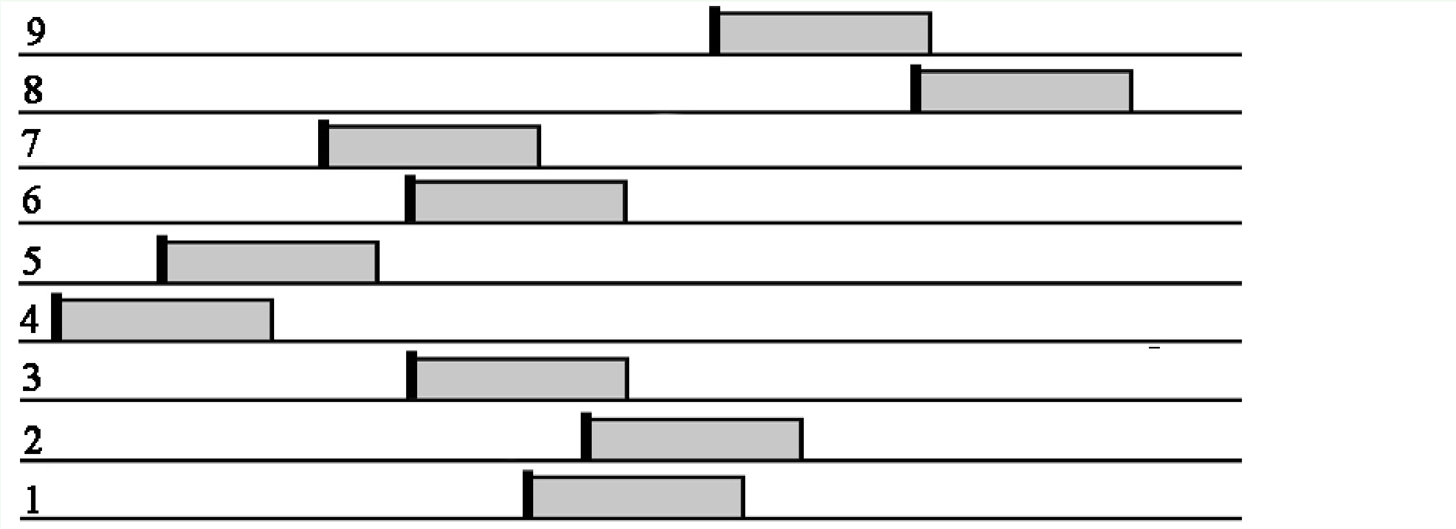


Рис.1. Пример входных данных задачи

Работы перенумерованы в порядке неубывания весов. Если веса одинаковые, то в порядке невозрастания времен поступления.

Свойства задачи

Лемма 1. Существует оптимальное расписание, в котором все моменты прерываний работ принадлежат множеству $\{r_1, r_2, \dots, r_n\}$.

Следствие 1. Существует оптимальное расписание, в котором суммарное число прерываний всех работ не превосходит $n - 1$.

Лемма 2. Прервать некоторую работу может только работа с большим весом.

Лемма 3. Переключение станка на выполнение другой работы происходит либо в момент r_i, либо в момент окончания некоторой работы C_j.

Свойства задачи

Пусть S_i – момент начала выполнения работы i, $i = 1, 2, \ldots, n$.

Лемма 4. Если $w_i < w_j$ и $r_i \geq r_j$, то в оптимальном расписании работа j целиком выполняется раньше работы i, то есть $C_j \leq S_i$.

При таком условии работы i и j конкурировать в расписании не будут.

Порядок выполнения может меняться только для пары работ i и j при условии $r_i < r_j$ и $w_i < w_j$, когда более важная работа поступает позднее. Такие работы назовем *конкурентными*.

Лемма 5. Интервал $[S_j, C_j]$ не может содержать фрагментов более легкой работы.

Это означает, что более легкая работа i, либо выполняется целиком до начала тяжелой работы j, либо целиком после, либо окаймляет ее.

Следствие. Если некоторая работа прервала более легкую, то она завершится раньше, чем более легкая возобновит свое выполнение.

Альтернативы в моменты r_i и C_j

В момент r_i имеет место альтернатива выбора между *текущей работой* j и *поступившей работой* i. Причем, если выбор происходит в пользу работы i, то, в соответствии с леммой 5, она целиком завершиться до возобновления работы j.

Если нет прерванных или доступных для выполнения работ, то альтернативы в C_j нет.

Если есть и прерванные и доступные работы. Пусть i последняя из прерванных работ, а j доступная работа с наибольшим весом.

Если $w_i > w_j$, то продолжаем выполнять работу i .

Если $w_i < w_j$, то возникает **альтернатива** выбора между i и j.

Лемма 6. При альтернативе выбора в момент C_j либо п*родолжит выполнение последняя из прерванных работ*, либо *начнет выполнение доступная работа с максимальным весом*.

Алгоритм формирования расписаний

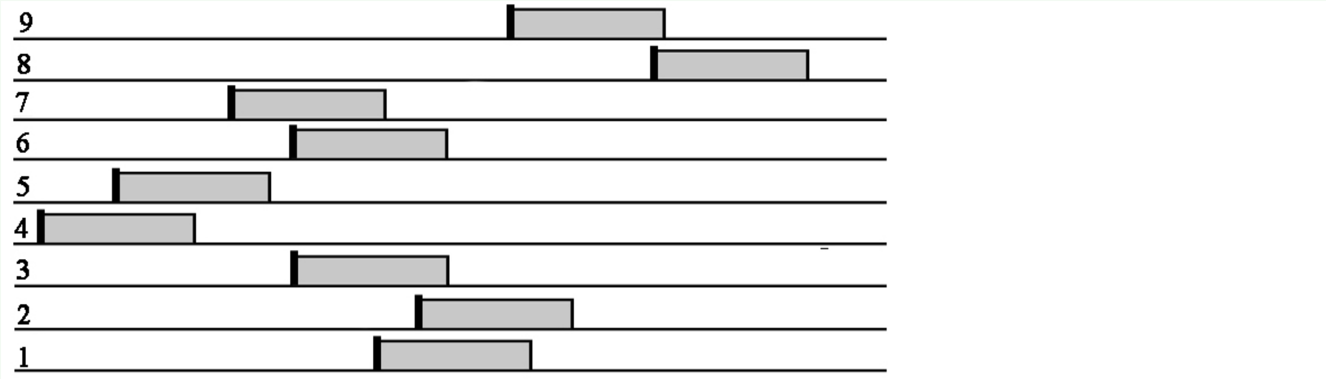


Рис.1. Входные данные

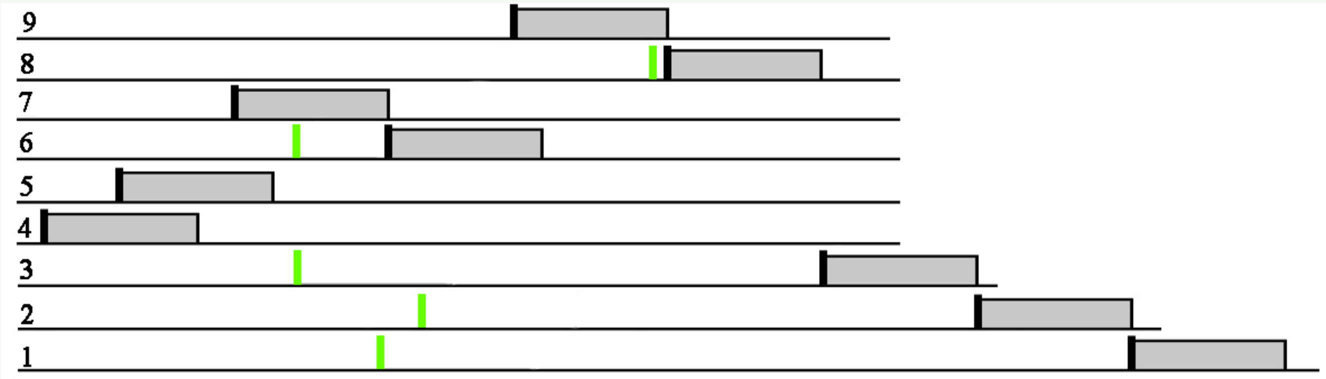


Рис.2. Входные данные после предобработки.

В результате предобработки все r_i будут различны.

Алгоритм формирования расписаний

Упорядочиваем работы в порядке неубывания весов $w_1 \leq \ldots \leq w_n$.

Алгоритм заключается в последовательном просмотре моментов r_j и C_j в порядке их возрастания, и ветвлении вариантов в случае альтернативы выбора работ.

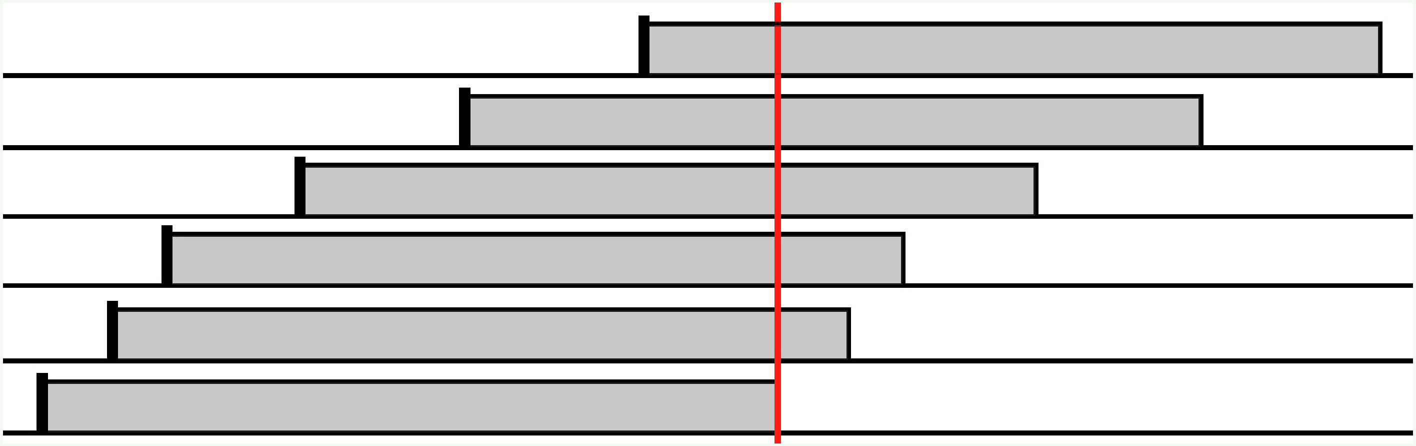
В момент r_j возможна альтернатива между текущей работой и работой j, а в момент C_j между последней прерванной работой и самой тяжелой из доступных работ.

В результате работы алгоритма будет построено не более 2^{2n-3} расписаний.

В процессе исследования задачи был рассмотрен пример с 5 работами, длительностью 5. Было сгенерировано 51 расписание, из которых только 25 могут быть оптимальными.

Полное перекрытие интервалов

Пусть все работы независимы и все интервалы $(r_i, r_i + p)$ попарно пересекаются:
$w_1 < w_2 < \ldots < w_n$ и $r_1 < r_2 < \ldots < r_n < p$.



Теорема о трех работах. Если все интервалы $[r_i, r_i + p]$ попарно пересекаются, то в каждой тройке работ, работа с наибольшим весом не может завершаться последней из них.

Алгоритм

Всего существует $\mathbf{2^{n-1}}$ расписаний, удовлетворяющих теореме в случае попарного пересечения всех интервалов $[\boldsymbol{r_i}, \boldsymbol{r_i} + \boldsymbol{p}]$. Причем самая тяжелая работа стоит либо на первом, либо на втором месте.

Шаг 1. Для первых работ имеем два расписания $\mathbf{1-2}$, $\mathbf{2-1}$ и выбираем оптимум $\boldsymbol{i_1^* - i_2^*}$.

Шаг 2. Формируем расписания для трех работ

$$\begin{array}{l} \boldsymbol{3 - i_1^* - i_2^*} \\ \mathbf{1 - 3 - 2} \\ \mathbf{2 - 3 - 1} \end{array}$$

и находим оптимум $\boldsymbol{j_1^* - j_2^* - j_3^*}$.

Шаг 3. Аналогично

$$\begin{array}{l} \boldsymbol{4 - j_1^* - j_2^* - j_3^*,} \\ \mathbf{1 - 4 - 3 - 2,} \\ \mathbf{2 - 4 - 3 - 1,} \\ \boldsymbol{3 - 4 - i_1^* - i_2^*.} \end{array}$$

Оптимум $\boldsymbol{k_1^* - k_2^* - k_3^* - k_4^*}$.

Алгоритм

Шаг 4.

$$\begin{aligned}&5 - k_1^* - k_2^* - k_3^* - k_4^*,\\ &1 - 5 - 4 - 3 - 2,\\ &2 - 5 - 4 - 3 - 1,\\ &3 - 5 - 4 - i_1^* - i_2^*,\\ &4 - 5 - j_1^* - j_2^* - j_3^*.\end{aligned}$$

Шаг k.

$k-$ "оптимум для $k-1$

$$\begin{aligned}&1 - k - (k-1) - (k-2) - \dots - 3 - 2,\\ &2 - k - (k-1) - (k-2) - \dots - 3 - 1,\\ &3 - k - (k-1) - (k-2) - \dots - 4 - i_1^* - i_2^*,\\ &4 - k - (k-1) - (k-2) - \dots - 5 - j_1^* - j_2^* - j_3^*.\\ &\dots\dots\end{aligned}$$

$k - (k-1)-$"оптимум для $k-2$".

Трудоемкость алгоритма $O(n^3)$ операций.

Единственное перекрытие

Рассмотрим подзадачу, когда интервалы $(r_i, r_i + p)$ попарно не пересекаются, кроме двух работ с наименьшими весами.

Переупорядочим работы по неубыванию весов: $w_1 < w_2 < \ldots < w_n$ и $r_1 < r_2 < \ldots < r_n$.

По условию $r_2 - r_1 < p$ и $r_{i+1} - r_i > p$, $i = 2, 3, \ldots, n - 1.$

$\varepsilon = p - (r_2 - r_1)$ – длительность фрагмента первой работы, не завершенной к моменту r_2,

Пусть $\alpha_i = r_{i+1} - r_i - p$.

Единственное перекрытие

Обозначим через $\varphi(i,j)$ оптимальное значение взвешенной суммы работ, завершившихся после момента r_i, если в этот момент из ранее начатых не завершена работа $j \in \{1,2,\ldots,i-1\}$. Такая работа только одна.

Нужно найти $\varphi(2,1)$. Начинаем с конца с момента r_n.
Для всех $j \in \{1,2,\ldots,n-1\}$

$$\varphi(n,j) = \min \begin{cases} (r_n + p)\cdot w_n + (r_n + p + \varepsilon - \sum_{i=2}^{n-1} \alpha_i)\cdot w_j, \\ (r_n + \varepsilon - \sum_{i=2}^{n-1} \alpha_i)\cdot w_j + (r_n + p + \varepsilon - \sum_{i=2}^{n-1} \alpha_i)\cdot w_n. \end{cases}$$

Далее для всех $i = n-1, n-2, \ldots, 2$
и всех $j = 1,2,\ldots,i-1$

$$\varphi(i,j) = \min \begin{cases} (r_i + p)\cdot w_i + \varphi(i+1,j), \\ (r_i + \varepsilon - \sum_{k=2}^{i-1} \alpha_k)\cdot w_j + \varphi(i+1,i). \end{cases}$$

С использованием алгоритма динамического программирования задача с единственным перекрытием может быть решена за $O(n^2)$ операций.

Заключение

1. Исследованы свойства задачи $\mathbf{1}|\boldsymbol{r_i, p_i = p, pmtn}|\sum \boldsymbol{w_i C_i}$.

2. Предложен и обоснован алгоритм построения конечного множества расписаний, в котором содержится оптимальное решение задачи для любого набора весов $\boldsymbol{w_i}$.

3. Выделены полиномиально разрешимые случаи рассматриваемой задачи.

Спасибо за внимание!