ИНФОРМАТИКА

2016 июль-сентябрь № 3

УДК 519.8

А.В. Кононов[1], И.Н. Лущакова[2]

ОПТИМАЛЬНОЕ ОБСЛУЖИВАНИЕ ТРЕБОВАНИЙ ДВУМЯ ПРИБОРАМИ ПРИ ЛИНЕЙНО УБЫВАЮЩИХ ФУНКЦИЯХ СТОИМОСТИ ВРЕМЕННЫХ ИНТЕРВАЛОВ

Рассматривается задача построения оптимального расписания обслуживания требований двумя параллельными приборами. В качестве целевой функции применяется линейная комбинация взвешенной суммы моментов завершения обслуживания требований и суммарной стоимости использования временных интервалов. В случае заданных для каждого из приборов линейно убывающих или постоянных последовательностей стоимостей временных интервалов предлагается точный псевдополиномиальный алгоритм динамического программирования.

Введение

В различных практических задачах качество расписания s обслуживания требований приборами может определяться целевой функцией вида $F_1(s) + F_2(s)$. Функция $F_1(s)$ – это какая-либо традиционная функция, рассматриваемая в теории расписаний, которая зависит от моментов завершения обслуживания требований, функция $F_2(s)$ – суммарная стоимость использования приборов в конкретные моменты времени. Например, стоимость заявки, выполненной во внеурочное время, будет выше. Стоимость доставки продуктов, заказанных в сетевом интернет-магазине, выше в вечернее время и выходные дни. Стоимость телевизионной рекламы зависит не только от длительности рекламного ролика, но и от времени суток трансляции ее в эфире. В облачных вычислениях разные вычислительные ресурсы (оборудование) имеют различную стоимость использования [1], причем пользователям могут предлагаться и различные схемы оплаты, зависящие от времени суток выполнения работ на удаленном сервере [2].

1. Постановка задачи и предварительные результаты

Разные задачи построения расписаний обслуживания множества требований одним прибором, минимизирующих функцию вида $F_1(s) + F_2(s)$, исследовались в работах [3, 4]. В настоящей статье обратимся к модели с параллельными приборами. Рассмотрим следующую задачу.

Множество $N = \{1, 2, ..., n\}$ требований необходимо обслужить на M параллельных приборах. Для каждого требования $i \in N$ заданы длительность его обслуживания $p_i > 0$, где p_i – целое число, и весовой коэффициент $w_i > 0$. Каждый прибор не может одновременно обслуживать более одного требования. Если в расписании s обслуживание требования i начинается на каком-либо приборе в момент времени $r_i(s)$, то оно протекает непрерывно на этом приборе и завершается в момент времени $C_i(s) = r_i(s) + p_i$. Будем рассматривать $F_1(s) = \sum_{i=1}^{n} w_i C_i(s)$.

Пусть горизонт планирования состоит из K интервалов единичной длины. Предположим, что $p_i \le K$ для каждого требования $i \in N$. Кроме того, предположим, что число K достаточно велико, так что все требования могут быть обслужены без прерываний на M приборах в пределах горизонта планирования. Интервал $(k-1, k]$ назовем k-м интервалом, $k = 1, 2, ..., K$. Введем следующие обозначения: π_k^L – стоимость использования k-го интервала времени прибором L, $1 \le L \le M$; $N_L(s)$ – множество требований, назначенных на прибор L в расписании s; $\overline{\pi}^L_{r_i(s), C_i(s)}$ – общая стоимость использования требованием $i \in N_L(s)$ временных интервалов для рассматриваемого горизонта планирования при расписании s, т. е.

$\bar{\pi}^L_{r_i(s),C_i(s)} = \bar{\pi}^L_{r_i(s),r_i(s)+p_i} = \pi^L_{r_i(s)+1} + \pi^L_{r_i(s)+2} + \ldots + \pi^L_{r_i(s)+p_i}$, где $r_i(s)$ – момент начала обслуживания требования i в расписании s. Тогда общая стоимость использования временных интервалов для рассматриваемого горизонта планирования при расписании s будет задана функцией $F_2(s) = \sum_{L=1}^{M} \sum_{i \in N_L(s)} \bar{\pi}^L_{r_i(s),C_i(s)}$.

Расписание s определяется указанием для каждого требования $i \in N$ прибора L, на котором оно будет обслужено, и момента $r_i(s)$ начала его обслуживания на приборе (либо, что то же самое, момента $C_i(s)$ окончания его обслуживания). Заметим, что при расписании s возможны простои приборов между интервалами времени, используемыми для обслуживания требований.

Необходимо построить расписание s^*, минимизирующее функцию

$$F_1(s) + F_2(s) = \sum_{i=1}^{n} w_i C_i(s) + \sum_{L=1}^{M} \sum_{i \in N_L(s)} \bar{\pi}^L_{r_i(s),C_i(s)} .$$

Обозначим эту задачу $PM / slotcost / \sum (w_i C_i + \bar{\pi}^L_{r_i,C_i})$. Аналогичная задача обслуживания требований одним прибором в работе [4] обозначена $1/ slotcost / \sum (w_i C_i + \bar{\pi}_{r_i,C_i})$.

Предположим, что стоимость использования временных интервалов не принимается во внимание. В этом случае задача $P2 / / \sum w_i C_i$ для двух параллельных приборов является NP-трудной [5]. Для задачи $PM / / \sum w_i C_i$ известен псевдополиномиальный алгоритм [6]. В монографии [7] отмечается, что задача $P / / \sum w_i C_i$ с произвольным числом приборов является NP-трудной в сильном смысле.

Для задачи $1/ slotcost / \sum (w_i C_i + \bar{\pi}_{r_i,C_i})$ в работах [3, 4] получены следующие результаты.

Теорема 1 [3]. *Задача* $1/ slotcost / \sum (C_i + \bar{\pi}_{r_i,C_i})$ *является NP-трудной в сильном смысле.*

Свойство 1 [3]. *Если последовательность* $\{\pi_k\}, k = 1, 2, \ldots, K,$ *неубывающая, то задача* $1/ slotcost / \sum (w_i C_i + \bar{\pi}_{r_i,C_i})$ *сводится к классической задаче* $1/ / \sum w_i C_i$.

Действительно, для минимизации регулярного критерия, каким является критерий $\sum w_i C_i$, требования должны быть обслужены как можно раньше, исключая простои на приборе. Этого же принципа в случае неубывающей последовательности $\{\pi_k\}, k = 1, 2, \ldots, K,$ следует придерживаться и с точки зрения минимизации общей стоимости использования временных интервалов. Поэтому приходим к выводу, что свойство, аналогичное свойству 1, может быть получено и для задачи $PM / slotcost / \sum (w_i C_i + \bar{\pi}^L_{r_i,C_i})$.

Теорема 2 [4]. *Если последовательность* $\{\pi_k\}, k = 1, 2, \ldots, K,$ *невозрастающая, то задача* $1/ slotcost / \sum (w_i C_i + \bar{\pi}_{r_i,C_i})$ *является NP-трудной в сильном смысле.*

Рассмотрим задачу $1/ slotcost / \sum (w_i C_i + \bar{\pi}_{r_i,C_i})$ при условии, что стоимость использования временных интервалов линейно убывает относительно индекса k, т. е. $\pi_k - \pi_{k+1} = \varepsilon,\ \varepsilon > 0,$ для $k = 1, 2, \ldots, K-1$.

Лемма 1 [4]. *Если последовательность* $\{\pi_k\}, k = 1,2,\ldots,K,$ *линейно убывающая, то для задачи* $1/ slotcost / \sum (w_i C_i + \bar{\pi}_{r_i,C_i})$ *существует оптимальное расписание, в котором требования назначены на обслуживание в порядке* $\frac{w_1}{p_1} \ge \frac{w_2}{p_2} \ge \ldots \ge \frac{w_n}{p_n}$.

Лемма 2 [4]. *Если* $\frac{w_i}{p_i} > \varepsilon$, *то в оптимальном расписании требование* i *должно быть обслужено как можно раньше. Если* $\frac{w_i}{p_i} < \varepsilon$, *то в оптимальном расписании требование* i *должно быть обслужено как можно позже. Если* $\frac{w_i}{p_i} = \varepsilon$, *то не имеет значения, в каком интервале времени обслуживается требование* i.

На основании лемм 1 и 2 для задачи $1/\,slotcost\,/\sum(w_iC_i + \overline{\pi}_{r_i,C_i})$ в случае линейно убывающей последовательности стоимостей временных интервалов в [4] предложен алгоритм сложности $O(n\log n)$ операций.

2. Метод динамического программирования для двух приборов

Пусть $M = 2$. Предположим, что приборы имеют постоянные или линейно убывающие последовательности $\{\pi_k^1\}$ и $\{\pi_k^2\}$ стоимостей временных интервалов: $\pi_k^1 - \pi_{k+1}^1 = \varepsilon_1$ и $\pi_k^2 - \pi_{k+1}^2 = \varepsilon_2$ для $k = 1, 2, \dots K-1$, причем $0 \le \varepsilon_1 \le \varepsilon_2$.

Разобьем множество N требований на три подмножества: $J_1 = \{i : \frac{w_i}{p_i} > \varepsilon_2\}$, $J_2 = \{i : \frac{w_i}{p_i} < \varepsilon_1\}$ и $J_3 = \{i : \varepsilon_1 \le \frac{w_i}{p_i} \le \varepsilon_2\}$. Для непустого множества J_j, $1 \le j \le 3$, полагаем $n_j = |J_j|$, $P_j = \sum_{i \in J_j} p_i$, в противном случае $n_j = 0, P_j = 0$.

Упорядочим и перенумеруем требования следующим образом: $J_1 = \{i_1,\ i_2, \dots, i_{n_1}\}$ и $\frac{w_{i_1}}{p_{i_1}} \ge \frac{w_{i_2}}{p_{i_2}} \ge \dots \ge \frac{w_{i_{n_1}}}{p_{i_{n_1}}} > \varepsilon_2$; $J_2 = \{k_1,\ k_2, \dots, k_{n_2}\}$ и $\frac{w_{k_1}}{p_{k_1}} \le \frac{w_{k_2}}{p_{k_2}} \le \dots \le \frac{w_{k_{n_2}}}{p_{k_{n_{12}}}} < \varepsilon_1$; $J_3 = \{l_1,\ l_2, \dots, l_{n_3}\}$ и $\varepsilon_2 \ge \frac{w_{l_1}}{p_{l_1}} \ge \frac{w_{l_2}}{p_{l_2}} \ge \dots \ge \frac{w_{l_{n_3}}}{p_{l_{n_3}}} \ge \varepsilon_1$.

Для решения задачи в случае $M = 2$ предлагается алгоритм динамического программирования *DPLinDec2*, состоящий из трех этапов.

На *этапе 1* алгоритм *DPLinDec2* назначает на обслуживание требования из множества J_1.

Пусть $J_1 \ne \varnothing$. Положим $\overline{p}_m^{(1)} = p_{i_1} + p_{i_2} + \dots + p_{i_m}$ для $m = 1, 2, \dots, n_1$; $P_1 = \overline{p}_{n_1}^{(1)}$. Рассмотрим подзадачу составления расписания для требований $i_1,\ i_2, \dots, i_m$ из множества J_1. Предположим, что приборы начинают обслуживание требований в момент времени 0. Пусть $f_1(x,m)$ – наименьшее значение целевой функции при условии, что прибор 1 завершает обслуживание в момент времени x, $0 \le x \le \min\{P_1,\ K\}$. Тогда прибор 2 должен завершить обслуживание в момент времени $\overline{p}_m^{(1)} - x$. Функция $f_1(x,m)$ вычисляется рекурсивно с учетом того, что требование i_m обслуживается последним либо на приборе 1, либо на приборе 2:

$$f_1(x,m) = \min\{\, f_1(x - p_{i_m}, m-1) + w_{i_m} x + \overline{\pi}^1_{x-p_{i_m},x};$$

$$f_1(x,m-1) + w_{i_m}(\overline{p}_m^{(1)} - x) + \overline{\pi}^2_{\overline{p}_{m-1}^{(1)}-x,\,\overline{p}_m^{(1)}-x}\} \ .$$

Зададим начальные условия:

$$f_1(x,1)=\begin{cases} w_{i_1}p_{i_1}+\overline{\pi}^2_{0,p_{i_1}}, \text{ если } x=0, \\ w_{i_1}p_{i_1}+\overline{\pi}^1_{0,p_{i_1}}, \text{ если } x=p_{i_1}, \\ +\infty \text{ в противном случае.} \end{cases}$$

Граничные условия определим следующим образом:

$$f_1(x,m)=+\infty \text{ для } x<0 \text{, либо } x>\min\{K,\overline{p}^{(1)}_m\} \text{, либо } \overline{p}^{(1)}_m - x > K .$$

Для каждого значения $x=0,\ 1,...,\min\{P_1,K\}$ будем хранить значение функции $f_1(x,n_1)$ и соответствующее ему расписание $\sigma'(x)$.

Если $J_1=\varnothing$, то полагаем $x=0,\ f_1(0,0)=0$ и $\sigma'(0)$ – пустое расписание.

Этап 1 алгоритма может быть выполнен за $O(n_1P_1)$ операций.

На *этапе 2* алгоритм *DPLinDec2* назначает на обслуживание требования из множества J_2.

Пусть $J_2\neq\varnothing$. Положим $\overline{p}^{(2)}_m=p_{k_1}+p_{k_2}+...+p_{k_m}$ для $m=1,2,...,n_2$; $P_2=\overline{p}^{(2)}_{n_2}$. Рассмотрим подзадачу составления расписания для требований $k_1,k_2,...,k_m$ из множества J_2. Пусть $f_2(y,m)$ – наименьшее значение целевой функции при условии, что прибор 2 начинает обслуживание некоторых из этих требований в момент времени y. Тогда прибор 1 начинает обслуживание оставшихся требований в момент времени $2K-y-\overline{p}^{(2)}_m$. Оба прибора завершают обслуживание требований в момент времени K. Функция $f_2(y,m)$ вычисляется рекурсивно с учетом того, что требование k_m обслуживается первым либо на приборе 2, либо на приборе 1:

$$f_2(y,m)=\min\{ f_2(y+p_{k_m},m-1)+w_{k_m}(y+p_{k_m})+\overline{\pi}^2_{y,y+p_{k_m}},$$

$$f_2(y,m-1)+w_{k_m}(2K-y-\overline{p}^{(2)}_{m-1})+\overline{\pi}^1_{2K-y-\overline{p}^{(2)}_m,2K-y-\overline{p}^{(2)}_{m-1}}\}.$$

Зададим начальные условия:

$$f_2(y,1)=\begin{cases} w_{k_1}K+\overline{\pi}^2_{K-p_{k_1},K}, \text{ если } y=K-p_{k_1}, \\ w_{k_1}K+\overline{\pi}^1_{K-p_{k_1},K}, \text{ если } y=K, \\ +\infty \text{ в противном случае.} \end{cases}$$

Граничные условия определим следующим образом:

$$f_2(y,m)=+\infty \text{ для } y>K \text{, либо } y<\max\{0,K-\overline{p}^{(2)}_m\} \text{, либо } 2K-y-\overline{p}^{(2)}_m<0 .$$

Для каждого значения $y=K,\ K-1,...,\max\{0,K-P_2\}$ будем хранить значение функции $f_2(y,n_2)$ и соответствующее ему расписание $\sigma''(y)$.

Если $J_2=\varnothing$, то полагаем $y=K,\ f_2(K,0)=0$ и $\sigma''(K)$ – пустое расписание.

Этап 2 алгоритма может быть выполнен за $O(n_2P_2)$ операций.

Этап 3 алгоритма выполняется для всех возможных значений переменных x и y, $0\le x\le\min\{P_1,K\},\max\{0,K-P_2\}\le y\le K$.

Рассмотрим конкретные значения переменных x и y из указанных диапазонов. Сначала алгоритм проверяет существование допустимого расписания $\sigma(x,y)$, в котором требования множеств J_1 и J_2 выполняются в тех же интервалах времени и на тех же приборах, что и в расписаниях $\sigma'(x)$ и $\sigma''(y)$. Расписание $\sigma(x,y)$ будет допустимым, если на каждом приборе в любой момент времени обслуживается не более одного требования. Чтобы убедиться, что расписание $\sigma(x,y)$ допустимо, достаточно проверить выполнение неравенства $P_1 - y \le x \le 2K - y - P_2$. Если расписание $\sigma(x,y)$ допустимо, алгоритм вставляет в него требования множества J_3.

Пусть $J_3 \ne \varnothing$. Положим $\bar{p}_m^{(3)} = p_{l_1} + p_{l_2} + \ldots + p_{l_m}$ для $m = 1, 2, \ldots, n_3$; $P_3 = \bar{p}_{n_3}^{(3)}$. Рассмотрим подзадачу составления расписания для требований $l_1,\ l_2, \ldots, l_m$ из множества J_3. Известно, что прибор 1 начинает обслуживание некоторых из этих требований в момент времени x и заканчивает в момент времени z, в то время как прибор 2 начинает обслуживание оставшихся требований в момент времени u. Пусть $f_3(x,y,z,u,m)$ – наименьшее значение целевой функции при условии, что прибор 1 завершает обслуживание части требований множества J_3 в момент времени z, а прибор 2 завершает обслуживание остальных требований множества J_3 в момент времени $u + \bar{p}_m^{(3)} - (z - x)$. Функция $f_3(x,y,z,u,m)$ вычисляется рекурсивно с учетом того, что l_m – это последнее требование из множества J_3, обслуженное либо на приборе 1, либо на приборе 2:

$$f_3(x,y,z,u,m) = \min\{ f_3(x,y,z - p_{l_m}, u, m-1) + w_{l_m} z + \bar{\pi}^1_{z - p_{l_m}, z};$$

$$f_3(x,y,z,u,m-1) + w_{l_m}(u + \bar{p}_m^{(3)} - (z - x)) + \bar{\pi}^2_{u + \bar{p}_{m-1}^{(3)} - (z-x),\, u + \bar{p}_m^{(3)} - (z-x)}\}.$$

Зададим начальные условия:

$$f_3(x,y,z,u,1) = \begin{cases} w_{l_1}(u + p_{l_1}) + \bar{\pi}^2_{u, u+p_{l_1}}, & \text{если } z = x \text{ и } u + p_{l_1} \le y, \\ w_{l_1}(x + p_{l_1}) + \bar{\pi}^1_{x, x+p_{l_1}}, & \text{если } z = x + p_{l_1} \le 2K - y - P_2, \\ +\infty \text{ в противном случае.} \end{cases}$$

Граничные условия определим следующим образом:

$$f_3(x,y,z,u,m) = +\infty \text{ для } z < x \text{, либо } z > \min\{2K - y - P_2, x + \bar{p}_m^{(3)}\},$$
$$\text{либо } u + \bar{p}_m^{(3)} - (z - x) > y.$$

Таким образом, для допустимого расписания $\sigma(x,y)$ алгоритм *DPLinDec2* строит семейство расписаний $\tilde{\sigma}(x,y,z,u)$, $x \le z \le \min\{x + P_3, 2K - y - P_2\}$, $\max\{y - P_3, P_1 - x\} \le u \le y$. Каждому расписанию $\tilde{\sigma}(x,y,z,u)$ соответствует значение целевой функции $f_3(x,y,z,u,n_3)$.

Если $J_3 = \varnothing$, полагаем $z = x,\ u = y,\ f_3(x,y,x,y,0) = 0$. В этом случае соответствующее расписание $\tilde{\sigma}(x,y,x,y)$ совпадает с расписанием $\sigma(x,y)$.

Отметим, что согласно лемме 2 в оптимальном расписании все требования множества J_3, которые обслуживаются на приборе 2, должны быть обслужены как можно позже, т. е. прибор 2 должен начать их обслуживание в момент времени $u = y - (P_3 - (z - x))$ и завершить в момент времени y. Поэтому будем отбрасывать все расписания $\tilde{\sigma}(x,y,z,u)$, у которых $u \ne y - (P_3 - (z - x))$.

На каждом шаге этапа 3, определяемом конкретными значениями переменных x, y, z, будем хранить только текущее расписание $\tilde{\sigma}(x,y,z,y - (P_3 - (z - x)))$ с минимальным значением целевой функции $f_1(x,n_1) + f_2(y,n_2) + f_3(x,y,z,y - (P_3 - (z - x)), n_3)$, которое находится по всем просмотренным к данному шагу значениям переменных x, y, z. По окончании этапа 3 алгоритм находит оптимальное расписание обслуживания требований множества N. Этап 3 алгоритма может быть выполнен за $O(n_3 P_1 P_2 P_3^2)$ операций.

Заметим, что если некоторые из подмножеств J_1, J_2, J_3 требований являются пустыми, то это отразится на общей вычислительной сложности алгоритма *DPLinDec2* (таблица).

Общая вычислительная сложность алгоритма *DPLinDec2*

Подмножества требований	Вычислительная сложность
$J_1 \neq \emptyset, J_2 \neq \emptyset, J_3 \neq \emptyset$	$O(n_1 P_1 + n_2 P_2 + n_3 P_1 P_2 P_3^2)$
$J_1 \neq \emptyset, J_2 \neq \emptyset, J_3 = \emptyset$	$O(n_1 P_1 + n_2 P_2 + P_1 P_2)$
$J_1 \neq \emptyset, J_2 = \emptyset, J_3 \neq \emptyset$	$O(n_1 P_1 + n_3 P_1 P_3^2)$
$J_1 = \emptyset, J_2 \neq \emptyset, J_3 \neq \emptyset$	$O(n_2 P_2 + n_3 P_2 P_3^2)$
$J_1 = \emptyset, J_2 = \emptyset, J_3 \neq \emptyset$	$O(n_3 P_3^2)$
$J_1 = \emptyset, J_2 \neq \emptyset, J_3 = \emptyset$	$O(n_2 P_2)$
$J_1 \neq \emptyset, J_2 = \emptyset, J_3 = \emptyset$	$O(n_1 P_1)$

На всех этапах алгоритма объем памяти, требуемый для поддержания его работы, не превышает $O(nK)$.

Заключение

В статье предложен точный псевдополиномиальный алгоритм динамического программирования для построения оптимального расписания обслуживания требований двумя параллельными приборами при линейно убывающих или постоянных функциях стоимости временных интервалов.

Отметим, что если у всех требований множества N длительности обслуживания единичные, то для непустого множества J_j, $1 \leq j \leq 3$, имеем $P_j = n_j$. В этом случае алгоритм *DPLinDec2* превращается в полиномиальный алгоритм с общей вычислительной сложностью $O(n_1^2 + n_2^2 + n_1 n_2 n_3^3)$.

Работа выполнена при частичной финансовой поддержке проекта № Ф15СО-043 БРФФИ.

Список литературы

1. Zhao, G. Cost-aware scheduling algorithm based on PSO in Cloud Computing Environment / G. Zhao // Intern. J. of Grid and Distributed Computing. – 2014. – Vol. 7, no. 1. – P. 33–42.

2. Amazon EC2 Pricing Options [Electronic resource]. – 2016. – Mode of access : https://aws.amazon.com/ec2/pricing. – Date of access : 10.04.2016.

3. Wan, G. Scheduling with Variable Time Slot Costs / G. Wan, X. Qi // Naval Research Logistics. – 2010. – Vol. 57, no. 2. – P. 159–171.

4. Zhao, Y. On scheduling with non-increasing time slot cost to minimize total weighted completion time / Y. Zhao, X. Qi, M. Li [Electronic resource]. – 2016. – Mode of access : http://link.springer.com/article/10.1007/s10951-015-0462-9#/page-1. – Date of access : 10.04.2016.

5. Bruno, J. Scheduling independent tasks to reduce mean finishing time/ J. Bruno, E.G. Coffman, Jr., R. Sethi // Communications of the ACM. – 1974. – Vol. 17. – P. 382–387.

6. Sequencing and Scheduling: Algorithms and Complexity / E.L. Lawler [et al.] // Handbooks in Operations Research and Management Science. – North-Holland, Amsterdam, 1993. – Vol. 4. – P. 445–522.

7. Brucker, P. Scheduling Algorithms / P. Brucker. – Springer, 2004. – 367 p.

Поступила 06.06.2016

[1]*Институт математики им. С. Л. Соболева СО РАН,*
Новосибирск, пр. академика Коптюга, 4
e-mail: alvenko@math.nsc.ru

[2]*Белорусский государственный университет*
информатики и радиоэлектроники,
Минск, ул. П. Бровки, 6
e-mail: IrinaLushchakova@yandex.ru

A.V. Kononov, I.N. Lushchakova

SCHEDULING JOBS ON TWO PARALLEL MACHINES WITH LINEAR DECREASING TIME SLOT COSTS

We consider a scheduling problem with two parallel machines to minimize the sum of total weighted completion time and total machine time slot costs. In the case of the constant or linear decreasing sequences of time slotcosts we suggest an exact pseudopolynomial DP algorithm.