УДК 519.87

РЕШЕНИЕ FLOW SHOP ЗАДАЧИ ТЕОРИИ РАСПИСАНИЙ В РЕКУРСИВНОМ ПРЕДСТАВЛЕНИИ

Б.В. Куприянов
Институт проблем управления им. В.А. Трапезникова РАН
Россия, 117997, Москва, Профсоюзная ул., 65
E-mail: kuprianovb@mail.ru

А.А. Рощин
Институт проблем управления им. В.А. Трапезникова РАН
Россия, 117997, Москва, Профсоюзная ул., 65
E-mail: rochinaa@mail.ru

Ключевые слова: permutation flow shop, branch-and-bound.

Аннотация: Рассматривается эквивалентность постановки flow shop задач теории расписаний в существующем представлении и представлении в виде специальной рекурсивной функции и рекурсивных вычислений. Для решения задачи вычисления оптимального расписания рассматривается метод ветвей и границ. Приводится определение варианта нижней границы и метода ветвления.

1. Введение

Количество публикаций, посвященных flow shop задачам (FSP) теории расписаний (ТР) значительно. Большинство современных работ основано на различных эвристиках и метаэвристиках, например [1]. В работе рассматривается новый подход через определение рекурсивных функций для описания временны́х характеристик FSP. В работе вводится понятие временно́го интервала, определяющего множество времен возможного завершения выполнения машиной заказа. Доказывается эквивалентность представления FSP в существующем и рекурсивном виде. Эквивалентность понимается как равенство множеств допустимых расписаний для каждой пары (заказ, машина). Осуществляется переход от интервальных рекурсивных функций к скалярным. Показано, что скалярная функция вычисляет одно из оптимальных значений. В качестве метода вычисления оптимального расписания рассматривается метод ветвей и границ (B&B). Приводится определение нижней границы и метода ветвления, основанный на специальном генераторе перестановок. Описаны алгоритм перестановок и реализация метода B&B.

Определение 1. *Пусть* $N = \{0, 1, 2, 3, \ldots, \hat{N}\}$ – *конечное множество натуральных чисел, от нуля до* $\hat{N}$. *Под интервалом* $[a, b]$, $a \leqslant b$ *понимается* *замкнутое ограниченное подмножество* N *вида* $[a,b] = \{x|\ x \in N\ \wedge\ a \leqslant x \leqslant b\}$.

Множество всех интервалов обозначим через $\mathbf{N}$. Пусть $*$ некоторая операция и $A, B \in \mathbf{N}$. В этом случае $A * B = \{a * b|\ a \in A,\ b \in B\}$.

2. Эквивалентность традиционной и рекурсивной постановки FSP

В общеим случае рекурсивная функция оределяется отображением $\mathcal{T} : J \times \mathcal{M} \to \mathbf{N}$, где
$\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ – множество заданий;
$\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$ – множество машин (операций).
$p_{j,k}$ – время обработки задания j на машине M_k.
Вычисляемой величиной является время завершения выполнения j-го заказа k-й машиной (makespan). Будем обозначать интервальную рекурсивную функцию как $\mathcal{T}(j,k)$, а скалярную как $C(j,k)$. Рассмотрим проблему эквивалентности традиционной формулировки FSP [3] и ее рекурсивного представления. Эквивалентность формулировок устанавливается через эквивалентность интервалов допустимых времени́ых значений завершения выполнения для каждой пары (j,k).

Рассмотрим подробнее (см. рис. 1a) FSP. Граф предшествования такой задачи представляет собой простую цепь, в которой вершины являются машинами, а дуги задают отношения предшествования машин. Если вершину графа отождествить с парой (j,k), то можно определить отношения предшествования между заказами $J \times J$ и между машинами $K \times K$. Такой развернутый граф, описывающий отношения

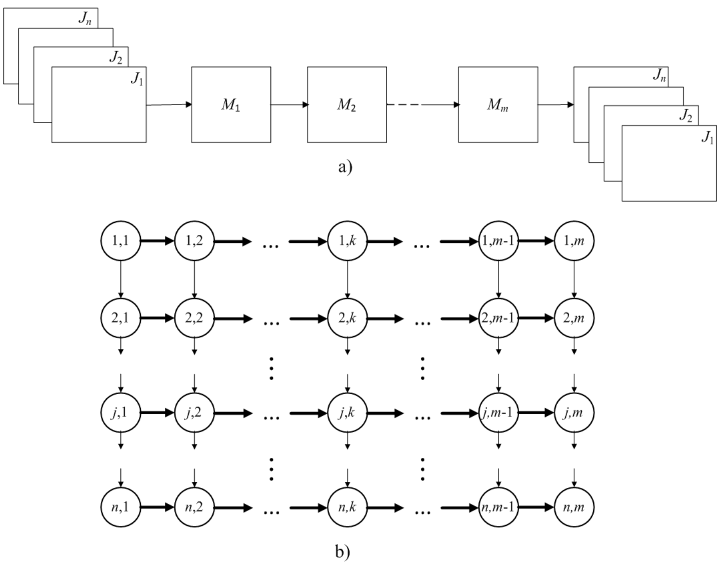

Рис. 1. Пример графа FSP.

предшествования для каждой пары (j,k) представлен на рис. 1b. Вертикальные дуги определяют отношения предшествования между заказами для конкретной машины. Горизонтальные дуги между машинами при фиксированном номере заказа. Данный граф является ациклическим и имеет одну начальную и одну конечную вершины. Обозначим $T_{j,k} = [\underline{t}_{j,k}, \overline{t}_{j,k}]$ как интервальное время, где $\underline{t}_{j,k}$ минимальное время, а $\overline{t}_{jk}$

как максимальное время завершения выполнения j-го заказа k-й машиной. В этом случае будут выполнятся следующие временńые отношения

$$(1)\qquad p_{1,1} \leqslant \underline{t}_{1,1},\ \ \underline{t}_{1,1} \leqslant \underline{t}_{1,2} - p_{1,2},\ \ldots,\ \underline{t}_{1,m-1} \leqslant \underline{t}_{1,m} - p_{1,m}.$$

$$(2)\qquad \underline{t}_{1,1} \leqslant \underline{t}_{2,1} - p_{2,1},\ \ \max(\underline{t}_{1,2}, \underline{t}_{2,1}) \leqslant \underline{t}_{2,2} - p_{2,2},\ \ldots,$$
$$\max(\underline{t}_{1,m}, \underline{t}_{2,m-1}) \leqslant \underline{t}_{2,m} - p_{2,m}.$$
$$\vdots$$
$$(3)\qquad \underline{t}_{n-1,1} \leqslant \underline{t}_{n,1} - p_{n,1},\ \ \max(\underline{t}_{n-1,2}, \underline{t}_{n,1}) \leqslant \underline{t}_{n,2} - p_{n,2},\ \ldots,$$
$$\max(\underline{t}_{n-1,m}, \underline{t}_{n,m-1}) \leqslant \underline{t}_{n,m} - p_{n,m}.$$

Можно показать, что все допустимые множества временńых значений являются единственным способом определяемыми и максимальными. Следовательно, при поиске оптимального решения не будет пропущено ни одно оптимальное решение.

Интервальные вычисления можно представить рекурсивной функцией

$$(4)\qquad \mathcal{T}(j,k) = \begin{cases} [p_{1,1}, Tmax] \cap T^0_{1,1}, & \text{при } j = 1 \text{ и } k = 1; \\ \mathcal{T}(1, k-1) + p_{1,k}, & \text{при } j = 1 \text{ и } k > 1; \\ \mathcal{T}(j-1, 1) + p_{j,1}, & \text{при } j > 1 \text{ и } k = 1; \\ \max(\mathcal{T}(j-1,k), \mathcal{T}(j,k-1)) + p_{j,k}, & \text{при } j > 1 \text{ и } k > 1. \end{cases}$$

Доказана следующая теорема.

Теорема 1. *Пусть для исходной и рекурсивной FSP будут выполнены следующие условия:*

- *для обоих задач задан один граф (см. рис. 1б), определяющий отношения предшествования между машинами и отношения предшествования между заказами;*
- *время выполнения k-й машиной j-го заказа равно $p_{j,k}$;*
- *задана одна матрица начальных временных интервалов*

$$\|T^0_{j,k}\| = \|[\underline{t}^0_{j,k}, \overline{t}^0_{j,k}]\|.$$

В этом случае справедливо следующее утверждение: множество интервалов $D_{j,k}$ – допустимых временńых значений для исходной FSP совпадает с множеством интервалов $D'_{j,k}$ рекурсивной функции, т.е. $D_{j,k} = D'_{j,k}$ для $1 \leqslant j \leqslant n$, $1 \leqslant k \leqslant m$. Иначе говоря, множества допустимых расписаний для каждой пары: (задание, машина) совпадают.

Основной FSP является нахождение порядка выполнения заказов, при котором makespan принимает минимальное значение. Подавляющее большинство таких задач, определяются как задачи с перестановкой заданий Permutation FSP (PFSP). Для эффективного решения задачи вычисления оптимального расписания необходимо от интервальных рекурсивных функций (4) перейти к скалярным рекурсивным функциям вида:

$$(5)\qquad C(j,k) = \begin{cases} p_{1,1} & \text{при } j = 1 \text{ и } k = 1; \quad a) \\ C(1, k-1) + p_{1,k} & \text{при } j = 1 \text{ и } k > 1; \quad b) \\ C(j-1, 1) + p_{j,1} & \text{при } j > 1 \text{ и } k = 1; \quad c) \\ \max(C(j-1,k), C(j,k-1)) + p_{j,k} & \text{при } j > 1 \text{ и } k > 1. \quad d) \end{cases}$$

Для того, чтобы от интервальных вычислений перейти к скалярным сформулируем следующую теорему.

Теорема 2. *Пусть для PFSP заданы m машин и некоторая последовательность из n заказов. В этом случае справедливо следующее равенство.*

$$\underline{\mathcal{T}}(n,m) = C(n,m). \tag{6}$$

Из данной теоремы вытекает важное следствие. Пусть Π множество всех перестановок из n заданий и $\pi \in \Pi$ перестановка, соответствующая оптимальному расписанию. Для PFSP множество оптимальных расписаний определяется любым из двух выражений равенства

$$\min_{\pi\in\Pi} \underline{\mathcal{T}}(n,m) = \min_{\pi\in\Pi} C(n,m).$$

При этом равенство для обоих вариантов справедливо для одного и того же множества порядков следования заказов. Данное следствие о том, что при вычислении оптимального расписания для PFSP от интервальных вычислений можно перейти к рекурсивной функции $C(j,k)$ (5).

Обозначим перестановкой $\pi = (j_1, j_2, \ldots, j_n)$ порядок номеров заказов. Ранее рекурсивная функция была определена для фиксированного порядка выполнения заказов. Внесем в определение функции перестановку π, чтобы можно было менять порядок номеров заказов

$$C(j,k) = \begin{cases} p_{\pi(1),1}, & j = 1 \text{ и } k = 1; \quad a) \\ C(1,k-1) + p_{\pi(1),k}, & j = 1 \text{ и } k > 1; \quad b) \\ C(j-1,1) + p_{\pi(j),1}, & j > 1 \text{ и } k = 1; \quad c) \\ \max(C(j-1,k), C(j,k-1)) + p_{\pi(j),k}, & j > 1 \text{ и } k > 1. \quad d) \end{cases} \tag{7}$$

В этой функции j определяет порядковый номер заказа.

3. Реализация метода ветвей и границ для PFSP

Применению метода B&B для решения PFSP посвящено большое количество работ. В [4] представлен обзор наиболее важных работ по алгоритмам B&B для PFSP. Основными вопросами разработки таких алгоритмов являются выборы методов вычисления границ LB и алгоритмов ветвления. Опишем в варианте, когда используется только нижняя граница LB1.

Оценка LB1. Для некоторой пары (j,k) LB можно определить следующим образом

$$LB(j,k) = C(j,k) + \sum_{i=k+1}^{m} p_{j,i} + \sum_{i=j+1}^{n} p_{i,m}. \tag{8}$$

Пусть $\mathcal{P}(j_1, j_2, \ldots, j_n)$ – множество всех перестановок из n заказов. Сформулируем точный смысл границы $LB(j,k)$. В данном случае она означает, что при фиксированном порядке части заказов $1, 2, \ldots, j$ время выполнения всех заказов не может быть меньше $LB(j,k)$ при любом порядке поступления оставшихся заказов.

Пусть Π некоторое множество перестановок из n заказов, для которых было вычислено значение $C(n,m)$. Тогда нижней границей $LB1$ для рассмотренных последовательностей заказов будет

$$LB1 = \min_{\pi\in\Pi} C(n,m).$$

Таким образом, если для текущей пары (j,k) выполняется неравенство $LB(j,k) \geqslant LB1$, то данная последовательность заказов $1,2,\ldots,j$ исключается из рассмотрения, и соответствующая ветвь перестановок обрезается. Следующей будет последовательность $1,2,\ldots,j-1,\pi'$, где

$$\pi' \in \mathcal{P}(j,j+1,\ldots,n) \text{ и } \pi' \neq (j,j+1,\ldots,n).$$

Ветвление. В первую очередь, ветвление необходимо согласовать с обходом, определяемым рекурсивной функцией. Если

$$LB(j,k) \geqslant LB1$$

необходимо изменить порядок следования заказов таким образом, чтобы минимизировать потерю информации при вычислении функций $C(1,1),\ldots,C(j,k)$, Пусть $x\mathcal{P}(\pi) = \{x\pi_1, x\pi_2, \ldots, x\pi_{n!}\}$. В данном случае, алгоритм генерации перестановок должен вычислять их в такой оследовательности, чтобы они удовлетворяли следующему рекурсивному свойству

$$\mathcal{P}(j_1,j_2,\ldots,j_n) = \bigcup_{i=1}^{n} j_i\mathcal{P}(j_1,j_2,\ldots,j_{i-1},j_{i+1},\ldots,j_n). \tag{9}$$

Данное свойство после вычисления функции $C(j,k)$ позволяет исследовать все перестановки «хвоста» очереди заказов на планирование $\mathcal{P}(j+1,j+2,\ldots,n)$ и только после этого произвести откат назад.

В работе представлены алгоритм генерации множества перестановок в последовательности, удовлетворяющей свойству 9 и алгоритм, включающий в описанный перебор метод B&B.

4. Заключение

В статье рассмотрен рекурсивный метод вычисления расписаний для FSP в существующей постановке [3]. В данном случае основное достоинство и перспективность рекурсивного представления заключается в легкой возможности модификации задачи дополнительными ограничениями и целевыми показателями, т.к. они инкапсулируются в тело рекурсивной функции. При этом дополнительные ограничения приводят к дополнительным сокращениям комбинаторных вариантов.

Список литературы

1. Riahi V., Kazemi M. A new hybrid ant colony algorithm for scheduling of no-wait flow shop. Operational Research, vol. 18, no. 1, pp. 55–74, 2018.
2. Калмыков С.А., Шокин Ю.И., Юлдашев З.Х. Методы интервального анализа. Новосибирск: Наука, 1986.
3. Лазарев А.А., Гафаров Е.Р. Теория расписаний. Задачи и алгоритмы. М.: Изд-во МГУ, 2011. 223с.
4. Gmys J., Mezmaz M., Melab N., Tuyttens D. A computationally efficient Branch-and-Bound algorithm for the permutation flow-shop scheduling problem // European Journal of Operational Research. 2020. Vol. 284, No. 3. P. 814-833.