УДК 519.8

Распределение множества заданий и определение оптимальных очередностей их выполнения на параллельных машинах методами динамического программирования

Ю.А. Зак

Аннотация

Предлагаются математические модели и изучаются свойства задач построения выполнения расписаний на параллельных и различных по техническим характеристикам машинах в условиях учета потерь на переналадки машин, ограничений на времена выполнения заданий и допустимые сроки работы машин. Конструируются операторы исключения из рассмотрения подмножеств расписаний, не содержащих допустимых и неоптимальных решений. На основе установленных свойств рассмотрены алгоритмы решения рассматриваемых задач методами динамического программирования.

Ключевые слова: *расписания на параллельных машинах, потери на переналадки, ограничения на сроки выполнения заданий и времена работы машин, динамическое программирование.*

1. Введение

Задачи распределения множества подлежащих выполнению заданий по рабочим станциям (машинам) и определения оптимальных последовательностей их выполнения на каждой машине в условиях потерь времени на переналадки имеют большое количество приложений в календарном планировании производства, маршрутизации перевозок, организации обслуживания и выполнения ремонтных работ, организации вычислительного процесса и т.п. (см. [1, 4, 17]).

Методам решения данного класса задач уделялось значительное место в монографиях и периодической литературе [1-8]. Известны подходы, связанные с построением линейных и нелинейных целочисленных моделей и с использованием методов математического программирования. Алгоритмы решения задач данного класса достаточно большой размерности, представляющих практический интерес, требуют больших объёмов вычислений [5, 7].

Наиболее широкое распространение получили приближенные методы решения задач данного класса с использованием генераторов случайных расписаний [4, 6, 8, 10], использующих различные правила предпочтения [4], эвристические подходы [7, 12], а также генетические алгоритмы и эволюциионные стратегии [8, 11, 13]. Алгоритмы решения данной задачи без учета ограничений на директивные сроки выполнения заданий методами построения кратчайших допустимых путей на графах приведены в работе автора [14-17]. Указанные подходы позволили в ряде случаев находить эффективные расписания выполнения заданий для многих практических приложений. Однако наличие жестких ограничений на директивные сроки выполнения заданий в ряде случаев затрудняет процесс генерирования допустимых расписаний и существенно увеличивает затраты на поиск. Кроме того, отсутствие нижних оценок значения критерия оптимальности для построенного расписания не позволяет объективно оценить эффективность полученного решения. В монографии автора [17] исследованы свойства допустимых и оптимальных планов сформулированной задачи и предложены алгоритмы ее решения задачи методами ветвей и границ.

В данной работе изучаются свойства задач данного класса, на основе которых конструируются операторы исключения из рассмотрения подмножеств расписаний, не содержащих допустимых и неоптимальных решений. На основе установленных свойств предлагаются алгоритмы решения рассматриваемых задач методами динамического программирования алгоритмами построения допустимых и экстремальных путей на графе.

2. Постановка и математическая модель задачи.

На K, $k=1,...,K$, рабочих станциях (машинах) должны быть выполнены N различных работ (заданий), $i,j=1,...,N$. Каждое из заданий должно выполняться только на одной машине и без разрывов времени в процессе его выполнения. Пусть заданы:

- директивные сроки допустимого времени наиболее раннего срока начала b_i и наиболее позднего срока завершения каждого из заданий B_i, $i=1,...,N$;

- матрица времён выполнения каждого из заданий на всех машинах $\bar{t}^k = (t_1^k, t_2^k, ..., t_i^k, ..., t_N^k)$, $k = 1,..., K$;

- h^k и H^k - наиболее ранние допустимые сроки начала и завершения выполнения работ на каждой k-й машине;

- $\bar{\tau}^k = \left|\tau_{ij}^k\right|$, $i, j = 0,1,..., N$, $k = 1,..., K$, - матрицы времён потерь времени на переналадки при переходе k-й машины от выполнения одного задания к другому. На пересечении i-й строки и j-го столбца этих матриц стоят потери времени на переналадки k-й машины при переходе после выполнения i-го задания к j-му. В 0-й строке каждой матрицы $\bar{\tau}^k$ заданы времена настройки машины при переходе ее из состояния, в котором она находится в момент начала выполнения расписания работ, в режим выполнения i-го задания. Элементы 0-го столбца определяют затраты времени на переход k-й машины после завершения выполнения j-го задания в режим простоя (или возвращения машины из j-го пункта на базу).

Необходимо найти распределение всего множества заданий по машинам, а также определить последовательности всех назначенных на каждой машине заданий, обеспечивающие выполнение всех ограничений на установленные сроки их выполнения T_i, и минимизировать время завершения всего комплекса работ (критерий оптимальности F_1). В качестве других критериев оптимальности могут быть выбраны F_2 - минимальное средневзвешенное время работы машин, необходимое для выполнения всего комплекса работ, а также F_3 - наиболее равномерная загрузка всех работающих машин.

Частным случаем рассматриваемых задач являются задачи построения последовательностей выполнения заданий N на K параллельных и различных по производительности машинах в условиях, когда потерями времени на переналадки машин можно пренебречь, т.е. $\tau_{ij}^k = 0$, $i, j = 0,1,..., N$; $k = 1,..., K$, а случаи, когда времена выполнения каждого из заданий и потерь на переналадки для всех машин одинаковы, т.е.

$t_j^k = t_j$ и $\tau_{ij}^k = a_{ij}$, $i, j = 0,1,...,N$; $k = 1,...,K$.

Построим матрицы суммарных затрат времени на выполнения j-го задания и переналадок для каждой k-й машины при переходе от выполнения i-го к j-му заданию, $A^k = \left\| \tau_{ij}^k + t_j^k \right\|$, $i, j = 0,1,...,N$, $k = 1,...,K$, где $i = 0$ - режим начального состояния (останова машины),

$$a_{ij}^k = \begin{cases} \tau_{ij}^k + t_j^k, \textit{если } i = 1,..., N, \\ \tau_{ij}^k, \textit{если } i = 0. \end{cases} \qquad (1)$$

Необходимо разбить все множество выполняемых заданий $\tilde{I}$ на K не пересекающихся подмножеств $\tilde{I}^k$, т.е. $\tilde{I}^k \bigcap \tilde{I}^l = \varnothing$, $k, l = 1,...,K$, $k \neq l$; $\bigcup_{k=1}^{K} \tilde{I}^k = \tilde{I}$, определить последовательности $\overline{U}^k$ выполнения подмножеств этих заданий на каждой машине, каждая из которых должна начинаться и завершаться в начальной вершине $i^k = 0$.

Пусть последовательность $\overline{U}^k$ имеет вид $\overline{U}^k = \{0, i_1^k,..., i_p^k,..., i_n^k, 0\}$. Времена начала и завершения стоящих в последовательности $\overline{U}^k$ заданий определяются согласно выражениям

$$\theta_{i_1}^k = \max(h^k + \tau_{0,i_1}^k, b_{i_1}), \ T_{i_1}^k = \theta_{i_1}^k + t_{i_1}^k; \quad \theta_{i_2}^k = \max(T_{i_1}^k + 1 + \tau_{i_1,i_2}^k, b_{i_2}), \ T_{i_2}^k = \theta_{i_2}^k + t_{i_2}^k, \ ...,$$

$$\theta_{i_n}^k = \max(T_{i_{n-1}}^k + 1 + \tau_{i_{n-1},i_n}^k, b_{i_n}), \ T_{i_n}^k = \theta_{i_n}^k + t_{i_n}^k. \qquad (2)$$

Обозначим $R^k = T_{i_n}^k + a_{i_n,0}^k$, $k = 1,...,K$, - время завершения выполнения расписания работ на k-й машине. Здесь $T_{i_n}^k$ и R^k - соответственно время завершения стоящего последним в последовательности $\overline{U}^k$ задания и время завершения выполнения расписания.

На строящиеся последовательности выполнения заданий накладываются ограничения:

- время работы каждой машины не должно превышать выделенный для ее работы ресурс времени

$$R^k \leq H^k, \ k = 1,...,K, \qquad (3)$$

- время завершения выполнения каждого из заданий не должно превышать заданные граничные значения B_i

$$T_{i_p}^k \le B_{i_p}\ ,\ \ p = 1,...,n\ ,\ \ k = 1,...,K\ . \tag{4}$$

В качестве критериев оптимальности могут рассматриваться:

- завершение выполнения расписания работ в кратчайшие сроки

$$F_1 = \min \max_{1 \le k \le K} R^k\ , \tag{5}$$

- минимизация средневзвешенного суммарного времени работы машин

$$F_2 = \min \sum_{k=1}^{K} w^k R^k\ , \text{ где } 0 \le w^k \le 1\ ,\ \ k = 1,...,K\ ;\ \ \sum_{k=1}^{K} w^k = 1\ . \tag{6}$$

В качестве частного случая может быть принято $w^k = 1$, $k = 1,...,K$.

Критерий наиболее равномерной загрузки машин имеет вид:

$$F_3 = \min \max_{1 \le k \le K} \max_{\substack{1 \le s \le K, \\ k > s}} \left| R^k - R^s \right|\ . \tag{7}$$

Здесь $\left| R^k - R^s \right|$ - абсолютная величина разности двух величин.

Ниже рассматриваются алгоритмы решения сформулированной задачи методами динамического программирования [2, 4, 5, 17]. В процессе решения строятся различные допустимые частичные планы (последовательности) выполнения работ на каждой машине. В процессе этих построений производится отсев недопустимых и не содержащих оптимальных решений последовательностей.

На втором этапе решения производится проверка, возможно ли из имеющихся последовательностей выполнения заданий на каждой машине сконструировать расписание выполнения всех заданий, удовлетворяющее всем ограничениям задачи и являющееся оптимальным согласно выбранному критерию оптимальности. В процессе решения задачи на 1-м и 2-м этапе используются сформулированные ниже правила сравнения и отсева неперспективных вариантов.

3. Свойства допустимых и оптимальных расписаний.

Правило отсева 1. Если при построении последовательности выполнения заданий на некоторой k-машине при включении i_l^k-го задания на последнее место в строящейся подпоследовательности $\overline{U}^k = \{0, i_1^k, ..., i_{l-1}^k, i_l^k\}$ выполняется одно из следующих неравенств

$$T_{i_l}^k = \theta_{i_l}^k + t_{i_l}^k > B_{i_l}, \quad T_{i_l}^k = T_{i_l}^k + a_{i_l,0}^k > H^k, \tag{8}$$

где значения $\theta_{i_l}^k$ определяются по формулам (2), то это задание не может быть включено на l-е место в последовательности выполнения заданий на k-машине.

<u>Правило отсева 2.</u> Если для двух последовательностей $\overline{U}^{1k} = \{0, i_{i_1}^{1k}, ..., i_{i_p}^{1k}\}$ и $\overline{U}^{2k} = \{0, i_{i_1}^{2k}, ..., i_{i_r}^{2k}\}$ выполнения заданий на одной и той же k-й машине справедливы соотношения

$$i_{i_p}^{1k} = i_{i_r}^{2k}, \quad \tilde{I}^{2k} \subseteq \tilde{I}^{1k}, \quad T_{i_p}^{1k} \le T_{i_r}^{2k}, \tag{9}$$

то последовательность $\overline{U}^{2k}$ может быть отброшена как неперспективная.

Следствия правила отсева 2.

При построении подпоследовательности $\overline{U}^k = \{0, i_1^k, ..., i_{l-1}^k, i_l^k\}$ выполнения заданий на k-машине производится на основании использования правила отсева 2 по результатам проверки соотношений (9) исключается из рассмотрения как неперспективна либо вновь построенная подпоследовательность, либо все ранее построенные неперспективные подпоследовательности.

Для продолженных во все допустимые вершины графа последовательностей выполнения заданий на одной и той же машине может использоваться следующее правило предпочтения.

<u>Правило отсева 3.</u> Если для двух построенных и не допустимых для дальнейшего продолжения последовательностей $\overline{U}^{1k} = \{0, i_{i_1}^{1k}, ..., i_{i_p}^{1k}\}$ и $\overline{U}^{2k} = \{0, i_{i_1}^{2k}, ..., i_{i_r}^{2k}\}$ выполнения заданий на одной и той же k-й машине, для которых индексы стоящих последними в последовательностях заданий $i_{i_p}^{1k}$ и $i_{i_r}^{2k}$ могут быть различными, справедливы соотношения

$$\tilde{I}^{2k} \subseteq \tilde{I}^{1k}, \quad R^{1k} \le R^{2k}, \tag{10}$$

то последовательность $\overline{U}^{2k}$ может быть отброшена как неперспективная.

Эти правила отсева используется на 1-м этапе решения задачи.

На втором этапе решения в качестве допустимых последовательностей выполнения заданий $\overline{U}^q$ на первых Q машинах, $q = 1,...,Q$, где $Q = 2,...,K$, рассматриваются только такие, у которых подмножества включенных в них заданий, обладают следующими свойствами

$$\tilde{J}^Q = \bigcup_{q=1}^{Q} \tilde{I}^q, \ \tilde{I}^q \cap \tilde{I}^p = \varnothing, q \neq p, \ q, p = 1,...,Q \ . \tag{11}$$

Для следующей и еще не рассмотренной k-й машины $\overline{U}^k$ с подмножеством заданий $\tilde{I}^k$ рассматриваются только такие метки, для которых справедливо соотношения $\tilde{J}^Q \cap \tilde{I}^k = \varnothing$. Обозначим такое объединение последовательностей Ω^Q.

<u>*Правило отсева 4.*</u> Если для двух построенных объединений последовательностей $\Omega^{1,q}$ и $\Omega^{2,q}$, $q = 1,...,Q$, подмножеств $\tilde{J}^{1,Q}$ и $\tilde{J}^{2,Q}$ при решении задачи по критерию F_1 справедливы соотношения

$$\tilde{J}^{2,Q} \subseteq \tilde{J}^{1,Q}, \quad \max_{1 \le q \le Q} R^{1,q} \le \max_{1 \le q \le Q} R^{2,q}, \tag{12}$$

либо при решении задачи по критерию F_2 справедливы соотношения

$$\tilde{J}^{2,Q} \subseteq \tilde{J}^{1,Q}, \quad \sum_{q=1}^{Q} w^q R^{1,q} \le \sum_{q=1}^{Q} w^q R^{2,q}, \tag{13}$$

то объединение частичных планов $\Omega^{2,q}$ может быть отброшено как неперспективное.

Сформулированные правила отсева неперспективных продолжений используются в работе описанного выше алгоритма.

4. Вычислительная схема алгоритма

При построении различных вариантов последовательностей выполнения заданий на каждой машине метка s-й строящейся последовательности содержит следующие признаки:

v_1^s - номер машины;

v_2^s - номер метки для этой машины;

v_3^s - номер задания, включенного на данное место в последовательность;

v_4^s - номер предыдущей метки (продолжаемой на этом шаге последовательности);

v_5^s - возможность и перспективность продолжения данной последовательности. При этом $v_5^s = 0$, если данная последовательность может быть продолжена, и $v_5^s = 1$ - в противном случае, $v_5^s = 2$, если эта метка исключена из рассмотрения как неперспективная и не должна рассматриваться на 2-м этапе работы алгоритма;

$\overline{U}^s = \{i_0^k, i_1^k, i_2^k, ..., i_n^k\}$ - построенная последовательность выполнения заданий;

$T^s = T_{i_n}^k$ - время завершения стоящего в данной последовательности задания;

R^s - время работы машины (с учетом возвращения в исходное состояние i_0^k), необходимое для выполнения данной последовательности работ.

На втором этапе алгоритма, осуществляющем объединение в общее расписание построенных последовательностей. При этом метка каждого рассматриваемого g-го варианта состоит из следующих признаков:

g - номер варианта;

$\overline{U}^{g,k} = \{i_1^{g,k}, i_2^{g,k}, ..., i_n^{g,k}\}$, $k = 1,...,K$, - последовательности выполнения заданий на каждой машине;

$R^{g,k}$, $k = 1,...,K$, - вектор суммарных времен работы каждой машины.

4.1. Алгоритм 1-го этапа.

В начале работы 1- го этапа алгоритма сформируем начальные ($s = 0$) метки для каждой k-й машины:

$v_1^s = k$, $v_2^s = 0$, $v_3^s = i_0$, $v_4^s = \varnothing$, $v_5^s = 0$; $\overline{U}^s = \{i_0\}$, $T^s = R^s = 0$.

Для каждого значения $k = 1,...,K$ выполняем следующие построения.

Шаг 1. Среди меток k-й машины находим s-ю метку, для которой $v_5^s = 0$. Если такую метку удается найти, то переходим к шагу 2. Если таких меток не существует, то построены все допустимые последовательности выполнения заданий для данной машины, и переходим к следующей машине. Если таких меток не существует ни для одной из машин, то построены все допустимые последовательности выполнения заданий на данной машине, и переходим к шагу 4.

Шаг 2. Рассматриваем s-ю метку. Подмножество заданий, для которых выполняются все условия

$$D_j^{s,k} = \max(b_j, T^{s,k} + \tau_{i_n,j}^{s,k}) + t_j^k \le B_j, \quad D_j^{s,k} + a_{j,0}^k \le H^k, \quad j \in \tilde{I} / \tilde{J}^{s,k}, \tag{14}$$

обозначим $\hat{J}^{s,k}$, $\hat{J}^{s,k} = \{j_1^{s,k}, ..., j_\lambda^{s,k}, ..., j_M^{s,k}\}$. В выражении (14) $\tilde{J}^{s,k}$ - подмножество заданий, входящих в последовательность $\overline{U}^s$.

Пусть $M^{s,k}$ - количество заданий в подмножестве $\hat{J}^{s,k}$. Если $\hat{J}^{s,k} = \varnothing$, то полагаем для s-й метки $v_5^s = 1$, и переходим к шагу 1.

Если $\hat{J}^{s,k} \ne \varnothing$, то образуем $M^{s,k}$ новых меток, продолжив и поставив на последнее место последовательности $\overline{U}^s$ каждое из заданий подмножества $\tilde{J}^{s,k}$.

Признаки каждой из этих меток имеют вид

$$v_1^{s+\lambda} = k, \ v_2^{s+\lambda} = s + \lambda, \ v_3^{s+\lambda} = j_\lambda^{s+\lambda}, \ v_4^{s+\lambda} = s, \ v_5^{s+\lambda} = 0; \quad \overline{U}^{s+\lambda} = \{i_0^{s,k}, ..., i_n^{s,k}, j_\lambda^{s,k}\};$$

$$T^{s+\lambda} = \max(T_{i_n}^{s,k} + \tau_{i_n,j_\lambda}^k, b_{j_\lambda}) + t_{j_\lambda}^k, \quad R^{s+\lambda} = T^{s+\lambda} + a_{j_\lambda i_0}^k; \quad \lambda = 1,2,...,M^{s,k}. \tag{15}$$

Полагаем для s-й метки признак $v_5^s = 1$, и переходим к шагу 3.

Шаг 3. Производим попарно сравнение каждой из вновь образованных последовательностей $\overline{U}^{s+\lambda}$ со всеми хранящимися в банке данных последовательностями. Если для одной из рассматриваемых пар выполняются условия правила отсева 2, то соответствующая неперспективная метка с индексом γ исключается из рассмотрения, т.е. полагаем для нее значение $v_5^\gamma = 2$. Производим переиндексацию всех рассматриваемых на данной машине вариантов. Исключив варианты, для которых $v_5^\gamma = 2$. Переходим к шагу 1.

Шаг 4. Производим попарно сравнение всех меток, для которых $v_5^s = 1$. Если в результате такого сравнения для какой-то из пар меток выполняется правило отсева 3, то для неперспективной и подлежащей отсеву метки полагаем значение $v_5^s = 2$. После выполнения этой процедуры и переиндексации всех перспективных меток, для которых $v_5^s = 1$, алгоритм 1-го этапа для k-машины завершает свою работу.

Выполнив описанный выше алгоритм для каждой из рассматриваемых машин, переходим к этапу 2 алгоритма. Отметим, что если все параллельные машины имеют одинаковые технические характеристики и одни и те же допустимые времена начала и завершения выполнения работ, то на 1-м этапе построение альтернативных последовательностей выполнения заданий строится только для одной машины.

4.2. Алгоритм второго этапа.

Метка рассматриваемых вариантов на 2-м этапе решения задачи имеет следующие признаки:

u^s - номер составной метки;

$Z^s = | z^{sk} |$ - номера включаемых меток каждой машины;

$\tilde{\Pi}^s$ - множество заданий, включенных в расписание выполнения работ на всех машинах;

$F_1^s = R_{\min}^s$ - минимальное время расписания выполнения работ на всех машинах;

F_2^s - средневзвешенное суммарное время работы машин.

Рассматриваются только допустимые и перспективные метки последовательностей выполнения заданий на каждой машине, для которых значение $v_5^{pk} = 1$. Каждая из этих меток имеет свой индекс $p^k = 1,...,P^k$. Обозначим $F_1^{\min}(s)$, $F_2^{\min}(s)$ - наилучшие значения соответствующих показателей на данном s-м шаге алгоритма. Введем вектор номеров рассмотренных меток каждой машины $L = | \lambda^k |$, $k = 1,...,K$, где $\lambda^k = 1,...,m^k$, а также счетчик номера анализируемой машины δ^k, $k = 1,...,K$. В начале процесса полагаем $s = 0$; $\lambda^k = 0$, $\delta^k = 0$, $k = 1,...,K$; $F_1^{\min}(s) = F_1^{\min}(s) = \infty$.

Алгоритм представляет собой последовательное выполнение следующих шагов.

<u>*Шаг 1.*</u> Полагаем $s := (s+1)$. Формируем начальные признаки новой метки. Для $k = K, (K-1), ..., 1$, выполняем:

1.1) если $\lambda^k \le (m^k + 1)$, то полагаем $z^{sk} = (\lambda^k + 1)$, $\lambda^k := (\lambda^k + 1)$; $z^{gk} = \lambda^g$, $g = (k-1), ..., 1$.

1.2) если $\lambda^k = (m^k + 1)$, то полагаем $z^{sk} = \lambda^k = 1$, $z^{s,k-1} = (\lambda^{k-1} + 1)$, $\lambda^{k-1} := (\lambda^{k-1} + 1)$. При этом, если $\lambda^{k+1} = (m^{k+1} + 1)$, то положив $k := (k-1)$, переходим к пункту 1.1. В противном случае полагаем $z^{gk} = \lambda^g$, $g = (k-2), ..., 1$.

Если $\lambda^k = m^k + 1$, $k = 1, ..., K$, то переходим к шагу 4. В противном случае переходим к шагу 2.

<u>*Шаг 2.*</u> Полагаем $\tilde{\Pi}^s = \varnothing$, $F_1^s = F_2^s = 0$. Последовательно для индексов машин $k = 1, ..., K$ выбираем метку, номер которой равен z^{sk}, подмножество выполняемых заданий для которой $\tilde{J}(z^{sk})$. Если $\tilde{J}(z^{sk}) \cap \tilde{\Pi}^s = \varnothing$, то определяем $\tilde{\Pi}^s = \tilde{\Pi}^s \cup \tilde{J}(z^{sk})$ и вычисляем в зависимости от выбранного критерия оптимальности значение

$$F_1^s = \max[F_1^s, R^k(z^{sk})] \text{ или } F_2^s = F_2^s + w^k R^k(z^{sk}), \text{ или } F_3^s = \max_{1 \le k \le K} \max_{\substack{1 \le s \le K, \\ k > s}} \left| R^k(z^{sk}) - R^s \right|. \quad (16)$$

Если $F_1^s \ge F_1^{\min}(s)$ или $F_2^s \ge F_2^{\min}(s)$, или $F_3^s \ge F_3^{\min}(s)$, либо если $\tilde{J}(z^{sk}) \cap \tilde{\Pi}^s \ne \varnothing$, то полагаем $\lambda^k := (\lambda^k + 1)$, и переходим к шагу 1. В противном случае переходим к шагу 3.

<u>*Шаг 3.*</u> Если $\tilde{\Pi}^s \ne \tilde{I}$, то переходим к шагу 1. Если $\tilde{\Pi}^s = \tilde{I}$, то, если $F_1^s < F_1^{\min}(s)$, полагаем $F_1^s = F_1^{\min}(s)$. Если $F_2^s < F_2^{\min}(s)$, то полагаем $F_2^s = F_2^{\min}(s)$. Запоминаем номер метки s^*, соответствующей полученному наилучшему расписанию на данном шаге. Переходим к шагу 1.

<u>*Шаг 4.*</u> Получено оптимальное решение задачи, которое содержится в метке с номером s^*. Находим номера меток с номерами z^{s^*k}, соответствующие последовательностям выполнения заданий на каждой k-й машине $\overline{U}^k(z^{s^*k})$, а

также времена работы машин $R^k(z^{s^*k})$, $k=1,2,...,K$. На основе этой информации могут быть определены времена начала и завершения выполнения каждого из заданий на данной машине. На этом алгоритм решения задачи завершает свою работу.

Следует отметить, что в практических расчетах решения задачи по критериям (5) или (7) минимизации времени выполнения расписания работ, с целью сокращения количества вариантов на 1-м этапе решения задачи, могут быть заданы вспомогательные дополнительные ограничения на времена работы машин $\hat{H}^k < H^k$, $k=1,...,K$. Это сократит также и объём вычислений на 2-м этапе алгоритма.

5. Иллюстративные примеры

4.1. Пример 1.

На 2 машинах необходимо выполнить 6 заданий. Суммарные времена выполнения заданий и потерь на переналадки при переходе от выполнения i-го заданий к j-му для каждой из машин сведены в табл. 1, 2.

Табл. 1: Суммарные времена выполнения заданий и потерь на переналадки для 1-й машины

№ заданий	Суммарные времена выполнения заданий и переналадок						
	0	1	2	3	4	5	6
0	∞	7	5	6	4	6	8
1	13	∞	14	15	12	16	17
2	20	18	∞	20	21	20	17
3	24	24	25	∞	26	23	27
4	18	14	16	15	∞	18	16
5	13	13	16	16	14	∞	15
6	23	21	22	20	24	23	∞

Табл. 2: Суммарные времена выполнения заданий и потерь на переналадки для 2-й машины

№ заданий	Суммарные времена выполнения заданий и переналадок						
	0	1	2	3	4	5	6
0	∞	8	6	5	4	7	9
1	16	∞	16	17	15	18	14
2	21	20	∞	22	21	23	21
3	28	27	26	∞	28	29	26
4	22	18	19	20	∞	21	17
5	16	17	16	18	19	∞	15

6	15	24	23	26	25	27	∞

Граничные сроки завершения каждого из заданий приведены в табл. 3.

Табл. 3: Граничные сроки завершения заданий

Граничные сроки завершения выполнения заданий					
1	2	3	4	5	6
51	22	65	50	37	75

Выполнение производственной программы должно начинаться и завершаться в состоянии $i = 0$. Необходимо определить подмножества заданий, выполняемых на каждой из машин, и последовательности их выполнения, которые обеспечили бы выполнение всех ограничений на сроки завершения заданий и минимизировали бы значение следующих критериев оптимальности:

1) выполнение производственной программы в кратчайшие сроки,

$F_1 = \min \max(T_1, T_2)$

2)минимизация суммарного времени работы машин $F_2 = \min(T_1 + T_2)$.

А) Строим допустимые последовательности выполнения заданий на каждой машине, осуществляя отсев неперспективных вариантов. Для сокращения объёма изложения признаки меток v_1^s, v_5^s в описании примера опущены.

1) 1, 0, $\overline{U}_1^1 = \{1\}$, 13, 20; 2) 2, 1, $\overline{U}_2^1 = \{1,3\}$, 37, 43; 3)3, 1, $\overline{U}_3^1 = \{1,4\}$, 27, 31;

4) 4, 1, $\overline{U}_4^1 = \{1,5\}$, 26, 31; 5) 5, 1, $\overline{U}_5^1 = \{1,6\}$, 34, 42; 6) 6, 2, $\overline{U}_6^1 = \{1,3,6\}$, 57, 65;

7) 7, 3, $\overline{U}_7^1 = \{1,4,3\}$, 53, 59; 8)8, 3, $\overline{U}_8^1 = \{1,4,6\}$, 51, 59; 9) 9, 4, $\overline{U}_9^1 = \{1,5,3\}$, 49, 55;

10) 10, 4, $\overline{U}_{10}^1 = \{1,5,4\}$, 44, 48; 11) 11, 4, $\overline{U}_{11}^1 = \{1,5,6\}$, 49, 57;

12) 12, 5, $\overline{U}_{12}^1 = \{1,6,4\}$, 50, 54; 13) 13, 7, $\overline{U}_{13}^1 = \{1,4,3,6\}$, 73, 81;

14) 14, 9, $\overline{U}_{14}^1 = \{1,5,3,6\}$, 69, 77; 15) 15, 10, $\overline{U}_{15}^1 = \{1,5,4,6\}$, 68,76;

2-я машина

1) 0, $\overline{U}_1^2 = \{2\}$, 21, 27; 2) 0, $\overline{U}_2^2 = \{3\}$, 28, 33; 3) 0, $\overline{U}_3^2 = \{4\}$, 22, 26;

4) 0, $\overline{U}_4^2 = \{5\}$, 16, 23; 5) 0, $\overline{U}_5^2 = \{6\}$, 26, 35; 6) 1, $\overline{U}_6^2 = \{2,3\}$, 48, 53;

7) 1, $\overline{U}_7^2 = \{2,4\}$, 40, 46; 8) 1, $\overline{U}_8^2 = \{2,6\}$, 46, 55; 9) 2, $\overline{U}_1^2 = \{3,4\}$, 48, 52;

10) 2, $\overline{U}_{10}^{2} = \{3,5\}$, 46,53; 11) 2, $\overline{U}_{11}^{2} = \{3,6\}$, 54, 63; 12) 3, $\overline{U}_{12}^{2} = \{4,3\}$, 50, 54;

13) 3, $\overline{U}_{13}^{2} = \{4,5\}$, 41, 48; 14) 3, $\overline{U}_{14}^{2} = \{4,6\}$, 47, 56; 15) 4, $\overline{U}_{15}^{2} = \{5,3\}$, 45, 52;

16) 4, $\overline{U}_{16}^{2} = \{5,4\}$, 41,51; 17) 4, $\overline{U}_{17}^{2} = \{5,6\}$, 43,52; 18) 5, $\overline{U}_{18}^{2} = \{6,3\}$, 52, 57;

19) 5, $\overline{U}_{19}^{2} = \{5,6\}$, 47,51; 20) 6, $\overline{U}_{20}^{2} = \{2,3,6\}$, 67,73; 21) 7, $\overline{U}_{21}^{2} = \{2,4,6\}$,65 ,74;

22) 9, $\overline{U}_{22}^{2} = \{3,4,6\}$, 73,8 2; 23) 10, $\overline{U}_{23}^{2} = \{3,5,6\}$, 63, 72;

24)12, $\overline{U}_{24}^{2} = \{4,3,6\}$, 76, 85; 25) 13, $\overline{U}_{25}^{2} = \{4,5,6\}$, 68, 75.

Следует отметить, что метка 24 рассматриваемых вариантов для 2-й машины, т.е. последовательность $\overline{U}_{24}^{2} = \{4,3,6\}$, исключается как неперспективная, так как метка 22 с последовательностью выполнения заданий $\overline{U}_{22}^{2} = \{3,4,6\}$ является более перспективной.

В) На втором этапе решения строим допустимые и наиболее эффективные объединения подмножеств выполняемых заданий на двух машинах.

Допустимыми являются только 4 объединения меток:

a) $\overline{U}_{9}^{1} = \{1,5,3\}$, $T_{9}^{1} = 55$, и $\overline{U}_{21}^{2} = \{2,4,6\}$, $T_{21}^{2} = 74$;

b) $\overline{U}_{10}^{1} = \{1,5,4\}$, $T_{10}^{1} = 54$, и $\overline{U}_{20}^{2} = \{2,3,6\}$, $T_{20}^{2} = 73$;

c) $\overline{U}_{14}^{1} = \{1,5,3,6\}$, $T_{14}^{1} = 77$, и $\overline{U}_{7}^{2} = \{2,4\}$, $T_{7}^{2} = 46$;

d) $\overline{U}_{15}^{1} = \{1,5,4,6\}$, $T_{15}^{1} = 76$, и $\overline{U}_{6}^{2} = \{2,3\}$, $T_{6}^{2} = 53$.

По критерию (4.38) минимизации времени завершения выполнения всех заданий оптимальным является расписание $\overline{U}_{10}^{1} = \{1,5,4\}$ & $\overline{U}_{20}^{2} = \{2,3,6\}$ со значением критерия оптимальности, равным $F_{1}^{b} = \max(54,73) = 73$.

По критерию (4.39) минимизации суммарного времени работы машин оптимальным является расписание $\overline{U}_{14}^{1} = \{1,5,3,6\}$ & $\overline{U}_{7}^{2} = \{2,4\}$ со значением критерия оптимальности, равным $F_{1}^{c} = (77+46) = 123$.

4.2. Пример 2 (одинаковые по техническим характеристикам машины).

Суммарные времена переналадок и выполнения заданий для каждой машины сведены в таблицу 4, а граничные сроки завершения выполнения у каждого из заданий – в таблицу 5. Время завершения выполнения всех работ каждой машиной не должно превышать значения 60.

Таблица 4: Суммарные времена переналадок и выполнения заданий для каждой машины

№	0	1	2	3	4	5
0	∞	5	7	8	4	6
1	15	∞	15	17	20	23
2	16	16	∞	21	19	16
3	17	16	19	∞	22	18
4	15	21	17	20	∞	19
5	16	22	14	16	20	∞

Таблица 5: Граничные сроки завершения выполнения заданий

1	2	3	4	5
30	25	48	52	43

Процесс решения иллюстративного примера 2.

А) Построения допустимых и перспективных расписаний работы каждой машины.

1) 0; (0, 1); 15, 20. 2) 0; (0, 2); 16, 23. 3) 0; (0,3); 17, 25. 4) 0; (0, 4); 15,19.

5) 0; (0,5); 16, 22.

6) 1; (0, 1, 3); 31, 39. 7) 1; (0, 1, 4); 36, 40. 8) 1; (0, 1, 5); 37, 43.

9) 2; (0, 2, 3); 37, 45. 10) 2; (0, 2, 4); 33, 40. 11) 2; (0, 2, 5); 30, 36.

12) 3; (0, 3, 4); 35, 39. 13) 3; (0, 3, 5); 33, 39. 14) 4; (0, 4, 3); 37, 45.

15) 4; (0, 4, 5); 35, 41. 16) 5; (0, 5, 3); 34, 42. 17) 5; (0, 5,4); 35, 39.

18) 6; (0, 1, 3, 4); 51, 55. 19) 11; (0, 2, 5, 3); 46, 54.

20)11; (0, 2, 5, 4); 49, 53. 21)13; (0, 3, 5, 4); 52, 56.

Б) Построение допустимых перспективных планов работы 2 машин (допустимые объединения подмножеств):

а) 6) 1; (0, 1, 3); 31, 39 и 20)11; (0, 2, 5, 4); 49, 53.

б) 7) 1; (0, 1, 4); 36, 40 и 19) 11; (0, 2, 5, 3); 46, 54.

По всем рассматриваемым критериям оптимальности наиболее предпочтительным является вариант а). В соответствии с этим решением 1-я машина выполняет задание 1 со сроком завершения, равным 15, затем задание 3 со сроком завершения, равным 31, и возвращается на базу в момент времени 39. 2-я машина выполняет задание 2 со сроком завершения, равным 16, затем задание 5 со сроком завершения, равным 30, после чего задание 3 со сроком завершения, равным 49, и возвращается на базу в момент времени 53.

Литература

1. Конвей Р. В., Максвелл В. Л., Миллер Л. В.: - Теория расписаний. – М-, Физматгиз, Наука, 1975, 359 с.

2. Танаев В.С., Шкурба В.В.: - Введение в теорию расписаний. – М., Физматгиз, Наука, 1975, 256 с.

3. Танаев В.С., Ковалёв М.Я., Шафранский Я.М.: - Теория расписаний. Групповые технологии. - Минск, Институт технической кибернетики НАН Беларуси, 1998, 289 с.

4. Domschke W., Scholl A., Voß S.: - Produktionsplanung. Ablauforganisatorische Aspekte. – Berlin, Heidelberg: Springer Verlag, 2005, 456 S.

5. Brucker P.: - Scheduling Algorithms. - Springer-Verlag, Berlin, Heidelberg und New York 1998.

6. Blazewicz J., Ecker K.H., Pesch, E., Schmidt G., Weglarz J.: - Scheduling Computer and Manufacturing Processes, Springer Verlag, Berlin-Heidelberg, 2001.

7. Herrmann J.: - Supply Chain Scheduling. Transaktionskostentheorie; Parallele Maschinen; Heuristik; Optimierungsmodelle. - Gabler Verlag, Berlin-Heidelberg, 2010, 162 S.

8. Pinedo M.: - Scheduling: Theory, Algorithms and Systems. - Springer Verlag, Berlin-Heidelberg 1999.

9. Blazewicz J., Domschke W. und Pesch E.: - The job shop scheduling problem: Conventional and new solution techniques. - Eur. J. Oper. Res., 1996, P. 93, 1-33

10. Herrmann J.: - Supply Chain Scheduling. Transaktionskostentheorie; Parallele Maschinen; Heuristik; Optimierungsmodelle. - Gabler Verlag, Berlin-Heidelberg, 2010, 162 S.

11. Szelke E., Kerr R.M.: - Artificial Intelligence in Reactive Scheduling. - Chapman & Hall, London, 1995.

12. Батищев Д.И., Гудман Э.Д., Норенков И.П., Прилуцкий М.Х .: - Метод комбинирования эвристик для решения комбинаторных задач упорядочения и распределения ресурсов. - М., Информационные технологии, 2, 1997. с.29-32.

13. Курейчик В.М.: - Генетические алгоритмы. Монография.- Таганрог: Изд. ТРТУ, 1998, 242с.

14. Зак Ю.А: - Алгоритмы решения задач «N коммивояжеров». - Кибернетика, К., 1972, № 2, С. 99-106.

15. Зак Ю.А: - Об одном классе задач разбиения на допустимые подмножества и упорядочения. – Сб. «Техническая кибернетика», Изд. АН УССР, Киев, 1970, № 15, с.27-36.

16. Зак Ю.А: - Определение порядка выполнения независимых операций на параллельных машинах. – Изв. АН СССР. Техническая кибернетика, 1969,

№ 2, С. 15-20.

17. Зак Ю.А: - Прикладные задачи теории расписаний и маршрутизации перевозок. – URSS, М., 2011, 394 с.