УДК 519.874

РАЗБИЕНИЕ НА ПОДМНОЖЕСТВА И ПОСТРОЕНИЕ ДОПУСТИМЫХ И ОПТИМАЛЬНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ВЫПОЛНЕНИЯ МНОЖЕСТВА ЗАДАНИЙ НА НЕСКОЛЬКИХ МАШИНАХ

Ю.А. ЗАК

Исследованы свойства задач построения допустимых и оптимальных расписаний выполнения N заданий на m машинах в условиях потерь времени на перекладки. На основе установленных свойств конструируются операторы исключения из рассмотрения подмножеств расписаний, не содержащих допустимых решений. Предложены алгоритмы вычисления нижних оценок различных критериев оптимальности, а также алгоритмы решения рассматриваемых задач последовательными алгоритмами оптимизации.

ВВЕДЕНИЕ

Задачи распределения множества заданий подлежащих выполнению 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, а также организации обслуживания и выполнения ремонтных работ, организации вычислительного процесса и т.д. Приведем только некоторые из таких постановок.

КАЛЕНДАРНОЕ ПЛАНИРОВАНИЕ ПРОИЗВОДСТВА

На K различных по техническим характеристикам единицах технологического оборудования (станках) необходимо произвести N различных партий деталей. Заданы времена обработки партии деталей каждого вида на каждом из станков, матрицы времен переналадок при переходе каждого станка от выпуска одной партии деталей на другую, а также временные ресурсы возможности использования оборудования и ограничения на сроки завершения обработки каждой партии деталей. Каждая партия деталей может обрабатываться только на одном станке. Необходимо определить, партии каких деталей должны обрабатываться на каждом станке, а также последовательности их обработки на каждой единице технологического оборудования, которые обеспечивают выполнение всех ограничений на временные ресурсы работы оборудования и сроки завершения выполнения заданий, а также экстремальное значение выбранного критерия оптимальности. В качестве критерия оптимальности может быть выбрано завершение обработки всех партий деталей в кратчайшие сроки или минимизация суммарного средневзвешенного времени занятости технологического оборудования.

ПЛАНИРОВАНИЕ РЕМОНТНО-ПРОФИЛАКТИЧЕСКИХ РАБОТ

K различных по своей квалификации и по уровню технического оснащения бригад должны обслужить N объектов (ремонтно-профилактические, монтажные работы, доставка, погрузка, разгрузка товаров и т.д.). Заданы времена, необходимые каждой бригаде для выполнения всего объема работ на каждом из объектов, а также потери времени при переходе каждой бригады от одного объекта на другой. Необходимо определить подмножества объектов, обслуживаемых каждой бригадой, а также последовательности выполнения этих подмножеств работ, которые обеспечили выполнение всех заданий в установленные ограничениями сроки. В качестве критериев оптимальности построенного расписания могут быть выбраны минимальные суммарные затраты рабочего времени и времени на переезды всех бригад или выполнение всего множества заданий в кратчайшие сроки.

Методам решения данного класса задач уделялось значительное место в монографиях и периодической литературе [1–3, 7, 9, 10]. Подходы, связанные с построением линейных и нелинейных целочисленных моделей и использование методов математического программирования для решения задач данного класса достаточно большой размерности, представляющих практический интерес, требуют больших объемов вычислений [6–8].

Наиболее широкое распространение получили приближенные методы решения задач данного класса с использованием генераторов случайных расписаний [6, 7, 12, 13], использующих различные правила предпочтения [7–11, 13], эвристические подходы [5, 9], а также генетические алгоритмы и эволюционные стратегии [4, 11]. Алгоритмы решения данной задачи без учета ограничений на директивные сроки выполнения заданий методами построения кратчайших допустимых путей на графах приведены в работе автора [12]. 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 [13]. Кроме того, отсутствие нижних оценок значения критерия оптимальности построенного расписания не позволяет объективно оценить эффективность полученного решения.

В данной работе изучаются свойства задач данного класса, на основе которых конструируются операторы исключения из рассмотрения подмножеств расписаний, не содержащих допустимых решений, предлагаются алгоритмы вычисления нижних оценок различных критериев оптимальности. На основе установленных свойств предлагаются алгоритмы решения рассматриваемых задач последовательными алгоритмами оптимизации. В качестве частного случая рассматриваются свойства и алгоритмы решения задачи определения допустимых и оптимальных последовательностей выполнения заданий на одной машине.

ПОСТАНОВКА 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 завершения каждого из заданий T_i, $i = 1,..., N.$
- матрица времен выполнения каждого из заданий на всех машинах $\bar{t}^k = (t_1^k, t_2^k, ..., t_i^k, ..., t_N^k)$, $k = 1,..., K;$
- θ^k — наиболее ранние допустимые сроки начала выполнения работ на k -й машине;
- $A^k = | a_{ij}^k |$, $i, j = 0,1,..., N$ — матрицы времен 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 A^k заданы времена настройки машины из состояния, в котором 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 другого критерия оптимальности F_2 может быть выбрано минимальное средневзвешенное время работы машин, необходимое для выполнения всего комплекса работ.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ЗАДАЧИ. СВОЙСТВА ДОПУСТИМЫХ И ОПТИМАЛЬНЫХ РАСПИСАНИЙ

Построим матрицы $B^k = | b_{ij}^k |$, $i, j = 0,1,..., N$; $k = 1,..., K$ суммарных затрат времени на выполнение заданий на каждой машине, учитывающие времена выполнения заданий, плюс потери времени на переналадки. Элементы этой матрицы определяются по формулам

$$b_{ij}^k = t_j^k + a_{ij}^k,\ i, j = 0,1,...,N;\ k = 1,...,K. \tag{1}$$

Определим также

$$\overline{b}_i^{k,\min} = \min_{0 \le j \le N} b_{ij}^k,\ i = 0,1,...,N,\ k = 1,...,K. \tag{2}$$

Построим вспомогательную матрицу $\mathbf{B} = |\beta_{ij}|$, $i, j = 0,1,...,N$, элементы которой определяются согласно выражению

$$\beta_{ij} = \min_{1 \le k \le K} b_{ij}^k,\ i, j = 0,1,...,N. \tag{3}$$

С учетом времен пусконаладочных работ и (или) переналадок каждое i-е из заданий на k-й машине не может быть завершено ранее чем за время $\overline{b}_i^{k,\min}$, не ранее чем через промежуток времени b_{ij}^k после завершения выполнения задания, стоящего в последовательности перед ним, а на любой из всего множества машин — за время β_{ij}.

Найдем минимальное значение элементов каждого столбца матрицы $\mathbf{B} = |\beta_{ij}|$

$$\beta_j^{\min} = \min_{0 \le i \le N} \beta_{ij},\ j = 0,1,...,N. \tag{4}$$

Упорядочим все множество выполняемых заданий $\widetilde{I} = \{1,2,...,i,...,N\}$ в порядке невозрастания граничных сроков их завершения

$$\widetilde{U}_1 = \{i_1, i_2,...,i_N \mid T_{i_1} \le T_{i_2} \le .... \le T_{i_N}\}. \tag{5}$$

Введем булевые переменные x_i^k: $x_i^k = 1$, если i-е задание выполняется на k-й машине, и $x_i^k = 0$ — в противном случае. Так как каждое из заданий может выполняться только на одной машине, то справедливо ограничение

$$\sum_{k=1}^{K} x_i^k = 1,\ i = 1,...,N. \tag{6}$$

Пусть определено подмножество заданий, выполняемых на k-й машине — $\widetilde{J}^k = \{i_1^k, i_2^k,...,i_N^k\}$. Рассмотрим последовательность выполнения этих заданий в порядке невозрастания граничных времен их завершения:

$$\widetilde{U}_1^k = \{j_1^k, j_2^k,...,j_N^k \mid T_{j_1^k} \le T_{j_2^k} \le ... \le T_{j_N^k}\}. \tag{7}$$

Утверждение 1. Если не выполняется хотя бы одно из неравенств системы

$$\theta^k + \sum_{l=1}^{r} \beta_{j_l^k}^{\min} \le T_{j_r^k},\ r = j_1^k, j_2^k;...,j_N^k, \tag{8}$$

то не существует допустимых расписаний выполнения данного комплекса на k-й машине.

Доказательство аналогичного утверждения приведено в [1, 13].

Если определены подмножества $\widetilde{J}^k$ и построены последовательности $\widetilde{U}_1^k$ для всех машин, $k=1,...,K$, то справедливо утверждение 2.

Утверждение 2. Если хотя бы для одного из индексов $k=1,...,K$ не выполняется хотя бы одно из неравенств системы (8), то для данного распределения заданий по рабочим станциям $\widetilde{J}^k$, $k=1,...,K$ не существует допустимых расписаний выполнения всех заданий в установленные ограничениями сроки.

Сформулируем задачу распределения заданий по рабочим станциям (машинам) и выполнения их на каждой из машин в последовательности невозрастания граничных сроков их завершения (последовательности (5)) в виде следующей задачи булевого линейного программирования

$$x_{i_l}^k = \begin{cases} 0 \\ 1 \end{cases}, \; l=1,...,N,\; k=1,...,K, \tag{9}$$

$$\sum_{k=1}^{K} x_{i_l}^k = 1,\; l=1,...,N, \tag{10}$$

$$\theta^k + \sum_{l=1}^{r} \beta_{i_l}^{\min} x_{i_l}^k \le T_{i_l},\; r=1,...,N,\; k=1,...,K, \tag{11}$$

$$\theta^k + \sum_{l=1}^{r} \overline{b}_{i_l}^{k,\min} x_{i_l}^k \le T_{i_l}\; r=1,...,N,\; k=1,...,K. \tag{12}$$

Утверждение 3. Если не выполняется система неравенств (9)–(11) или (9), (10), (12), то система ограничений задачи является несовместной.

Утверждение 4. Время завершения выполнения всех заданий на рабочих станциях не может быть меньше значения η, которое определяется в результате решения следующей задачи булевого линейного программирования

$$\min\left\{\eta \mid \theta^k + \sum_{l=1}^{N} \beta_{i_l}^{\min} x_{i_l}^k - \eta_1 \le 0,\; k=1,...,K\right\} \tag{13}$$

в условиях ограничений (9)–(11) или задачи

$$\min\{\eta_1 \mid \theta^k + \sum_{l=1}^{N} \overline{b}_{i_l}^{k,\min} x_{i_l}^k - \eta_1 \le 0, k=1,...,K\} \tag{14}$$

в условиях ограничений (9), (10), (12).

Так как $\beta_{i_l}^{\min} \le \overline{b}_{i_l}^{k,\min}$ для всех $l=1,...,N$, $k=1,...,K$, то система ограничений (9), (10), (12) является более жесткой, чем система ограничений (9)–(11), и не все значения переменных, удовлетворяющие системе ограничений (9)–(11), обеспечат выполнение условий (9), (10), (12). Поэтому для значений η, определяемым в результате решения задачи (9)–(11), (13) и значения η_1, определяемого из условий (9), (10), (12), (14), справедливо соотношение $\eta_1 \ge \eta$.

Пусть на некотором шаге решения задачи определены подмножества $\widetilde{I}^{k,+}$ и частичные последовательности выполнения заданий на каждой машине $\widetilde{V}^k = \{j_1^k, j_2^k, \ldots, j_{m_k}^k\}$, каждая из которых включает m_k, $k = 1,\ldots,K$ заданий. Здесь $\sum_{k=1}^{K} m_k = m < N$, где $j_{m_1}^k$ — последнее задание, выполняемое на k-й машине в построенной подпоследовательности $\widetilde{V}^k$.

Пусть $P = N - m$, $\widetilde{I}^+ = \bigcup_{k=1}^{K} \widetilde{I}^{k,+}$ — подмножество всех включенных во все подпоследовательности заданий. Вычислим для каждой из машин времена завершения, выполняемых в этих частичных последовательностях заданий:

$$\overline{\theta}^k = \theta^k + \sum_{l=1}^{m_k} (a_{i_{l-1}i_l}^k + t_{i_l}^k), \ k = 1,\ldots,K. \tag{15}$$

Обозначим $\hat{J} = \{\gamma_1, \gamma_2, \ldots, \gamma_P\}$ — подмножество не включенных в частичные последовательности и подлежащих выполнению заданий. Упорядочим эти задания в последовательность по невозрастанию граничных значений времен их завершения

$$\hat{V}^- = \{v_1, v_2, \ldots, v_P \mid T_{v_1} \le T_{v_1} \le \ldots \le T_{v_P}\}. \tag{16}$$

Преобразуем матрицы B^k следующим образом:

- вычеркнем 0-й столбец, а также строки, соответствующие индексам заданий, входящих в подмножество заданий $\widetilde{I}^+$;
- удалим в каждой из матриц B^k строки, соответствующие индексам заданий, принадлежащим подмножеству $\widetilde{I}^+(k) = \widetilde{I}^+ / j_{m_1}^k$.

Назовем такое преобразование матриц B^k преобразованием 1.

Вновь образованные прямоугольные матрицы меньших размеров обозначим $D^k = |d_{ij}^k|$, $k = 1,\ldots,K$. Найдем

$$\overline{d}_i^{k,\min} = \min_{j \in \hat{V}^-} d_{ij}^k, \ i \in \hat{V}^-, \ k = 1,\ldots,K. \tag{17}$$

На основе этих вновь образованных матриц построим матрицу $D = |d_{ij}|$, элементы которой определяются по формулам:

$$d_{ij} = \min_{1 \le k \le K} d_{ij}^k, \ i \in \hat{V}^- \cup \{0\}, \ j \in \hat{V}^-. \tag{18}$$

Определим

$$\overline{\beta}_i^{\min} = \min_{j \in \hat{J}} d_{ij}, \ i \in \hat{V}^- \cup \{0\}. \tag{19}$$

Следствием утверждений 3 и 4 является утверждение 5.

Утверждение 5. Если не выполняется система неравенств

$$x_{v_l}^k = \begin{cases} 0 \\ 1 \end{cases}, \ v_l \in \hat{V}^-, \ k = 1,...,K, \tag{20}$$

$$\sum_{k=1}^{K} x_{v_l}^k = 1, \ v_l \in \hat{V}^-, \tag{21}$$

$$\overline{\theta}^k + \sum_{l=1}^{p} \overline{\beta}_{v_l}^{\min} x_{v_l}^k \le T_{v_l}, \ v_l \in \hat{V}^-, \ p = 1,...,P, \ k = 1,...,K, \tag{22}$$

где значения $\overline{\theta}^k$ определяются по формулам (15), а величины $\overline{\beta}_{v_l}^{\min}$ — в соответствии с выражениями (19), или система неравенств (20), (21), (23):

$$\overline{\theta}^k + \sum_{l=1}^{p} \overline{d}_{v_l}^{k,\min} x_{v_l}^k \le T_{v_l}, \ v_l \in \hat{V}^-, \ p = 1,...,P, \ k = 1,...,K, \tag{23}$$

где значения $\overline{d}_{v_l}^{k,\min}$ вычисляются по формулам (17), то при выбранном распределении заданий по рабочим станциям и выполнении их на каждой машине в последовательности $\widetilde{V}^k$, $k = 1,...,K$ не существует допустимых расписаний, обеспечивающих выполнение всего оставшегося подмножества заданий $\hat{J}$ в установленные ограничениями задачи сроки.

Утверждение 6. Если построены частичные последовательности выполнения некоторого подмножества заданий $\widetilde{V}^k$ на машинах $k = 1,...,K$, определены сроки завершения выполнения этих заданий на каждой из машин $\overline{\theta}^k$, и необходимо распределить по машинам и построить на них расписание выполнения оставшегося подмножества заданий $\hat{V}^-$, то время завершения расписания выполнения всех заданий на этих машинах не может быть меньше величины $\varsigma_1 \ge \varsigma$, определяемых соответственно в результате решения следующих задач линейного булевого программирования:

$$\left\{ \min \varsigma \mid \overline{\theta}^k + \sum_{l=1}^{P} \overline{\beta}_{v_l}^{\min} x_{v_l}^k - \varsigma \le T_{v_l}, \ v_l \in \hat{V}^-, \ k = 1,...,K \right\} \tag{24}$$

в условиях ограничений (20)–(22) или

$$\left\{ \min \varsigma_1 \mid \overline{\theta}^k + \sum_{l=1}^{P} \overline{d}_{v_l}^{k,\min} x_{v_l}^k - \varsigma_1 \le T_{v_l}, \ v_l \in \hat{V}^-, \ k = 1,...,K \right\} \tag{25}$$

в условиях ограничений (20), (21), (23).

Следовательно, результаты решения задач (9)–(11) (или (9), (10), (12)) и (20)–(22) (или (20), (21), (23)) являются соответственно необходимыми, но не достаточными условиями возможности выполнения всей системы ограничений как на начальном этапе решения, так и этапе решения, когда сформированы некоторые частичные подпоследовательности выполнения непересекающихся подмножеств заданий на каждой машине.

Результаты решения задач (9)–(11), (13) (или (9), (10), (12), (14)) и (20)–(22), (24) (или (20), (21), (23), (25)) позволяют вычислить нижнюю границу длины оптимального расписания в решении.

Решение сформулированных оценочных задач булевого линейного программирования при больших размерностях K и N может потребовать значительных объемов вычислений. Поэтому в качестве грубой оценки возможности выполнения всей системы ограничений на различных этапах решения могут рассматриваться результаты решения задач (9)–(11) (или (9), (10), (12)) и (20)–(22) (или (20), (21), (23)) не для всего множества переменных (N — на начальном этапе или $P = N - m$ — в процессе решения), а для некоторой части подлежащих выполнению заданий, стоящих в левой части соответственно последовательностей $\widetilde{I}$ и $\hat{J}$, т.е. для количества заданий $M < N$ или $P_1 < P$).

Для вычисления более грубой, но требующей существенно меньшего объема вычислений оценки, нижней границы функции цели F_1 можно воспользоваться алгоритмом.

Вычислим

$$E = \sum_{l=1}^{P} \beta_{v_l}^{\min}, \; \overline{\theta}_{\max}^{k} = \max_{1 \le k \le K} \overline{\theta}^{k},$$

$$\Delta_k = \overline{\theta}_{\max}^{k} - \overline{\theta}^{k}, \; k = 1,\dots,K; \; E_1 = E - \sum_{k=1}^{K} \Delta^k. \tag{26}$$

Тогда

$$\xi(F_1) = \overline{\theta}_{\max}^{k} + \left| \frac{E_1}{K} \right|. \tag{27}$$

Здесь $|\cdot|$ — целая часть частного от деления.

Пусть $0 \le \lambda^k \le 1$, $k = 1,\dots,K$, $\sum_{k=1}^{K} \lambda^k = 1$ — весовые коэффициенты, определяющие степень важности резерва свободного времени машин.

Утверждение 6. Минимальное средневзвешенное время работы машин, необходимое для выполнения всего комплекса работ, как на начальном этапе решения, так и этапе решения, когда сформированы некоторые частичные подпоследовательности выполнения непересекающихся подмножеств заданий на каждой машине, не может быть меньше значения

$$\xi(F_2) = \sum_{k=1}^{K} \lambda^k \left[\theta^k + \sum_{l=1}^{N} \overline{b}_{i_l}^{k,\min} x_{i_l}^{k,\min} \right] \to \min \tag{28}$$

в условиях ограничений (9)–(11) или (9), (10), (12), либо определяется в результате решения задачи

$$\xi(F_2) = \sum_{k=1}^{K} \left[\overline{\theta}^{k} + \sum_{l=1}^{P} \overline{d}_{v_l}^{k,\min} x_{v_l}^{k} \right] \to \min \tag{29}$$

в условиях ограничений (20)–(22) или (20), (21), (23).

Следовательно, значение $\xi(F_2)$ является нижней границей второго критерия оптимальности на различных этапах решения.

Для вычисления грубой оценки нижней границы функции цели $\xi_1(F_2)$ можно воспользоваться выражениями

$$\xi_1(F_2)=\sum_{i=1}^{N}\beta_i^{\min} \text{ или } \xi_1(F_2)=\sum_{k=1}^{K}\overline{\theta}^{\,k}+\sum_{\substack{l=1,\\ v_l\in\hat{J}}}^{P}\overline{\beta}_{v_l}^{\min}. \qquad (30)$$

Пусть

$$(j^*, k)=\arg\min_{j\in V^-} d_{ij}^k, \qquad (31)$$

т.е. (j^*, k^*) — пара индексов $j\in\hat{J}$, $k=1,\ldots,K$, на которых достигается $\overline{\beta}_i^{\min}=\min_{j\in\hat{J}} d_{ij}$.

Обозначим $i^s(k)$, $k=1,\ldots,K$ — номера заданий, выполняемых в строящихся последовательностях на каждой машине последними.

Вычислим

$$(\hat{j},\hat{k})=\arg\delta_i^{\min}=\arg\{\min_{j\in\hat{J}}\min_{1\le k\le K}\min_{j\in\hat{V}^-} d^{\,k}_{i^s(k),j} \mid (\hat{j},\hat{k})\ne(j^*,k^*)\}. \qquad (32)$$

Ясно, что если в строящемся расписании выполнения заданий будет выбрана не пара индексов (j^*, k^*), а некоторая другая пара, то суммарное время выпонения всех оставшихся невыполненных заданий на всех машинах будет увеличено не менее чем на величину $\Delta_{i^s(k)}=\delta^{\min}_{i^s(k)}-\overline{\beta}^{\min}_{i^s(k)}$. Следовательно, значение $\xi_1(F_2)$ увеличиться не менее чем на величину $\Delta_{i^s(k)}$, а значение $\xi_1(F_1)$ — не менее чем на величину $\left|\dfrac{\Delta_{i^s(k)}}{K}\right|$, где $|.|$ — целая часть частного от деления этих величин. Найдем

$$(i^*, j^*, k^*)=\arg\delta^{\min}_{i^s(k)}=\arg\min_{i\in\hat{V}^-}(\delta^{\min}_{i^s(k)}-\overline{\beta}^{\min}_{i^s(k)})=$$

$$=\min_{i\in\hat{V}^-}[\{\min_{i^s(k)\in\hat{J}}\min_{1\le k\le K}\min_{j\in\hat{V}^-} d^{\,k}_{i^s(k),j} \mid (\hat{j},\hat{k})\}-\min_{i^s(k)\in\hat{J}}\min_{1\le k\le K}\min_{j\in\hat{V}^-} d^{\,k}_{i^s(k),j}]. \qquad (33)$$

Выбор на данном шаге алгоритма для включения в строящиеся последовательности выполнения заданий соответствующего элемента (i^*, j^*, k^*) (т.е. выполнение задания j^* на машине k^*) обеспечит уменьшение границы функции цели задачи на минимальную величину.

АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ

Сформулированные оптимизационные задачи относятся к классу NP-полных задач.

Ниже рассматриваются алгоритмы решения задачи последовательными алгоритмами оптимизации: методом «ветвей и границ» [1, 12] и методом последовательного анализа и отсева неперспективных вариантов [2].

Рассматриваемые ниже методы строят на каждой машине альтернативные допустимые подпоследовательности заданий в порядке их выполнения. Каждое из заданий может и должно входить только в одну из строящихся последовательностей, выполняемых на одной какой-либо машине. На каждом этапе решения задачи эти подпоследовательности расширяются путем включения для какой-то одной из машин нового задания. При этом рассчитываются времена начала и завершения выполнения этого задания и на основании утверждения 3 проверяется допустимость выполнения всех ограничений для этих построенных подпоследовательностей. Если установка выбираемого задания на данной машине на соответствующее место в последовательности является недопустимым, то выбирается следующее альтернативное задание для этой или другой машины. В противном случае на основании утверждений 4, 5 вычисляется нижняя оценка времени выполнения расписания. В методе ветвей и границ на каждом шаге алгоритма выбирается для развития множество подпоследовательностей с наименьшей нижней границей (оценкой), а в методе последовательного анализа вариантов — конструируются и рассматриваются все допустимые множества построенных подпоследовательностей с заданным количеством заданий. Если соответствующее множество подпоследовательностей выполняемых заданий на всех машинах является недопустимым, то значение нижней оценки полагается равным ∞. В методе ветвей и границ факт несовместимости системы ограничений задачи устанавливается в случае, если для всех вершин дерева значение нижней границы равно ∞, а факт получения оптимального решения (если построены последовательности выполнения всех заданий на всех машинах с временем завершения) не большим чем значения оценок всех множеств не построенных до конца подпоследовательностей.

Ниже приводится формальное описание алгоритмов решения задачи.

Упорядочим множество подлежащих выполнению заданий $\widetilde{I}$ в последовательность $\widetilde{U}_1$, т.е. в порядке возрастания граничных времен их завершения в соответствии с выражением (5). Вычислим значения величин b_{ij}^{k}, $\bar{b}_i^{k,\min}$ и β_{ij} по формулам (1)–(2). Сформулируем и решим систему неравенств (9), (10), (12) относительно булевых переменных $x_{i_l}^{k}$. Если данная система неравенств несовместима, то не существует расписаний, обеспечивающих выполнение всей системы ограничений задачи, и алгоритм с этим сообщением заканчивает свою работу. Заметим, что, если сформулированная задача большого размера и требует значительных объемов у вычислений, то она может быть решена для части последовательности, содержащей $N_1 < N$ членов. Решение этой задачи даст более грубую оценку совместности ограничений, так как учитывает только первые N_1 членов последовательности $\widetilde{U}_1$. Если получено решение системы линейных неравенств, то может быть вычислена либо грубая оценка минимальной длины расписания выполнения заданий по формулам (26), (27), либо более точная как результат решения задачи (9), (10), (12), (14). Менее точное значение нижней границы средневзвешенного времени работы машин вычисляется по формуле (30), а более точная оценка определяется в результате решения задачи (9), (10), (12), (28).

ВЫЧИСЛИТЕЛЬНАЯ СХЕМА МЕТОДА ВЕТВЕЙ И ГРАНИЦ

Каждая метка s -й вершины дерева состоит из следующих компонент: q_1^s — номер вершины дерева; q_2^s — признак того, целесообразно ли дальнейшее развитие данной вершины дерева $(q_2^s = 1)$ или нет $(q_2^s = 0)$; $q_3^s = F^s$ — значение нижней границы функции цели для данной вершины дерева; q_4^s — признак того, получено допустимое решение задачи $(q_4^s = 1)$ или нет $(q_4^s = 0)$; $q_5^s = i^s(1)$, $q_6^s = i^s(2),\ldots,q_{4+k}^s = i^s(k),\ldots,q_{4+K}^s = i^s(K)$ — номера заданий, выполняемых в строящихся последовательностях на каждой машине последними; $G^s =| g_i^{sk} |$ — матрица размерностью K строк и N столбцов, элементами которой является место, которое занимает задание i в последовательности выполнения его на машине k. Если для вершины s еще не установлены, что задание i выполняется на k -й машине то $g_i^{sk} = 0$. Если назначено, что это задание выполняется на какой-то другой машине, то соответствующий элемент матрицы равен «–1»; $\overline{\theta}^{sk}$, $k = 1,...,K$ — вектор допустимых времен начала выполнения на k -й машине следующего задания.

Пусть S — количество вершин дерева на данном этапе решения. Если выполнены все описанные в начале данного раздела вычисления и не установлен факт несовместности исходной системы ограничений, то выполняем начальный предварительный шаг.

Шаг 0. Формируем метку корневой вершины дерева $s = 1$ следующим образом:

$q_1^s = 1$; $q_2^s = 1$; $q_4^s = 0$; $q_{4+k}^s = 0$, $k = 1,...,K$; $g_{ki}^s = 0$, $k = 1,...,K$, $i = 1,...,N$.

Значения $\overline{\theta}^{sk}$ полагаем равными $\overline{\theta}^{sk} = \theta^k$, $k = 1,...,K$.

Значение $q_3^s = F^s$ определяем в виде грубой или точной оценки (по результатам решения задачи булевого линейного программирования) и в зависимости от выбранного критерия оптимальности $(F_1^s$ или $F_2^s)$.

Алгоритм решения задачи состоит из некоторого количества итераций, на каждой из которых в определенной логической последовательности выполняются следующие шаги.

Шаг 1. Среди меток вершин дерева, для которых $q_2^s = 1$, $q_4^s = 0$, выбираем метку с наименьшим значением q_3^s, т.е. метку с номером γ, для которой $g_3^\gamma = \{\min_{1\le s\le S} q_3^s \mid q_2^s = 1,\ q_4^s = 0\}$.

Если таких не существует, то переходим к шагу 2. В противном случае переходим к шагу 3.

Шаг 2. Находим среди меток дерева, для которых $q_4^s = 1$, метку с номером ρ, для которой $g_3^\rho = \{\min_{1 \le s \le S} q_3^s \mid q_2^s = 0,\ q_4^s = 1\}$. Если таких меток не существует, то переходим к шагу 5. В противном случае метка дерева с номером ρ является оптимальным решением и переходим к шагу 5.

Шаг 3.

- Определим из матрицы G^γ подмножества заданий $\hat{I}^{\gamma,k,+}$, выбранных в γ-м частичном плане для выполнения на каждой машине (т.е. номера столбцов в каждой k-й строке матрицы G^γ метки с номером γ, элемент которых $g_i^{\gamma k} > 0$). Пусть количество таких элементов в каждой строке соответственно равно $h^{\gamma k}$. Найдем номера заданий, стоящих в подпоследовательностях и выполняемых последними (т.е. $\{\bar{i}(\gamma,k) \mid g_i^{\gamma k} = h^{\gamma k}\}$). Определим также множество всех выполненных на различных машинах заданий $\hat{I}^{\gamma,+} = \bigcup_{k=1}^{K} \hat{I}^{\gamma,k,+}$.
- В каждой матрице $B^k = \left|b_{ij}^k\right|$, $k = 1,\dots,K$ вычеркиваем все столбцы подмножества $j \in \hat{I}^{\gamma,+}$, а также все строки $i \in \hat{I}^{\gamma,+} / \bar{i}(\gamma,k)$ (т.е. все строки, соответствующие индексам выполненных на всех машинах заданий, кроме задания, выполненного на данной машине последним).
- Выполним преобразование 1 вновь полученных матриц B^k и построение матриц D^k.
- Определяем значения $\bar{d}_{v_l}^{k,\min}$, d_{ij} и $\bar{\beta}_{v_l}^{\min}$ по формулам (17)–(19).

4) Определяем грубую или точную оценку возможности выполнения системы ограничений задачи в результате решения системы линейных неравенств с булевыми переменными (20), (21), (23) размерностью P или $P_1 < P$. Если система неравенств несовместна, то подмножество не содержит допустимых планов. Полагаем $q_2^\gamma = 0$, $F^\gamma = \infty$ и переходим к шагу 1. Если получено решение системы неравенств, то вычисляем грубую (по формулам (30)) или точную оценку функции цели задачи (в результате решения задачи булевого линейного программирования (20), (21), (23) с критерием оптимальности (28) или (29)).

По результатам решения одной из оптимизационных задач корректируем для γ-й метки вершины дерева значение оценки функции цели (признак $q_3^\gamma = F^\gamma$) и переходим к шагу 4.

Шаг 4 (разбиение на подмножества). Для метки с номером γ, выполнив вычисления (31)–(33), находим элемент $(i^*,\ j^*,\ k^*)$, т.е. номер машины k^* и номер задания j^*, которое должно быть выбрано на этой машине и установлено на следующее место последовательности после выполненного на этой же машине задания $i^* = i^\gamma(k^*) = q^\gamma_{4+k^*}$.

Формируем $(S+1)$-ю вершину дерева, признаки которой определяются следующим образом:

$$q_1^{S+1} = S+1,\ q_2^{S+1} = 1,\ q_3^{S+1} = F^\gamma,\ q_{4+k}^{S+1} = q_{4+k}^\gamma,\ k = 1,\dots,K,\ k \neq k^*;$$

$$q_{4+k}^{S+1} = j^*;\ q_i^{S+1,k} = q_i^{\gamma,k},\ i = 1,\dots,N,\ k = 1,\dots,K,\ k \neq k^*;$$

$$g_i^{S+1,k^*} = g_i^{\gamma,k^*},\ i = 1,\dots,N,\ i \neq j^*;$$

$$g_i^{S+1,k^*} = (g_i^{\gamma,k^*} + 1);$$

$$\overline{\theta}^{S+1,k} = \overline{\theta}^{\gamma,k},\ k = 1,\dots,K,\ k \neq k^*;\ \overline{\theta}^{S+1,k^*} = \overline{\theta}^{\gamma,k^*} + b^{k^*}_{i(k^*),j^*}.$$

Формируем $(S+2)$-ю вершину дерева, признаки которой определяются следующим образом:

$$q_1^{S+2} = S+2,\ q_2^{S+2} = 1,\ q_3^{S+2} = F_1^\gamma + \left|\frac{\Delta_{i(k^*)}}{K}\right| \text{ (или } q_3^{S+2} = F_2^\gamma + \Delta_{i(k^*)});$$

$$q_{4+k}^{S+2} = q_{4+k}^\gamma,\ k = 1,\dots,K;$$

$$g_i^{S+2,k} = g_i^{\gamma,k},\ i = 1,\dots,N,\ k = 1,\dots,K,\ k \neq k^*;$$

$$g_i^{S+1,k^*} = g_i^{\gamma,k^*},\ i = 1,\dots,N,\ i \neq j^*;\ g_i^{S+1,k^*} = -1;$$

$$\overline{\theta}^{S+2,k} = \overline{\theta}^{\gamma,k},\ k = 1,\dots,K.$$

В метке вершины γ изменяем только один признак, полагая $q_2^\gamma = 0$.

Полагаем $(S+2) := S$ и переходим к шагу 1.

Шаг 5. Восстанавливаем оптимальные расписания выполнения заданий на каждой машине и рассчитываем времена начала и завершения выполнения каждого из этих заданий, а также соответствующие потери времени на переналадки. Формируем матрицу последовательностей выполнения заданий $\widetilde{W}$ на каждой машине, включающую K строк и N столбцов, а также трехмерный массив времен выполнения заданий $T = \left|\tau_{lj}^k\right|$, размерностью $k = 1,\dots,K$, $l = 1,\dots,4$, $j = 1,\dots,N$. В первой строке каждой двухмерной матрицы T^k указываются: τ_{1j}^k — время начала настройки k-й машины для выполнения задания j; τ_{2j}^k, τ_{3j}^k — соответственно времена начала и завершения на k-й машине выполнения задания j.

В начале процесса полагаем, что все члены этой матрицы последовательностей равными 0 — $w_i^k = 0$, $k = 1,...,K$, $i = 1,...,N$. Все элементы трехмерной матрицы T положим также равными 0. Введем также некоторое вспомогательный вектор H^k, компоненты которого в начале процесса положим равными $H^k = \theta^k$.

Введем также вектор $\pi = (\pi^k)$. В начале процесса положим $\pi^k = 0$, $k = 1,...,K$.

Для каждой k-й строки матрицы G^γ выполняем следующий объем вычислений.

Для $i = 1,...,N$ выполним: если $g_i^{\gamma k} > \pi^k$, полагаем $\pi^k = g_i^{\gamma k}$. После завершения этой процедуры значения π^k определят количество заданий, выполняемых в оптимальном расписании на каждой машине.

Полагаем $\lambda^k = 1$. Пусть K — текущий номер задания, соответствующего предыдущему члену последовательности. Значение ω^k в начале процесса полагаем равным нулю. Переходим к пункту (а).

а) Среди элементов вектора $G^{\gamma k}$ находим элемент $g_\mu^{\gamma k} = \lambda^k$ (μ-й элемент вектора). Полагаем $\omega_i^k = \lambda^k$. Полагаем $\tau_{1\mu}^k = H^k$, $\tau_{2\mu}^k = \tau_{1\mu}^k + a_{\omega\mu}^k$, $\tau_{3\mu}^k = \tau_{2\mu}^k + t_\mu^k$. Полагаем $\omega^k = \mu^k$, $H^k = \tau_{3\mu}^k + 1$. Переходим к пункту (б).

б) Полагаем $\lambda^k = \lambda^k + 1$. Если $\lambda^k = \pi^k + 1$, то определены все характеристики оптимального расписания выполнения заданий, и алгоритм заканчивает работу. В противном случае переходим к пункту (а).

Заметим, что если поставлена задача построения допустимого расписания выполнения заданий, то 2-й шаг алгоритма выглядит таким же образом.

В качестве частных случаев рассматриваемой задачи в работе автора [14] предложены математические модели и более простые алгоритмы построения допустимых и оптимальных расписаний выполнения заданий на одной машине, требуещие существенно меньшего объема вычислений.

ВЫВОД

В работе предложены математические модели оптимального разбиения на непересекающиеся подмножества всего комплекса заданий, подлежащих выполнению на нескольких машинах, и построения оптимальных последовательностей их выполнения на каждой из машин. Сформулированная задача решается в условиях учета потерь времени на переналадки или потери времени при переходе от выполнения одного задания к другому, а также с учетом заданных ограничений на начальные и конечные сроки выполнения каждого из заданий и относится к классу NP-сложных задач. На основе установленных в работе свойств допустимых и оптимальных решений предложены эффективные точные и приближенные (в случае использования левосторонней схемы ветвления) алгоритмы решения последовательными методами оптимизации, которые уже на ранних этапах решения позволяют установить факт несовместности исходной системы ограничений и при небольшом объеме вычислений получить эффективное приближенное решение. Полученные в работе результаты могут найти широкое применение в календарном планировании производства, организации технического обслуживания объектов, планировании параллельных вычислений и в маршрутизации перевозок.

ЛИТЕРАТУРА.

1. *Танаев В.С., Шкурба В.С.* Введение в теорию расписаний. — М.: Наука, 1975. — 256 с.
2. *Танаев В.С., Ковалев М.Я., Шафранский Я.М.* Теория расписаний. Групповые технологии. — Минск: ИТК НАН Беларуси, 1998. — 289 с.
3. *Конвей Р.В., Максвелл В.Л., Миллер Л.В.* Теория расписаний. — М.: Наука, 1975. — 360 с.
4. *Курейчик В.М.* Генетические алгоритмы: монография. — Таганрог: Изд. ТРТУ, 1998. — 242 с.
5. *Батищев Д.И., Гудман Э.Д., Норенков И.П., Прилуцкий М.Х.* Метод комбинирования эвристик для решения комбинаторных задач упорядочения и распределения ресурсов // Информационные технологии. — № 2. — 1997. — С. 29–32.
6. *Pinedo M.* Scheduling: Theory, Algorithms and Systems. — Berlin-Heidelberg: Springer-Verlag, 1999. — 671 p.
7. *Brucker P.* Scheduling algorithms. — Berlin: Springer-Verlag, 2007. — 371 p.
8. *Blazewicz J., Domschke W., Pesch E.* The job shop scheduling problem: Conventional and new solution techniques // European Journal of Operational Research. — 1996. — Vol. 93. — P. 1–33.
9. *Herrmann J.* Supply Chain Scheduling. Transaktionskostentheorie; Parallele Maschinen; Heuristik; Optimierungsmodelle. — Berlin-Heidelberg: Gabler Verlag, 2010. — 162 p.
10. *Blazewicz J., Ecker K.H., Pesch E., Schmidt G., Weglarz J.* Scheduling Computer and Manufacturing Processes. — Berlin-Heidelberg: Springer-Verlag, 2001. — 485 p.
11. *Szelke E., Kerr R.M.* Artificial Intelligence in Reactive Scheduling. — Chapman & Hall, London, 1995. — 164 p.
12. *Зак Ю.А.* Определение порядка выполнения независимых операций на параллельных машинах // Изв. АН СССР. Техническая кибернетика. — 1969. — № 2. — С. 15–20.
13. *Зак Ю.А.* Решение обобщенной задачи Джонсона с ограничениями на сроки выполнения заданий и времена работы машин. Ч.1. Точные методы решения // Проблемы управления. — 2010. — № 3. — С. 17–25; Ч.2. Приближенные методы решения // Проблемы управления. — 2010. — № 4. — С. 12–19.
14. *Зак Ю.А.* Оптимизация планирования производства и раскроя бумажной продукции // Управляющие системы и машины. — 2010. — № 5. — С. 82–93.

Поступила 07.10.2010