

МИНИМИЗАЦИЯ СУММАРНОГО ВРЕМЕНИ ОБСЛУЖИВАНИЯ ТРЕБОВАНИЙ С ЕДИНИЧНЫМИ ДЛИТЕЛЬНОСТЯМИ ОПЕРАЦИЙ НА ОСНОВЕ РАСКРАСКИ СМЕШАННОГО ГРАФА

*Оптимальная раскраска φ смешанного графа $G=(V,A,E)$ определяет расписание, которое минимизирует среднее время обслуживания n требований в системе *job shop* с единичными длительностями операций. Подграф (V,A,\emptyset) смешанного графа G представляет собой объединение путей, а подграф (V,\emptyset,E) – объединение клик. Разработан метод ветвей и границ для оптимальной раскраски смешанного графа G с критерием минимизации суммы номеров цветов, используемых для n требований. Проведен вычислительный эксперимент на ПЭВМ по построению оптимальной раскраски вершин смешанных графов порядка $n \leq 200$, сгенерированных случайным образом.*

Введение

В 1976 г. В.С. Танаев предложил для исследования следующую задачу раскраски (вершин) смешанного графа [1]. Пусть $G = (V,A,E)$ – конечный смешанный граф с непустым множеством вершин V , множеством дуг A и множеством ребер E . Функция φ называется раскраской смешанного графа G , если для каждой вершины $v_i \in V$ она определяет натуральное число (называемое цветом) $\varphi(v_i) \in N$ так, что из включения $(v_i, v_j) \in A$ следует соотношение

$$\varphi(v_i) \leq \varphi(v_j), \quad (1)$$

а из включения $[v_i, v_j] \in E$ – соотношение

$$\varphi(v_i) \neq \varphi(v_j). \quad (2)$$

Здесь и далее N обозначает множество натуральных чисел. Как и в случае обычной раскраски вершин неориентированного графа (V,\emptyset,E) , в задаче требуется построить такую раскраску смешанного графа $G = (V,A,E)$, $A \neq \emptyset$, $E \neq \emptyset$, которая содержит минимальное число различных цветов. В настоящей статье показано, как оптимальная раскраска вершин смешанного графа с суммарным критерием может быть использована для решения задачи минимизации среднего времени обслуживания m приборами n требований с единичными длительностями операций в многостадийной системе с различными маршрутами. Далее вместо соотношения (1) в определении допустимой раскраски $\varphi: V \rightarrow N$ смешанного графа G будет использоваться строгое неравенство

$$\varphi(v_i) < \varphi(v_j) \quad (3)$$

в соответствии с определением раскраски смешанного графа, приведенным в работе [2]. Используемая далее терминология по теории расписаний соответствует терминологии монографии [3], а по теории графов – монографии [4].

1. Постановка и сложность задачи построения оптимального расписания

Рассмотрим следующую задачу минимизации суммарного времени обслуживания n требований m приборами в системе *job shop* [3]. Множество требований $J = \{J_1, J_2, \dots, J_n\}$ должно быть обслужено в детерминированной системе, состоящей из множества последовательных приборов $M = \{M_1, M_2, \dots, M_m\}$. В любой момент времени каждый прибор $M_k \in M$ может обслуживать не более одного требования, а каждое требование $J_i \in J$ может обслуживаться не более чем одним прибором. В системе *job shop* обслуживание требования $J_i \in J$ включает $r_i > 1$ операций $(v_{i1}, v_{i2}, \dots, v_{ir_i})$, которые должны выполняться последовательно приборами

$M_{i_1}, M_{i_2}, \dots, M_{i_r}$ множества M в соответствии с заданным (технологическим) маршрутом. В маршруте $(M_{i_1}, M_{i_2}, \dots, M_{i_r})$ обслуживания требования $J_i \in J$ допускаются повторения прибором и (или) отсутствие некоторых приборов множества M . Прерывания операции запрещены, и для выполнения любой операции v_{ij} соответствующим прибором M_{i_j} требуется единица времени. Иными словами, для каждой операции v_{ij} должно выполняться равенство $c_{ij} = s_{ij} + 1$, где s_{ij} и c_{ij} обозначают соответственно момент начала и момент окончания операции v_{ij} . Рассматриваемая задача состоит в построении такого расписания обслуживания n требований m приборами, при котором целевая функция $\Phi(C_1, C_2, \dots, C_n) = \sum_{i=1}^n C_i$ принимает наименьшее

значение. Здесь и далее C_i обозначает момент завершения обслуживания требования $J_i \in J$. Следовательно, имеет место равенство $C_i = c_{ir_i}$. Для такой задачи используется обозначение $J/p_{ij}=1/\sum C_i$, если маршруты обслуживания требований могут быть различными, и обозначение $F/p_{ij}=1/\sum C_i$, если маршруты обслуживания всех требований одинаковы и каждый прибор $M_k \in M$ встречается в маршруте обслуживания каждого требования точно один раз.

В статье [5] было доказано, что задача $F/p_{ij} \in \{0,1\}/\sum C_i$ является NP-трудной. Здесь включение $p_{ij} \in \{0,1\}$ означает, что длительность операции может равняться либо нулю, либо единице. При этом нулевая длительность $p_{ij} = 0$ операции v_{ij} означает, что требование $J_i \in J$ соответствующим прибором M_{i_j} вообще не обслуживается. Нетрудно видеть, что задача $F/p_{ij} \in \{0,1\}/\sum C_i$ является частным случаем задачи $J/p_{ij}=1/\sum C_i$ и, следовательно, задача $J/p_{ij}=1/\sum C_i$ также является NP-трудной.

В работах [6–10] было показано, как задачи теории расписаний с критерием минимизации максимального времени обслуживания требований $\Phi(C_1, C_2, \dots, C_n) = \max\{C_i : J_i \in J\} = C_{max}$ и единичными длительностями операций сводятся к задаче раскраски неориентированного графа (V, \emptyset, E) или смешанного графа G в минимальное число цветов. В следующем разделе показано, что задача $J/p_{ij}=1/\sum C_i$ может интерпретироваться как задача построения оптимальной раскраски смешанного графа с суммарным критерием. На основе такой интерпретации разработан метод ветвей и границ для решения задачи $J/p_{ij}=1/\sum C_i$, который оказался достаточно эффективным при решении серии задач, сгенерированных случайным образом.

2. Сведение задачи $J/p_{ij}=1/\sum C_i$ к раскраске смешанного графа G

Задача $J/p_{ij}=1/\sum C_i$ может быть сформулирована как задача построения оптимальной раскраски смешанного графа G , в котором множество вершин V представляет собой множество операций. Ребра множества E связывают попарно операции, выполняемые одним и тем же прибором. В результате получается следующее разбиение неориентированного подграфа (V, \emptyset, E) смешанного графа G :

$$(V, \emptyset, E) = (V^1, \emptyset, E_1) \cap (V^2, \emptyset, E_2) \cap \dots \cap (V^m, \emptyset, E_m), \quad (4)$$

в котором неориентированный граф (V^k, \emptyset, E_k) представляет собой клику порядка $|V^k|$ с множеством операций (вершин) $V^k \subset V$, выполняемых прибором $M_k \in M$. Множество дуг A определяет ограничения на порядок выполнения операций множества V . Поскольку маршрут обслуживания каждого требования $J_i \in J$ определяет линейно упорядоченное множество V_i всех операций по обслуживанию требования J_i , то получаем следующее разбиение ориентированного подграфа (V, A, \emptyset) смешанного графа G :

$$(V, A, \emptyset) = (V_1, A_1, \emptyset) \cap (V_2, A_2, \emptyset) \cap \dots \cap (V_n, A_n, \emptyset), \quad (5)$$

в котором ориентированный граф (V, A, \emptyset) представляет собой путь $(v_{i1}, v_{i2}, \dots, v_{ir_i})$ (т.е. определяет линейный порядок операций). Таким образом, для вершин V смешанного графа G выполняются следующие равенства:

$$V = \bigcup_{i=1}^n V_i = \bigcup_{k=1}^m V^k. \quad (6)$$

Поскольку длительность каждой операции в задаче $J/p_{ij}=1/\sum C_i$ равна единице, то раскраска $\varphi: V \rightarrow \{1, 2, \dots, t\} \subset N$ вершин смешанного графа G определяет допустимое назначение (расписание) операций множества V единичным интервалам: $[0, 1], (1, 2], (2, 3], \dots, (t-1, t]$.

Из соотношений (2) и (3) следует, что оптимальное расписание задачи $J/p_{ij}=1/\sum C_i$ соответствует раскраске, для которой минимальное значение принимает величина

$$\sum_{i=1}^n \varphi(v_{ir_i}). \quad (7)$$

В дальнейшем такую раскраску смешанного графа G будем называть оптимальной.

Итак, получено следующее взаимно-однозначное соответствие между раскрасками φ смешанного графа G , удовлетворяющего условиям (4) – (6), и расписаниями для задачи $J/p_{ij}=1/\sum C_i$:

$$\begin{aligned} \{ \text{вершина } v_j \} &\leftrightarrow \{ \text{операция } v_j \}; \\ \{ \text{множество вершин пути } (V_i, A_i, \emptyset) \} &\leftrightarrow \{ \text{множество операций требования } J_i \}; \\ \{ \text{множество вершин } V^k \text{ клики} \} &\leftrightarrow \{ \text{множество операций, выполняемых прибором } M_k \}; \\ \{ \text{раскраска } \varphi \} &\leftrightarrow \{ \text{расписание } \varphi \}; \\ \{ \text{оптимальная раскраска } \varphi \} &\leftrightarrow \{ \text{оптимальное расписание } \varphi \}. \end{aligned}$$

Отметим, что в случае произвольного регулярного критерия $\Phi(C_1, C_2, \dots, C_n)$ [3] задача job shop полиномиально разрешима при $n=2$. В статье [11] были разработаны полиномиальные алгоритмы построения оптимальных расписаний обслуживания двух требований, основанные на геометрическом представлении расписаний для задач $J/n=2/\Phi$ и $J/n=2, Pr/\Phi$. Здесь и далее Pr обозначает допустимость прерываний операции в процессе ее выполнения.

Пусть $P_i = \sum_{j=1}^{r_i} p_{ij}$. В геометрическом алгоритме длительности выполнения операций требования J_1 (требования J_2) откладываются по оси абсцисс (по оси ординат). Точка (x, y) в прямоугольнике Z , ограниченном точками $(0, 0)$, $(P_1, 0)$, $(0, P_2)$ и (P_1, P_2) , определяет состояние обслуживания обоих требований в момент времени, определенный длиной траектории от точки $(0, 0)$ до точки (x, y) . При этом используется чебышевская метрика, т. е. расстояние $d((x, y), (x', y'))$ между точками (x, y) и (x', y') определяется по следующей формуле:

$$d((x, y), (x', y')) = \max\{|x - x'|, |y - y'|\}.$$

В работе [11] было доказано, что задачи $J/n=2/\Phi$ и $J/n=2, Pr/\Phi$ полиномиально сводятся к построению ориентированного графа в системе координат на плоскости и поиску кратчайшего пути из вершины $(0, 0)$ в вершину (P_1, P_2) .

Основная идея геометрического алгоритма [11] состоит в стремлении строить участки траектории в прямоугольнике Z под углом 45° к оси абсцисс всякий раз, когда возможно одновременное обслуживание требований J_1 и J_2 разными приборами из множества M . В следующем разделе статьи, по сути, эта же идея используется для случая произвольного числа требований. Если $n > 2$, то геометрическое представление из работы [11] процесса обслуживания требований множества J уже не удастся использовать так эффективно, как в случае $n = 2$.

Следует отметить, что в работе [12] была доказана NP-трудность задач $J/n=3/\sum C_i$ и $J/n=3, Pr/\sum C_i$. В работе [13] была доказана NP-трудность и задачи $J2/n=3, Pr/\sum C_i$. Заметим, что все известные результаты по NP-полноте и NP-трудности задач получены при предположении целочисленности числовых параметров. Учитывая это предположение, нетрудно убедиться в том, что задача $J2/n=3, Pr/\sum C_i$ полиномиально сводится к задаче $J2/n=3, p_{ij}=1/\sum C_i$. Следовательно, точное решение задачи $Jm/n=n^0, p_{ij}=1/\sum C_i$ при $m \geq 2$ и $n^0 \geq 3$ подразумевает использование явного или неявного перебора активных расписаний [3], например, по схеме метода ветвей и границ.

3. Описание алгоритма ветвей и границ

Для решения задачи $J/p_{ij}=1/\sum C_i$ разработан алгоритм ветвей и границ (будем называть его алгоритмом MINSUM) построения оптимальной раскраски смешанного графа G , допускающего разбиения (4) и (5). В алгоритме MINSUM используются нижние и верхние оценки величины (7), а также процедура ветвления (разбиения множества допустимых раскрасок на несколько непересекающихся подмножеств) с целью разрешения конфликтов, возникающих при раскраске смежных вершин.

В результате реализации алгоритма MINSUM получается (ориентированное) дерево решений T . Для того чтобы избежать излишних формальностей при описании алгоритма MINSUM, будем использовать пример задачи $J5/p_{ij}=1, n=3/\sum C_i$ (рис. 1). Для этого примера маршруты обслуживания требований J_1, J_2 и J_3 заданы соответственно следующими: $(M_1, M_2, M_3, M_4, M_2, M_2, M_5)$, $(M_2, M_1, M_2, M_4, M_3, M_5, M_5)$ и $(M_3, M_1, M_4, M_3, M_5)$.

На рис. 1 вершины множества V представлены в виде квадратов, причем обозначение вершины $v_{ij} \in V$ указано под соответствующим квадратом. Изображения операций множества V^i имеют одинаковый рисунок внутри соответствующих им квадратов. Маршрут требования J_1 указан над квадратами, изображающими его операции.

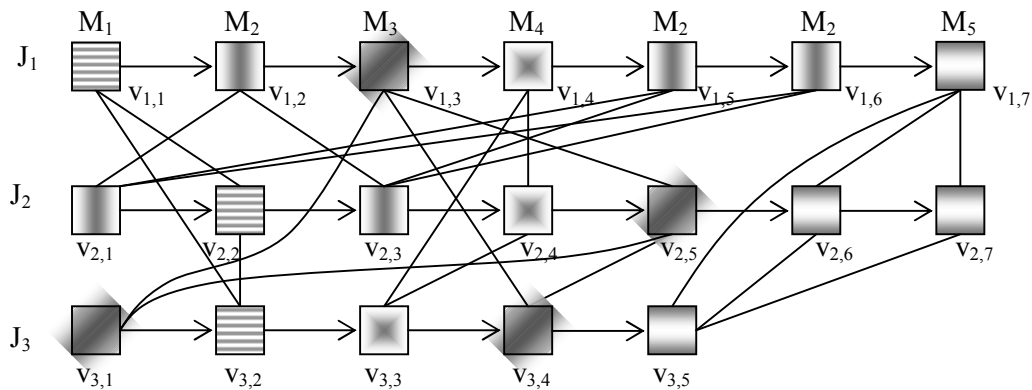


Рис. 1. Смешанный граф G для задачи $J5/p_{ij}=1, n=3/\sum C_i$

Дерево решений $T = (W, R, \emptyset)$ для рассматриваемого примера $J5/p_{ij}=1, n=3/\sum C_i$ изображено на рис. 2. Каждая вершина $w^{(i)} \in W$ дерева решений T представляет собой целочисленный n -мерный вектор, определяющий состояние процесса раскраски вершин:

$$w^{(i)} = \begin{pmatrix} w_1^{(i)} \\ w_2^{(i)} \\ \dots \\ w_n^{(i)} \end{pmatrix}.$$

Первая компонента $w_1^{(i)}$ вектора $w^{(i)}$ соответствует требованию J_1 , вторая компонента $w_2^{(i)}$ – требованию J_2 и т. д. Последняя компонента $w_n^{(i)}$ вектора $w^{(i)}$ соответствует требованию J_n . Под очередной итерацией алгоритма будем подразумевать построение вершины дерева решений, поэтому номер итерации совпадает с номером вершины. На начальной (нулевой) итерации алгоритма нулевой вектор $w^{(0)}$ (рис. 2) указывает на то, что ни одна вершина смешанного графа G еще не окрашена.

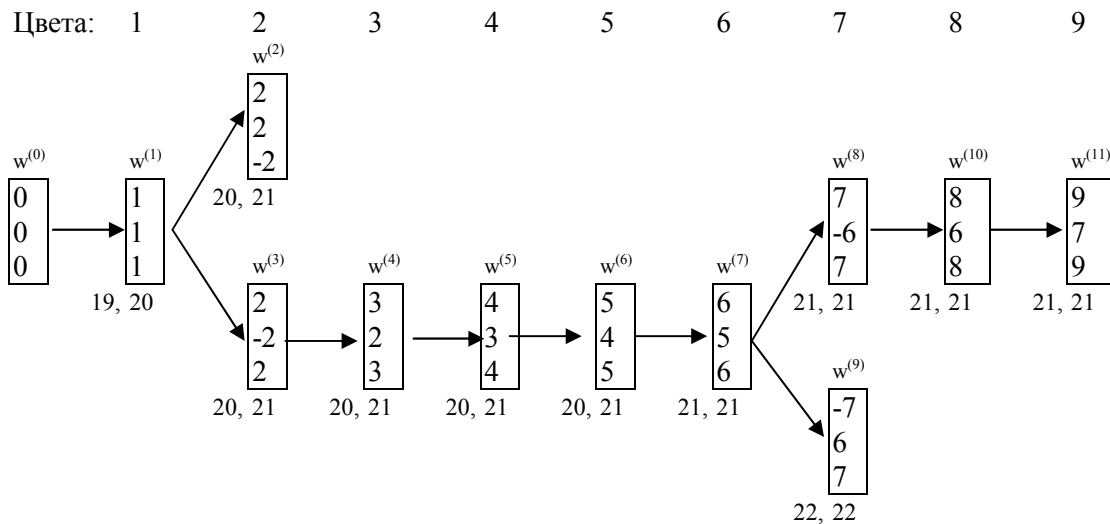


Рис. 2. Дерево решений $T = (W, R, \emptyset)$

На первой итерации максимальное количество вершин множества $\{v_{1,1}, v_{2,1}, \dots, v_{n1}\}$ красится в цвет 1 при соблюдении условия (2). В рассматриваемом примере на первой итерации всем трем вершинам $v_{1,1}$, $v_{2,1}$ и $v_{3,1}$ приписан цвет 1: $\varphi(v_{1,1}) = \varphi(v_{2,1}) = \varphi(v_{3,1}) = 1$, поскольку между ними нет ни одного ребра. В общем случае на i -й итерации каждая компонента $w_k^{(i)}$ вектора полагается равной либо индексу l вершины v_{kl} из пути $(v_{k1}, \dots, v_{kl}, \dots, v_{kk})$, если вершина v_{kl} окрашивается на этой итерации, либо $-l$ в случае, если на этой итерации очередная вершина v_{kl} из множества V_k не была покрашена. Последняя ситуация имеет место, если в очередной цвет c , используемый на i -й итерации, покрашена вершина $v_{lu} \in V$, такая, что $[v_{kb}, v_{lu}] \in E$. В частности, в нашем примере на второй итерации возник конфликт, поскольку вершины $v_{2,2}$ и $v_{3,2}$ соединены ребром: $[v_{2,2}, v_{3,2}] \in E$. В результате разрешения этого конфликта построены две вершины дерева решений T , определяемые векторами

$$w^{(2)} = \begin{pmatrix} 2 \\ 2 \\ -2 \end{pmatrix} \text{ и } w^{(3)} = \begin{pmatrix} 2 \\ -2 \\ 2 \end{pmatrix}.$$

Здесь равенство $w_3^{(2)} = -2$ (равенство $w_2^{(3)} = -2$ соответственно) указывает на то, что в случае $w^{(2)}$ (в случае $w^{(3)}$) неокрашенной вершиной требования J_3 (требования J_2) была вершина $v_{3,2}$ (вершина $v_{2,2}$). Каждая дуга $(w^{(i)}, w^{(k)})$ дерева решений T переводит состояние $w^{(i)}$ процесса раскраски вершин в цвет $c-1$ в состояние $w^{(k)}$ раскраски очередных вершин из множества V в цвет c . Число дуг множества R , исходящих из вершины $w^{(i)} \in W$, равно произведению чисел вершин из множества V , покраска которых в цвет $c-1$ приводит к конфликту в силу условия (2). Таким образом, в дереве $T = (W, R, \emptyset)$ дуга $(w^{(i)}, w^{(k)})$ принадлежит множеству R тогда и только тогда, когда вектор $w^{(k)}$ образован непосредственно из вектора $w^{(i)}$ в результате использования очередного цвета c . В свою очередь, вектор $w^{(i)}$ образован в результате использования цвета $c-1$. Используемые на очередной итерации цвета для рассматриваемого примера указаны в верхней части рис. 2. Вершина дерева решений T является висячей в случае, когда она опреде-

ляет раскраску всех вершин смешанного графа G или, нижняя оценка величины (7), которая описана в следующем разделе, не меньше полученной к этому моменту верхней оценки величины (7).

4. Оценки целевой функции

В алгоритме MINSUM использованы три оценки снизу величины (7): одна локальная $LB0$ и две глобальные $LB1$ и $LB2$. Если процесс раскраски находится в состоянии, которое описывается вектором $w^{(j)}$, то для окраски вершин множества V_i требуется не менее $r_i + l_i^j$ цветов, где l_i^j обозначает число цветов, пропущенных для вершин V_i в процессе их раскраски вплоть до состояния $w^{(j)}$ включительно. Следовательно, в качестве локальной оценки величины (7) в состоянии $w^{(j)}$ можно использовать следующую сумму:

$$LB0 = \sum_{i=1}^n (r_i + l_i^j). \quad (8)$$

Важно отметить, что для вычисления оценки $LB0$ достаточно локальной информации, а именно, достаточно знать вектор $w^{(j)}$ и вектор (r_1, r_2, \dots, r_n) .

Вычисление глобальных оценок $LB1$ и $LB2$ более трудоемко и требует определенного поиска информации по дереву решений T . Вначале докажем справедливость оценки $LB1$ в случае существования прибора $M_k \in M$, который обслуживает каждое требование $J_i \in J$ точно один раз, т. е. для всех требований $J_i \in J$ выполняется равенство

$$V_i \cap V^k = \{v_{ij}\}, \quad (9)$$

где $M_{i_j} = M_k$. Обозначим $h_{\min} = \min_{J_i \in J} \{j-1 : M_{i_j} = M_k\}$, $t_{\min} = \min_{J_i \in J} \{r_i - j : M_{i_j} = M_k\}$. Содержательно h_{\min} (t_{\min}) означает длину минимального пути в орграфе (V, A, \emptyset) , заканчивающегося (начинающегося) в вершине, принадлежащей множеству V_k . Поскольку каждый прибор может выполнять не более одной операции в единицу времени, то операции множества V^k , выполняемые прибором M_k , должны быть линейно упорядочены. Следовательно, сумма цветов для раскраски всех вершин V^k должна быть больше или равна $\frac{n(n+1)}{2}$. Соответственно сумма цветов

для всех вершин множества V должна быть больше или равна величине $LB1(k) = nh_{\min} + \frac{n(n+1)}{2} + nt_{\min}$. В результате получаем следующую оценку снизу величины (7):

$$\sum_{i=1}^n \varphi(v_{i r_i}) \geq LB1(k) = nh_{\min} + \frac{n(n+1)}{2} + nt_{\min}. \quad (10)$$

Если условие (9) не выполняется, то аналитическое выражение оценки $LB1(k)$ можно получить в результате следующей модификации равенства (10). Пусть $J(M_k)$ обозначает подмножество требований $\{J_{i_1}, J_{i_2}, \dots, J_{i_r}\} \subseteq J$, для каждого из которых выполняется соотношение $V_i \cap V^k \neq \emptyset$, $t \in \{1, 2, \dots, s\}$. Очевидно, что для системы с одним прибором M_k оптимальной для критерия $\sum C_i$ будет перестановка требований $J(M_k)$, расположенных в порядке невозрастания числа операций на приборе M_k . Пусть $\sum_{J_i \in J(M_k)} C_i^*$ обозначает значение целевой функции, определенное для такой перестановки. Тогда вместо равенства (10) получаем следующее представление оценки $LB1(k)$ в общем случае:

$$LB1(k) = sh_{\min}^* + \sum_{J_i \in J(M_k)} C_i^* + st_{\min}^* + \sum_{J_j \in J \setminus J(M_k)} r_j. \quad (11)$$

В равенстве (11) значения h_{\min}^* и t_{\min}^* вычисляются следующим образом:

$$h_{\min}^* = \min_{J_i \in J(M_k)} \{j-1 : v_{ij} \in V_i \cap V^k, M_{i_j} = M_k\};$$

$$t_{\min}^* = \min_{J_i \in J(M_k)} \{r_i - j : v_{ij} \in V_i \cap V^k, M_{i_j} = M_k\}.$$

В качестве нижней оценки $LB1$ величины (7) берется наибольшее значение величины $LB1(k)$ по всем приборам $M_k \in M$. Приведенные оценки (10) и (11) достижимы для первой итерации алгоритма. Нетрудно видеть, что на последующих итерациях значение оценки $LB1(k)$ можно увеличить, поскольку некоторые вершины из множества V^k будут уже покрашены и соответственно может увеличиться значение h_{\min}^* . Кроме того, определенный до текущей итерации порядок обслуживания требований множества $J(M_k)$ может быть отличен от порядка, определенного невозрастанием числа операций на приборе M_k .

Оценка $LB2$ получается как обобщение локальной оценки $LB0$. Пусть $n(c)$ обозначает максимально возможное число требований, вершины которых могут быть окрашены в процессе реализации алгоритма MINSUM в цвет c . В частности, полагаем $n(1)=n$. На последующих итерациях значение $n(c)$ может стать меньше n , если все вершины одного или нескольких требований будут покрашены в цвета, меньшие чем c . Пусть $m(c)$ обозначает число приборов, требуемых для выполнения операций в момент времени c , а также операций, пропущенных при раскраске на предыдущих итерациях. Значение $m(c)$ определяется на основе соответствующего вектора $w^{(i)}$.

В частности, $m(1)$ равно числу приборов, необходимых для выполнения операций множества $\{v_{1,l}, v_{2,l}, \dots, v_{n,l}\}$. Если на первой итерации (т. е. при раскраске в цвет 1) алгоритма MINSUM требуется $m(1)$ приборов, то в оценке

$$\sum_{i=1}^n \varphi(v_{ir_i}) \geq \sum_{i=1}^n r_i \quad (12)$$

можно добавить к правой части неравенства (12) разность $n(1)-m(1)$ (поскольку в единицу времени каждый прибор может выполнить только одну операцию).

Соответственно при раскраске в цвет 2 к правой части неравенства (12) можно добавить величину $\max\{0, n(2)-m(2)\}$. В общем случае при раскраске в цвет c к правой части неравенства (12) можно добавить также величину $\max\{0, n(c)-m(c)\}$. В результате получается следующая оценка:

$$\sum_{i=1}^n \varphi(v_{ir_i}) \geq LB2 = \sum_{i=1}^n r_i + \sum_{j=1}^r \max\{0, n(j) - m(j)\}, \quad (13)$$

где $r = \max\{r_i : J_i \in J\}$. Оценка $LB2$ (так же, как и оценка $LB1$) должна вычисляться на каждой итерации алгоритма. На рис. 2 оценки $LB1$ и $LB2$ представлены под соответствующими вершинами дерева решений.

Для вершин $w^{(0)}$, $w^{(1)}$, $w^{(2)}$ и т. д. вплоть до построения первой раскраски смешанного графа верхняя оценка полагается равной сумме $r_1 + (r_1 + r_2) + \dots + \sum_{i=1}^n r_i$, определяющей значение целевой функции для раскраски, в которой требования J_1, J_2, \dots, J_n обслуживаются последовательно одно за другим в порядке возрастания их номеров. Если построена хотя бы одна раскраска всех вершин смешанного графа, то для получения верхней оценки величины (7) используется рекордная раскраска вершин (т. е. наилучшая на данной итерации алгоритма).

5. Результаты вычислительных экспериментов

Алгоритм MINSUM реализован на алгоритмическом языке C++ и протестирован на PC Pentium III (600 MHz) для псевдослучайных смешанных графов, удовлетворяющих условиям (4) и (5). Во всех экспериментах, результаты которых представлены в табл. 1 – 5, рассматривался наиболее сложный для данного алгоритма случай, когда $r_1 = r_2 = \dots = r_n$. Были реализованы две версии алгоритма MINSUM. В первой версии использовалась только глобальная оценка $LB1$, которая показала лучшие результаты на предварительной стадии вычислительного эксперимента. Во второй версии алгоритма использовались обе глобальные оценки $LB1$ и $LB2$. Локальная оценка $LB0$, которая вычисляется достаточно просто, применялась в обеих версиях алгоритма MINSUM.

Основные результаты проведенных экспериментов представлены в табл. 1 – 5 для смешанных графов порядка $n \in \{70, 90, 100, 120, 200\}$. В первом столбце каждой таблицы указано произведение $n \times m$ числа требований на число приборов. В каждой серии размерности $n \times m$ было решено по 10 случайно сгенерированных примеров. В столбце 2 каждой таблицы указано среднее число ребер смешанного графа, в столбце 3 – среднее значение оценки $LB1$, полученное на первой итерации, в столбце 4 – среднее значение целевой функции (7) для оптимальной раскраски, которое вычислялось только для точно решенных примеров, в столбце 5 – среднее значение числа вершин $|W|$ в дереве решений, в столбце 6 – процент точно решенных примеров данной серии, в столбце 7 – среднее время решения одного примера в секундах работы процессора.

Таблица 1

Смешанные графы порядка 70

$m \times n$	$ E $	Оценка $LB1$					Оценки $LB1$ и $LB2$					
		3	4	5	6	7	8	9	10	11	12	
1	2											
7×5	479.5	93.0	101.7	3245658	80	64.73	93.2	103.0	1034828	90	33.64	
7×7	345.6	82.9	93.2	461247	100	5.71	82.9	93.2	89037	100	2.72	
7×10	240.0	78.3	84.6	711	100	0.06	78.3	84.6	632	100	0.22	
10×6	396.6	100.2	108.0	8066796	50	292.77	100.2	109.0	4089110	80	249.08	
10×8	304.5	91.2	100.5	2108315	100	54.20	92.3	100.5	206872	100	9.60	
10×10	238.4	84.3	93.8	24106	100	0.49	85.7	93.8	12323	100	0.47	
14×10	244.2	96.4	108.4	5737391	60	429.16	98.5	108.4	384453	100	64.85	
14×11	208.3	91.8	99.3	1148966	100	70.05	91.9	99.3	256644	100	21.49	
14×12	199.3	91.0	100.2	812531	90	35.51	92.6	100.1	320603	100	20.51	
14×14	168.3	86.4	96.2	151375	100	7.30	88.3	96.2	94528	100	7.06	

В столбцах 8 – 12 приведены результаты для версии алгоритма, использовавшей обе глобальные оценки $LB1$ и $LB2$. В столбце 8 указано среднее значение наибольшей из оценок $LB1$ и $LB2$, полученных на первой итерации, в столбце 9 – среднее значение целевой функции (7) для оптимальной раскраски, которое вычислялось только для точно решенных примеров данной версии алгоритма, в столбце 10 – среднее значение числа вершин в дереве решений, в столбце 11 – процент точно решенных примеров данной серии, в столбце 12 – среднее время решения одного примера в секундах работы процессора. Из таблиц следует, что использование двух глобальных оценок приводит к лучшим результатам как по числу точно решенных задач, так и по числу $|W|$ вершин дерева решений, построенного при решении одной задачи.

В эксперименте было ограничено максимальное число вершин в дереве решений $T = (W, R, \emptyset)$ для каждого примера, а именно $|W| \leq 20\,000\,000$. Время, затраченное на решение одной задачи, целесообразно сравнивать только в случаях совпадения процентов решенных задач для обеих версий алгоритма. Действительно, в столбцах 7 и 12 среднее время вычисляется только для точно решенных задач (поскольку именно нерешенные одной из версий алгоритма задачи требуют, как правило, наибольшее время для решения другой версией алгоритма).

Таблица 2

Смешанные графы порядка 90

$m \times n$	$ E $	Оценка $LB1$					Оценки $LB1$ и $LB2$					
1	2	3	4	5	6	7	8	9	10	11	12	
6×4	983.5	121.6	131.0	10138166	40	176.40	121.6	131.6	12286352	80	453.59	
6×5	796.6	110.3	121.6	4964368	90	131.15	110.3	121.6	3890109	90	142156	
6×6	662.0	102.6	116.0	120778	100	1.77	102.6	116.0	107283	100	2.73	
10×8	489.0	115.0	124.8	5857087	60	137.22	115.0	125.9	3722763	80	163.19	
10×9	447.3	111.3	122.2	4053500	90	131.58	111.3	122.8	1703791	100	83.77	
10×10	396.4	106.9	118.5	1309243	100	38.60	107.1	118.5	146865	100	6.75	
15×12	334.3	117.5	126.0	1055275	30	62.25	119.0	128.9	2809480	80	572.87	
15×14	292.1	112.5	122.9	2636349	70	139.20	114.0	126.1	2929057	100	432.30	
15×55	257.9	109.6	119.6	150841	100	7.16	109.8	119.6	59930	100	4.29	

Таблица 3

Смешанные графы порядка 100

$m \times n$	$ E $	Оценка $LB1$					Оценки $LB1$ и $LB2$					
1	2	3	4	5	6	7	8	9	10	11	12	
5×4	1228.7	124.0	135.4	1609875	70	19.85	124.0	135.4	1597596	70	34.37	
5×5	999.2	114.3	128.4	683916	100	8.19	114.3	128.4	594437	100	13.87	
5×6	830.8	111.6	124.2	248935	100	4.11	111.6	124.2	248889	100	7.42	
10×9	544.9	122.3	133.2	2187751	80	47.50	122.3	134.4	3396962	100	169.03	
10×10	489.2	118.5	132.5	421152	80	10.29	118.5	132.7	1569142	90	100.22	
10×11	440.1	114.1	126.9	2475391	100	83.41	114.2	126.9	475550	100	21.05	
20×20	244.3	124.0	136.0	5187153	90	1040.21	124.0	136.0	2412107	90	636.56	
20×22	225.3	120.7	130.2	1820208	90	296.14	122.5	131.4	195214	100	56.10	
20×25	196.7	116.3	127.1	1068121	100	168.50	116.6	127.1	900975	100	176.57	

Таблица 4

Смешанные графы порядка 120

$m \times n$	$ E $	Оценка $LB1$					Оценки $LB1$ и $LB2$					
1	2	3	4	5	6	7	8	9	10	11	12	
6×5	1427.0	145.8	160.5	6800490	60	276.46	145.8	160.5	6586068	60	342.07	
6×6	1185.8	136.3	152.7	3955279	90	177.90	136.3	152.7	3926692	90	229.83	
6×7	1015.1	131.7	149.2	862342	100	16.88	131.7	149.2	843899	100	28.25	
8×8	875.6	136.9	153.7	3635017	80	162.25	136.9	153.7	3495129	80	215.51	
8×9	799.4	136.3	152.2	2154475	80	67.72	136.3	152.2	2081932	80	107.58	
8×10	699.2	131.0	146.6	234380	100	5.40	131.0	146.6	230118	100	9.20	
8×11	648.4	131.4	144.9	303533	100	7.19	131.4	144.9	291114	100	12.62	
10×9	791.9	145.6	161.0	9066021	20	310.16	145.6	161.0	9529912	30	787.78	
10×10	703.8	139.4	156.4	5925586	90	280.44	139.4	156.4	2768785	90	160.52	
10×12	599.3	135.5	150.2	853598	100	23.98	135.5	150.2	644894	100	33.73	
15×15	465.5	142.2	155.9	2989791	70	192.09	143.9	158.4	4037277	100	424.27	
15×17	417.7	139.2	154.4	2799784	100	154.46	139.4	154.8	2068091	100	252.19	
15×20	358.4	136.0	151.6	2816778	100	155.50	136.8	151.6	361953	100	36.40	
20×20	353.6	146.7	158.5	3708888	40	419.62	146.7	159.7	3059639	70	782.00	
20×21	337.5	145.8	158.5	4930280	80	654.64	145.8	158.5	4316874	80	783.72	
20×22	319.2	142.4	155.1	2664345	70	410.07	143.5	155.9	2112155	80	614.44	
20×23	310.6	142.1	155.7	4098524	90	520.63	142.8	155.8	1046320	100	238.30	
20×24	288.0	138.7	152.2	1199904	100	147.80	138.7	152.2	991529	100	155.57	
20×25	283.4	138.6	151.6	755786	90	67.08	138.6	151.6	283544	90	45.05	
20×26	273.6	138.2	152.0	1208435	100	117.12	138.2	152.0	792385	100	121.41	
20×27	266.5	137.3	149.9	174786	90	23.33	137.3	150.8	196138	100	52.22	

Несмотря на то, что версия алгоритма, основанная на использовании обеих оценок $LB1$ и $LB2$ строит дерево решений в среднем меньшего размера, чем версия алгоритма, основанного на использовании одной оценки $LB1$, по среднему времени решения одной задачи нельзя отдать предпочтения ни одной из версий алгоритма MINSUM. Этот факт объясняется тем, что дополнительное вычисление оценки $LB2$ оказалось достаточно трудоемким.

Таблица 5

Смешанные графы порядка 200

$m \times n$	$ E $	Оценка $LB1$					Оценки $LB1$ и $LB2$				
1	2	3	4	5	6	7	8	9	10	11	12
10×15	1332.5	217.3	236.0	6364626	50	441.61	217.3	236.0	6361378	50	666.52
10×17	1174.4	215.1	233.4	3168299	90	170.07	215.1	233.4	3154827	90	276.60
15×18	1097.0	212.9	231.6	4439601	100	304.58	212.9	231.6	4432942	100	409.25
20×33	603.8	217.6	236.9	5194660	70	799.67	217.9	237.4	5041105	80	1514.92
20×35	571.7	217.2	235.5	5069522	90	926.98	217.8	235.5	4633708	90	1720.68
20×40	502.5	215.2	232.5	1381569	100	200.55	215.9	232.5	1336430	100	331.92

Заключение

Поскольку в системе job shop допускается последовательное повторение одного и того же прибора из множества M , то описанное в данной статье сведение к оптимальной раскраске смешанного графа G применимо и для задачи $J/[p_{ij}], Pr/\sum C_i$ с произвольными целочисленными длительностями операций (обозначение $[p_{ij}]$) при разрешении прерываний процесса выполнения операций.

Отметим также, что в работах [8–10] исследовалась задача $J/p_{ij}=1/C_{max}$ на основе сведения ее к минимальной (по числу цветов) раскраске смешанного графа $G = (V, A, E)$. В частности, в статье [9] исследована асимптотическая сложность задачи построения минимальной раскраски смешанного графа G в случае, когда (V, A, \emptyset) представляет собой объединение путей, а (V, \emptyset, E) – объединение клик. Для такого смешанного графа предложены три алгоритма раскраски типа ветвей и границ, которые были протестированы на случайно сгенерированных смешанных графах порядка $n \leq 200$ для точного решения и порядка $n \leq 900$ для приближенного. Как следует из теоретических и экспериментальных исследований, рассмотренная выше задача $J/p_{ij}=1/\sum C_i$, вообще говоря, сложнее задачи $J/p_{ij}=1/C_{max}$. В связи с этим полученные в настоящей статье вычислительные результаты для критерия $\sum C_i$ в ряде случаев уступают результатам, полученным в работе [9] для критерия C_{max} . Предметом дальнейших исследований может стать получение более точных оценок сверху и снизу указанных целевых функций с целью увеличения размерности задач, для которых будет возможно получение точного решения за приемлемое время.

Работа выполнена частично при финансовой поддержке Фонда фундаментальных исследований Республики Беларусь в рамках проектов Ф03МС-039 и Ф02Р-013. Авторы благодарны рецензенту за полезные замечания, которые были учтены в последней редакции статьи.

Список литературы

1. Сотсков Ю.Н., Танаев В.С. Хроматический многочлен смешанного графа // Весці АН БССР. Сер. фіз.-мат. навук. – 1976. – № 6. – С. 20-23.
2. Hansen P., Kuplinsky J., de Werra D. Mixed graph colorings // Mathematical Methods of Operations Research. – 1997. – V. 45. – P. 145-160.
3. Танаев В.С., Сотсков Ю.Н., Струсевич В.А. Теория расписаний. Многостадийные системы. – М.: Наука, 1989.
4. Харари Ф. Теория графов. – М.: Мир, 1973.

5. Gonzalez T. Unit execution time shop problems // *Mathematical Methods of Operations Research*. – 1982. – V. 7. – № 1. – P. 57-66.
6. Leighton F.T. A graph coloring algorithm for large scheduling problems // *Journal of Research of the National Bureau of Standards*. – 1979. – V. 84. – P. 489-506.
7. De Werra D. An introduction to timetabling // *European Journal of Operations Research*. – 1985. – V. 19. – P. 151-162.
8. Sotskov Yu.N., Tanaev V.S, Werner F. Scheduling problems and mixed graph colorings // *Optimization*. – 2002. – V. 5. – № 3. – P. 597-624.
9. Sotskov Yu.N., Dolgui A., Werner F. Mixed graph coloring for unit-time job-shop scheduling // *International Journal of Mathematical Algorithms*. – 2001. – V. 2. – P. 289-323.
10. Андреев Г.В., Сотсков Ю.Н. Раскраска вершин смешанного графа методом ветвей и границ // *Весці НАН Беларусі. Сер. фіз.-мат. навук*. – 2001. – № 1. – С. 124-129.
11. Сотсков Ю.Н. Оптимальное обслуживание двух требований при регулярном критерии // *Автоматизация процессов проектирования*. - Мн.: Ин-т техн. кибернетики АН БССР, 1985. – С. 86-95.
12. Сотсков Ю.Н. Сложность задач оптимального обслуживания трех требований // *Кибернетика*. – 1990. – № 5. – С. 50-54.
13. Brucker P., Kravchenko C.A., Sotskov Yu.N. Preemptive job-shop scheduling problems with a fixed number of jobs // *Mathematical Methods of Operations Research*. – 1999. – № 49. – P.41-76.

Поступила в редакцию

Объединенный институт проблем информатики НАН Беларуси, Минск, Сурганова, 6
e-mail: sotskov@newman.bas-net.by

Yu. N. Sotskov, G. V. Andreev

MEAN FLOW TIME MINIMIZATION OF JOBS WITH UNIT-TIME OPERATION DURATIONS VIA MIXED GRAPH COLORING

It is shown that optimal coloring φ of mixed graph G with criterion MINSUM defines a schedule minimizing mean flow time objective in a job shop problem with unit-time operation duration. Subgraph (V, A, \emptyset) of such a mixed graph $G = (V, A, E)$ is a union of paths, and subgraph (V, \emptyset, E) is a union of cliques. A branch and bound method is developed for optimal coloring of such a mixed graph G . This method is realized in C++ and tested for randomly generated mixed graphs of order $n \leq 200$.