ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Апрель—июнь 2016. Том 23, № 2. С. 41–62

УДК 519.8 DOI: 10.17377/daio.2016.23.524

О СЛОЖНОСТИ ОПТИМАЛЬНОЙ РЕКОМБИНАЦИИ ДЛЯ ЗАДАЧ СОСТАВЛЕНИЯ РАСПИСАНИЙ В МНОГОСТАДИЙНОЙ СИСТЕМЕ ПОТОЧНОГО ТИПА *)

Ю. В. Коваленко[1]
[1]Институт математики им. С. Л. Соболева,
пр. Коптюга, 4, 630090 Новосибирск, Россия
e-mail: julia.kovalenko.ya@yandex.ru

Аннотация. Рассматривается вычислительная сложность оптимальной рекомбинации для различных вариантов задачи составления поточного расписания (flowshop) с критериями минимизации общего времени завершения работ и максимального временного смещения. Доказана NP-трудность этих задач, и предложен точный алгоритм их решения. Показано, что в случае перестановочной задачи flowshop трудоёмкость предложенного алгоритма полиномиальна для «почти всех» пар родительских решений при числе работ, стремящемся к бесконечности. Ил. 4, библиогр. 26

Ключевые слова: задача flowshop, перестановка, генетический алгоритм, оптимальная рекомбинация.

Введение

Работоспособность генетических алгоритмов (см., например, [6, 19]) существенно зависит от выбора оператора кроссинговера, где комбинируются элементы родительских решений при построении решений-потомков. Задача оптимальной рекомбинации (ЗОР) [3, 15–17] состоит в отыскании наилучшего возможного результата кроссинговера при заданных двух родительских решениях задачи оптимизации. ЗОР является вспомогательной задачей, как правило, меньшей размерности, чем исходная задача, и формулируется с учётом основных принципов построения оператора кроссинговера [22]. Результаты, содержащиеся в [2, 9] и других работах, дают экспериментальное подтверждение целесообразности решения ЗОР в операторах кроссинговера.

*) Исследование выполнено при финансовой поддержке Российского научного фонда (проект 15–11–10009).

ЗОР для задач, где допустимые решения представлены в виде перестановок, основана на вычислении оптимального потомка, который наследует значения в позициях перестановок родительских решений. В генетических алгоритмах для задач на перестановках алгоритмы кроссинговера подобного типа применялись в [12, 13, 23, 24].

Классическая задача flowshop [8, 18] заключается в составлении расписания выполнения n работ на m машинах. Для всех работ маршруты обслуживания (порядки прохождения машин) одинаковы. В таком случае говорят, что обслуживающая система является системой поточного типа [8]. Данная задача относится к числу NP-трудных задач оптимизации [1], которые могут могут быть сформулированы как задачи на перестановках. Результаты экспериментальных исследований из [10, 20, 25, 26] и др. свидетельствуют о перспективности использования в генетических алгоритмах для задачи flowshop таких операторов кроссинговера, где скрещивание родительских решений происходит с сохранением в потомках абсолютного и/или относительного порядка обслуживания работ.

При рассмотрении задачи flowshop на перестановках [8, 18] предполагается, что порядок обслуживания работ на всех машинах идентичен. В настоящей статье доказывается NP-трудность оптимальной рекомбинации для задачи flowshop на перестановках (в том числе без задержек) с критериями минимизации общего времени завершения работ и максимального временного смещения. Также исследуется сложность оптимальной рекомбинации для классической задачи flowshop с указанными критериями.

Предлагается точный алгоритм решения ЗОР для произвольной задачи на перестановках, основанный на переборе совершенных паросочетаний в специальном двудольном графе. С помощью данного алгоритма для различных вариантов задачи flowshop оценивается трудоёмкость ЗОР и устанавливается её полиномиальная разрешимость для «почти всех» пар родительских решений при числе работ, стремящемся к бесконечности.

Статья построена следующим образом. В разд. 1 формулируется ЗОР для задач на перестановках, предлагается точный алгоритм её решения и исследуются асимптотические свойства. Разд. 2 посвящён описанию различных вариантов задачи flowshop и постановок ЗОР для них. В разд. 3 представлен анализ вычислительной сложности ЗОР для задачи flowshop. В разд. 4 исследуется сложность ЗОР для задачи flowshop на перестановках и без задержек.

1. Задача оптимальной рекомбинации на перестановках

Рассмотрим задачу оптимизации на перестановках:

$$F(\pi) \to \min(\max), \quad \pi = (\pi_1, \dots, \pi_k) \in \Pi.$$

Здесь Π — множество всех возможных перестановок элементов из $X = \{x_1, \dots, x_k\}$, $F : \Pi \to R$ — некоторая функция.

Задача оптимальной рекомбинации [16]. Вход: *индивидуальная задача оптимизации на перестановках и два родительских решения* $\pi^1 = \big(\pi^1_1, \dots, \pi^1_k\big)$, $\pi^2 = \big(\pi^2_1, \dots, \pi^2_k\big)$.

Найти: *перестановку (потомка)* $\pi' \in \Pi$ *такую, что*

(I) $\pi'_j = \pi^1_j$ *или* $\pi'_j = \pi^2_j$ *для всех* $j = 1, \dots, k$;

(II) *для любой* $\pi \in \Pi$ *такой, что* $\pi_j = \pi^1_j$ *или* $\pi_j = \pi^2_j$ *для всех* $j = 1, \dots, k$, *выполняется неравенство*

$$F(\pi') \leqslant F(\pi) \quad \text{(\textit{в случае задачи на минимум})}$$

или

$$F(\pi') \geqslant F(\pi) \quad \text{(\textit{в случае задачи на максимум})}.$$

Построим алгоритм решения ЗОР (I), (II), используя метод А. И. Сердюкова [7], разработанный для оценки мощности множества допустимых решений в задаче коммивояжёра с предписаниями вершин.

Рассмотрим двудольный граф $G = (X_k, X, U)$ с равными по мощности долями вершин X_k, X и множеством рёбер $U = \big\{\{i, x\} \mid i \in X_k, x = \pi^1_i \text{ или } x = \pi^2_i\big\}$, где $X_k = \{1, \dots, k\}$. Между множеством допустимых решений рассматриваемой ЗОР и множеством совершенных паросочетаний в графе G существует взаимно однозначное соответствие: совершенное паросочетание $w^\pi = \{\{1, x^1\}, \dots, \{k, x^k\}\}$ определяет перестановку $\pi^w = (x^1, \dots, x^k)$, и наоборот.

Ребро $\{i, x\} \in U$ назовём *особым*, если $\{i, x\}$ принадлежит любому совершенному паросочетанию в графе G. Заметим, что максимальный связный подграф графа G, содержащий не менее двух рёбер, представляет собой цикл. Обозначим через $q(G)$ число циклов в G. Рёбра $\{i, x\} \in U$ такие, что $\pi^1_i = \pi^2_i$, являются особыми и циклам не принадлежат, а рёбра $\{i, x\} \in U$ такие, что $\pi^1_i \neq \pi^2_i$, содержатся в циклах. Кроме того, каждый цикл $j = 1, \dots, q(G)$ графа G имеет ровно два максимальных (совершенных) паросочетания, наборы рёбер которых различны, следовательно, он не содержит особых рёбер. Таким образом, ребро $\{i, x\} \in U$ особое, только если $\pi^1_i = \pi^2_i$, и любое совершенное паросочетание в графе G взаимно

однозначно определяется набором максимальных паросочетаний (по одному из каждого цикла) и совокупностью особых рёбер.

Циклы графа G могут быть вычислены за время $O(k)$, например, с помощью алгоритма «поиск в глубину» [5]. Особые рёбра и максимальные паросочетания в циклах находятся очевидным образом за время $O(k)$.

Таким образом, для решения ЗОР (I), (II) можно предложить следующий алгоритм. Строим двудольный граф G, определяем в нём набор особых рёбер и циклов, а также находим максимальные паросочетания в циклах. Перебираем все совершенные паросочетания в графе G (формируя их из максимальных паросочетаний в циклах и особых рёбер). В процессе перебора каждому совершенному паросочетанию w ставим в соответствие перестановку π и вычисляем $F(\pi)$. В результате находим требуемую перестановку $\pi' \in \Pi$. Трудоёмкость этого алгоритма равна $O(2^{q(G)}T(F))$, где $q(G) \leqslant \lfloor \frac{k}{2} \rfloor$, и данная оценка достижима, $T(F)$ — время вычисления $F(\pi)$ для перестановки $\pi \in \Pi$.

В [17] доказано, что $q(G) \leqslant \frac{\ln(k)}{\ln(2)}$ для «почти всех» пар родительских перестановок, т. е. ЗОР (I), (II) имеет не более k допустимых решений. Приведём более подробное описание данного результата.

Определение 1 [7]. Граф $G = (X_k, X, U)$ назовём *хорошим*, если для него выполняется неравенство $q(G) \leqslant \frac{\ln(k)}{\ln(2)}$.

Обозначим через $\widetilde{\Re}_k$ множество пар родительских перестановок с k элементами, которые соответствуют хорошим двудольным графам G, а через $\Re_k$ — множество всех возможных пар родительских перестановок с k элементами. Из результатов [17] следует

Теорема 1. $|\widetilde{\Re}_k|/|\Re_k| \longrightarrow 1$ *при* $k \to \infty$.

Согласно известной терминологии (см., например, [11]) данная теорема означает, что «почти все» индивидуальные ЗОР (I), (II) имеют не более k допустимых решений и разрешимы за время $O(kT(F))$.

Оператор кроссинговера, основанный на решении ЗОР (I), (II) с помощью представленного алгоритма, можно рассматривать как дерандомизацию равномерного циклического кроссинговера (Uniform Cycle Crossover) [14].

2. Задачи составления расписаний в многостадийной системе поточного типа

В обслуживающую систему поточного типа, состоящую из m последовательных машин, в момент времени $t = 0$ поступает множество работ

$N = \{1, \ldots, n\}$. Каждая работа обслуживается на машинах в последовательности $1 \to 2 \to \cdots \to m$. Обозначим через $t_{il} \in \mathbb{R}^+$ длительность обслуживания работы i машиной l, где $i \in N$, $l \in M = \{1, \ldots, m\}$ (здесь и далее $\mathbb{R}^+$ — множество положительных вещественных чисел). Для каждой работы прерывания процесса обслуживания на любой машине и одновременное выполнение несколькими машинами не допускаются. Каждая машина обслуживает не более одной работы в любой момент времени. Данная задача в теории расписаний обозначается через $F \parallel \gamma$ [18], где символ γ определяет критерий оптимизации.

Пусть $t^{st}(i,l)$ — момент начала работы i на машине l, а $t^f(i,l) = t^{st}(i,l) + t_{il}$ — момент окончания работы i на машине l, $i \in N$, $l \in M$. Тогда расписание s однозначно определяется заданием $\{t^{st}(i,l)\}$ или $\{t^f(i,l)\}$, и момент завершения обслуживания всех работ всеми машинами равен

$$C_{\max}(s) = \max\{t^f(i,m) \mid i \in N\} = \max\{t^{st}(i,m) + t_{im} \mid i \in N\}.$$

При рассмотрении критерия минимизации максимального временнóго смещения для каждой работы $i \in N$ задаётся директивный срок d_i, к которому желательно завершить обслуживание данной работы всеми машинами. Максимальное временнóе смещение для расписания s равно

$$L_{\max}(s) = \max\{t^f(i,m) - d_i \mid i \in N\} = \max\{t^{st}(i,m) + t_{im} - d_i \mid i \in N\}.$$

Требуется составить расписание s^* выполнения работ на машинах такое, что $C_{\max}(s^*)$ (или $L_{\max}(s^*)$) имеет наименьшее значение. Задача $F \parallel \gamma,\ \gamma \in \{C_{\max},\ L_{\max}\}$, NP-трудная в сильном смысле уже при $m = 3$ (и $m = 2$ соответственно) [1].

Пусть перестановка $\pi^l = \left(\pi^l_1, \ldots, \pi^l_n\right)$ определяет порядок выполнения работ на машине l, где π^l_i — работа, выполняемая i-й по счёту на машине $l \in M$. Для набора перестановок $\{\pi^l\}_{l \in M}$ раннее расписание может быть вычислено за время $O(nm)$:

$$t^{st}\left(\pi^1_1, 1\right) = 0, \quad t^{st}\left(\pi^1_i, 1\right) = t^f\left(\pi^1_{i-1}, 1\right), \quad i = 2, \ldots, n,$$

$$t^{st}\left(\pi^l_1, l\right) = t^f\left(\pi^l_1, l-1\right), \quad l = 2, \ldots, m,$$

$$t^{st}\left(\pi^l_i, l\right) = \max\left\{t^f\left(\pi^l_{i-1}, l\right), t^f\left(\pi^l_i, l-1\right)\right\}, \quad i = 2, \ldots, n,\ l = 2, \ldots, m,$$

$$C_{\max}(\{\pi^l\}_{l \in M}) = t^f\left(\pi^m_n, m\right) = t^{st}\left(\pi^m_n, m\right) + t_{\pi^m_n, m},$$

$$L_{\max}(\{\pi^l\}_{l \in M}) = \max\left\{t^f\left(\pi^m_i, m\right) - d_{\pi^m_i} \mid i = 1, \ldots, n\right\}.$$

Далее под расписанием для набора перестановок $\{\pi^l\}_{l\in M}$ будем понимать раннее расписание, а под значением целевой функции — значение целевой функции, соответствующее раннему расписанию.

Задача $F \parallel \gamma,\ \gamma \in \{C_{\max},\ L_{\max}\}$, может быть сформулирована как задача поиска набора перестановок $\{\pi^{*l}\}_{l\in M}$ такого, что $C_{\max}(\{\pi^{*l}\}_{l\in M})$ (и $L_{\max}(\{\pi^{*l}\}_{l\in M})$ соответственно) имеет минимальное значение.

Сформулируем задачу оптимальной рекомбинации для $F \parallel \gamma$. Для заданных родительских наборов перестановок $\{\pi^{1l}\}_{l\in M}$ и $\{\pi^{2l}\}_{l\in M}$ требуется найти набор перестановок $\{\pi'^l\}_{l\in M}$ такой, что

(i) $\pi_i'^l = \pi_i^{1l}$ или $\pi_i'^l = \pi_i^{2l}$ для всех $i = 1, \ldots, n,\ l \in M$;

(ii) $\{\pi'^l\}_{l\in M}$ имеет минимальное значение целевой функции среди всех наборов перестановок, удовлетворяющих условию (i).

В задаче flowshop на перестановках $F|prmu|\gamma$ [21] вводится дополнительное ограничение: каждая машина обслуживает работы в одинаковой последовательности. Задача $F|prmu|\gamma,\ \gamma \in \{C_{\max},\ L_{\max}\}$, NP-трудна в сильном смысле уже при $m = 3$ (и $m = 2$ соответственно) [1]. Заметим, что при $m \leqslant 3$ в задаче $F \parallel \gamma,\ \gamma \in \{C_{\max},\ L_{\max}\}$, оптимальное решение достаточно искать среди допустимых расписаний соответствующей задачи $F|prmu|\gamma,\ \gamma \in \{C_{\max},\ L_{\max}\}$ [4].

Для перестановки $\pi = (\pi_1, \ldots, \pi_n)$, определяющей порядок выполнения работ на всех машинах (π_i — работа, выполняемая i-й по счёту), раннее расписание задачи $F|prmu|\gamma,\ \gamma \in \{C_{\max},\ L_{\max}\}$, вычисляется за время $O(mn)$:

$$t^{st}(\pi_1, 1) = 0, \quad t^{st}(\pi_i, 1) = t^f(\pi_{i-1}, 1), \quad i = 2, \ldots, n,$$
$$t^{st}(\pi_1, l) = t^f(\pi_1, l-1), \quad l = 2, \ldots, m,$$
$$t^{st}(\pi_i, l) = \max\{t^f(\pi_{i-1}, l), t^f(\pi_i, l-1)\}, \quad i = 2, \ldots, n,\ l = 2, \ldots, m,$$
$$C_{\max}(\pi) = t^f(\pi_n, m) = t^{st}(\pi_n, m) + t_{\pi_n, m},$$
$$L_{\max}(\pi) = \max\{t^f(\pi_i, m) - d_{\pi_i} \mid i = 1, \ldots, n\}.$$

Далее под расписанием для перестановки π будем понимать раннее расписание, а под значением целевой функции — значение целевой функции, соответствующее раннему расписанию.

Сформулируем задачу оптимальной рекомбинации для $F|prmu|\gamma$. Для произвольных заданных родительских перестановок π^1 и π^2 требуется найти перестановку π' такую, что

(i$'$) $\pi_i' = \pi_i^1$ или $\pi_i' = \pi_i^2$ для всех $i = 1, \ldots, n$;

(ii$'$) π' имеет минимальное значение целевой функции среди всех перестановок, удовлетворяющих условию (i$'$).

3. Вычислительная сложность оптимальной рекомбинации

При доказательстве теорем о сложности ЗОР будет использоваться NP-трудная задача Упорядоченное Разбиение [1] в следующей постановке. Заданы упорядоченное множество $A = \{a_1, a_2, \ldots, a_{2n_0}\}$ и вес $e_i \in \mathbb{Z}^+$ каждого элемента $a_i \in A$, причём $\sum\limits_{a_i \in A} e_i = 2E$ (здесь и далее $\mathbb{Z}^+$ — множество положительных целых чисел). Существует ли разбиение множества A на два подмножества A_1 и A_2 таких, что

$$\sum_{a_i \in A_1} e_i = \sum_{a_i \in A_2} e_i = E, \quad |A_1| = |A_2| = n_0$$

и множество A_1 включает в себя только один элемент из каждой пары $a_{2i-1},\ a_{2i},\ i = 1, \ldots, n_0$?

Теорема 2. *Задача оптимальной рекомбинации* (i′), (ii′) *для* $F3|prmu|C_{\max}$ *NP-трудна.*

Доказательство. Сформулируем для ЗОР (i′), (ii′) соответствующую задачу распознавания: определить, существует ли перестановка π', удовлетворяющая условию (i′), такая, что $C_{\max}(\pi') \leqslant y$ для заданного числа y. Покажем, что к этой задаче полиномиально сводится задача Упорядоченное Разбиение.

В задаче $F|prmu|C_{\max}$ положим $m = 3$ — число машин, $n = 2n_0 + 3$ — число работ. Работы множества N разобьём на две группы: U-работы, соответствующие элементам $a_i \in A$, $i = 1, \ldots, 2n_0$, и обозначаемые через U_i, и V-работы, обозначаемые через V_i, $i = 0, 1, 2$. Пусть

$$t_{U_i,1} = t_{U_i,3} = 1, \quad t_{U_i,2} = e_i, \quad i = 1, \ldots, 2n_0,$$
$$t_{V_0,1} = t_{V_2,3} = 1, \quad t_{V_0,2} = t_{V_1,2} = t_{V_2,2} = E,$$
$$t_{V_0,3} = t_{V_1,1} = t_{V_1,3} = t_{V_2,1} = 2E - n_0.$$

Определим родительские перестановки для ЗОР:

$$\pi^1 = (V_0, U_1, U_3, \ldots, U_{2n_0-1}, V_1, U_2, U_4, \ldots, U_{2n_0}, V_2),$$
$$\pi^2 = (V_0, U_2, U_4, \ldots, U_{2n_0}, V_1, U_1, U_3, \ldots, U_{2n_0-1}, V_2).$$

Покажем, что в построенной ЗОР перестановка работ π', для которой выполняется условие (i′) и $C_{\max}(\pi') \leqslant 5E + 2$, существует тогда и только тогда, когда на вопрос в задаче Упорядоченное Разбиение ответ «да».

(I) Пусть на вопрос в задаче Упорядоченное Разбиение ответ «да» и $A_1,\ A_2$ — подмножества множества A такие, что $\sum\limits_{a_i\in A_1} e_i = \sum\limits_{a_i\in A_2} e_i = E$ и $|A_1| = |A_2| = n_0$.

Обозначим через $N_1(U) = (U_{j_1},\ldots,U_{j_{n_0}})$ и $N_2(U) = (U_{k_1},\ldots,U_{k_{n_0}})$ перестановки U-работ, соответствующих элементам из A_1 и A_2 соответственно, где $j_i < j_{i+1}$ и $k_i < k_{i+1}$ для всех $i = 1,\ldots,n_0-1$. Тогда для перестановки $\pi' = (V_0, N_1(U), V_1, N_2(U), V_2)$ выполняется условие (i′) и $C_{\max}(\pi') = 5E+2$ (рис. 1). При этом все три машины функционируют без простоев: машина 1 — в интервале $(0, 4E+1]$, машина 2 — в интервале $(1, 5E+1]$ и машина 3 — в интервале $(E+1, 5E+2]$.

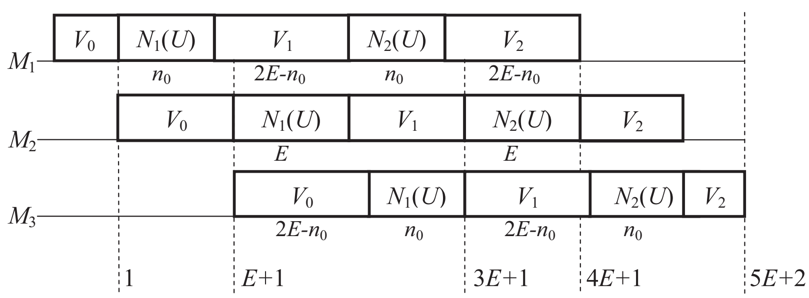

Рис. 1. Структура расписания для перестановки $(V_0, N_1(U), V_1, N_2(U), V_2)$ в задаче $F3|prmu|C_{\max}$

(II) Пусть существует перестановка π', для которой выполняется условие (i′) и $C_{\max}(\pi') \leqslant 5E+2$. Будем полагать, что в задаче Упорядоченное Разбиение $E > n_0$, так как в противном случае данная задача тривиальна. Обозначим через $N'_j(U)$ множество работ, стоящих в перестановке π' между работами V_{j-1} и V_j, $j = 1, 2$. При этом $|N'_1(U)| = |N'_2(U)| = n_0$, суммарная длительность работ из $N'_j(U)$ на первой и третьей машинах равна n_0, а на второй машине — $\sum\limits_{U_i\in N'_j(U)} e_i$, где $j = 1, 2$.

В расписании, соответствующем π', все три машины функционируют без простоев. Действительно, суммарная длительность работ на первой машине равна $4E+1$, а так как общая длительность работы V_2 на второй и третьей машинах равна $E+1$, машина 1 функционирует без простоев в интервале $(0, 4E+1]$. Аналогичным образом можно показать, что машина 2 функционирует без простоев в интервале $(1, 5E+1]$, а машина 3 — в интервале $(E+1, 5E+2]$.

Докажем, что суммарная длительность работ из $N_1'(U)$ на второй машине $\sum\limits_{U_i \in N_1'(U)} e_i$ равна E. Идея доказательства заключается в следующем. Если $\sum\limits_{U_i \in N_1'(U)} e_i > E$, то для работы V_1 для моментов начала верны оценки $t^{st}(V_1, 2) > 2E + 1$, $t^{st}(V_1, 3) > 3E + 1$, и в результате $C_{\max}(\pi') > 5E + 2$. Если $\sum\limits_{U_i \in N_1'(U)} e_i < E$, то для работы V_1 $t^{st}(V_1, 2) \geqslant 2E + 1$ (поскольку суммарная длительность работ V_0, $N_1'(U)$ и V_1 на первой машине равна $2E + 1$), а так как $\sum\limits_{U_i \in N_2'(U)} e_i > E$, то $C_{\max}(\pi') > 5E + 2$.

Приведём формальное доказательство. Рассмотрим работы V_0 и V_1. Так как все три машины функционируют без простоев в соответствующих интервалах, имеют место соотношения

$$t^{st}(V_1, 1) = t^f(V_0, 1) + n_0, \tag{1}$$

$$t^{st}(V_1, 2) = t^f(V_0, 2) + \sum_{U_i \in N_1'(U)} e_i \geqslant t^f(V_1, 1) = t^f(V_0, 1) + 2E, \tag{2}$$

$$t^{st}(V_1, 3) = t^f(V_0, 3) + n_0 \geqslant t^f(V_1, 2) = t^f(V_0, 2) + E + \sum_{U_i \in N_1'(U)} e_i, \tag{3}$$

$$t^f(V_0, 2) = t^f(V_0, 1) + E, \tag{4}$$

$$t^f(V_0, 3) = t^f(V_0, 1) + 3E - n_0. \tag{5}$$

Следовательно,

$$t^f(V_1, 2) \overset{(2)}{=} t^f(V_0, 2) + \sum_{U_i \in N_1'(U)} e_i + E \overset{(4)}{=} t^f(V_0, 1) + 2E + \sum_{U_i \in N_1'(U)} e_i, \tag{6}$$

$$t^{st}(V_1, 3) \overset{(3),(5)}{=} t^f(V_0, 1) + 3E \overset{(3)}{\geqslant} t^f(V_1, 2) \overset{(6)}{=} t^f(V_0, 1) + 2E + \sum_{U_i \in N_1'(U)} e_i, \tag{7}$$

$$t^{st}(V_1, 2) \overset{(2),(4)}{=} t^f(V_0, 1) + E + \sum_{U_i \in N_1'(U)} e_i \overset{(2)}{\geqslant} t^f(V_1, 1) = t^f(V_0, 1) + 2E, \tag{8}$$

а значит, из (7) получаем $\sum\limits_{U_i \in N_1'(U)} e_i \leqslant E$, а из (8) — $\sum\limits_{U_i \in N_1'(U)} e_i \geqslant E$.

Таким образом, $\sum\limits_{U_i \in N_1'(U)} e_i = E$. Поскольку $\sum\limits_{U_i \in N_1'(U) \cup N_2'(U)} e_i = 2E$, имеем $\sum\limits_{U_i \in N_2'(U)} e_i = E$.

В результате, множества

$$A_1 = \{a_i \in A \mid U_i \in N'_1(U)\}, quad A_2 = \{a_i \in A \mid U_i \in N'_2(U)\}$$

соответствуют условиям задачи Упорядоченное Разбиение, и данная индивидуальная задача имеет ответ «да».

Сведе́ние задачи Упорядоченное Разбиение к рассматриваемой задаче распознавания построено за $O(n_0)$ операций, следовательно, ЗОР (i′), (ii′) для $F3|prmu|C_{\max}$ NP-трудна. Теорема 2 доказана.

Следствие 1. *Задача оптимальной рекомбинации* (i), (ii) *для* $F3 \parallel C_{\max}$ NP-*трудна.*

Доказательство. Пусть в ЗОР (i), (ii) для $F3 \parallel C_{\max}$ родительские наборы перестановок таковы, что $\pi^{11} = \pi^{12} = \pi^{13}$ и $\pi^{21} = \pi^{22} = \pi^{23}$. Если в оптимальном решении данной ЗОР перестановки π'^{l}, $l = 1, 2, 3$, не совпадают, то, используя подход из [4], можно показать, что набор перестановок $\{\pi''^{l}\}_{l\in M}$, где $\pi''^{1} = \pi''^{2} = \pi''^{3} = \pi'^{2}$, также оптимальный для рассматриваемой ЗОР. Тогда из теоремы 2 непосредственно следует требуемое утверждение. Следствие 1 доказано.

Теорема 3. *Задача оптимальной рекомбинации* (i′), (ii′) *для* $F2|prmu|L_{\max}$ *NP-трудна.*

Доказательство. Сформулируем для ЗОР (i′), (ii′) соответствующую задачу распознавания: определить, существует ли такая перестановка π', удовлетворяющая условию (i′), при которой $L_{\max}(\pi') \leqslant y$ для заданного числа y. Покажем, что к этой задаче полиномиально сводится задача Упорядоченное Разбиение.

В задаче $F|prmu|L_{\max}$ положим $m = 2$ — число машин, $n = 2n_0 + 2$ — число работ. Работы множества N разобьём на две группы: U-работы, соответствующие элементам $a_i \in A$, $i = 1, \dots, 2n_0$, и обозначаемые через U_i, и V-работы, обозначаемые через V_i, $i = 0, 1$. Пусть

$$t_{U_i,1} = e_i, \quad t_{U_i,2} = 2e_i, \quad d_{U_i} = 8E, \quad i = 1, \dots, 2n_0,$$

$$t_{V_0,1} = t_{V_1,1} = 2E, \quad t_{V_0,2} = t_{V_1,2} = E, \quad d_{V_0} = 3E, \quad d_{V_1} = 6E.$$

Определим родительские перестановки для ЗОР:

$$\pi^1 = (V_0, U_1, U_3, \dots, U_{2n_0-1}, V_1, U_2, U_4, \dots, U_{2n_0}),$$

$$\pi^2 = (V_0, U_2, U_4, \dots, U_{2n_0}, V_1, U_1, U_3, \dots, U_{2n_0-1}).$$

Покажем, что в построенной ЗОР перестановка работ π', для которой выполняется условие (i′) и $L_{\max}(\pi') \leqslant 0$, существует тогда и только тогда, когда на вопрос в задаче Упорядоченное Разбиение ответ «да».

(I) Пусть на вопрос в задаче Упорядоченное Разбиение ответ «да» и A_1, A_2 — подмножества элементов множества A такие, что $\sum\limits_{a_i \in A_1} e_i = \sum\limits_{a_i \in A_2} e_i = E$ и $|A_1| = |A_2| = n_0$.

Обозначим через $N_1(U) = (U_{j_1}, \ldots, U_{j_{n_0}})$ и $N_2(U) = (U_{k_1}, \ldots, U_{k_{n_0}})$ перестановки U-работ, соответствующих элементам из A_1 и A_2 соответственно, где $j_i < j_{i+1}$ и $k_i < k_{i+1}$ для всех $i = 1, \ldots, n_0 - 1$. Тогда для перестановки $\pi' = (V_0, N_1(U), V_1, N_2(U))$ выполняется условие (i$'$) и $L_{\max}(\pi') = 0$, так как в расписании для π' соблюдены директивные сроки всех работ и $t^f(V_0, 2) = d_{V_0}$ (рис. 2). При этом обе машины функционируют без простоев: машина 1 — в интервале $(0, 6E]$, а машина 2 — в интервале $(2E, 8E]$.

(II) Пусть существует перестановка π', для которой выполняется условие (i$'$) и $L_{\max}(\pi') \leqslant 0$. Обозначим через $N'_j(U)$ множество U-работ, стоящих в перестановке π' после работы V_{j-1}, $j = 1, 2$.

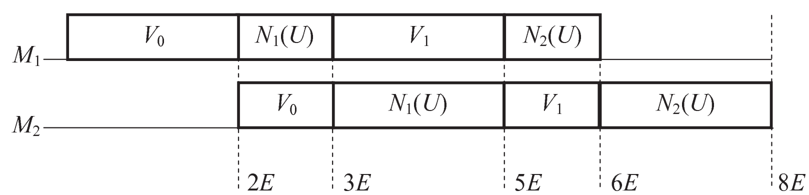

Рис. 2. Структура расписания для перестановки $(V_0, N_1(U), V_1, N_2(U))$ в задаче $F2|prmu|L_{\max}$

Поскольку $t_{V_0,1} + t_{V_0,2} = 3E = d_{V_0}$, работа V_0 обслуживается машиной 1 в интервале $(0, 2E]$, а машиной 2 — в $(2E, 3E]$. Так как суммарная длительность обслуживания всех работ на машине 2 равна $6E$, данная машина функционирует без простоев в интервале $(2E, 8E]$.

Рассмотрим временно́й интервал $(3E, 6E]$. В этом интервале машина 2 должна выполнить работу V_1 в течение E единиц времени (так как $d_{V_1} = 6E$) и U-работы в течение $2E$ единиц времени.

Очевидно, что $t^f(V_1, 1) \leqslant 5E$, иначе $t^f(V_1, 2) > 6E$. Предположим, что $t^f(V_1, 1) = 5E - C$, где $0 < C < E$. Тогда машина 1 выполняет в $(2E, 3E - C]$ работы из $N'_1(U)$ в течение не более $E - C$ единиц времени, и эти же работы обслуживаются на машине 2 в течение не более $2E - 2C$ единиц времени в $(3E, 5E - C]$ (рис. 3). Однако на машине 2 момент начала $t^{st}(V_1, 2) \geqslant 5E - C$, а значит, после обслуживания V_1 на машине 2 останется не более $2E + C$ единиц времени до момента $8E$,

которых не хватит на выполнение работ из $N'_2(U)$, суммарная длительность которых на машине 2 составит не менее $2E + 2C$. Следовательно, $t^f(V_1, 1) = t^{st}(V_1, 2) = 5E$, $\sum\limits_{U_i \in N'_1(U)} e_i = E$ и $\sum\limits_{U_i \in N'_2(U)} e_i = E$.

Таким образом, множества

$$A_1 = \{a_i \in A \mid U_i \in N'_1(U)\}, \quad A_2 = \{a_i \in A \mid U_i \in N'_2(U)\}$$

соответствуют условиям задачи Упорядоченное Разбиение, и данная индивидуальная задача имеет ответ «да».

Сведе́ние задачи Упорядоченное Разбиение к рассматриваемой задаче распознавания построено за $O(n_0)$ операций, следовательно, ЗОР (i′), (ii′) для $F2|prmu|L_{\max}$ NP-трудна. Теорема 3 доказана.

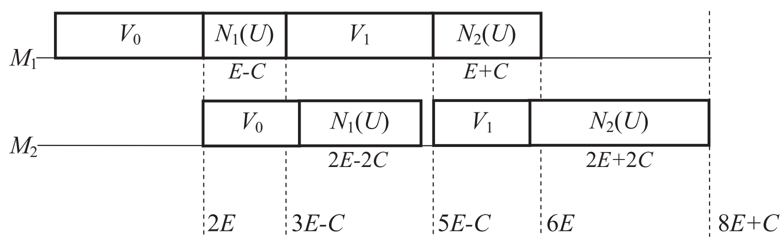

Рис. 3. Иллюстрация к доказательству от противного для перестановки π' в задаче $F2|prmu|L_{\max}$

Следствие 2. *Задача оптимальной рекомбинации* (i), (ii) *для* $F2 \parallel L_{\max}$ *NP-трудна.*

Доказательство. Пусть в ЗОР (i), (ii) для $F2 \parallel L_{\max}$ родительские наборы перестановок таковы, что $\pi^{11} = \pi^{12}$ и $\pi^{21} = \pi^{22}$. Если в оптимальном решении данной ЗОР перестановки π'^1 и π'^2 не совпадают, то, используя подход из [4], можно показать, что набор перестановок $\{\pi''^l\}_{l \in M}$, где $\pi''^1 = \pi''^2 = \pi'^2$, также оптимальный для рассматриваемой ЗОР. Тогда из теоремы 3 непосредственно следует требуемое утверждение. Следствие 2 доказано.

Задача оптимальной рекомбинации (i′), (ii′) для $F|prmu|\gamma$, $\gamma \in \{C_{\max}, L_{\max}\}$, может быть решена с помощью алгоритма, описанного в разд. 1. Так как для произвольной перестановки π значение целевой функции $C_{\max}(\pi)$ (или $L_{\max}(\pi)$) вычисляется за время $O(nm)$ (см. разд. 2), имеет место

Теорема 4. *Задача оптимальной рекомбинации* (i′), (ii′) *для* $F|prmu|\gamma$, $\gamma \in \{C_{\max}, L_{\max}\}$, *разрешима за время* $O(1.42^n nm)$.

Аналогичным образом из теоремы 1 вытекает

Следствие 3. *Задача оптимальной рекомбинации* (i′), (ii′) *для* $F|prmu|\gamma$, $\gamma \in \{C_{\max}, L_{\max}\}$, *для «почти всех» пар родительских перестановок разрешима за время* $O(n^2m)$.

Теорема 5. *Задача оптимальной рекомбинации* (i), (ii) *для* $F \parallel \gamma$, $\gamma \in \{C_{\max}, L_{\max}\}$, *разрешима за время* $O(1.42^{n(m-2)}nm)$.

Доказательство. В задаче $F \parallel \gamma$, $\gamma \in \{C_{\max}, L_{\max}\}$, существует оптимальное решение, при котором сохраняется порядок выполнения работ на двух первых и двух последних машинах [4]. Поэтому будем рассматривать только такие ЗОР (i), (ii), где $\pi^{11} = \pi^{12}$, $\pi^{21} = \pi^{22}$, $\pi^{1(m-1)} = \pi^{1m}$, $\pi^{2(m-1)} = \pi^{2m}$.

С помощью подхода, описанного в разд. 1, для машины $l \in M \setminus \{1, m\}$ перебираем перестановки π^l такие, что $\pi^l_i = \pi^{1l}_i$ или $\pi^l_i = \pi^{2l}_i$ для всех $i = 1, \dots, n$. Число таких перестановок для машины $l \in M \setminus \{1, m\}$ есть $O(1.42^n)$. Для каждого набора перестановок $\{\pi^l\}_{l\in M}$, где $\pi^1 = \pi^2$, $\pi^m = \pi^{(m-1)}$, вычисляем значение целевой функции за время $O(nm)$ (см. разд. 2). В результате, за время $O(1.42^{n(m-2)}nm)$ находим требуемый набор перестановок $\{\pi'^l\}_{l\in M}$. Теорема 5 доказана.

Рассуждая аналогично доказательству теоремы 5 и используя теорему 1, получаем

Следствие 4. *Задача оптимальной рекомбинации* (i), (ii) *для* $F \parallel \gamma$, $\gamma \in \{C_{\max}, L_{\max}\}$, *разрешима за время* $O(n^{m-1}m)$ *для «почти всех» пар наборов родительских перестановок.*

Таким образом, ЗОР (i), (ii) для $Fm \parallel \gamma$, $\gamma \in \{C_{\max}, L_{\max}\}$, эффективно разрешима для «почти всех» пар наборов родительских перестановок.

4. Вычислительная сложность оптимальной рекомбинации для задачи без задержек

В задаче flowshop без задержек $F|no\text{-}wait|\gamma$ [18] предполагается, что каждая работа выполняется непрерывно одной машиной за другой. Предположим, что в задаче $F|no\text{-}wait|\gamma$ допускается $t_{il} = 0$, и это означает, что работа i на машине l не выполняется. При указанной интерпретации записи $t_{il} = 0$ в оптимальном решении задачи $F|no\text{-}wait|\gamma$ машины могут выполнять работы ненулевой длительности в различном порядке.

Будем рассматривать задачу $F|no\text{-}wait,\ prmu|\gamma$, $\gamma \in \{C_{\max},\ L_{\max}\}$, которая NP-трудна в сильном смысле уже при $m = 2$ [1]. Для перестановки $\pi = (\pi_1, \dots, \pi_n)$, определяющей порядок выполнения работ, раннее

расписание задачи $F|no\text{-}wait,\ prmu|\gamma,\ \gamma \in \{C_{\max},\ L_{\max}\}$, вычисляется за время $O(mn)$. Обозначим через $M_i = \bigl(l^i_1, \ldots, l^i_{m_i}\bigr)$ последовательность машин, на которых выполняется работа $i \in N$, а через $\pi^k_{j'}$ — работу, которая предшествует работе π_j на машине $l^{\pi_j}_k$ согласно последовательности π (если работа π_j выполняется на машине $l^{\pi_j}_k$ первой, то полагаем $\pi^k_{j'} := 0,\ t^f\bigl(\pi^k_{j'}, l^{\pi_j}_k\bigr) := 0)$, $j = 1, \ldots, n,\ k = 1, \ldots, m_j$. Тогда

$$t^{st}\bigl(\pi_1, l^{\pi_1}_1\bigr) = 0, \quad t^{st}\bigl(\pi_1, l^{\pi_1}_j\bigr) = t^f\bigl(\pi_1, l^{\pi_1}_{j-1}\bigr), \quad j = 2, \ldots, m_1,$$

$$t^{st}\bigl(\pi_i, l^{\pi_i}_1\bigr) = \max_{k=1,\ldots,m_i} \left\{ t^f\bigl(\pi^k_{i'}, l^{\pi_i}_k\bigr) - \sum_{j=1}^{k-1} t_{\pi_i, l^{\pi_i}_j} \right\}, \quad i = 2, \ldots, n,$$

$$t^{st}\bigl(\pi_i, l^{\pi_i}_j\bigr) = t^f\bigl(\pi_i, l^{\pi_i}_{j-1}\bigr), \quad i = 2, \ldots, n,\ j = 2, \ldots, m_i,$$

$$C_{\max}(\pi) = \max\bigl\{t^f\bigl(i, l^i_{m_i}\bigr) \mid i \in N\bigr\},$$

$$L_{\max}(\pi) = \max\bigl\{t^f\bigl(i, l^i_{m_i}\bigr) - d_i \mid i \in N\bigr\}.$$

Задача оптимальной рекомбинации для $F|no\text{-}wait,\ prmu|\gamma$ формулируется так же, как и для задачи $F|prmu|\gamma$.

Теорема 6. *Задача оптимальной рекомбинации* (i′), (ii′) *для* $F2|no\text{-}wait,\ prmu|C_{\max}$ NP*-трудна, если запись* $t_{il} = 0$ *означает, что работа* i *на машине* l *не обслуживается.*

Доказательство. Сформулируем для ЗОР (i′), (ii′) соответствующую задачу распознавания: определить, существует ли такая перестановка π', удовлетворяющая условию (i′), при которой $C_{\max}(\pi') \leqslant y$ для заданного числа y. Покажем, что к этой задаче полиномиально сводится задача Упорядоченное Разбиение.

В задаче $F|no\text{-}wait,\ prmu|C_{\max}$ положим $m = 2$ — число машин, $n = 2n_0+3$ — число работ. Работы множества N разобьём на две группы: U-работы, соответствующие элементам $a_i \in A$, $i = 1, \ldots, 2n_0$, и обозначаемые через U_i, и V-работы, обозначаемые через V_i, $i = 0, 1, 2$. Пусть

$$t_{U_i,1} = 0, \quad t_{U_i,2} = e_i, \quad i = 1, \ldots, 2n_0,$$

$$t_{V_0,1} = t_{V_1,1} = t_{V_2,1} = 2E, \quad t_{V_0,2} = t_{V_1,2} = t_{V_2,2} = E.$$

Определим родительские перестановки для ЗОР:

$$\pi^1 = (V_0, U_1, U_3, \ldots, U_{2n_0-1}, V_1, U_2, U_4, \ldots, U_{2n_0}, V_2),$$

$$\pi^2 = (V_0, U_2, U_4, \ldots, U_{2n_0}, V_1, U_1, U_3, \ldots, U_{2n_0-1}, V_2).$$

Поскольку U-работы не выполняются на машине 1, перестановки π^1 и π^2 порождают следующие подпоследовательности для указанной машины: $\widetilde{\pi}^1 = (V_0, V_1, V_2)$, $\widetilde{\pi}^2 = (V_0, V_1, V_2)$.

Покажем, что в построенной ЗОР перестановка работ π', для которой выполняется условие (i$'$) и $C_{\max}(\pi') \leqslant 7E$, существует тогда и только тогда, когда на вопрос в задаче Упорядоченное Разбиение ответ «да».

(I) Пусть на вопрос в задаче Упорядоченное Разбиение ответ «да» и A_1, A_2 — подмножества множества A такие, что $\sum\limits_{a_i \in A_1} e_i = \sum\limits_{a_i \in A_2} e_i = E$ и $|A_1| = |A_2| = n_0$.

Обозначим через $N_1(U) = (U_{j_1}, \ldots, U_{j_{n_0}})$ и $N_2(U) = (U_{k_1}, \ldots, U_{k_{n_0}})$ перестановки U-работ, соответствующих элементам из A_1 и A_2 соответственно, где $j_i < j_{i+1}$ и $k_i < k_{i+1}$ для всех $i = 1, \ldots, n_0 - 1$. Тогда для перестановки $\pi' = (V_0, N_1(U), V_1, N_2(U), V_2)$ выполняется условие (i$'$) и $C_{\max}(\pi') = 7E$ (рис. 4). При этом машины функционируют без простоев: машина 1 — в интервале $(0, 6E]$ (обслуживает работы в последовательности $\tilde{\pi}' = (V_0, V_1, V_2)$), а машина 2 — в интервале $(2E, 7E]$ (обслуживает работы в последовательности π').

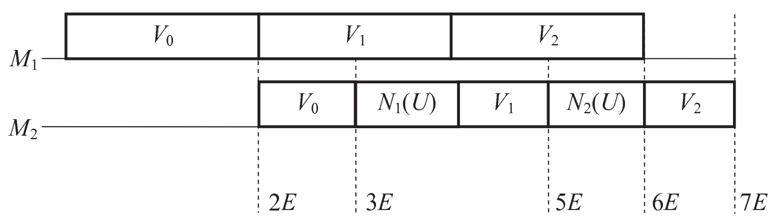

Рис. 4. Структура расписания для перестановки $(V_0, N_1(U), V_1, N_2(U), V_2)$ в задаче $F2|no\text{-}wait,\ prmu|C_{\max}$

(II) Пусть существует перестановка π', для которой выполняется условие (i$'$) и $C_{\max}(\pi') \leqslant 7E$. Обозначим через $N'_j(U)$ множество работ, стоящих в перестановке π' между работами V_{j-1} и V_j, $j = 1, 2$.

Поскольку суммарная длительность всех работ на первой машине равна $6E$, на второй машине — $5E$, $t_{V_0,1} = 2E$, $t_{V_2,2} = E$, обе машины функционируют без простоев: машина 1 — в интервале $(0, 6E]$, а машина 2 — в интервале $(2E, 7E]$.

Так как задержки при выполнении работ запрещены, работа V_0 выполняется на машине 1 в интервале $(0, 2E]$, а на машине 2 — в интервале $(2E, 3E]$; работа V_1 выполняется на машине 1 в интервале $(2E, 4E]$, а на машине 2 — в интервале $(4E, 5E]$; работа V_2 выполняется на машине 1 в интервале $(4E, 6E]$, а на машине 2 — в интервале $(6E, 7E]$. Тогда

U-работы обслуживаются на машине 2 в интервалах $(3E, 4E]$ и $(5E, 6E]$, следовательно, $\sum\limits_{U_i \in N'_1(U)} e_i = E$ и $\sum\limits_{U_i \in N'_2(U)} e_i = E$.

В результате, множества $A_1 = \{a_i \in A \mid U_i \in N'_1(U)\}$ и $A_2 = \{a_i \in A \mid U_i \in N'_2(U)\}$ соответствуют условиям задачи Упорядоченное Разбиение, и данная индивидуальная задача имеет ответ «да».

Све́дение задачи Упорядоченное Разбиение к рассматриваемой задаче распознавания построено за $O(n_0)$ операций, следовательно, ЗОР (i$'$), (ii$'$) для $F|no\text{-}wait,\ prmu|C_{\max}$ NP-трудна. Теорема 6 доказана.

Если в задаче $F|no\text{-}wait,\ prmu|L_{\max}$ положить $d_i = 0,\ i \in N$, то из теоремы 6 непосредственно вытекает

Следствие 5. *Задача оптимальной рекомбинации* (i$'$), (ii$'$) *для* $F2|no\text{-}wait,\ prmu|L_{\max}$ NP-*трудна, если запись* $t_{il} = 0$ *означает, что работа* i *на машине* l *не обслуживается.*

Задача оптимальной рекомбинации (i$'$), (ii$'$) для $F|no\text{-}wait,\ prmu|\gamma$, $\gamma \in \{C_{\max},\ L_{\max}\}$, может быть решена с помощью алгоритма, описанного в разд. 1. Поскольку для произвольной перестановки π значение целевой функции $C_{\max}(\pi)$ (или $L_{\max}(\pi)$) вычисляется за время $O(nm)$, имеет место

Теорема 7. *Задача оптимальной рекомбинации* (i$'$), (ii$'$) *для* $F|no\text{-}wait,\ prmu|\gamma$, $\gamma \in \{C_{\max},\ L_{\max}\}$, *разрешима за время* $O(1.42^n nm)$, *если запись* $t_{il} = 0$ *означает, что работа* i *на машине* l *не обслуживается.*

Аналогичным образом из теоремы 1 получаем

Следствие 6. *Задача оптимальной рекомбинации* (i$'$), (ii$'$) *для* $F|no\text{-}wait,\ prmu|\gamma$, $\gamma \in \{C_{\max},\ L_{\max}\}$, *разрешима за время* $O(n^2 m)$ *для «почти всех» пар родительских перестановок, если запись* $t_{il} = 0$ *означает, что работа* i *на машине* l *не обслуживается.*

Заключение

Представлены оценка числа допустимых решений задачи оптимальной рекомбинации для задач на перестановках и точный алгоритм её решения, основанный на переборе совершенных паросочетаний в специальном двудольном графе.

Полученные результаты показывают, что оптимальная рекомбинация для задач $Fm \parallel \gamma$, $F|prmu|\gamma$, $F|no\text{-}wait,\ prmu|\gamma$, где $\gamma \in \{C_{\max},\ L_{\max}\}$, NP-трудна, однако оптимальный потомок может быть найден за полиномиальное время для «почти всех» пар родительских решений при числе работ, стремящемся к бесконечности.

Для дальнейшего анализа представляет интерес экспериментальное исследование предложенного оператора оптимальной рекомбинации и исследование NP-трудности в сильном смысле рассматриваемых задач оптимальной рекомбинации. Также открытым является вопрос о сложности задачи оптимальной рекомбинации для $F|no\text{-}wait|\gamma$, если запись $t_{il} = 0$ означает, что работа i на машине l обслуживается, но длительностью этого обслуживания можно пренебречь.

Автор благодарен А. В. Еремееву за полезные замечания.

ЛИТЕРАТУРА

1. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
2. **Еремеев А. В., Коваленко Ю. В.** О задаче составления расписаний с группировкой машин по технологиям // Дискрет. анализ и исслед. операций. 2011. Т. 18, № 5. С. 54–79.
3. **Еремеев А. В., Коваленко Ю. В.** О сложности оптимальной рекомбинации для одной задачи составления расписаний с переналадками // Дискрет. анализ и исслед. операций. 2012. Т. 19, № 3. С. 13–26.
4. **Конвей Р. В., Максвелл В. Л., Миллер Л. В.** Теория расписаний. М.: Наука, 1975. 360 с.
5. **Кормен Т., Лейзерсон Ч., Ривест Р.** Алгоритмы: построение и анализ. М.: МЦНМО, 2001. 960 с.
6. **Рутковская Д., Пилиньский М., Рутковский Л.** Нейронные сети, генетические алгоритмы и нечёткие системы. М.: Горячая линия-Телеком, 2006. 452 с.
7. **Сердюков А. И.** О задаче коммивояжёра при наличии запретов // Управляемые системы. 1978. Вып. 17. С. 80–86.
8. **Танаев В. С., Сотсков Ю. Н., Струсевич В. А.** Теория расписаний. Многостадийные системы. М.: Наука, 1989. 328 с.
9. **Balas E., Niehaus W.** Optimized crossover-based genetic algorithms for the maximum cardinality and maximum weight clique problems // J. Heuristics. 1998. Vol. 4, No. 2. P. 107–122.
10. **Cheng B.-W., Chang C.-L.** A study on flowshop scheduling problem combining Taguchi experimental design and genetic algorithm // Expert. Syst. Appl. 2007. Vol. 32, No. 2. P. 415–421.
11. **Chvatal V.** Probabilistic methods in graph theory // Ann. Oper. Res. 1984. Vol. 1, No. 3. P. 171–182.
12. **Cook W., Seymour P.** Tour merging via branch-decomposition // INFORMS J. Comput. 2003. Vol. 15, No. 3. P. 233–248.

13. **Cotta C., Alba E., Troya J. M.** Utilizing dynastically optimal forma recombination in hybrid genetic algorithms // Proc. 5th Int. Conf. Parallel Problem Solving from Nature (Amsterdam, Sept. 27–30, 1998). Heidelberg: Springer-Verl., 1998. P. 305–314. (Lect. Notes Comput. Sci.; Vol. 1498).
14. **Cotta C., Troya J. M.** Genetic forma recombination in permutation flowshop problems // Evol. Comput. 1998. Vol. 6, No. 1. P. 25–44.
15. **Eremeev A. V.** On complexity of optimal recombination for binary representations of solutions // Evol. Comput. 2008. Vol. 16, No. 1. P. 127–147.
16. **Eremeev A. V., Kovalenko Yu. V.** Optimal recombination in genetic algorithms for combinatorial optimization problems: Part I // Yugoslav J. Oper. Res. 2014. Vol. 24, No. 1. P. 1–20.
17. **Eremeev A. V., Kovalenko Yu. V.** Optimal recombination in genetic algorithms for combinatorial optimization problems: Part II // Yugoslav J. Oper. Res. 2014. Vol. 24, No. 2. P. 165–186.
18. **Graham R. L., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G.** Optimization and approximation in deterministic sequencing and scheduling: A survey // Discrete Optimization II. Amsterdam, North-Holland: 1979. P. 287–326. (Ann. Discrete Math., Vol. 5).
19. **Holland J. H.** Adaptation in natural and artificial systems. Ann Arbor: Univ. Michigan Press, 1975. 183 p.
20. **Nagano M. S., Ruiz R., Lorena L. A. N.** A constructive genetic algorithm for permutation flowshop scheduling // Comput. Ind. Eng. 2008. Vol. 55, No. 1. P. 195–207.
21. **Pinedo M. L.** Scheduling: theory, algorithms and systems. Upper Saddle River, NJ: Prentice Hall, 2002. 676 p.
22. **Radcliffe N. J.** The algebra of genetic algorithms // Ann. Math. Artif. Intell. 1994. Vol. 10, No. 4. P. 339–384.
23. **Tinós R., Whitley D., Ochoa G.** Generalized asymmetric partition crossover (GAPX) for the asymmetric TSP // Proc. 2014 Annu. Conf. Genetic and Evolutionary Computation (Vancouver, Canada, July 12–16, 2014). New York: ACM, 2014. P. 501–508.
24. **Yagiura M., Ibaraki T.** The use of dynamic programming in genetic algorithms for permutation problems // Eur. J. Oper. Res. 1996. Vol. 92, No. 2. P. 387–401.
25. **Wang L., Zhang L.** Determining optimal combination of genetic operators for flow shop scheduling // Int. J. Adv. Manuf. Technol. 2006. Vol. 30, No. 3–4. P. 302–308.

26. **Zhang L., Wang L., Zheng D.-Z.** An adaptive genetic algorithm with multiple operators for flowshop scheduling // Int. J. Adv. Manuf. Technol. 2006. Vol. 27, No. 5–6. P. 580–587.

Коваленко Юлия Викторовна

Статья поступила
21 января 2016 г.

Исправленный вариант —
11 февраля 2016 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
April–June 2016. Volume 23, No. 2. P. 41–62

UDC 519.8 DOI: 10.17377/daio.2016.23.524

ON COMPLEXITY OF OPTIMAL RECOMBINATION FOR FLOWSHOP SCHEDULING PROBLEMS

Yu. V. Kovalenko[1]
[1]Sobolev Institute of Mathematics,
4 Koptyug Ave., 630090 Novosibirsk, Russia
e-mail: julia.kovalenko.ya@yandex.ru

Abstract. Under study is the complexity of optimal recombination for various flowshop scheduling problems with the makespan criterion and the criterion of maximum lateness. The problems are proved to be NP-hard, and a solution algorithm is proposed. In the case of a flowshop problem on permutations, the algorithm is shown to have polynomial complexity for "almost all" pairs of parent solutions as the number of jobs tends to infinity. Ill. 4, bibliogr. 26.

Keywords: flowshop problem, permutation, genetic algorithm, optimal recombination.

REFERENCES

1. **M. R. Garey** and **D. S. Johnson,** *Computers and Intractability: A Guide to the Theory of NP-Completeness,* Freeman, San Francisco, 1979. Translated under the title *Vychislitel'nye mashiny i trudnoreshaemye zadachi,* Mir, Moscow, 1982.
2. **A. V. Eremeev** and **Yu. V. Kovalenko,** On scheduling with technology based machines grouping, *Diskretn. Anal. Issled. Oper.,* **18**, No. 5, 54–79, 2011.
3. **A. V. Eremeev** and **Yu. V. Kovalenko,** On complexity of optimal recombination for one scheduling problem with setup times, *Diskretn. Anal. Issled. Oper.,* **19**, No. 3, 13–26, 2012.
4. **R. W. Conway, W. L. Maxwell,** and **L. W. Miller,** *Theory of Scheduling,* Addison-Wesley, Reading, MA, USA, 1967. Translated under the title *Teoriya raspisanii,* Nauka, Moscow, 1975.
5. **T. H. Cormen, C. E. Leiserson,** and **R. L. Rivest,** *Introduction to Algorithms,* MIT Press, Cambridge, MA, 1990. Translated under the title *Algoritmy: postroenie i analiz,* MTsNMO, Moscow, 2001.

6. D. Rutkowska, M. Piliński, and **L. Rutkowski,** *Neural Networks, Genetic Algorithms and Fuzzy Systems,* Naukowe PWN, Warsaw, 1997 [Polish]. Translated under the title *Neironnye seti, geneticheskie algoritmy i nechyotkie sistemy,* Goryachaya liniya–Telekom, Moscow, 2006.

7. A. I. Serdyukov, On travelling salesman problem with prohibitions, in *Upravlyaemye sistemy* (Controlled Systems), Vol. 17, pp. 80–86, Inst. Mat. SO AN SSSR, Novosibirsk, 1978.

8. V. S. Tanaev, Yu. N. Sotskov, and **V. A. Strusevich,** *Teoriya raspisanii. Mnogostadiinye sistemy* (Theory of Scheduling. Multi-Stage Systems), Nauka, Moscow, 1989.

9. E. Balas and **W. Niehaus,** Optimized crossover-based genetic algorithms for the maximum cardinality and maximum weight clique problems, *J. Heuristics,* **4**, No. 2, 107–122, 1998.

10. B.-W. Cheng and **C.-L. Chang,** A study on flowshop scheduling problem combining Taguchi experimental design and genetic algorithm, *Expert Syst. Appl.,* **32**, No. 2, 415–421, 2007.

11. V. Chvátal, Probabilistic methods in graph theory, *Ann. Oper. Res.,* **1**, No. 3, 171–182, 1984.

12. W. Cook and **P. Seymour,** Tour merging via branch-decomposition, *INFORMS J. Comput.,* **15**, No. 3, 233–248, 2003.

13. C. Cotta, E. Alba, and **J. M. Troya,** Utilizing dynastically optimal forma recombination in hybrid genetic algorithms, in *Parallel Problem Solving from Nature — PPSN V* (Proc. 5th Int. Conf. PPSN, Amsterdam, The Netherlands, Sept. 27–30, 1998), pp. 305–314, Springer, Heidelberg, 1998 (Lect. Notes Comput. Sci., Vol. 1498).

14. C. Cotta and **J. M. Troya,** Genetic forma recombination in permutation flowshop problems, *Evol. Comput.,* **6**, No. 1, 25–44, 1998.

15. A. V. Eremeev, On complexity of optimal recombination for binary representations of solutions, *Evol. Comput.,* **16**, No. 1, 127–147, 2008.

16. A. V. Eremeev and **Ju. V. Kovalenko,** Optimal recombination in genetic algorithms for combinatorial optimization problems: Part I, *Yugoslav J. Oper. Res.,* **24**, No. 1, 1–20, 2014.

17. A. V. Eremeev and **Ju. V. Kovalenko,** Optimal recombination in genetic algorithms for combinatorial optimization problems: Part II, *Yugoslav J. Oper. Res.,* **24**, No. 2, 165–186, 2014.

18. R. L. Graham, E. L. Lawler, J. K. Lenstra, and **A. H. G. Rinnooy Kan,** Optimization and approximation in deterministic sequencing and scheduling: A survey, in *Discrete Optimization II,* pp. 287–326, North-Holland, Amsterdam, 1979 (Ann. Discrete Math., Vol. 5).

19. J. H. Holland, *Adaptation in Natural and Artificial Systems,* University of Michigan Press, Ann Arbor, 1975.

20. **M. S. Nagano, R. Ruiz,** and **L. A. N. Lorena,** A constructive genetic algorithm for permutation flowshop scheduling, *Comput. Ind. Eng.,* **55**, No. 1, 195–207, 2008.
21. **M. L. Pinedo,** *Scheduling: Theory, Algorithms, and Systems,* Prentice Hall, Upper Saddle River, USA, 2002.
22. **N. J. Radcliffe,** The algebra of genetic algorithms, *Ann. Math. Artif. Intell.,* **10**, No. 4, 339–384, 1994.
23. **R. Tinós, D. Whitley,** and **G. Ochoa,** Generalized asymmetric partition crossover (GAPX) for the asymmetric TSP, in *Proc. 2014 Annual Conf. Genetic Evol. Comput., Vancouver, Canada, July 12–16, 2014,* pp. 501–508, ACM, New York, 2014.
24. **M. Yagiura** and **T. Ibaraki,** The use of dynamic programming in genetic algorithms for permutation problems, *Eur. J. Oper. Res.,* **92**, No. 2, 387–401, 1996.
25. **L. Wang** and **L. Zhang,** Determining optimal combination of genetic operators for flow shop scheduling, *Int. J. Adv. Manuf. Technol.,* **30**, No. 3–4, 302–308, 2006.
26. **L. Zhang, L. Wang,** and **D.-Z. Zheng,** An adaptive genetic algorithm with multiple operators for flowshop scheduling, *Int. J. Adv. Manuf. Technol.,* **27**, No. 5–6, 580–587, 2006.

Yulia V. Kovalenko

Received
21 January 2016
Revised
11 February 2016