УДК 519.8
DOI: 10.20535/SRIT.2308-8893.2019.3.09

АЛГОРИТМЫ ПРИБЛИЖЕННОГО РЕШЕНИЯ МНОГОСТАДИЙНЫХ FLOW-SHOP-PROBLEM

Ю.А. ЗАК

Аннотация. Построение многостадийных расписаний выполнения заданий на расположенных в последовательную цепочку системах машин имеет много практических приложений в календарном планировании дискретного производства. Получены оценки нижней границы критерия эффективности для оптимальной последовательности выполнения заданий и два алгоритма приближенного решения задач, обеспечивающие выполнение всех работ на всех стадиях обработки в кратчайшие сроки. Алгоритмы решения проиллюстрированы числовым примером. Приведены оценки сложности предложенных алгоритмов. Алгоритмы решения задачи могут быть использованы в календарном планировании мелко- и среднесерийного дискретного производства.

Ключевые слова: многостадийные расписания**,** Flow-Shop-Problem, оптимальные последовательности, нижняя граница времени выполнения заданий, приближенное решение, эвристический алгоритм.

ВВЕДЕНИЕ. ПОСТАНОВКА ЗАДАЧИ

Технологический процесс предусматривает выполнение заданий и изготовление изделий на S последовательных стадиях обработки. На этих стадиях все изделия обрабатываются на каждой стоящей в последовательной цепочке системе машин и как на первой стадии обработки, так и на каждой S-й, $s=1,...,S$ в одной и той же последовательности. Каждая стадия обработки включает m_s машин, $s=1,...,S$. Количество машин на каждой стадии обработки может быть различным. Более сложной постановкой задачи может быть ситуация, когда допускаются отличные друг от друга последовательности обработки изделий на каждой стадии изготовления.

Необходимо определить оптимальную последовательность обработки изделий, а также время начала и время завершения обработки группы, состоящей из n $\left(i=1,...,n\right)$ изделий, одинаковую на всех машинах S стадий обработки, обеспечивающих выполнение всего комплекса работ в кратчайшие сроки. Задача построения расписания на одной стадии обработки широко известна в литературе [1–4, 6, 7] и относится к классу Flow-Shop-Problem. Все перемещения обрабатываемых изделий, связанные с окончанием ее на одной машине и началом выполнения на другой, следуют только в одном направлении.

Обобщенная задача Джонсона (Flow-Shop-Problem) формулируется следующим образом [2, 3, 6,7]:

— на некоторой последовательности, состоящей из m, $k=1,...,m$ рабочих станций (машин), необходимо выполнить n, $i=1,...,n$, заданий (обработку n изделий);

— каждое из заданий состоит из некоторой упорядоченной последовательности выполнения m работ (операций) на различных рабочих станциях (машинах). Никакая машина не может выполнять более одной операции одновременно;

— длительности выполнения каждой из этих операций t_{ij} известны и не зависят от последовательности выполнения остальных операций на этой или других машинах;

— никакая из этих операций не допускает прерываний в процессе ее выполнения;

— каждое из заданий выполняется в строго заданной и одинаковой для всех заданий последовательности $M = (k = 1,...,m)$, $K = (k = 1,...,m)$;

— каждая из операций назначается только на одну определенную машину;

— последовательность прохождения рабочих станций является заданной и одинаковой для всех заданий.

Необходимо определить последовательность выполнения заданий, обеспечивающую минимальное время выполнения всех работ.

Одностадийные расписания Flow-Shop-Problem относятся к классу NP-полных задач экспоненциальной сложности. Рассматриваемые в литературе многочисленные эвристические методы и правила предпочтения [1–3, 6–10] позволяют зачастую, как показывают вычислительные эксперименты, получить хорошие приближения к оптимальному решению задачи. Правила последовательного улучшения последовательности выполнения заданий путем обмена местами двух индексов [2, 3] имеют либо локальный, либо статистический характер и поэтому не могут рассматриваться как методы локального спуска в зону глобального минимума. Создание гибридных методов, которые на основе хороших эвристик позволяют осуществить попадание в зону глобального минимума, а затем методами локальных перестановок местами заданий осуществить спуск в точку локального минимума этой области. Никакая из предложенных в литературе эвристик не гарантирует попадания в область глобального минимума. Сравнительно небольшой объем вычислений эвристических алгоритмов и методов локальных вариаций обеспечивают их широкое применение в практических приложениях и в условиях большой размерности. Эффективные точные и приближенные методы решения Flow-Shop-Problem в условиях ограничений на сроки и частичные порядки выполнения заданий рассмотрены в работах [6, 7, 10] и в работах автора [2, 3]. Несмотря на большое количество приложений в календарном планировании производства, задачам многостадийного построения расписаний в этих условиях не уделялось достаточного внимания в литературе. Сформулированная задача относится к классу NP-полных проблем. В данной работе рассматриваются приближенные методы решения сформулированной проблемы, которые могут найти практическое применение в календарном планировании мелко- и среднесерийного дискретного производства

ОЦЕНКА НИЖНЕЙ ГРАНИЦЫ ДЛИНЫ РАСПИСАНИЯ FLOW-SHOP-PROBLEM

Рассмотрим вначале оценку минимальной длины расписания Flow-Shop-Problem. Ясно, что общая продолжительность расписания не может быть меньше длительности выполнения каждого из заданий:

$$F^0 = F \geq \max_{1\leq i\leq n} \sum_{k=1}^{m} t_{ik} .$$

Остальные задания при этом должны быть выполнены либо перед, либо после выполнения этого задания. Следовательно, дополнительные затраты времени равны

$$q_i = \sum_{i=1}^{n} \min\{(t_{l,1} t_{,l,m}) \mid i \neq l\} , \; l = 1,\ldots,n .$$

Нижняя граница вычисляется по формуле

$$F^1 = \max_{1\leq i\leq n}[q_i + \sum_{k=1}^{m} t_{ik}] .$$

Нижняя граница длины оптимального расписания может быть вычислена из следующих соображений.

Для начала работы каждой k-й машины необходимо, чтобы все операции $l = 1,\ldots,k-1$, предшествующие этой k-й операции, были уже выполнены. Для этого в i-м задании требуется времени не менее чем $\beta_1(i,k) = \sum_{l=1}^{k-1} t_{il}$ даже в том случае, если в этом задании ни одна из машин не теряет времени на простой после выполнения предыдущей операции. Суммарное время выполнения всех операций на k-й машине равно $\sum_{i=1}^{n} t_{ik}$. После выполнения операций всех заданий на k-й машине требуется дополнительное время работы машин, $l = k+1,\ldots,m$: $\beta_2(i,k) = \sum_{l=k+1}^{m} t_{il}$. Это время необходимо для завершения выполнения i-го задания. Следовательно, оценка времени выполнения расписания из условия занятости рабочих систем определяется следующим образом:

$$F^2 = \max_{1\leq k\leq m}\left\{ \sum_{l}^{m} t_{il} + \min_{1\leq i\leq n} \beta_1(i,k) + \min_{1\leq i\leq n} \beta_2(i,k) \right\} .$$

Таким образом значение нижней границы критерия оптимальности определяется выражением

$$\varphi(F) = \max(F^1, F^2) . \tag{1}$$

Выражение нижней границы длины расписания в виде (1) известно и широко применяется в литературе для получения точных и приближенных оценок решений этой задачи [2, 3].

Flow-Shop-Problem для двух машин решена Джонсоном еще в 1950 г. [5]. Точное решение задачи получено с помощью алгоритма, сложность которого равна $O(n \log n)$. Алгоритм упорядочивает по возрастанию все задания, для которых $t_{i1} \le t_{i2}$, по возрастанию значений t_{i1}, а задания, для которых $t_{i1} > t_{i2}$, — и по убыванию времени выполнения задания. В первую очередь выполняются задания из первой подпоследовательности, а затем из второй. Эти подходы в практических приложениях используются в алгоритмах получения приближенных решений задачи.

ВРЕМЯ НАЧАЛА И ВРЕМЯ ЗАВЕРШЕНИЯ ВЫПОЛНЕНИЯ ЗАДАНИЙ НА ВСЕХ МАШИНАХ КАЖДОЙ СТАДИИ ОБРАБОТКИ

Пусть $\widetilde{I} = \{i = 1,...,n\}$ — множество всех изделий, подлежащих обработке; m_s, $s = 1,...,S$ — соответственно количество машин на каждой стадии обработки. Обозначим:

$x_{ik}(s)$, $\theta_{ik}(s)$ — соответственно время начала и время завершения обработки i-го изделия на s-й стадии обработки на k-й машине, $k = 1,...,m_s$;

$t_{ik}(s)$ — время продолжительности обработки i-го изделия на k-й машине на s-й стадии обработки;

$T_i(s)$ — время завершения изготовления i-го изделия на s-й стадии обработки;

$\overline{T}_i$ — время завершения изготовления i-го изделия на последней S-й стадии обработки;

Ф — время завершения выполнения всех заданий на всех стадиях обработки;

$\widetilde{I}^1, \widetilde{I}^2$ — подмножество изделий, для которых в процессе выполнения алгоритма определено и не определено место в последовательности многостадийной обработки $\widetilde{I}^1 \bigcup \widetilde{I}^2 = \widetilde{I}$, $\widetilde{I}^1 \bigcap \widetilde{I}^2 = \varnothing$.

Пусть на некотором шаге вычислительного процесса определены подмножества изделий $\widetilde{I}^1, \widetilde{I}^2$, а также последовательность многостадийной обработки подмножества изделий $\widetilde{I}^1$ — $\widetilde{U} = \{u_1, u_2, ..., u_j, ..., u_p\}$. Время начала и время завершения обработки всех изделий на каждой из машин всех стадий обработки определяются по формулам:

$$x_{u_1,1}(1) = 1, \ \ \theta_{u_1,1}(1) = t_{u_1,1}(1); \tag{2}$$

$$x_{u_k,1}(1) = \theta_{u_{k-1},1}(1) + 1, \ \ \theta_{u_k,1}(1) = x_{u_k,1}(1) + t_{u_k,1}(1) - 1, \ \ k = 2,...,m. \tag{3}$$

$$x_{u_1,1}(s) = \theta_{u_1,m}(s-1) + 1, \ \ \theta_{u_1,1}(s) = x_{u_1,1}(s) + t_{u_1,1}(s) - 1, \ \ s = 1,...,S; \tag{4}$$

$$x_{u_j,1}(s) = \max[\theta_{u_j,1}(s-1), \theta_{u_{j-1},1}(s)] + 1,$$

$$\theta_{u_j,1}(s) = x_{u_j,1}(s) + t_{u_j,1}(s) - 1\,,\ s = 2,...,S\,; \tag{5}$$

$$x_{u_j,k}(s) = \max[\theta_{u_{j-1},k}(s);\ \theta_{u_j,k-1}(s)] + 1\,,$$

$$\theta_{u_j,k}(s) = x_{u_j,k}(s) + t_{u_j,k}(s) - 1\,,\ j = 1,...,P\,,\ s = 2,...,S\,. \tag{6}$$

$$\overline{T}_i = \overline{T}_i(s)\,,\quad i = 1,2,...,n\,,\ s = 1,2,...,S\,;\quad \Phi = \overline{T}_n(S)\,.$$

ЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ ПОЛУЧЕНИЯ ПРИБЛИЖЕННЫХ РЕШЕНИЙ ЗАДАЧИ

Алгоритм 1. Приближенное решение может быть получено в соответствии с алгоритмом, аналогичным алгоритму Джонсона [5] построения расписаний для двух машин.

Определим $Q = \left|\frac{S}{2}\right|$, $G = Q + 1$, а также $M = \sum_{s=1}^{S} m_s$, $M_1 = \left|0{,}5 \cdot \sum_{s=1}^{S} m_s\right|$ и $M_2 = M_1 + 1$. Здесь $|\ .\ |$ — целая часть частного от деления двух чисел, M_2, M_1 — суммарные количества машин на различных (первой и второй) стадиях обработки.

Вычислим:

$$R_1^1(i) = \sum_{s=1}^{Q}\sum_{k=1}^{m_s} t_{ik}\,,\ R_2^1(i) = \sum_{s=1}^{G}\sum_{k=1}^{m_s} t_{ik}\,,\ R_1^2(i) = \sum_{s=Q+1}^{S}\sum_{k=1}^{m_s} t_{ik}\,;$$

$$R_2^2(i) = \sum_{s=G+1}^{S}\sum_{k=1}^{m_s} t_{ik}\,.\ i = 1,...,n\,; \tag{7}$$

$$P_1^1(i) = \sum_{k=1}^{M_1} t_{ik}\,,\ P_2^1(i) = \sum_{k=M_2}^{M} t_{ik}\,,\ P_1^2(i) = \sum_{k=1}^{M_2} t_{ik}\,,\ P_2^2(i) = \sum_{k=M_2+1}^{M_2} t_{ik}\,,\ i = 1,...,n\,.$$

Определим четыре подмножества выполняемых заданий:

$$\tilde{J}_1^1 = \{i \in \tilde{I} \mid R_1^1(i) \le R_2^1(i)\}\,,\quad \tilde{J}_2^1 = \{i \in \tilde{I} \mid R_1^1(i) > R_2^1(i)\}\,,$$

$$\tilde{J}_1^2 = \{i \in \tilde{I} \mid R_1^2(i) \le R_2^2(i)\}\,,\quad \tilde{J}_2^2 = \{i \in \tilde{I} \mid R_1^2(i) > R_2^2(i)\}\,;$$

$$\tilde{J}_1^3 = \{i \in \tilde{I} \mid P_1^1(i) \le P_2^1(i)\}\,,\ \tilde{J}_2^3 = \{i \in \tilde{I} \mid P_1^1(i) > P_2^1(i)\}\,,$$

$$\tilde{J}_1^4 = \{i \in \tilde{I} \mid P_1^2(i) \le P_2^2(i)\}\,,\ \tilde{J}_2^4 = \{i \in \tilde{I} \mid P_1^2(i) > P_2^2(i)\}\,.$$

$$\tilde{J}_1^3 = \{i \in \tilde{I} \mid P_1^1(i) \le P_2^1(i)\}\,,\ \tilde{J}_2^3 = \{i \in \tilde{I} \mid P_1^1(i) > P_2^1(i)\}\,,$$

$$\tilde{J}_2^4 = \{i \in \tilde{I} \mid P_1^2(i) > P_2^2(i)\}\,.$$

Для простоты изложения обозначим $w_j, v_j, p_j, r_j, c_j, u_j, \rho_j, \mu_j$ — индексы заданий, входящих в подмножества $\tilde{J}_1^1, \tilde{J}_2^1, \tilde{J}_1^2, \tilde{J}_2^2, \tilde{J}_1^3, \tilde{J}_2^3, \tilde{J}_1^4, \tilde{J}_2^4$.

Упорядочим подмножества заданий $\widetilde{J}_1^1$ и $\widetilde{J}_1^2$, а также $\widetilde{J}_1^3$, $\widetilde{J}_1^4 \ldots$ по возрастанию соответственно значений $R_1^1(i)$ и $R_1^2(i)$ и $P_1^2(i)$ и $P_2^2(i)$ в последовательности

$$\widetilde{W}_1^1 = \{w_1, w_2, ..., w_l, ..., w_{L_1} \mid w_l \in \widetilde{J}_1^1;\ w_1 \le w_2 \le ... \le w_l \le ... \le w_{L_1}\};$$

$$\widetilde{W}_1^2 = \{v_1, v_2, ..., v_l, ..., v_{L_2} \mid v_l \in \widetilde{J}_1^2;\ v_1 \le v_2 \le ... \le v_l \le ... \le v_{L_2}\};$$

$$\widetilde{W}_1^3 = \{\rho_1, \rho_2, ..., \rho_l, ..., \rho_{L_3} \mid \rho_l \in \widetilde{J}_1^3;\ \rho_1 \le \rho_2 \le ... \le \rho_l \le ... \le \rho_{L_3}\};$$

$$\widetilde{W}_1^4 = \{\mu_1, \mu_2, ..., \mu_l, ..., \mu_{L_4} \mid \mu_j \in \widetilde{J}_1^4;\ \mu_1 \le \mu_2 \le ... \le \mu_l \le ... \le \mu_{L_4}\},$$

а подмножества заданий $\widetilde{J}_2^1$ и $\widetilde{J}_2^2$, а также $\widetilde{J}_1^3$ и $\widetilde{J}_1^4$ по убыванию соответственно значений $R_1^2(i)$ и $R_2^2(i)$, а также $P_2^1(i)$ и $P_2^2(i)$ в последовательности

$$\widetilde{W}_2^1 = \{p_1, p_2, ..., p_j, ..., p_{L_5} \mid p_j \in \widetilde{J}_1^2;\ p_1 \ge p_2 \ge ... \ge p_j \ge ... \ge p_{L_5}\};$$

$$\widetilde{W}_2^2 = \{r_1, r_2, ..., r_j, ..., r_{L_6} \mid r_j \in \widetilde{J}_2^2;\ r_1 \ge r_2 \ge ... \ge r_j \ge ... \ge r_{L_6}\};$$

$$\widetilde{W}_2^3 = \{c_1, c_2, ..., c_j, ..., c_{L_7} \mid c_j \in \widetilde{J}_1^3;\ c_1 \ge c_2 \ge ... \ge c_j \ge ... \ge c_{L_7}\};$$

$$\widetilde{W}_2^4 = \{u_1, u_2, ..., u_j, ..., u_{L_8} \mid u_j \in \widetilde{J}_1^4;\ u_1 \ge u_2 \ge ... \ge u_j \ge ... \ge u_{L_8}\}.$$

Здесь $L_1, L_2, ..., L_8$ — количества заданий в соответствующих подмножествах.

Построим 8 расписаний выполнения заданий. В каждом из этих расписаний вначале расположены задания подмножеств $\widetilde{J}_1^d$, $d = 1,2,3,4$ в порядке, расположенном в последовательности $\widetilde{W}_1^d$, а затем подмножества заданий $\widetilde{J}_1^d$ в порядке, расположенном в последовательности $\widetilde{W}_2^d$, $d = 1,2,3,4$. Для каждого из этих расписаний выполним расчет времени начала и времени завершения выполнения всех заданий на каждой машине и на всех стадиях разработки по формулам (2)–(6). Определим для каждого из этих расписаний время завершения выполнения всех работ и обозначим их: $\overline{T}_g = \theta_{u_{L_g}, m_s}(S)$, $g = 1,...,8$. В качестве решения задачи выбираем расписание с наименьшим значением критерия оптимальности.

Для получения грубых приближенных решений может использоваться только одна из последовательностей $\widetilde{U}^r = \{\widetilde{W}_1^r, \widetilde{W}_2^r\}$, $r = 1,2,3,4$. Предложенный алгоритм является алгоритмом полиномиальной сложности с объемом вычислений того же порядка, что и алгоритм Джонсона $O(n \log n)$.

Алгоритм 2. В начале вычислительного процесса, когда $l = 0$, положим $\widetilde{I}^1(l = 0) = \varnothing$, $\widetilde{I}^2(l = 0) = \widetilde{I}$.

Вычислим $H^0(i)=\sum_{s=1}^{S}\sum_{k=1}^{m_s} t_{ik}(s)$. Определим последовательность выполнения заданий $\widetilde{U}^0=\{u\in\widetilde{I}\mid H^0(u_1)\le H^0(u_2)\le .....\le H^0(u_n)\}$.

На первое место в расписании выполнения заданий поставим задание с индексом u_1 — $\widetilde{Z}(l=1)=\{u_1\}$. Здесь $\widetilde{Z}(l)$ — строящаяся последовательность выполнения заданий.

Вычислим время начала и время завершения выполнения этого задания на всех машинах и на всех стадиях обработки деталей по формулам (2)–(4). Определяем $\widetilde{I}^1(l=1)=\{u_1\}$, $\widetilde{I}^2(l=1)=\{\widetilde{I}/u_1\}$.

На следующих шагах алгоритма $l=1,2,...,(n-1)$ выполняются следующие вычисления.

1. Для каждого задания из подмножества $i\in\widetilde{I}^2(l)$ рассчитываем по формулам (5), (6) время начала и время завершения выполнения каждого из заданий, поставив их на первое место после стоящего последним в подпоследовательности $\widetilde{Z}(l-1)=\{v_1,v_2,...,v_{l-1}\}$, т.е. зная значения $x_{v_{l-1},k}(s)$, $\theta_{v_{l-1},k}(s)$, $k=1,2,...,m_s$, $s=1,2,...,S$ рассчитываем

$$x_{i,1}(1)=\theta_{v_{l-1},1}(1)+1;\quad \theta_{i,1}(1)=x_{i,1}(1)+t_{i,1}(1)-1;$$

$$x_{i,1}(1)=\max[\theta_{i,k-1}(1),\, x_{v_{l-1},k}(1)]+1;\quad x_{i,1}(s)=\max[\theta_{v_{l-1},1}(s),\, \theta_{i,m_{s-1}}(s-1)]+1;$$

$$\theta_{i,k}(s)=x_{i,k}(s)+t_{i,k}(s)-1;\quad k=1,2,...,m_s;\ s=1,2,...,S.$$

2. Определяем

$$F_i(l)=\theta_{i,m_S}(S),\quad i\in\widetilde{I}^2(l);\quad F_\sigma(l)=\min_{i\in\widetilde{I}^2(l)} F_i(l).$$

Устанавливаем задание с индексом $i=\sigma$ на последнее место в строящейся последовательности $\widetilde{Z}(l)=\{v_1,v_1,...,v_{l-1},v_l=\sigma\}$. Определяем

$$\widetilde{I}^1(l)=\{\widetilde{I}^1(l-1)\cup\sigma\},\quad \widetilde{I}^2(l)=\{\widetilde{I}^2(l-1)/\sigma\}.$$

Если $\widetilde{I}^1(l)=\{1,2,...,n\}$, $\widetilde{I}^2(l)=\varnothing$, то получено решение задачи и значение $F_\sigma(l)$ — время выполнения расписания, т.е. значение критерия оптимальности. В противном случае переходим к выполнению оценки нижней границы длины расписания Flow-Shop-Problem.

Предложенный алгоритм также является алгоритмом полиномиальной сложности с объемом вычислений $O([n\cdot\log n]^2)$. Алгоритм 2 требует существенно большего объема вычислений, чем алгоритм 1, но, как правило, обеспечивает более точное решение задачи.

НИЖНЯЯ ГРАНИЦА ДЛИНЫ МНОГОСТАДИЙНОГО РАСПИСАНИЯ

Время выполнения многостадийного расписания не может быть меньше максимального значения нижней границы времени выполнения всех работ на какой-либо одной стадии обработки изделий. Это значение должно быть увеличено как минимум на сумму времени выполнения на всех машинах одного заданий на стадиях, предшествующих этой s -й $l = 1, 2, \ldots, s-1$, т.е. на величину $D(s) = \sum_{l=1}^{s\text{-}1} \min_{1 \le i \le n} \sum_{k=1}^{m_l} t_{ik}(l)$, а также на сумму времени выполнения заданий на последней машине каждого из заданий на стадиях обработки $q = (s+1), (s+2), \ldots, S$, т.е. величины $C(s) = \sum_{\text{q}=(s+1)}^{\text{S}} \min_{1 \le i \le n} t_{\,i, m_q}(q)$.

Следовательно, нижняя граница длины многостадийного расписания равна $\varphi(\Phi_1) = \max_{1 \le s \le S} \{\Phi_1(\text{s}) + D(s) + C(s)\}$.

ИЛЛЮСТРАТИВНЫЙ ПРИМЕР

Значения времени выполнения заданий на всех машинах на трех стадиях обработки приведены в табл. 1.

Таблица 1. Значения времени выполнения заданий на всех машинах трехстадийной обработки

Номер заданий	**Время выполнения заданий**													
	Первая стадия обработки					Вторая стадия обработки				Третья стадия обработки				
	1	2	3	4	$\sum_{i=1}^{n} t_{ik}$	1	2	3	$\sum_{i=1}^{n} t_{ik}$	1	2	3	4	$\sum_{i=1}^{n} t_{ik}$
1	2	10	5	6	23	10	8	9	27	6	8	9	5	28
2	4	5	6	3	18	4	6	7	17	10	3	2	9	24
3	7	8	1	4	20	8	5	12	25	7	7	8	5	27
4	5	6	3	7	21	3	2	6	11	4	9	8	7	28
5	2	6	8	5	21	9	10	3	22	5	10	9	3	27
$\sum_{i=1}^{n} t_{ik}$	20	35	23	28		34	31	37		32	37	36	29	

Нижние границы времени завершения выполнения всех заданий на каждой стадии обработки, рассчитанные по формулам (1)–(3), соответственно равны (42, 42, 52). Нижняя граница трехстадийного расписания составляет 81.

Найдем решение задачи в соответствии с алгоритмом 1, используя только одну последовательность $\widetilde{U}^1 = \{\widetilde{W}_1^1, \widetilde{W}_2^1\}$, $M = 3$, $M_1 = 1$, $M_2 = 2$. Вычисляем: $R_1^1(i) = (23,18,20,21,21)$ $R_2^1(i) = (55,41,52,39,49)$. Так как $R_1^1(i) <$ $< R_2^1(i)$, то $\widetilde{U} = \{2,3,4,5,1\}$. Время завершения выполнения заданий на всех машинах трехстадийной обработки приведены в табл. 2. Время завершения выполнения расписания для этой последовательности равно $\Phi = \overline{\overline{T}}_1 = 108$.

Таблица 2. Последовательность выполнения заданий, полученная с использованием алгоритма 1

Номер заданий	Время выполнения заданий на различных машинах										
	Первая стадия обработки				Вторая стадия обработки			Третья стадия обработки			
	1	2	3	4	1	2	3	1	2	3	4
2	4	9	15	18	22	28	35	41	49	58	64
3	11	19	20	24	32	37	49	56	63	71	76
4	16	25	28	31	35	39	55	60	69	77	84
5	18	31	39	44	53	63	66	71	81	90	93
1	20	41	46	52	63	71	80	86	94	103	**108**

Найдем решение задачи алгоритмом 2. На 0-м шаге алгоритма значения времени завершения выполнения заданий равны $H^0(i) = (78, 59, 72, 60, 70)$. Выбираем задание с минимальным временем завершения, т.е. задание 2, и устанавливаем его на первое место в строящейся последовательности $H^l(i)$, $i \in \widetilde{I}^2(l)$, $l = 1,...,4$, табл. 3, в которой жирными цифрами выделены минимальные значения. Выбранные минимальные по времени завершения выполнения задания на каждом шаге алгоритма включены в строящеюся последовательность выполнения заданий. Значения времени завершения выполнения заданий на всех машинах трехстадийной обработки приведены в табл. 4.

Таблица 3. Значения времени завершения выполнения заданий

Номер шагов	Расчетные значения $H^l(i)$			
	1	3	4	5
1	93	84	**73**	77
2	91	**80**		83
3	100			**92**
4	**107**			

Поскольку значение критерия оптимальности, полученное с использованием алгоритма 2 меньше, чем соответствующее значение, полученное по алгоритму 1, то это расписание со значением критерия эффективности $\Phi = 107$ принимается как решение задачи.

Таблица 4. Последовательность выполнения заданий, полученная с использованием алгоритма 2

Номер заданий	Время завершения выполнения заданий на различных машинах										
	Первая стадия обработки				Вторая стадия обработки			Третья стадия обработки			
	1	2	3	4	1	2	3	1	2	3	4
2	4	9	15	18	22	28	35	45	48	50	59
4	9	15	18	25	28	30	41	49	58	66	73
3	16	24	25	29	37	42	54	61	67	75	80
5	18	30	38	43	52	62	65	70	80	89	92
1	20	40	45	51	62	70	79	86	93	102	**107**

ЗАКЛЮЧЕНИЕ

Одностадийные и многостадийные задачи Flow-Shop-Problem широко применяются для решения задач оперативно-календарного планирования в машиностроении, приборостроении, электронной, деревообрабатывающей, легкой и других отраслях промышленности. Поскольку точное решение таких задач может быть получено только с помощью алгоритмов экспоненциальной сложности и требует больших объемов вычислений, построение эффективных приближенных методов решения этих задач может найти широкое практическое применение в системах управления производством.

ЛИТЕРАТУРА

1. *Конвей Р.В.* Теория расписаний / Р.В. Конвей, В.Л. Максвелл, Л.В. Миллер. — М.: Физматгиз, Наука. 1975. — 359 с.
2. *Зак Ю.А.* Прикладные задачи теории расписаний и маршрутизации перевозок / Ю.А. Зак. — М.: URSS, 2012. — 394 с.
3. *Зак Ю.А.* Решение обобщенной задачи Джонсона с ограничениями на сроки выполнения заданий и времена работы машин. Ч.1. Точные методы решения / Ю.А. Зак // Проблемы управления. — 2010 — № 3. — С. 17–25. Ч. 2. Приближенные методы // Проблемы управления. — № 4. — 2010. — С. 12–19.
4. *Згуровский М.З.* Принятие решений в сетевых системах с ограниченными ресурсами / М.З. Згуровский, А.А. Павлов. — К.: Наук. думка, 2010. — 573 с.
5. *Johnson S.M.* Optimal two- and three-stage production schedules with setup times included / S.M. Johnson // Naval research logistics quarterly. — 1954. — **1(1)**. P. 61–68.
6. *Brucker P.* Scheduling Algorithms / P. Brucker. — Berlin, Heidelberg und New York: Springer-Verlag, 1998.
7. *Domschke W.* Produktionsplanung. Ablauforganisatorische Aspekte / W. Domschke, A. Scholl, S. Voß. — Berlin, Heidelberg: Springer Verlag, 2005. — 456 p.
8. *Ho J.C.* A new heuristic for the *n*-job, *M*-machine problem / J.C. Ho, Y.-L. Chang // European Journal of Operational Research. — 1991. — **52**. — P. 194–202.
9. *Ogbu F.A.* The application of the simulated annealing algorithm to the solution of the $n/m/C_{\max}$ flow-shop problem / F.A. Ogbu, D.K. Smith // Computer & Operations Research. — 1990. — **17**. — P. 243–253.
10. *Hundal T.S.* An extension of Palmer's heuristic for the flow-shop scheduling problem / T.S. Hundal, J. Rajgopal // International Journal of Production Research. — 1988. — **26**. — P. 1119–1124.

Поступила 11.04.2019