

ДВУХСТАДИЙНЫЕ ЗАДАЧИ ПЛАНИРОВАНИЯ ДЛЯ ПОТОЧНОЙ ЛИНИИ

Ю.А. Зак

Аннотация. Сформулирована задача построения расписаний двухстадийной обработки множества изделий на двух расположенных в последовательную цепочку системах машин в одной и той же и одинаковой для всех изделий последовательности. Число машин на каждой стадии обработки может быть различным. Ни одна из выполняемых операций не допускает прерываний в процессе ее выполнения. Учтены также потери времени на постобработку после завершения первой стадии изготовления. В качестве критерия оптимальности рассмотрено выполнение всего комплекса работ в кратчайшие сроки. Получены оценки нижней границы оптимальной последовательности обработки изделий, обеспечивающие выполнение всех работ на двух стадиях обработки в кратчайшие сроки. Предложены алгоритмы точного и приближенного решения задачи методами ветвей и границ и динамического программирования, а также эвристический алгоритм полиномиальной сложности получения приближенных решений. Алгоритмы решения проиллюстрированы на числовом примере. Полученные результаты могут найти широкое применение в системах календарного планирования работы производственных участков и цехов машиностроительного и приборостроительного производства, а также в процессах деревообработки, электронной и легкой промышленности.

Ключевые слова: двухстадийные расписания, flow-shop-problem, оптимальные последовательности, метод ветвей и границ, динамическое программирование, эвристический алгоритм.

ВВЕДЕНИЕ. ПОСТАНОВКА ЗАДАЧИ

Рассмотрим технологический процесс, предусматривающий выполнение заданий и изготовление изделий на двух последовательных стадиях обработки. На этих стадиях все изделия обрабатываются на каждой стоящей в последовательной цепочке системе машин и, как на первой, так и на второй стадии обработки, в одной и той же последовательности. После первой стадии производится постобработка каждого изделия, связанная с испытанием и контролем качества, необходимым временем пролёживания (например, с охлаждением или нагреванием, оформлением необходимой документации, транспортными потерями времени и др.), что также требует затрат времени, различного для каждого из изделий. На первой стадии последовательная цепочка включает в себя m_1 машин, на второй — m_2 . Более сложной может быть ситуация, когда допускаются отличные друг от дру-

га последовательности обработки изделий на каждой стадии изготовления. Без ограничения общности затраты времени на постобработку при решении задачи могут рассматриваться либо как время обработки на последней машине первой стадии изготовления, либо как время обработки на включенной в цепочку первой машине второй стадии изготовления.

Необходимо определить оптимальную последовательность обработки изделий, а также времена начала и завершения обработки группы, состоящей из n , $i = 1, \dots, n$, изделий, одинаковую на всех машинах двух стадий обработки, обеспечивающие выполнение всего комплекса работ в кратчайшие сроки. Задача построения расписания на одной стадии обработки широко известна в литературе (см., например, работы [1–6]) и относится к классу задач планирования для поточной линии (англ. *Flow-Shop-Problem*). Все перемещения обрабатываемых изделий, связанные с окончанием их обработки на одной машине и началом обработки на другой, следуют только в одном направлении.



Обобщенная задача Джонсона (Flow-Shop-Problem) формулируется таким образом:

- на некоторой последовательности, состоящей из m , $k = 1, \dots, m$, машин, необходимо выполнить n , $i = 1, \dots, n$, заданий (обработку n изделий);
- каждое из заданий состоит из некоторой упорядоченной последовательности выполнения m работ (операций) на различных машинах; никакая машина не может выполнять более одной операции одновременно;
- длительности выполнения t_{ik} каждой из этих операций постоянны, известны и не зависят от последовательности выполнения остальных операций на этой или других машинах;
- ни одна из этих операций не допускает прерываний в процессе ее выполнения;
- каждое из заданий выполняется в строго заданной и одинаковой для всех заданий последовательности $K = \{1, \dots, m\}$;
- каждая из операций назначается только на одну определенную машину;
- последовательность прохождения всех машин задана и одинакова для всех заданий.

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

Одностадийные расписания Flow-Shop-Problem относятся к классу *NP*-сложных проблем, однако некоторые из них могут быть решены за полиномиальное время. Рассматриваемые в литературе многочисленные эвристические методы и правила предпочтения [3, 4, 6–11], генетические алгоритмы (см., например, работы [6, 7]) позволяют зачастую, как показывают вычислительные эксперименты, получить хорошие приближения к оптимальному решению задачи. Правила последовательного улучшения последовательности выполнения заданий путем обмена местами двух индексов [3, 4, 6] не могут рассматриваться как методы спуска в зону глобального минимума. Создание гибридных методов, которые на основе хороших эвристик позволяют осуществить попадание в зону глобального минимума, а затем методами перестановки местами двух индексов заданий осуществить спуск в точку локального минимума этой области. Ни одна из предложенных в литературе эвристик не гарантирует попадание в область глобального минимума. Сравнительно небольшой объем вычислений эвристических алгоритмов и методов локальных вариаций обеспечивает их широкое применение в практических приложениях и в условиях большой размерности [2–4]. Эффективные точные и приближенные методы решения Flow-Shop-Problem в условиях ограничений на сроки и частичные порядки выполнения заданий

рассмотрены в работах автора [3, 4]. Несмотря на большое число приложений в календарном планировании производства, задачам двухстадийного построения расписаний в этих условиях не уделялось достаточного внимания в литературе. Можно отметить лишь несколько публикаций в периодической литературе по данной тематике. В работе [12] предлагается алгоритм построения оптимального по быстрдействию расписания обслуживания множества требований в двухстадийной системе с нефиксированными маршрутами и ресурсными ограничениями. В работе [13] рассматривались свойства и приближенные методы решения задач построения расписаний многостадийной обработки изделий на одной машине в различных постановках и с различными критериями оптимальности. Многоступенчатые расписания на одной машине в условиях переналадки рассматривались в работе [14]. В зарубежных публикациях, в основном, рассматривались прикладные задачи по данной тематике: рассмотрен эвристический алгоритм для расписания действий экипажа при управлении самолетом [15]; предложен двухступенчатый подход к планированию квантовых цепей [16]; приведены приложения таких постановок для двухступенчатого режима холодной прокатки [17]. Определение размеров партий и составление графиков производства нескольких продуктов в многоступенчатых поточных производственных системах, каждая из ступеней которых состоит из разнородных параллельных производственных линий, рассматривались в работе [18].

1. 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 максимального по суммарной длительности выполнения на всех машинах задания

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

Для начала работы каждой k -й машины необходимо, чтобы все операции $l = 1, \dots, k - 1$, предшествующие этой k -й операции, были уже выполнены. Для этого в i -м задании требуется времени не менее чем $\eta_1(i, k) = \sum_{l=1}^{k-1} t_{il}$ даже в том случае, если в этом задании ни одна из машин не теряет времени на простой после выполнения предыдущей операции. Суммарное время выпол-

нения всех операций на k -й машине равно $\sum_{i=1}^n t_{ik}$.

После того, как операции всех заданий на k -й машине будут выполнены, еще требуется дополнительное время работы машин, $l = k + 1, \dots, m$, равное $\eta_2(i, k) = \sum_{l=k+1}^m t_{il}$. Это время необходимо для

завершения выполнения i -го задания. Следовательно, оценка времени выполнения расписания из условия занятости всех машин определяется выражением

$$F^0 \geq \xi(F) = \max_{1 \leq k \leq n} \left\{ \sum_{i=1}^n t_{ik} + \min_{1 \leq i \leq n} \eta_1(i, k) + \min_{1 \leq i \leq n} \eta_2(i, k) \right\}, \quad (2)$$

а с учетом оценки завершения выполнения отдельных заданий — выражением

$$F^0 \geq \xi(F) = \max \left[\max_{1 \leq k \leq m} \left\{ \sum_{i=1}^n t_{ik} + \min_{1 \leq i \leq n} \eta_1(i, k) + \min_{1 \leq i \leq n} \eta_2(i, k) \right\}, \max_{1 \leq i \leq n} \sum_{k=1}^n t_{ik} \right]. \quad (3)$$

Выражение нижней границы длины расписания в виде (3) известно и широко применяется в литературе для получения точных и оценки приближенных решений этой задачи (см., например, работы [3, 4]).

2. НИЖНЯЯ ГРАНИЦА МИНИМАЛЬНОЙ ДЛИТЕЛЬНОСТИ ДВУХСТАДИЙНОГО РАСПИСАНИЯ

Рассмотрим алгоритмы решения задачи, основанные на последовательных алгоритмах оптимизации. При построении последовательности выполнения заданий вначале определяются альтернативы установки задания на первое место в последовательности, потом на второе место и т. д. Предположим, что время постобработки, которое обозначим b_p , $i = 1, \dots, n$, рассматривается как время выполнения операций на последней машине первой стадии обработки или первой машине второй стадии изготовления изделий, т. е. $m_1 := (m_1 + 1)$ или $m_2 := (m_2 + 1)$.

Пусть $\tilde{I} = \{i = 1, \dots, n\}$ — множество всех изделий, подлежащих обработке, m_1 и m_2 — соответственно число машин на первой и второй стадиях обработки. Обозначим для каждой s -й ветви дерева решения задачи, $s = 0, 1, 2, \dots, S$, где S — число ветвей (вершин) дерева: $x_{ik}^1(s)$ и $x_{ik}^2(s)$ — вре-

мена начала обработки i -го изделия на первой и второй стадиях обработки соответственно на k -й, $k = 1, \dots, m_1$, и l -й, $l = 1, \dots, m_2$, машинах; t_{ik}^1 и t_{il}^2 — времена продолжительности обработки i -го изделия соответственно на первой и второй стадиях обработки на k -й и l -й машинах; $\theta_{ik}^1(s)$ и $\theta_{il}^2(s)$ — времена завершения обработки i -го изделия на k -й и l -й машинах соответственно на первой и второй стадиях обработки; $T_i^1(s)$ и $T_i^2(s)$ — времена завершения изготовления i -го изделия на первой и второй стадиях обработки; $\bar{T}^1(s)$, $\bar{T}^2(s)$ — времена выполнения всех работ на первой и второй стадиях обработки; $\bar{T}(s)$ — время завершения выполнения всех работ двухстадийного расписания; $\xi[\bar{T}^1(s)]$, $\xi[\bar{T}^2(s)]$ и $\xi[\bar{T}(s)]$ — нижняя граница длины расписания на первой и второй стадиях обработки и длины двухстадийного расписания; $\tilde{I}^1(s)$, $\tilde{I}^2(s)$ — подмножество изделий в s -й ветви дерева решения, для которых уже в процессе выполнения алгоритма определено и не определено место в последовательности обработки; $\tilde{I}^1(s) \cup \tilde{I}^2(s) = \tilde{I}$, $\tilde{I}^1(s) \cap \tilde{I}^2(s) = \emptyset$; $\tilde{U}(s) = \{u_1(s), \dots, u_p(s), \dots, u_p(s)\}$ — выбранная для данной s -й ветви дерева решений задачи последовательность двухстадийной обработки изделий.

В начале вычислительного процесса, когда $s = 0$, положим $\tilde{I}^1(s) = \tilde{U}(s) = \emptyset$, $\tilde{I}^2(s) = \tilde{I} = \{i = 1, \dots, n\}$, а также обозначим $\xi(\bar{T}^1) = \xi[\bar{T}^1(s)]$, $\xi(\bar{T}^2) = \xi[\bar{T}^2(s)]$ и $\xi[\bar{T}(s)] = \max\{\xi[\bar{T}^1(s)], \xi[\bar{T}^2(s)]\}$.

Определим нижнюю границу длины расписания в начале работы алгоритма решения задачи. Длина двухстадийного расписания \bar{T} не может быть меньше нижней границы длины расписания на каждой из стадий выполнения работ, вычисленной по формуле (3), т. е. $\bar{T} \geq \max\{\xi(\bar{T}^1), \xi(\bar{T}^2)\}$. Выполнение работ на второй стадии не может начаться ранее завершения изготовления хотя бы одного изделия на первой стадии обработки, т. е. не

ранее времени $\sigma^1 = \min_{1 \leq i \leq n} \sum_{k=1}^{m_1} t_{ik}^1$. Только после завершения обработки одного изделия на первой стадии обработки необходимо начать и завершить выполнение всех работ на второй стадии обработки, что потребует затрат времени не менее, чем

$$\sigma^2 = \min_{1 \leq i \leq n} \sum_{l=1}^{m_2} t_{il}^2. \text{ Следовательно, нижняя граница}$$



длины двухстадийного расписания \bar{T} определяется выражением

$$\xi(\bar{T}) = \max \left\{ \min_{1 \leq i \leq n} \sum_{k=1}^{m_1} t_{ik}^1 + \xi(\bar{T}^2); \right. \\ \left. \xi(\bar{T}^1) + \min_{1 \leq i \leq n} \sum_{l=1}^{m_2} t_{il}^2 \right\}. \quad (4)$$

Пусть на некотором шаге вычислительного процесса в s -й ветви дерева решений определены подмножества изделий $\tilde{I}^1(s)$, $\tilde{I}^2(s)$, а также последовательность двухстадийной обработки подмножества изделий $\tilde{I}^1(s)$: $\tilde{U}(s) = \{u_{1,s}, u_{p,s}, \dots, u_{p,s}\}$. Здесь $u_{p,s}$ — задание, стоящее на p -м месте в последовательности, построенной в s -й вершине дерева решений задачи. Времена начала и завершения обработки всех изделий на каждой из машин первой и второй стадий обработки определяются по формулам:

$$x_{u_{1,1}}^1 = 1, \quad x_{u_{1,k}}^1(s) = \theta_{u_{1,k-1}}^1(s) + 1, \\ \theta_{u_{1,k}}^1(s) = \theta_{u_{1,k-1}}^1(s) + t_{u_{1,k}}^1, \quad (5)$$

$$x_{u_{1,1}}^2(s) = \bar{T}_{u_{1,1}}^1(s) + 1, \\ \theta_{u_{1,1}}^2(s) = \bar{T}_{u_{1,1}}^1(s) + t_{u_{1,1}}^2, \quad (6)$$

$$x_{u_{p-1}}^2(s) = \max \left[\bar{T}_{u_{p-1}}^1(s), \theta_{u_{p-1,1}}^2(s) \right] + 1, \\ \theta_{u_{p-1}}^2(s) = x_{u_{p-1}}^2(s) + t_{u_{p-1}}^2 - 1; \quad (7)$$

$$x_{u_{p,k}}^\gamma(s) = \max \left[\theta_{u_{p,k-1}}^\gamma(s), \theta_{u_{p-1,k}}^\gamma(s) \right] + 1, \\ \bar{T}_{u_p}^\gamma(s) = \theta_{u_{p,k}}^\gamma(s) = x_{u_{p,k}}^\gamma(s) + t_{u_{p,k}}^\gamma(s) - 1, \\ \gamma = 1, 2, \quad p = 2, \dots, P; \quad k = 2, \dots, m_1, \text{ если } \gamma = 1; \\ k = 2, \dots, m_2, \text{ если } \gamma = 2. \quad (8)$$

Обозначим $X_k^1(s) = \theta_{u_{p,k}}^1(s) + 1, k = 1, \dots, m_1;$
 $X_k^2(s) = \theta_{u_{p,k}}^2(s) + 1, k = 1, \dots, m_2,$ — соответственно времена начала работы машин изготовления подмножества изделий $\tilde{I}^2(s)$ на первой и второй стадиях обработки в s -й ветви дерева. Пусть последовательность $\tilde{U}(s)$ содержит P изделий. Минимальные времена завершения обработки подмножества изделий $\tilde{I}^2(s) = \{j = i_1^2, i_2^2, \dots, i_g^2, \dots, i_G^2\}$ на первой

и второй стадиях обработки ($\gamma = 1, 2$) на всех машинах не может быть меньше величин

$$\xi[T^1(s)] = \max \left\{ \max_{i \in \tilde{I}^2(s)} \left[X_i^1(s) + \sum_{k=1}^{m_1} t_{ik}^1 \right]; \right. \\ \left. \max_{1 \leq r \leq m_1} \left[X_r^1(s) + \sum_{t \in \tilde{I}^1(s)} t_{ir}^1 + \min_{1 \leq r \leq m_1} \sum_{k=r+1}^{m_1} t_{ik}^1 \right] \right\}, \quad (9)$$

$$\xi[T^2(s)] = \max_{1 \leq r \leq m_2} \left[X_r^2(s) + \sum_{t \in \tilde{I}^1(s)} t_{ir}^2 + \right. \\ \left. + \min_{1 \leq r \leq m_1} \sum_{k=r+1}^{m_2} t_{ik}^1 \right], \quad (10)$$

$$\xi[\bar{T}(s)] = \max \left\{ \xi[T^2(s)]; \xi[T^1(s) + \min_{i \in \tilde{I}^2} \sum_{k=1}^{m_2} t_{ik}^2] \right\}. \quad (11)$$

Оценки (9)—(11) могут быть вычислены для двух вариантов: когда времена постобработки b_i рассматриваются как времена обработки на последней машине первой стадии изготовления; когда они рассматриваются как времена обработки на первой машине второй стадии изготовления. Обозначив оценки этих вариантов соответственно $\xi_1[\bar{T}(s)]$ и $\xi_2[\bar{T}(s)]$, оценка нижней границы решения может быть определена как минимальная из этих оценок, т. е. $\xi[\bar{T}(s)] = \max\{\xi_1[\bar{T}(s)], \xi_2[\bar{T}(s)]\}$.

Кроме того, процесс двухстадийной обработки может рассматриваться как процесс обработки на одной стадии последовательности, состоящей из $M = (m_1 + 1 + m_2)$ машин. Здесь времена постобработки включены в качестве времен обработки на машине, стоящей между первой и второй стадиями изготовления, и для вычисления нижней границы можно воспользоваться формулами (1)—(3) для случая M машин. Оценку, вычисленную в этих условиях, обозначим $\xi_3[\bar{T}(s)]$. Тогда нижняя граница длины двухстадийного расписания определяется выражением

$$\xi[\bar{T}(s)] = \max\{\xi_3[\bar{T}(s)]; \max\{\xi_1[\bar{T}(s)], \xi_2[\bar{T}(s)]\}\}.$$

Рассмотрим два альтернативных варианта построенных подпоследовательностей в процессе решения задачи:

$$\tilde{I}^1(s), \tilde{U}(s), \tilde{I}^2(s) \text{ с параметрами } X_k^1(s), X_k^2(s), \\ i \in \tilde{I}^2(s), \text{ и } \xi[T^1(s)], \xi[T^2(s)], \xi[\bar{T}(s)];$$

$$\tilde{I}^1(p), \tilde{U}(p), \tilde{I}^2(p) \text{ с параметрами } X_k^1(p), X_k^2(p), \\ i \in \tilde{I}^2(p), \text{ и } \xi[T^1(p)], \xi[T^2(p)].$$

Правило отсева 1. Если для двух вариантов s и p выполняются условия

$$\tilde{I}^1(s) \subseteq \tilde{I}^1(p), \quad X_k^1(p) \leq X_k^1(s), \quad k = 1, \dots, m_1;$$

$$X_k^2(p) \leq X_k^2(s), \quad k = 1, \dots, m_2,$$

то p -й вариант более предпочтителен, чем вариант s , и вариант s может быть отброшен как неперспективный.

Правило отсева 2 (нестрогое предпочтение). Если для двух вариантов p и s справедливы условия

$$\tilde{I}^1(s) \subseteq \tilde{I}^1(p), \quad \xi[T^1(p)] \leq \xi[T^1(s)]$$

и $X_k^2(p) \leq X_k^2(s), \quad k = 1, \dots, m_2,$

то p -й вариант более предпочтителен, и вариант s может быть отброшен как неперспективный.

Правило отсева 3 (правило нестрогого предпочтения). Если для двух вариантов (ветвей дерева) p и s справедливы условия

$$\tilde{I}^1(s) \subseteq \tilde{I}^1(p), \text{ и } X_{i,m_1}^1(p) \leq X_{i,m_1}^1(s),$$

$$X_{i,m_2}^2(p) \leq X_{i,m_2}^2(s),$$

то вариант p более предпочтителен, чем вариант s , и вариант s может быть отброшен как неперспективный.

Правило предпочтения 1 может применяться в алгоритмах точного решения задачи, а правила 2 и 3 — для получения приближенных решений.

3. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ

В начале работы каждого из алгоритмов $s = 0$, $\tilde{I}^1(s) = \tilde{U}(s) = \emptyset$, $\tilde{I}^2(s) = \tilde{I}$. Для каждого из вариантов (вершины дерева решений) введем некоторую метку $w(s)$: $w(s) = 0$, если вариант перспективный и не продолжен во всех возможных направлениях; $w(s) = 1$ — вариант продолжен во всех возможных направлениях; $w(s) = 2$ — вариант отброшен как неперспективный.

Рассмотрим алгоритм решения задачи, основанный на методе ветвей и границ.

Алгоритм 1. Вычислим граничное значение минимального времени завершения выполнения двухстадийного расписания по формуле (4) и определим границу $\xi(\bar{T}) = \xi[\bar{T}(s = 0)]$. Пусть при рассмотрении некоторой s -й вершины дерева уже получено несколько решений задачи $F_l(s), l = 1, \dots, L(s)$, лучшее из которых обозначим $\bar{F}(s) = \min_{1 \leq l \leq L} F(l)$ и назовем рекордом. Последовательность выполнения заданий, соответствующую этому расписа-

нию, обозначим $\tilde{\Pi}^s$. Полагаем для $s = 0$, $\tilde{I}^1(s) = \tilde{U}(s) = \emptyset$, $\tilde{I}^2(s) = \tilde{I}$.

Шаг 1. Устанавливаем на первое место в последовательности обработки изделий каждое из этих изделий, образуя ветви дерева $s = 1, 2, \dots, n$. Для каждого из вариантов рассчитываем времена начала и завершения выполнения операций на всех машинах двухстадийного расписания по формулам (5)—(7). Определяем подмножества индексов изделий $\tilde{I}^1(s) = \tilde{U}(s) = \{s\}$, $\tilde{I}^2(s) = (\tilde{I}/s)$, $s = 1, 2, \dots, n$. Вычисляем значения нижней границы длины двухстадийного расписания для каждого из вариантов по формулам (9)—(11). Полагаем для каждого из этих вариантов $w(s) = 0$. Число ветвей дерева $S = n$. Полагаем $w(s = 0) = 1$. Переходим к выполнению шага $s - a$.

Шаг $s - a$. Среди всех ветвей дерева, для которых $w(s) = 0$, находим ветвь с наименьшим значением нижней границы $\xi[\bar{T}^*(s)] = \min_{1 \leq g \leq S} \{\xi[\bar{T}(g)] | w(g) = 0\}$.

Если таких вершин несколько, то выбираем среди них вершину, у которой подмножество $\tilde{I}^1(s)$ содержит максимальное число элементов. Для выбранной s -й вершины дерева определены подмножество $\tilde{I}^1(s)$ и последовательность $\tilde{U}(s) = \{u_{1,s}, \dots, u_{p,s}, \dots, u_{p,s}\}$. Полагаем для выбранной вершины дерева $w(s) = 1$. Переходим к шагу $s - b$.

Шаг $s - b$. Вычислим (или известны) времена начала и завершения обработки последнего стоящего в последовательности $\tilde{U}(s)$ изделия $u_{p,s}$ на всех машинах первой и второй стадий обработки. Устанавливаем на последнее, т. е. $(u_{p,s} + 1)$ -е, место в последовательности обработки изделий каждое из изделий подмножества $\tilde{I}^2(s)$, число которых равно $R(s) = n - P(s)$, образуя ветви дерева с индексами $q = S + 1, S + 2, \dots, S + R(s)$. Для каждого из вновь образованных вариантов рассчитываем времена начала и завершения выполнения операций на всех машинах двухстадийного расписания по формулам (5)—(8). Определяем подмножества индексов изделий $\tilde{I}^1(q) = \{\tilde{I}^1(S) \cup q\}$, $\tilde{I}^2(q) = \{\tilde{I}^2(S)/q\}$, $\tilde{U}(q) = \{u_1(s), \dots, u_{R(s)}(s), q\}$. Вычисляем по формулам (9)—(11) значения нижней границы длины двухстадийного расписания каждого из вариантов $\xi[\bar{T}(s)], s = S + 1, S + 2, \dots, S + R(s)$. Переходим к шагу $s - c$.

Шаг $s - c$. Если для какой-то из вновь образованной вершины $\xi[\bar{T}(s)] \geq \bar{F}(s)$, то полагаем для этой вершины $w(s) = 2$. Если для некоторой из вновь образованных вершин $\tilde{I}^2(s) = \emptyset$, то полу-



чено одно из решений задачи и $\xi[\bar{T}(\lambda)] = F(\lambda)$. Если $F(\lambda) < \bar{F}(s)$, то полагаем $\bar{F}(s) = F(\lambda)$. Если при этом $\bar{F}(s) \leq \min_{1 \leq s \leq S} \{\xi[\bar{T}(s)|w(s) = 0]\}$, то получено решение задачи, и алгоритм завершает свою работу. В противном случае, если для какой-то из этих вершин дерева выполняется правило отсева 1 (или выполняются правила отсева 2 или 3 для получения приближенных решений в условиях большой размерности), то полагаем для этого варианта $w(s) = 2$. Для всех остальных вновь образованных вершин дерева полагаем $w(s) = 0$. Переходим к шагу $s - a$.

Ясно, что через некоторое число шагов будет получено оптимальное решение.

Приближенное решение задачи на основе метода ветвей и границ может быть получено при реализации односторонней схемы ветвления до получения первого решения «рекорда», не обращая внимание на то, что в дереве решений есть вершины, для которых $w(s) = 0$ и $\bar{F}(s) > \bar{T}(s)$. При этом первое полученное двухстадийное расписание, для которого $\tilde{I}^1(s) = \tilde{I}$, принимается в качестве решения задачи.

Алгоритм 2 (решение задачи методом динамического программирования). Пусть S — суммарное число построенных вариантов (вершин дерева решений). В начале работы алгоритма $S = 0$. Наилучшее из построенных решений на предыдущих шагах алгоритма обозначим $\bar{F}(S)$. На каждом s -м шаге работы алгоритма находим вариант, у которого $w(s) = 0$, определены подмножества $\tilde{I}^1(s)$, $\tilde{I}^2(s)$ и последовательность $\tilde{U}(s) = \{u_1(s), u_2(s), \dots, u_R(s)\}$, а также вычисленные значения $X_k^1(s)$ и $X_k^2(s)$, вариант с наименьшим значением длины выполнения расписания для уже построенной подпоследовательности обработки $\tilde{U}(s)$, т. е. с наименьшим значением $\bar{T}[\tilde{U}(s)] = \min_{1 \leq s \leq S} \bar{T}[\tilde{U}(s)]$. Если $\tilde{I}^1(s) = \tilde{I}$ и $\tilde{I}^2(s) = \emptyset$, то переходим к п. 3. В противном случае переходим к п. 1. В дальнейшем для простоты изложения полагаем $s := \bar{s}$ и выполняем вычисления п. 1.

1. На последнее место в подпоследовательности $\tilde{U}(s)$ устанавливаем каждое из изделий подмножества $\tilde{I}^2(s) = \{i_1^2(s), \dots, i_g^2(s), \dots, i_G^2(s)\}$ и определяем новые подмножества вариантов

$$\begin{aligned}\tilde{I}^1(s+g) &= \{\tilde{I}^1(s) \cup i_g^2(s)\}, \\ \tilde{I}^2(s+g) &= \{\tilde{I}^2(s)/i_g^2(s)\},\end{aligned}$$

$$\begin{aligned}\tilde{U}(s+g) &= \{u_1(s), u_2(s), \dots, u_R(s), i_g^2(s)\}, \\ g &= 1, \dots, G.\end{aligned}$$

Вычисляем по ниже приведенным формулам времена начала и завершения задания, стоящего последним в каждой из этих подпоследовательностей, на всех машинах:

$$\begin{aligned}x(i_g^2(s), k) &= \max\{\theta(u_{R_p}^s, k), \theta(i_g^2(s), k-1)\} + 1, \\ \theta(i_g^2(s), k) &= x(i_g^2(s), k) + t(i_g^2(s), k) - 1, \\ g &= 1, \dots, G.\end{aligned}$$

Полагаем $w(s) = 1$, $w(s+g) = 0$. Образует новые варианты $p = S + g$, для которых

$$\begin{aligned}\tilde{I}^1(S+g) &= \tilde{I}^1(s+g) \cup i_g^2(s), \\ \tilde{I}^2(S+g) &= \{\tilde{I}^2(S+g)/i_g^2(s)\},\end{aligned}$$

$$\begin{aligned}\tilde{U}(S+g) &= \{u_1(s), u_2(s), \dots, u_R(s), u_{R+1}(s) = i_g^2(s)\}, \\ g &= 1, \dots, G.\end{aligned}$$

Вычисляем для каждого из вновь образованных вариантов времена начала и завершения обработки изделия с индексом $i_g^2(s) = u_{R+1}(s)$, стоящего последним в последовательности $\tilde{U}(S+g)$ на всех машинах первой и второй стадий обработки, а также значения $X_k^1(S+g)$ и $X_k^2(S+g)$. Переходим к п. 2.

2. Последовательно для каждого из вариантов $s = (S+g)$, $g = 1, \dots, G$, выполняем следующие действия. Если построена последовательность, для которой $\tilde{I}^1(S+g) = \tilde{I}$, то получено решение задачи, которое обозначим $F[\tilde{U}(S+g)]$. Если $F[(S+g)] \leq \{\bar{T}[\tilde{U}(s)|w(s) = 0]\}$, то получено решение задачи и переходим к п. 3. В противном случае, если $F[(S+g)] \leq \bar{F}(S)$, то полагаем $\bar{F}(S) = F[(S+g)]$. Для всех вариантов, для которых $w(s) = 0$ или $w(s) = 1$ и выполняются условия $\bar{T}[\tilde{U}(s)] \geq \bar{F}(S)$ либо выполняются правила отсева 1 (либо в случае получения приближенных решений правила отсева 2 или 3), полагаем $w(s) = 2$. Вычисляем новое значение числа образованных вариантов $S := (S+g)$. Переходим к п. 1.

3. Для полученной оптимальной последовательности обработки изделий $\tilde{U}^*(s) = \arg \bar{F}(S)$ вычисляем времена начала и завершения обработки изделий на всех машинах каждой стадии их изготовления, а также времена начала и завершения постобработки между этими стадиями. Здесь

$\arg \bar{F}(S)$ — последовательность обработки изделий в оптимальном решении. На этом алгоритм решения задачи завершает свою работу.

4. ЭВРИСТИЧЕСКИЙ АЛГОРИТМ ПРИБЛИЖЕННОГО РЕШЕНИЯ ЗАДАЧИ

Для решения задачи на двух машинах наиболее эффективен полиномиальный по времени алгоритм Джонсона [9, 10]. Все изделия разбиваются на две группы. Первая из них включает в себя изделия, для которых $\bar{t}_i^1 \leq \bar{t}_i^2$, а вторая — для которых $\bar{t}_i^1 > \bar{t}_i^2$. Изделия первой группы упорядочиваются по неубыванию времен \bar{t}_i^1 в последовательности $\tilde{V}^1 = \{v_1^1, v_2^1, \dots, v_{l-1}^1, v_l^1, \dots, v_L^1 | \bar{t}_{l-1}^1 \leq \bar{t}_l^1, l = 2, \dots, L\}$, а изделия второй группы — по невозрастанию времен \bar{t}_i^2 в последовательности $\tilde{V}^2 = \{v_1^2, v_2^2, \dots, v_{r-1}^2, v_r^2, \dots, v_R^2 | \bar{t}_{r-1}^2 \geq \bar{t}_r^2, r = 2, \dots, R\}$. Обработка изделий на двух машинах осуществляется в последовательности

$$\tilde{V} = \{v_1^1, v_2^1, \dots, v_{l-1}^1, v_l^1, \dots, v_L^1, v_1^2, v_2^2, \dots, v_{r-1}^2, v_r^2, \dots, v_R^2\}.$$

Доказано [10], что в случае двух машин данный алгоритм обеспечивает оптимальное решение задачи и имеет полиномиальную сложность порядка $O(n^2)$. Данная эвристика может быть применена для получения приближенного решения сформулированной в данной работе задачи. Рассмотрим два варианта расчета значений \bar{t}_i^1 и \bar{t}_i^2 :

$$\begin{aligned} \bar{t}_i^1 &= \sum_{k=1}^{m_1} \bar{t}_{ik}^1, & \bar{t}_i^2 &= \sum_{k=1}^{m_2} \bar{t}_{ik}^2 + b_i; \\ \bar{t}_i^1 &= \sum_{k=1}^{m_1} \bar{t}_{ik}^1 + b_i, & \bar{t}_i^2 &= \sum_{k=1}^{m_2} \bar{t}_{ik}^2. \end{aligned} \quad (12)$$

Для каждого из этих вариантов решаем задачу построения расписания для двух машин описанным выше алгоритмом Джонсона. При этом значения времен выполнения заданий рассчитываются по формулам (5)—(8). Пусть для построенных последовательностей значения критериев оптимальности соответственно равны \bar{T}^1 и \bar{T}^2 . В качестве решения задачи выбираем вариант и соответствующую последовательность двухстадийной обработки на всех машинах с наименьшим значением

критерия оптимальности, т. е. $\bar{T} = \min(\bar{T}^1, \bar{T}^2)$. Описанный алгоритм, как и алгоритм Джонсона, имеет полиномиальную сложность и требует только в два раза большего объема вычислений.

Для получения более точных приближенных решений могут быть рассмотрены и другие варианты вычисления суммарного времени выполнения операций на каждой из двух машин:

$$\begin{aligned} \bar{t}_i^1 &= \sum_{k=1}^{\delta_1} \bar{t}_{ik}^1, & \bar{t}_i^2 &= \sum_{k=\delta_1+1}^{m_1} \bar{t}_{ik}^1 + \sum_{k=1}^{m_2} \bar{t}_{ik}^2 + b_i; \\ \bar{t}_i^1 &= \sum_{k=1}^{m_1} \bar{t}_{ik}^1 + b_i + \sum_{k=1}^{\delta_2} \bar{t}_{ik}^1, & \bar{t}_i^2 &= \sum_{k=\delta_2+1}^{m_2} \bar{t}_{ik}^2, \end{aligned} \quad (13)$$

где $\delta_1 = 1, 2, \dots, (m_1 - 1)$, $\delta_2 = 1, 2, \dots, (m_2 - 1)$.

Для каждого из этих вариантов строится последовательность выполнения заданий алгоритмом Джонсона. В каждой из построенных последовательностей все времена начала и завершения заданий вычисляем по формулам (5)—(8). Среди всех рассмотренных вариантов, включая варианты, образованные по правилам (12), (13), выбирается последовательность с наименьшим значением критерия оптимальности.

Отметим, что если критерий оптимальности существенно отличается от значения нижней границы оптимального решения, то полученное решение можно попытаться улучшить алгоритмом локальных вариаций, последовательно меняя два члена, стоящие рядом в последовательности.

5. ИЛЛЮСТРАТИВНЫЙ ПРИМЕР

Времена обработки пяти изделий на каждой из машин первой и второй стадий обработки представлены в табл. 1. Первая и вторая стадии предусматривают выполнение трех операций на трех машинах. Времена постобработки изделий $\bar{b} = (5, 6, 8, 10, 9)$. Обработка каждого из изделий должна вестись на машинах, расположенных в заданной последовательности. Не допускается прерываний в процессе обработки на каждой из машин. Начало обработки изделий на первой машине второй стадии может начаться только после завершения обработки его на третьей машине первой стадии и последующей постобработки. Необходимо найти последовательность обработки изделий, обеспечивающую минимальное время выполнения расписания, т. е. всех работ на двух стадиях.

Времена работы машин для случаев, когда времена постобработки \bar{b} рассматриваются как времена работы машин первой или второй стадий изготовления, приведены соответственно в табл. 1 и 2.



Вычислим нижнюю границу минимального времени выполнения расписания. По формулам (9)–(11) вычисляем

$$\xi_1[T_1(s)] = \max(25 + 13, 3 + 38 + 8, 10 + 45) = 55,$$

$$\xi_1[T_2(s)] = \max(38 + 17, 5 + 33 + 10, 10 + 34 + 3, 18 + 30) = 55,$$

$$\xi_1[\bar{T}(s)] = \max(55 + 22, 55 + 1) = 77;$$

$$\xi_2[T_1(s)] = \max(25 + 19, 4 + 38 + 13, 10 + 45 + 5, 19 + 38) = 60,$$

$$\xi_2[T_2(s)] = \max(33 + 10, 2 + 34 + 3, 11 + 30) = 43,$$

$$\xi_2[\bar{T}(s)] = \max(60 + 17, 43 + 19) = 77;$$

$$\xi_3[\bar{T}(s)] = \max(25 + 13 + 22, 3 + 38 + 8 + 22, 10 + 45 + 22, 19 + 38 + 19, 19 + 33 + 10, 5 + 19 + 34 + 3, 19 + 17 + 30) = 77;$$

$$\xi[\bar{T}(s)] = \max(77; \max[77, 77]) = 77.$$

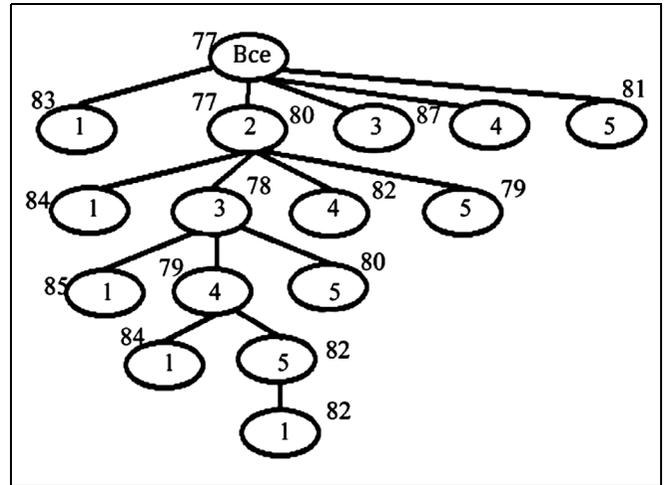


Рис. 1. Дерево приближенного решения задачи

Таблица 1

Времена выполнения работ на машинах первой и второй стадий (первый вариант)

Номер задания	Времена выполнения заданий на машинах первой и второй стадий (t_{ik}^1 и t_{ik}^2)								
	Машины первой стадии				Машины второй стадии				
	1	2	3	$\sum_{k=1}^3 t_{ik}^1$	1	2	3	4	$\sum_{k=1}^4 t_{ik}^2$
1	5	7	8	20	5	7	6	4	22
2	6	4	9	19	6	10	7	3	26
3	3	10	8	21	8	2	9	10	29
4	7	7	11	25	10	8	5	6	29
5	4	10	9	23	9	6	7	7	29
$\sum_{k=1}^5 t_{ik}^r$	25	38	45	—	38	33	34	30	—

Таблица 2

Времена выполнения работ на машинах первой и второй стадий (второй вариант)

Номер задания	Времена выполнения заданий на машинах первой и второй стадий (t_{ik}^1 и t_{ik}^2)								
	Машины первой стадии					Машины второй стадии			
	1	2	3	4	$\sum_{k=1}^3 t_{ik}^1$	1	2	3	$\sum_{k=1}^4 t_{ik}^2$
1	5	7	8	5	25	7	6	4	17
2	6	4	9	6	25	10	7	3	20
3	3	10	8	8	29	2	9	10	21
4	7	7	11	10	35	8	5	6	19
5	4	10	9	9	32	6	7	7	20
$\sum_{k=1}^5 t_{ik}^r$	25	38	45	38	—	33	34	30	—

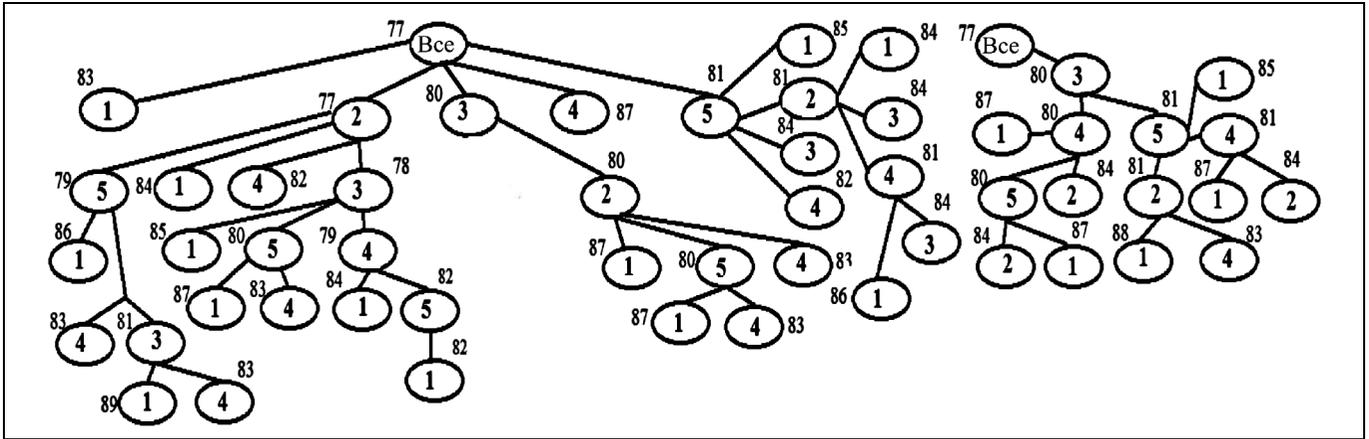


Рис. 2. Дерево точного решения задачи

Приближенное решение задачи получено с помощью алгоритма ветвей и границ в условиях, когда развивается только одна ветвь дерева решений с выбором вершины с наилучшим значением нижней границы (алгоритм одностороннего ветвления с целью получения «рекорда»). Дерево решений задачи приведено на рис. 1. Цифры, расположенные рядом с вершинами дерева, — значения нижней границы данной ветви. Полученное оптимальное значение последовательности обработки изделий $\tilde{U} = \{2, 3, 4, 5, 1\}$. Время выполнения расписа-

ния $T(\tilde{U}) = 82$. Времена завершения выполнения всех заданий на каждой из машин первой и второй стадий обработки приведены в табл. 2.

В табл. 3 приведены времена завершения выполнения заданий на машинах первой и второй стадий в построенном расписании.

Отметим, что полученное приближенное решение задачи методом ветвей и границ является оптимальным и совпадает с точным решением задачи, полученным методом ветвей и границ (рис. 2). Большой объем вы-

Таблица 3

Времена завершения выполнения заданий в оптимальном решении

Номер задания	Времена завершения выполнения заданий на машинах первой и второй стадий (θ_{ik}^1 и θ_{ik}^2)						
	Машины первой стадии			Машины второй стадии			
	1	2	3	1	2	3	4
2	6	10	19	25	35	42	45
3	9	20	28	36	38	47	57
4	16	27	39	49	57	62	68
5	20	37	48	58	64	71	78
1	25	44	56	63	70	78	82

Таблица 4

Времена завершения выполнения заданий на машинах (первое решение)

Номер задания	Времена завершения выполнения заданий на машинах первой и второй стадий (θ_{ik}^1 и θ_{ik}^2)						
	Машины первой стадии				Машины второй стадии		
	1	2	3	4	1	2	3
3	3	13	21	25	31	40	50
2	9	11	30	36	46	53	56
5	13	27	39	48	52	60	68
4	20	34	50	60	68	73	79
1	25	41	58	65	72	78	83

Времена завершения выполнения заданий на машинах (второе решение)

Номер задания	Времена завершения выполнения заданий на машинах первой и второй стадий (θ_{ik}^1 и θ_{ik}^2)						
	Машины первой стадии			Машины второй стадии			
	1	2	3	1	2	3	4
1	5	20	21	25	32	38	42
2	11	16	29	35	45	52	55
3	14	26	37	45	47	61	71
5	18	36	46	55	61	68	75
4	25	43	57	67	75	80	86

числений был необходим только для доказательства оптимальности полученного приближенного решения задачи. Такое большое число вершин дерева объясняется близкими значениями времен обработки изделий на машинах в рассматриваемом примере.

Рассмотрим приближенное решение задачи. Если времена постобработки включены в первую стадию обработки изделий, то $\bar{t}^1 = \{25, 25, 29, 35, 32\}$ и $\bar{t}^2 = \{17, 20, 19, 23, 22\}$. Так как $\bar{t}_i^1 > \bar{t}_i^2$, $i = 1, \dots, 5$, то последовательность по неубыванию времен \bar{t}_i^1 имеет вид $\bar{U}^1 = \{2, 3, 1, 5, 4\}$. Расписание выполнения работ обработки изделий в данной последовательности представлено в табл. 4; $\bar{T}(\bar{U}^1) = 83$.

Если времена постобработки включены во вторую стадию обработки изделий, то $\bar{t}^1 = \{20, 19, 21, 25, 23\}$ и $\bar{t}^2 = \{22, 26, 29, 29, 29\}$. Так как $\bar{t}_i^1 < \bar{t}_i^2$, $i = 1, \dots, 5$, то последовательность по невозрастанию времен \bar{t}_i^1 имеет вид $\bar{U}^2 = \{1, 2, 3, 5, 4\}$, и расписание выполнения работ обработки изделий в данной последовательности представлено в табл. 5; $\bar{T}(\bar{U}^2) = 86$. Так как $\bar{T}(\bar{U}^1) < \bar{T}(\bar{U}^2)$, то $\bar{T}(\bar{U}^1) = 83$ принимается в качестве приближенного решения задачи.

ЗАКЛЮЧЕНИЕ

Построение расписаний двухстадийной обработки множества изделий на двух последовательно расположенных в цепочку системах машин имеет много практических приложений в календарном планировании. Необходимо найти последовательность обработки изделий, времена начала и завершения обработки каждого изделия на всех машинах двух стадий изготовления, обеспечивающие выполнение всех работ на обеих стадиях обработки в кратчайшие сроки. Даже в условиях, когда все

изделия как на первой стадии, так и на второй стадии изготовления обрабатываются в одной и той же и одинаковой для всех изделий последовательности, рассматриваемая задача относится к классу *NP*-полных задач и является обобщением известной в теории расписаний *Flow-Shop-Problem*. Рассмотренные постановки задачи и предложенные в работе точные методы, а также приближенные методы и эвристические алгоритмы полиномиальной сложности решений этих задач могут найти широкое применение в системах календарного планирования работы участков и цехов машиностроительного и приборостроительного производства, а также в процессах деревообработки, электронной и легкой промышленности. Оценка эффективности полученных приближенных решений может быть выполнена в сравнении с вычисленной нижней границей оптимальной длительности выполнения расписания.

Решение рассмотренных проблем в условиях ограничений на времена завершения выполнения заданий и допустимые сроки работы машин, которое может быть предметом дальнейших исследований, позволит также более обоснованно оценить возможности выполнения заказов и планировать время технического обслуживания машин.

ЛИТЕРАТУРА

1. Конвей Р.В., Максвелл В.Л., Миллер Л.В. Теория расписаний. — М.: Наука, 1975. — 359 с. [Konvey, R.W., Maxwell, W.L., Miller, L.W. Teorija raspisanij. — М.: Nauka, 1975. — 359 s. (In Russian)]
2. Танаев В.С., Сотсков Ю.Н., Струевич В.А. Теория расписаний. Многостадийные системы. — М.: URSS, 1989. — 328 с. [Tanajev, V.S., Sotskov, Yu.N., Strusevitsh, V.A. Teorija raspisanij. Mnogostadijnye sistemi. — М.: URSS, 1989. — 328 s. (In Russian)]
3. Зак Ю.А. Прикладные задачи теории расписаний и маршрутизации перевозок. — М.: URSS, 2012. — 394 с. [Zack, Yu.A. Prikladnija zadachi teorii raspisanij i marshrutizazii perevosok. — М.: URSS, 2012. — 394 s. (In Russian)]

4. *Zack Yu.A.* Решение обобщенной задачи Джонсона с ограничениями на сроки выполнения заданий и времена работы машин. Ч. 1. Точные методы решения. // Проблемы управления. — 2010. — № 3. — С. 17–25. Ч. 2. Приближенные методы // Проблемы управления. — 2010. — № 4. — С. 12–19. [*Zack, Yu.A.* Reshenije obobchennoj zadachi Dczonsona s ogranichenijam na sroki vipolnenija zadaniy i vremena raboti mashin. Ch. 1. Tochnije metodi reshenija. // Control Sciences. — 2010. — No. 3. — S. 17–25. Ch. 2; Pbliczennije metodi. — 2010. — No. 4. — S. 12–19. (In Russian)]
 5. *Brucker, P.* Scheduling Algorithms. — Berlin, Heidelberg und New-York: Springer-Verlag, 1998. — 312 p.
 6. *Domschke, W., Scholl, A., Voß, S.* Produktionsplanung. Ablauforganisatorische Aspekte. — Berlin, Heidelberg: Springer Verlag, 2005. — 456 p.
 7. *Ho, J.C., Chang, Y.-L.* A New Heuristic for the n -job, M -machine Problem // European Journal of Operational Research. — 1991. — Vol. 52, iss. 2. — P. 194–202.
 8. *Ogbu, F.A., Smith, D.K.* The Application of the Simulated Annealing Algorithm to the Solution of the $n/m/C_{\max}$ Flow-shop Problem // Computer & Operations Research. — 1990. — Vol. 17. — P. 243–253.
 9. *Ogbu, F.A., Smith, D.K.* Simulated Annealing for the Permutation Flow-shop Problem // OMEGA. — 1991. — P. 64–67.
 10. *Hundal, T.S., Rajgopal, J.* An Extension of Palmer's Heuristic for the Flow-shop Scheduling Problem // International Journal of Production Research. — 1988. — Vol. 26. — P. 1119–1124.
 11. *Garey, M.R., Johnson, D.S., & Sethi, R.* The Complexity of Flowshop and Jobshop Scheduling // Mathematics of Operations Research. — 1976. — Vol. 1, iss. 2. — P. 117–129.
 12. *Лушакова И.Н., Струевич В.А.* Двухстадийные системы с нефиксированными маршрутами и ресурсными ограничениями // Журнал вычислительной математики и математической физики. — 1989. — Т. 29, № 9. — С. 1393–1407. [*Lushakova, I.L., Strusevitch, V.A.* Dwuchstadijnije sistemi s nefiksirovannimi marshrutami i resursnimi ogranichenijami // Zhurnal vychislitelnoj matematiki i matematicheskoj fiziki. — 1989. — Vol. 29, No. 9. — S. 1393–1407. (In Russian)]
 13. *Zack Yu.A.* Построение двухстадийных расписаний обработки изделий на одной машине // Системні дослідження та інформаційні технології. — 2018. — № 4. — С. 19–36. [*Zack, Yu.A.* Postrojenije dvuchstadijnich raspisanij obrabotki izdelij na odnoj mashine // Sistemi doslidzhennya ta informazijnijni tehnologiji. — 2018. — No. 4. — P. 19–36. (In Russian)]
 14. *Johnson, S.M.* Optimal Two-and Three-stage Production Schedules with Setup Times Included // Naval Research Logistics Quarterly. — 1954. — Vol. 1, iss. 1. — P. 61–68.
 15. *Henry, H.Y.K., Yu, L.Ch.* A Two Stage Heuristic Algorithm for the Integrated Aircraft and Crew Schedule Recovery Problems // Computer & Industrial Engineering. — 2015. — Vol. 87. — P. 436–453.
 16. *Guerreschim, G.G., Park, Jo.* Two-step Approach to Scheduling Quantum Circuits // Quantum Science and Technology. — 2018. — Vol. 3, No. 4. — P. 212–219.
 17. *Yin, Ai-H., Zhao, Xi-P.* A Two-step Approach for Solving the Flexible Job Shop Scheduling Problem // The 2009 IEEE Int. Conf. on Granular Computing, GrC, Lushan Mountain, Nanchang, China, 17–19 August, 2009.
 18. *Seenner, F., Mey, H.* Multi-stage Simultaneous Lot-sizing and Scheduling for Flow Line Production // OR Spectrum. — 2013. — Vol. 35, No. 1. — P. 33–73.
- Статья представлена к публикации членом редколлегии А.А. Лазаревым.*
- Поступила в редакцию 15.02.2019, после доработки 31.07.2019.
Принята к публикации 31.07.2019.*
- Зак Юрий Александрович** — д-р техн. наук, науч. эксперт и консультант, г. Аахен, Германия, ✉ yuriy_zack@hotmail.com.

TWO-STAGE PLANNING TASKS FOR THE FLOW LINE

Yu.A. Zack

✉ yuriy_zack@hotmail.com

Abstract. The problem is formulated of constructing schedules of the two-stage processing of large number of products on two machine systems arranged into a sequential chain, keeping the same sequence for all products. The number of machines at each stage of processing can be different. None of the operations performed allows for interruptions during its execution. Post-processing time losses after completion of the 1st stage of manufacturing are also taken into account. As the optimality criterion, the implementation of the whole range of work in the shortest possible time is considered. Estimates are obtained of the lower bound of the optimal sequence of products processing, ensuring the implementation of all works at two stages of the processing in the shortest possible time. Algorithms are proposed of the exact and the approximate solution of the problem by the branch and bound method and the dynamic programming method, as well as a heuristic algorithm of polynomial complexity for obtaining the approximate solutions. The decision algorithms are illustrated with a numerical example. The results obtained can be widely used in work scheduling systems for the production sectors and workshops of machine-building and instrument-making production, as well as for the processes in woodworking, electronics and light industry.

Keywords: two-stage schedules, flow-shop-problem, optimal sequences, branch and bound method, dynamic programming, heuristic algorithm.