

# РЕШЕНИЕ ОБОБЩЕННОЙ ЗАДАЧИ ДЖОНСОНА С ОГРАНИЧЕНИЯМИ НА СРОКИ ВЫПОЛНЕНИЯ ЗАДАНИЙ И ВРЕМЕНА РАБОТЫ МАШИН. Ч. 1. Точные методы решения

Ю.А. Зак

Задача поиска оптимальной перестановки, определяющей последовательность выполнения некоторого множества заданий в условиях фиксированной и одинаковой для всех заданий последовательности выполнения отдельных работ на различных машинах, обобщена на случай, когда заданы ограничения на стартовые и завершающие сроки как выполнения отдельных заданий, так и на времена работы оборудования. Исследованы свойства допустимых и оптимальных последовательностей выполнения заданий. Приведены расчетные формулы для вычисления нижней границы суммарной длины расписания. Разработаны точные и приближенные методы решения задачи.

**Ключевые слова:** оптимальное расписание, последовательность выполнения заданий, ограничения на времена начала и завершения.

## ВВЕДЕНИЕ

Классическая задача теории расписаний, сформулированная Джонсоном («flow-shop problem») [1, 2], заключается в построении оптимальной последовательности выполнения комплекса заданий, а также в определении начальных и завершающих сроков выполнения всех работ (операций), входящих в каждое из этих заданий, обеспечивающих минимальное время завершения всего комплекса работ. Существенная особенность заключается в том, что состав и последовательности комплексов работ на рабочих станциях (машинах) для каждого из заданий одинаковы. Оптимальное решение ищется в подмножестве расписаний, для которого последовательность выполнения всего комплекса работ для всех рабочих станций (машин) одинакова.

Большинство рассматриваемых в литературе приближенных методов решения задачи в традиционной формулировке [1–4] не позволяют оценить точность полученного приближенного решения даже в случае, когда не заданы ограничения на конечные сроки выполнения отдельных заданий.

Эффективные формулы вычисления нижних границ оптимального решения позволили бы останавливать процесс решения при получении последовательности выполнения заданий с суммарной длительностью, удовлетворяющей поставленным требованиям точности.

Для конструирования алгоритмов получения точного решения задачи на основе метода «ветвей и границ» («branch-and-bound-method» [5–9]) должны быть получены эффективные оценки оптимального решения и для случаев, когда построена подпоследовательность выполнения только некоторого подмножества заданий, а остальные еще неупорядоченные задания должны продолжить эту последовательность.

Для практических приложений представляет существенный интерес рассмотрение обобщенной задачи Джонсона в условиях, когда заданы ограничения на начальные и конечные сроки выполнения заданий и использования рабочих станций. В такой постановке сформулированная задача имеет много приложений в планировании производства и организации обслуживания объектов. Исследованиям математических свойств таких задач не уделялось должного внимания в литературе. За-

частую очень сложно на начальных этапах решения установить факт несовместности исходной системы ограничений, что, как правило, также сопряжено со значительным объемом вычислений.

## 1. ПОСТАНОВКА ЗАДАЧИ

На  $m$  рабочих станциях (машинах) должно быть выполнено  $n$  заданий, каждое из которых состоит из фиксированного и одного и того же для каждого задания числа операций. Последовательность выполнения операций каждого задания строго задана и одинакова для всех заданий. Каждая из  $m$  операций задания выполняется на некоторой конкретной рабочей станции и не допускает разрывов во времени выполнения. Следовательно, пара индексов  $(i, k)$ ,  $i = 1, \dots, n$ ;  $k = 1, \dots, m$  однозначно идентифицируют каждую операцию любого задания.

Заданы:

$\Pi = \{1, \dots, k, \dots, m\}$  — последовательность выполнения операций, одинаковая для каждого задания;

$t(i, k)$  — время выполнения операции  $(i, k)$ ;

$b_i$  и  $B_i$  — соответственно граничные сроки на начало и завершение выполнения  $i$ -го задания;

$h_k$  и  $H_k$  — соответственно граничные начальные и конечные времена возможности использования  $k$ -й рабочей станции.

Если величины  $b_i$  или  $h_k$  не заданы, то они полагаются равными нулю. В случае отсутствия граничных значений  $B_i$  или  $H_k$  они полагаются равными  $\infty$ .

Все граничные значения и времена выполнения операций предполагаются целочисленными величинами.

Обозначим:

$x(i, k)$  — время начала выполнения операции  $(i, k)$ ;

$\theta(i, k)$  — допустимый наиболее ранний срок начала выполнения операции  $(i, k)$ ;

$\sigma(i, k) = x(i, k) + t(i, k)$  — время завершения выполнения операции  $(i, k)$ ;

$T_i = \sigma(i, m)$  — время завершения выполнения операции  $i$ -го задания,

$T = \max_{1 \leq i \leq m} T_i$  — время окончания выполнения всего множества заданий.

Необходимо найти оптимальную последовательность выполнения заданий и определяемые ею целочисленные значения времен начала и окончания выполнения всех операций, обеспечивающие соблюдение всех ограничений на допустимые сроки выполнения всех заданий и времена работы рабочих станций, которая минимизирует время выполнения всего комплекса работ  $T$ .

На основе исследования и анализа свойств, допустимых и оптимальных последовательностей выполнения заданий предлагаются как формулы, так и алгоритмы, требующие незначительных объемов вычислений, для оценки совместности системы ограничений и минимальной длительности расписания в условиях, когда построена последовательность выполнения некоторого подмножества заданий (некоторый частичный план). В процессе развития частичного плана (с увеличением длины последовательности, т. е. числа заданий, включенных в нее) эти оценки становятся все более точными и при построении полной последовательности совпадают с решением. На основе предложенных оценок уже на этапе предварительного анализа и на каждом из шагов алгоритма происходит эффективный отсев подмножеств, не содержащих допустимых расписаний. Во многих случаях, когда система ограничений является несовместной, определяются номера рабочих станций, ресурс времени которых должен быть раздвинут и альтернативные номера заданий, начальные и конечные сроки выполнения которых должны быть скорректированы.

Рассматриваются методы решения задачи, в которых оптимальная последовательность заданий строится последовательно и вначале состоит только из одного задания, выполняемого первым. Затем эта последовательность расширяется включением некоторого задания, которое выполняется вторым и т. д.

Частичным планом  $P^s$  назовем некоторую упорядоченную подпоследовательность выполнения подмножества  $I_1^s$ , включающего в себя  $R^s$  заданий, —  $J^s = \{i_1^s, i_2^s, \dots, i_R^s\}$ , в которой определены допустимые сроки начала и окончания выполнения каждой операции всех входящих в эту последовательность заданий. В частичном плане первым выполняется задание  $i_1^s$ , затем задание  $i_2^s$  и последним — задание  $i_R^s$ . Множество различных последовательностей выполнения остальных заданий расписания  $I_2^s = I/I_1^s$ , которое необходимо определить, назовем дополнением этого частичного плана и обозначим его  $D^s$ . Для каждого частичного плана могут быть определены нижние границы суммарной длительности расписания, оценки возможности выполнения системы ограничений на начальные и конечные сроки выполнения заданий и на времена работы оборудования. В подмножестве дополняющих планов могут быть исключены подмножества последовательностей, не содержащих допустимых решений.



## 2. ОЦЕНКА НИЖНЕЙ ГРАНИЦЫ ДЛИНЫ РАСПИСАНИЯ

Рассмотрим вначале оценку минимальной длины расписания в традиционной постановке flow-shop problem без учета ограничений на начальные и конечные сроки выполнения заданий и возможности использования машин. Ясно, что общая продолжительность расписания не может быть меньше длительности выполнения каждого из заданий

$$E^0 = F \geq \max_{1 \leq i \leq n} \sum_{k=1}^m t(i, k).$$

Для начала работы каждой  $k$ -й рабочей системы необходимо, чтобы все операции  $l = 1, \dots, k - 1$ , предшествующие этой  $k$ -й операции, были уже выполнены. Для этого в  $i$ -м задании требуется време-

ни не менее чем  $\eta_1(i, k) = \sum_{l=1}^{k-1} t(i, l)$  даже в том

случае, если при выполнении этого задания ни одна из рабочих станций не теряет времени на простой после выполнения предыдущей операции. Суммарное время выполнения всех операций

на  $k$ -й машине равно  $\sum_{i=1}^n t(i, k)$ . После того, как

операции всех заданий на  $k$ -й рабочей станции будут выполнены, еще требуется дополнительное время работы машин,  $l = k + 1, \dots, m$ , равное

$\eta_2(i, k) = \sum_{l=k+1}^m t(i, l)$ , для того чтобы завершить вы-

полнение  $i$ -го задания. Следовательно, оценка времени выполнения расписания из условия занятости рабочих систем определяется выражением

$$E^0 = F \geq \max_{1 \leq k \leq m} \left\{ \sum_{i=1}^n t(i, k) + \min_{1 \leq i \leq n} \eta_1(i, k) + \min_{1 \leq i \leq n} \eta_2(i, k) \right\},$$

а с учетом оценки завершения выполнения отдельных заданий — выражением

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

Выражение для нижней границы длины расписания в виде (1) известно и широко применяется в литературе (см., например, работы [1, 2, 10]). Од-

нако оно не годится при наличии ограничений на сроки выполнения заданий и времена работы оборудования, а также в случаях, когда необходимо оценить эффективность отдельных частичных планов.

Для вычисления нижней границы оптимальной длины расписания в условиях ограничений рассмотрим расчетные формулы вычисления сроков начала и завершения выполнения операций заданий некоторого частичного плана  $P^s$ , которые упорядочены в последовательности  $P^s = \{i_1^s, i_2^s, \dots, i_{l-1}^s, i_l^s, \dots, i_R^s\}$ :

$$\theta(i_1^s, 1) = \max[h_1, b(i_1^s)],$$

$$\sigma(i_1^s, 1) = \theta(i_1^s, 1) + t(i_1^s, 1) - 1, \quad (2)$$

$$\theta(i_1^s, k) = \max[h_k, \sigma(i_1^s, k - 1) + 1],$$

$$\sigma(i_1^s, k) = \theta(i_1^s, k) + t(i_1^s, k) - 1, \quad k = 2, \dots, m, \quad (3)$$

$$\theta(i_l^s, k) = \max[\sigma(i_l^s, k), \sigma(i_l^s, k - 1)] + 1,$$

$$\sigma(i_l^s, k) = \theta(i_l^s, k) + t(i_l^s, k) - 1, \quad (4)$$

$$k = 1, \dots, m, \quad l = i_2^s, \dots, i_R^s.$$

В формуле (4) полагается  $\sigma(i_l^s, k - 1) = 0$  при  $k = 0$ .

Определим следующие величины:

— время выполнения рабочей системой  $k$  всего подмножества заданий  $I_2^s$

$$\tau^s(k) = \sum_{i \in I_2^s} t(i, k), \quad k = 1, \dots, m; \quad (5)$$

— наиболее ранние сроки начала выполнения подмножества заданий  $I_2^s$  на каждой  $k$ -й рабочей системе

$$\mu^s(i, 0) = 0, \quad \mu^s(i, 1) = \sigma(i_R^s, 1) + 1, \quad (6)$$

$$\mu^s(i, k) = \max[\sigma(i_R^s, k) + 1, \mu^s(i, k - 1) + t(i, k - 1)], \quad k = 2, \dots, m, \quad i \in I_2^s; \quad (7)$$

— наиболее ранний срок завершения выполнения всех операций каждого из заданий  $i \in I_2^s$  после окончания работы  $k$ -й рабочей системы

$$g^s(i, k) = \sum_{j=k+1}^m t(i, j), \quad k = 1, \dots, m, \quad i \in I_2^s; \quad (8)$$

— наиболее ранний срок завершения выполнения каждого задания  $i \in I_2^s$

$$\lambda^s(i, m) = \mu^s(i, m) + t(i, m) - 1, \quad i \in I_2^s, \quad (9)$$

или более точная оценка

$$\lambda^s(i, m) = \max_{0 \leq k \leq m-1} \left[ \mu^s(i, m-k) + \sum_{r=0}^k t(i, m-r) - 1 \right]. \quad (10)$$

**Утверждение 1.** Продолжительность выполнения всего комплекса работ для частичного плана не может быть меньше величины

$$F^s = \max \left\{ \max_{i \in I_2^s} \lambda^s(i, m); \max_{1 \leq k \leq m} \left[ \min_{i \in I_2^s} \mu^s(i, k) + \min_{i \in I_2^s} g(i, k) + \sum_{i \in I_2^s} t(i, k) - 1 \right] \right\}. \quad (11)$$

Для начального частичного плана  $P^0$  ( $I_1^s = \emptyset$ ,  $I_2^s = I$ ) выражение (10) имеет вид

$$F^0 = \max \left\{ \max_{1 \leq i \leq n} \sigma^0(i, m); \max_{1 \leq k \leq m} \left[ \min_{1 \leq i \leq n} \sigma^0(i, k) + \min_{i \in I} \sum_{l=k+1}^m t(i, l) + \sum_{i \in I} t(i, k) - 1 \right] \right\}. \quad (12)$$

Выражение (12), в котором значения  $\sigma^0(i, k)$  и  $\sigma^0(i, m)$  вычисляются по формулам (2) и (3), эквивалентно выражению (1).

### 3. ОЦЕНКИ ВОЗМОЖНОСТИ ВЫПОЛНЕНИЯ ЗАДАННОЙ СИСТЕМЫ ОГРАНИЧЕНИЙ ФОРМУЛИРУЕМОЙ ЗАДАЧИ И ПОСТРОЕННЫХ ЧАСТИЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ВЫПОЛНЕНИЯ МНОЖЕСТВА ЗАДАНИЙ

Далее исследуются свойства допустимых частичных последовательностей выполнения заданий. Это позволит уже на первых шагах алгоритма исключить из рассмотрения подмножество последовательностей выполнения заданий, не содержащих допустимых планов, а также на начальном этапе решения установить факт несовместности системы ограничений исходной задачи.

**Утверждение 2.** Если для частичного плана  $P^s$  существует хотя бы одно задание  $i \in I_2^s$ , для которого справедливы условия

$$\lambda^s(i, m) > B_i, \quad (13)$$

то он не содержит допустимых последовательностей и может быть исключен из рассмотрения. ♦

**Утверждение 3.** Если для частичного плана  $P^s$  существует хотя бы одна рабочая система  $k = 1, \dots, m$ , для которой справедливы условия

$$\max_{i \in I_2^s} \mu^s(i, k) + \sum_{i \in I_2^s} t(i, k) - 1 > H_k, \quad (14)$$

он не содержит допустимых последовательностей и может быть исключен из рассмотрения. ♦

**Утверждение 4.** Если для некоторого частичного плана  $P^s$  справедливо неравенство

$$\max_{i \in I_2^s} B_i < F^s, \quad (15)$$

то он не содержит допустимых решений. ♦

Ясно, что для частичного плана  $P^s$  время завершения выполнения любого из заданий  $i \in I_2^s$  не может быть меньше времени  $\delta^s(i, m) = \sigma^s(i_R^s, m) + t(i, m) - 1$  или (более точная оценка)

$$\delta^s(i, m) = \max_{0 \leq k \leq m-1} \left[ \sigma^s(i_R^s, m-k) + \sum_{r=0}^k t(i, m-r) - 1 \right].$$

Упорядочим все задания подмножества  $I_2^s$  по возрастанию граничных значений времени завершения их выполнения  $B_i$ :

$$B_{j_1} \leq B_{j_2} \leq \dots \leq B_{j_g} \leq \dots \leq B_{j_G} \quad (16)$$

и определим значения

$$\begin{aligned} \delta^s(j_1, m) &= \mu^s(j_1, m); \\ \delta^s(j_g, m) &= \mu^s(j_{g-1}, m) + t(j_g, m) - 1, \\ &g = 2, \dots, G. \end{aligned} \quad (17)$$

Более точные оценки могут быть получены из формул

$$\begin{aligned} \delta^s(j_0, k) &= \sigma^s(i_R, k), \quad \delta^s(j_g, k) = \\ &= \max[\delta^s(j_{g-1}, k), \delta^s(j_g, k-1) + \\ &+ t(j_g, k-1) - 1], \end{aligned} \quad (18)$$

$$k = 1, \dots, m, \quad g = 1, \dots, G.$$

**Утверждение 5.** Если на этапе анализа некоторого частичного плана  $P^s$  в процессе вычисления значений  $\delta^s(j_g, m)$  согласно последовательности (16) обнаружится справедливость хотя бы одного из неравенств

$$\delta^s(j_g, m) > B_{j_g}, \quad g = 1, \dots, G, \quad (19)$$



то он не содержит допустимых последовательностей и может быть исключен из рассмотрения. ♦

Сформулируем очевидные условия несовместности системы ограничений на сроки выполнения заданий и возможностей использования рабочих систем обобщенной flow-shop problem.

**Утверждение 6.** Пусть исходное множество заданий  $I$  упорядочено в порядке возрастания граничных значений окончания выполнения отдельных заданий  $B_{i_1} \leq B_{i_2} \leq \dots \leq B_{i_g} \leq \dots \leq B_{i_n}$  и согласно формулам (2) и (3) определены наиболее ранние возможные сроки начала и завершения всех операций каждого задания  $\theta^0(i, k)$ ,  $\sigma^0(i, k)$ ,  $k = 1, \dots, m$ ;  $i = 1, \dots, n$ ; согласно выражению (1) вычислена нижняя граница общей длительности завершения выполнения всех заданий  $F^0$ . Если для вычисленных граничных значений справедливо хотя бы одно из следующих неравенств на сроки завершения выполнения отдельных операций или заданий:

$$\sigma^0(i, m) > B_i, \quad i = 1, \dots, n, \quad (20)$$

$$\sigma_{\min}^0 + \sum_{l=2}^g t(i_l, m) - 1 > B_{i_g}, \quad g = 2, \dots, n, \quad (21)$$

где  $\sigma_{\min}^0 = \min_{i \in I} \{\sigma^0(i, m) : B_i \neq \infty\}$ ,

$$\min_{1 \leq i \leq n} \theta^0(i, k) + \sum_{i=1}^n t(i, k) - 1 > H_k, \quad k = 1, \dots, m, \quad (22)$$

$$F^0 > \max_{1 \leq i \leq n} B_i, \quad (23)$$

то исходная система ограничений задачи несовместна. ♦

Утверждение 6 позволяет уже на начальном этапе решения установить факт несовместности сформулированной системы ограничений, определить индексы заданий и рабочих систем, являющихся «узким местом» и граничный временной ресурс которых должен быть расширен.

#### 4. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ

В основе алгоритмов точного решения задачи лежит метод ветвей и границ. Впервые подобный эффективный алгоритм был предложен в работе [5]. Дальнейшее развитие алгоритмов данного класса нашло отражение в работах [6, 7, 11–13] и др. Во всех алгоритмах при вычислении границ и выборе переменной для ветвления используются модифицированные времена выполнения заданий на двух машинах и алгоритмы Джонсона. При огра-

ничениях на сроки выполнения заданий и времена работы машин данные алгоритмы не позволяют эффективно отсеивать не содержащие допустимые расписания вариантов. Описываемые далее подходы лишены указанных недостатков.

#### 4.1. Параметры частичных планов

Каждый развиваемый частичный план  $P^s$ , определяемый подпоследовательностью выполнения некоторых  $R^s$  заданий и включаемый в дерево решений задачи, характеризуется следующими признаками:

$P_0^s$  — номер плана в дереве решений;

$P_1^s, \dots, P_i^s, \dots, P_n^s$  — информация о состоянии

выполняемых заданий,  $i = 1, \dots, n$ . Если  $P_i^s$  равно некоторому положительному числу  $r$ , где  $1 \leq r \leq n$ , то это определяет, что  $i$ -е задание стоит на  $r$ -м месте в подпоследовательности  $J^s$ . Если  $P_i^s = 0$ , то место этого задания в последовательности еще не определено и должно быть выбрано. Если  $P_i^s = -1$ , то установка  $i$ -го задания на  $(R^s + 1)$ -е место последовательности в данном частичном плане недопустима либо в дереве решений уже существует продолжение частичного плана  $P^s$ , в котором это задание стоит на этом  $(R^s + 1)$ -м месте;

$P_{n+1}^s, \dots, P_{n+i}^s, \dots, P_{2n}^s$  — значения оценки длины расписания, если  $i$ -е задание будет установлено в этой последовательности на  $(R^s + 1)$ -м месте —  $(F^s|i)$ . Для всех заданий, у которых  $P_i^s \neq 0$ ,  $P_{n+i}^s = \infty$ ;

$P_{2n+1}^s, \dots, P_{2n+k}^s, \dots, P_{2n+m}^s$  — времена завершения операций на соответствующих машинах для последнего  $R^s$ -го задания, стоящего в подпоследовательности  $J^s$  данного частичного плана;

$P_{2n+m+1}^s = \min_{1 \leq i \leq n} P_{n+i}^s$  — нижняя оценка критерия оптимальности для данного частичного плана;

$P_{2n+m+2}^s$  — признак возможности и целесообразности продолжения данного частичного плана.

Если  $P_{2n+m+1}^s = \infty$  и все значения признаков  $P_i^s$ ,  $i = 1, \dots, n$ , отличны от нуля, то  $P_{2n+m+2}^s = -1$ , и данный частичный план уже продолжен во все перспективные направления и больше (либо вообще) не содержит допустимых расписаний. В противном случае  $P_{2n+m+2}^s = 0$ , и еще не рассмотрены все возможные допустимые продолжения.



**4.2. Преобразование параметров при продолжении частичного плана  $P^s$  добавлением в подпоследовательность  $J^s$  некоторого  $j$ -го задания**

Пусть на некотором шаге алгоритма построено  $N$  частичных планов. При продолжении частичного плана  $P^s$  назначением на последнее  $(R^s + 1)$ -е место в подпоследовательности  $J^s$  некоторого задания  $j \in I_2^s$  признаки вновь образованного частичного плана  $P^q$  определяются следующим образом:

$$P_0^q = N + 1,$$

$$P_j^q = R^s + 1,$$

если  $1 \leq P_i^s \leq n$  и  $i \neq j$ ;  $P_i^q = 0$  во всех остальных случаях,  $i = 1, \dots, n$ ;

$$P_{n+i}^q = P_{n+j}^s,$$

если  $P_i^q = 0$ , и  $P_{n+i}^q = \infty$  в противном случае,  $i = 1, \dots, n$ ;

$$P_{2n+k}^q = \sigma^q(j, k) = \max[P_{2n+k}^s, P_{2n+k-1}^q + t(j, k - 1)] + t(j, k) - 1, \quad k = 1, \dots, m. \quad (24)$$

В формуле (24) предполагается  $P_{2n}^q = 0$ .

$$P_{2n+m+1}^q = P_{n+j}^s; \quad P_{2n+m+2}^q = 0,$$

если существуют признаки  $P_i^q = 0$ ;

$$P_{2n+m+1}^q = \infty, \quad P_i^q = -1 \text{ в противном случае.}$$

Число членов во вновь образованной подпоследовательности  $J^q$  равно  $R^q = R^s + 1$ . Параметры  $P_j^s$  и  $P_{n+j}^s$  частичного плана  $P^s$  полагаем равными  $P_j^s = -1, P_{n+j}^s = \infty$ . Корректируем также параметр  $P_{2n+m+1}^s = \min_{1 \leq i \leq n} P_{n+i}^s$ . Если  $P_{2n+m+1}^s = \infty$  и все  $P_i^s$  отличны от нуля, то полагаем  $P_{2n+m+2}^s = -1$ .

После выполнения данных преобразований число частичных планов  $N$  также увеличивается на единицу.

**4.3. Выбор задания, включаемого последним в подпоследовательность (ветвление)**

Для каждого из заданий подмножества  $j \in I_2^s$ , у которых  $P_j^s = 0$ , выполняем следующие вычисления.

Рассчитываем сроки завершения выполнения задания по формулам

$$\sigma^s(j, k|J^s) = \max[\sigma^s(j, k - 1|J^s) + t(j, k - 1); P_{2n+k}^s] + t(j, k) - 1.$$

Полагаем  $I_2^s = I_2^s/j$ ,  $\delta^s(i, m) = \sigma^s(j, m|J^s)$  и проверяем возможность выполнения системы ограничений задачи в соответствии с системой неравенств (13)–(19). Если справедливы условия хотя бы одного из утверждений 1–4, то назначение в  $P^s$ -м частичном плане задания  $j$  на  $(R^s + 1)$ -е место в последовательности недопустимо. Полагаем  $P_j^s = -1$  и  $P_{n+j}^s = \infty$ . Если не выполняется ни одно из неравенств системы (13)–(19), то вычисляем оценку длины расписания при условиях, что на  $(R^s + 1)$ -м месте в последовательности  $J^s$  стоит задание  $j$  по формулам (11). Рассчитанное значение  $(F^s|j)$  присваиваем параметру  $P_{n+j}^s = (F^s|j)$ .

Находим

$$P_{n+w}^s = \min\{P_{n+j}^s; j \in I_1^s, P_j^s = 0\}. \quad (25)$$

Соответствующий номер задания  $j = w$  определяет уточненную нижнюю границу критерия оптимальности для частичного плана  $P^s$  ( $P_{2n+m+1}^s = P_{n+w}^s$ ) и выбирается при продолжении данного частичного плана (см. п. 4.2).

**4.4. Алгоритмы получения точных и приближенных решений задачи**

Алгоритмы решения основываются на последовательном развитии некоторого допустимого и перспективного частичного плана  $P^s$  и назначении некоторого задания  $j \in I_2^s$  на  $(R^s + 1)$ -е место, т. е. после всех заданий данной подпоследовательности  $J^s$ . На каждом шаге алгоритма выбирается некоторый допустимый план с эффективной (минимальной или наилучшей среди некоторого подмножества частичных планов) оценкой. В этом плане в соответствии с выражением (25) определяется наиболее перспективное задание, которое целесообразно поставить на данное место в последовательности, проверяются возможности выполнения всей системы ограничений, связанные с данным назначением, исключаются продолжения, не содержащие допустимых расписаний, и уточняется оценка длины кратчайшего расписания, связанного с этим назначением.

Каждый частичный план представляется в виде вершины дерева анализируемых вариантов. Корнем этого дерева служит начальный частичный план  $P^0$ , у которого  $I_1^0 = J^0 = \emptyset, I_2^0 = I$ . Для на-



частичного плана  $P^0$  по формулам (2)—(4) рассчитываются значения  $\theta^0(i, k)$  и  $\sigma^0(i, k)$ , а также согласно выражениям (5)—(10) значения  $\tau^0(k)$ ,  $\mu^0(i, k)$ ,  $g^0(i, k)$  и  $\lambda^0(i, k)$ . Нижняя граница длины допустимого расписания выполнения всех заданий  $F^0$  рассчитывается в соответствии с формулой (12). Затем оценивается совместность исходной системы ограничений задачи в соответствии с выражениями (20)—(23). Если выполняется хотя бы одно из неравенств утверждения 6, то система ограничений задачи несовместима, и алгоритм с соответствующим сообщением заканчивает работу. В противном случае построенный начальный план является корнем дерева решений задачи. Первый полученный допустимый частичный план, у которого  $I_1^s = I$ , является допустимым расписанием выполнения заданий. Номер этого плана и значение его оценки (показатель  $P_{2n+m+1}^s$ ) запоминается как «рекорд», т. е. наилучшее из полученных допустимых решений. При получении следующего допустимого расписания со значением критерия оптимальности лучшим, чем в прежних допустимых решениях, производится замена «рекорда». Если получено значение «рекорда» не хуже, чем значение оценки длины оптимального расписания (показатель  $P_{2n+m+1}^s$ ) у любого из частичных планов, то рассчитанный «рекорд» является решением задачи и алгоритм заканчивает работу.

Если на какой-то итерации алгоритма еще не получено допустимых расписаний и во всех частичных планах значение показателя  $P_{2n+m+2}^s = -1$ , то не существует допустимых расписаний, и алгоритм с соответствующим сообщением заканчивает свою работу.

Далее следует формальное описание алгоритма. Обозначим:  $N$  — число построенных частичных планов,  $E$  — длина расписания наилучшего из полученных допустимых решений.

#### 4.4.1. Алгоритм А

Данный алгоритм состоит из отдельных шагов, которые в некоторой логической последовательности выполняются на каждой из итераций.

*Шаг 0* (предварительный). Полагаем  $N = 0$ ,  $E = \infty$ . Определяем начальный частичный план  $P^0$ , у которого  $I_1^0 = J^0 = \emptyset$ ,  $I_2^0 = I$ . Вычисляем оценку длины допустимого расписания  $F^0$  по формуле (12) и оцениваем допустимость этого частичного плана в соответствии с условиями утверждения 6. Если в результате вычисления будет установлено, что система ограничений задачи несовместима, то алгоритм с соответствующим сообщением заканчивает работу. В противном случае по-

казатели метки плана  $P^0$  определяются следующим образом:

$$P_1^0 = 0; \quad P_i^0 = 0, \quad i = 1, \dots, n; \quad P_{n+i}^0 = F^0, \quad i = 1, \dots, n; \\ P_{2n+k}^0 = 0, \quad k = 1, \dots, m; \quad P_{2n+k+1}^0 = F^0; \quad P_{2n+k+2}^0 = 0.$$

Выбираем частичный план  $P^0$  и переходим к шагу 1.

*Шаг 1.* Для выбранного частичного плана  $P^s$  выбираем задание  $j = w$ , включаемое в последовательность в соответствии с алгоритмом, описанным в п. 4.3. Если такого допустимого задания не существует, то полагаем  $P_{2n+k+1}^s = \infty$ ,  $P_{2n+k+2}^s = -1$  и переходим к шагу 4. В противном случае — к шагу 2.

*Шаг 2.* Образует новый частичный план  $P^q$  и выполняем преобразование параметров при продолжении частичного плана  $P^s$  добавлением в подпоследовательность  $J^s$  некоторого  $j$ -го задания ( $j = w$ ) в соответствии с алгоритмом, описанным в разделе 5.2. Если во вновь построенном частичном плане  $P^q$  все показатели  $P_i^q = b_i$ , где  $1 \leq b_i \leq n$ ,  $i = 1, \dots, n$ , переходим к шагу 5. В противном случае — к шагу 3.

*Шаг 3.* Если  $P_{2n+k+1}^q < E$ , то, зафиксировав этот частичный план  $P^q$  как выбранный план для дальнейшего продолжения (обозначим его  $P^s$ ), переходим к шагу 1. Если  $P_{2n+k+1}^q \geq E$ , то — к шагу 4.

*Шаг 4.* Среди всех частичных планов  $P^s$ , у которых  $P_{2n+k+1}^s \neq \infty$ ,  $P_{2n+k+2}^s = 0$ , находим некоторый частичный план  $P^l$ , у которого

$$P_{2n+k+1}^l = \min_{1 \leq l \leq N} \{P_{2n+k+1}^s \mid P_{2n+k+2}^s = 0\}; \\ P_{2n+k+1}^s < E\} = F_{\min}.$$

Если такой частичный план существует, то, зафиксировав этот частичный план  $P^q$  как выбранный план для дальнейшего продолжения (обозначим его  $P^s$ ), переходим к шагу 1. В противном случае — к шагу 6.

*Шаг 5.* Если  $P_{2n+k+1}^q < E$ , то полагаем  $E = P_{2n+k+1}^q$ , запоминаем номер плана  $P^q$  как наилучшее текущее допустимое расписание. Если  $E \leq F_{\min}$ , то переходим к шагу 6. В противном случае — к шагу 4.

*Шаг 6.* Если  $E = \infty$ , то задача не имеет допустимых решений, и алгоритм с соответствующим сообщением заканчивает работу. В противном случае в памяти сохранен номер вершины дерева вариантов  $P^l$ , определяющей оптимальное расписание, длительность которого равна  $E$ . По признакам метки  $P_i^l = b_i$  восстанавливается оптимальная последовательность выполнения заданий, в соответствии с которой по формулам (2)—(4) (при условии  $R^l = n$ ) рассчитываются сроки начала и завершения выполнения всех заданий. На этом алгоритм завершает свою работу.

Заметим, что алгоритм А может применяться для получения приближенных (с заданной наперед точностью  $\varepsilon = F/F_{\min}$ ) решений задачи. В этом случае на ша-

ге 0 полагается  $F_{\min} = F^0$ . Это значение корректируется в сторону увеличения, например, на каждом шаге 3 или шаге 4  $F_{\min} = \min_{1 \leq s \leq N} F_{2n+k+1}^s$ . На шаге 5 проверка условия  $E \leq F_{\min}$  заменяется проверкой  $E \leq \epsilon * F_{\min}$ .

Алгоритм А может также применяться для получения точных и приближенных решений для задач job-shop problem (см., например, [14, 15]), а также flow-shop problem в традиционной постановке, т. е. при отсутствии ограничений на сроки выполнения заданий и времена работы оборудования. В этом случае на шагах 0 и 1 алгоритма не проверяются возможности выполнения системы ограничений задачи, а времена начала и окончания выполнения заданий вычисляются не согласно выражениям (2)–(4), а по упрощенным формулам

$$\theta^s(i_1, 1) = 0, \sigma^s(i_1, 1) = t(i_1, 1) - 1,$$

$$\theta^s(i_1, k) = \sigma^s(i_1, k - 1) + 1,$$

$$\sigma^s(i_1, k) = \sigma^s t(i_1, k - 1) + t(i_1, k) - 1, \quad k = 1, \dots, m.$$

### 5. ИЛЛЮСТРАТИВНЫЙ ПРИМЕР

Пусть необходимо построить расписание выполнения восьми заданий, выполняемых на четырех машинах. Последовательность выполнения этих четырех опе-

Таблица 1

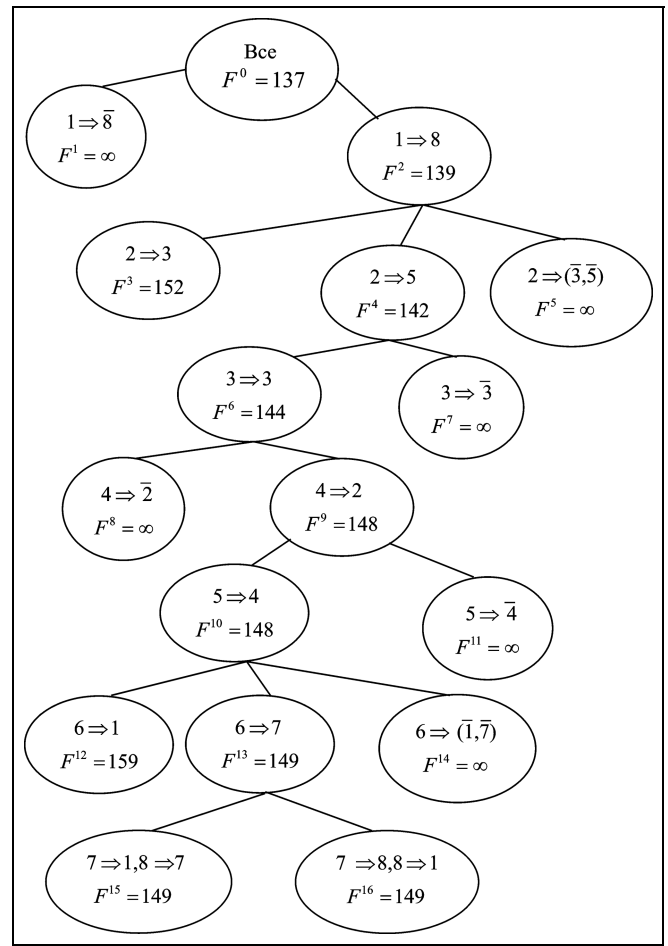
Исходные данные

<i>i</i>	<i>k</i> = 1	<i>k</i> = 2	<i>k</i> = 3	<i>k</i> = 4	<i>b<sub>i</sub></i>	<i>B<sub>i</sub></i>
1	10	15	12	7	15	170
2	6	12	20	10	5	110
3	8	7	21	15	10	98
4	15	4	3	25	0	130
5	6	9	10	15	7	(68) 70
6	4	18	15	6	2	175
7	13	18	15	10	10	145
8	3	5	13	15	0	65
<i>h<sub>k</sub></i>	5	10	10	20	—	—
<i>H<sub>k</sub></i>	75	120	150	190	—	—

Таблица 2

Наиболее ранние сроки начала и завершения всех операций заданий

<i>i</i>	<i>k</i> = 1		<i>k</i> = 2		<i>k</i> = 3		<i>k</i> = 4	
	$\theta$	$\sigma$	$\theta$	$\sigma$	$\theta$	$\sigma$	$\theta$	$\sigma$
1	10	24	25	39	40	51	52	58
2	5	11	12	23	24	43	44	53
3	10	17	18	24	25	45	46	60
4	5	19	20	23	24	26	27	51
5	7	12	13	21	22	31	32	46
6	5	8	9	26	27	31	32	37
7	10	22	23	27	28	30	31	40
8	5	7	10	23	24	39	40	54



Дерево решений задачи

раций для каждого из заданий одинакова. Все исходные данные примера — времена выполнения операций, граничные сроки начала и завершения отдельных заданий и возможностей использования оборудования — сведены в табл. 1.

Наиболее ранние сроки начала и завершения всех операций каждого задания, рассчитанные в соответствии с формулами (2)–(4), приведены в табл. 2.

Нижняя граница длины расписания, рассчитанная по формуле (12),  $F^0 = 137$ . Далее была проверена совместность системы ограничений. Все условия утверждений 2–4 выполняются. Последовательность (16) в условиях этого примера имеет вид:  $B_8 < B_5 < B_3 < B_2 < B_4 < B_7 < B_1 < B_6$ . В системе неравенств (17) для пятого задания значение  $\delta^0(i_5, 4)$  равно  $55 + 15 - 1 = 69$ . Так как  $69 > 68$ , то система ограничений примера несовместна. При корректировке ограничения на время завершения пятого задания ( $B_5 = 70$ ) может быть получено оптимальное решение, которое представлено на рисунке и в табл. 3.

На рисунке приняты следующие обозначения:  $r \Rightarrow i$  означает, что на *r*-е место в последовательности устанавливается *i*-е задание;  $r \Rightarrow \bar{j}$  — на *r*-е место в последовательности установка *j*-го задания запрещается.





Таблица 3

**Оптимальная последовательность выполнения заданий.  
Сроки начала и завершения выполнения операций**

$i$	$k = 1$		$k = 2$		$k = 3$		$k = 4$	
	$\theta$	$\sigma$	$\theta$	$\sigma$	$\theta$	$\sigma$	$\theta$	$\sigma$
8	5	7	10	23	24	39	40	54
5	8	13	24	33	40	50	55	69
3	14	21	34	40	51	71	72	86
2	22	27	41	52	72	91	92	101
4	28	42	53	56	92	94	102	126
7	43	55	57	61	95	107	127	136
1	56	65	66	80	108	119	137	143
6	66	69	81	98	120	134	144	149

В полученном решении выполнены все ограничения на сроки выполнения заданий и времена работы машин. Длина расписания равна 149.

С помощью предложенных алгоритмов были решены приведенные в литературе тестовые примеры flow-shop problem без учета ограничений на завершающие сроки выполнения заданий и времена работы машин. Результаты вычислительных экспериментов показали, что эффективность предложенных точных методов для этого класса задач не ниже известных в литературе.

### ЗАКЛЮЧЕНИЕ

Сформулирована обобщенная задача «flow-shop problem» в условиях ограничений на начальные и конечные сроки выполнения заданий и времена работы машин.

Установлены свойства допустимых расписаний сформулированной задачи и получены эффективные оценки нижней границы длины расписания как на начальном этапе процесса решения, так и в процессе решения при построенной подпоследовательности выполнения некоторого подмножества заданий.

Установленные свойства допустимых расписаний позволяют уже на начальных этапах решения установить несовместность исходной системы ограничений, выявить узкие места в ресурсах машин, определить перечни заданий, которые не могут быть выполнены в установленные сроки, а в случае совместности условий задачи, последовательно, на каждом шаге алгоритма существенно сокращать область поиска.

Разработаны алгоритмы точных и приближенных решений задачи на основе метода ветвей и границ. Алгоритмы сконструированы таким образом, что наличие дополнительных условий ограничивает область поиска, делая ее более направленной, благодаря чему в ряде случаев способствует сокращению объема вычислений по сравнению с решением задач на безусловный экстремум.

Эффективность предложенных алгоритмов проиллюстрирована на числовом примере.

Разработано программное обеспечение, позволяющее решать практические задачи размерностью до нескольких десятков заданий в условиях ограничений на времена работы машин и сроки выполнения заданий, выводить результаты решения как в форме таблиц, так и в графической форме — в виде диаграмм Ганта.

Во второй части работы будут предложены некоторые гибридные алгоритмы и приближенные методы решения задач большой размерности, использующие различные известные ранее эвристики и правила предпочтения, а также локальные улучшения построенных на основе эвристик последовательностей.

### ЛИТЕРАТУРА

1. Domschke W., Scholl A., Voß S. Produktionsplanung. Ablauforganisatorische Aspekte. — Berlin: Springer Verlag, 1997. — 456 p.
2. Johnson S.M. Optimal-two-and tree stage production schedules with setup times included // Research Logistics Quarterly. — 1954. — N 1. — P. 61–68.
3. Gupta J.N.D. A functional heuristic algorithm for the flow-shop scheduling problem // Operational Research Quarterly. — 1971. — N 2. — P. 39–47.
4. Cambell H.G., Dudek R.A., Smith M.L. A heuristic algorithm for the  $n$  job,  $m$  machine sequencing problem // Management Science. — 1970. — Vol. 16. — P. 630–637.
5. Ignall E., Schrage L. Application of the branch and bound technique to some flow-shop scheduling problems // Operations Research. — 1965. — N 3. — P. 400–412.
6. Potts C.N. A adaptive branching rule for the permutation flow-shop problem // European Journal of Operational Research. — 1980. — N 5. — P. 19–25.
7. Lomnicki Z.A. A branch-and-bound algorithm for the exact solution of the three-machine scheduling problem // Operational Research Quarterly. — 1965. — Vol. 16. — P. 89–100.
8. Brucker P. Scheduling algorithms. — Berlin: Springer Verlag, 1995.
9. Blazewicz J., Domschke W., Pesch E. The job shop scheduling problem: Conventional and new solution techniques // European Journal of Operational Research. — 1996. — Vol. 93. — P. 1–33.
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. McMahon G.B. Optimal production schedules for flow shops // Canadian Operations Research Society Journal. — 1969. — N 7. — P. 141–151.
12. Lageweg B.J., Lenstra J.K., Rinnooy Kann A.H.G. A general bounding scheme for the permutation flow-shop problem // Operations Research. — 1978. — Vol. 26. — P. 53–67.
13. Carlier J., Rebai I. Two branch and bound algorithms for the permutation flow-shop problem // European Journal of Operational Research. — 1996. — Vol. 90. — P. 238–251.
14. Танаев В.С., Сотсков Ю.Н., Струсович В.А. Теория расписаний. Многостадийные системы. — М.: URSS, 1989. — 328 с.
15. Зак Ю.А. Некоторые свойства задач теории расписаний // Автоматика и телемеханика. — 1978. — № 1. — С. 123–132.

Статья представлена к публикации членом редколлегии В.Н. Бурковым.

**Зак Юрий Александрович** — д-р техн. наук, пенсионер, последнее место работы — Европейский центр по мехатронике (European Centre for Mechatronics Reutershagweg), г. Аахен, Германия,  
☎ +49 (0) 241/543255, ✉ yuriy\_zack@hotmail.com.