УДК 519.8 DOI: 10.17587/it.27.249-258

Ю. А. Зак, д-р техн. наук, e-mail: yuriy_zack@hotmail.com,
Аахен, Германия

Многостадийные open-shop-problem: расписания выполнения *N* заданий на *K* различных по техническим характеристикам машинах при произвольном порядке выполнения заданий

Рассматриваются постановки и математические модели open-shop-problem: выполнение N заданий на K машинах при произвольном порядке обработки каждого из заданий на всех машинах. В отличие от большинства публикаций, посвященных решению данной проблемы, поставленная задача решается в многостадийной постановке в условиях работы последовательной цепочки участков и цехов предприятия, а также в условиях наличия ограничений на времена начала и завершения выполнения заданий и допустимые времена работы машин. Кроме того, для систем календарного планирования работы предприятий чрезвычайно актуальным является рассмотрение этой задачи в многостадийной постановке, т. е. в условиях работы последовательной цепочки участков и цехов. Исследуются свойства допустимых и оптимальных расписаний, а также алгоритмы точных и приближенных методов решения этих задач последовательными алгоритмами оптимизации. На ранних этапах решения задачи устанавливается факт несовместности системы ограничений задачи и определяется подмножество заданий или ресурсов, временной диапазон которых должен быть расширен. Описанные алгоритмы решения задачи иллюстрируются числовыми примерами.

Ключевые слова: *многостадийные расписания, open-shop-problet, ограничения на времена выполнения заданий, нижняя граница длины расписания, последовательные алгоритмы оптимизации, эвристический алгоритм*

Введение

Рассмотрим задачу построения расписаний выполнения N заданий на K машинах в условиях, когда каждое из заданий $i = 1, ..., N$ должно пройти обработку на каждой из $k = 1, ..., K$ машин. При этом допустим произвольный порядок обработки каждого из заданий на всех машинах. Эта задача известна в литературе как open-shop-problem. В отличие от данной задачи в задачах job-shop-problem для каждого задания определен строгий порядок выполнения операций и последовательность машин, на которых они выполняются [1—5, 7, 8, 10]. Эти два класса задач теории расписаний имеют много практических приложений в календарном планировании производства и организации технического и сервисного обслуживания объектов.

В литературе [1, 4, 7, 8, 10, 12—14] задачи open-shop-problem рассматривались, в основном, в постановке без учета ограничений на времена завершения выполнения заданий. В такой постановке заданы времена выполнения всех заданий на каждой машине t_{ik}, и эти времена не могут претерпевать разрывы во времени выполнения. Одновременно на каждой машине может выполняться только одно задание. Необходимо определить последовательности выполнения всех заданий на каждой машине, а также времена начала и завершения выполнения каждого из заданий на всех машинах. В качестве критериев оптимальности задачи рассматривались завершение выполнения заданий в кратчайшие сроки, а также минимизация суммарного средневзвешенного времени работы машин.

Рассматриваемые задачи относятся к классу NP-сложных проблем. По сравнению с большим числом публикаций по решению wan-machine-problem, flow-shop, job-shop-problem, а также задач построения расписаний выполнения работ на параллельных машинах, задачам open-shop-problem уделялось очень мало внимания в монографиях и периодической литературе. Публикации касались, в основном, решения частных задач для случаев, когда число машин $K = 2$ (см., например, работы [1, 7, 9, 11]), по критерию минимизации времени цикла (выполнение расписаний в кратчайшие сроки). В литературе не рассматривались постановки задач в условиях заданной системы ограничений на времена начала и завершения выполнения заданий и допустимые времена работы машин. Для практических приложений, когда перед предприятиями в первую очередь стоят задачи выполнения в установленные сроки принятых портфелей заказов, решение сформулированных проблем в такой постановке является чрезвычайно актуальным. В ряде случаев такая постановка задачи важнее, чем оптимизация какого-либо из критериев эффективности.

В данной статье рассматриваются постановки и математические модели open-shop-problem в самой общей постановке (выполнение N заданий на K машинах) в условиях, когда заданы ограничения на времена начала и завершения выполнения заданий и допустимые времена работы машин. Кроме того, для систем календарного планирования работы предприятий чрезвычайно актуальным является рассмотрение этой задачи в многостадийной постановке, т.е. в условиях работы последовательной цепочки участков и цехов. В данной работе исследуются свойства допустимых и оптимальных расписаний, а также точные и приближенные методы решения таких задач последовательными алгоритмами оптимизации. На ранних этапах решения задачи устанавливается факт несовместности системы ограничений задачи и определение подмножества заданий или ресурсов, временной диапазон которых должен быть расширен. Предложенные в работе алгоритмы решения этих задач с достаточно высокой эффективностью могут быть использованы и для решения open-shop-problem при отсутствии каких-либо ограничений. Описанные алгоритмы решения задачи иллюстрируются числовыми примерами.

1. Постановка и математическая формулировка задачи

Пусть на K рабочих станциях (машинах), $k = 1, ..., K$, должно быть выполнено N заданий. i-е задание состоит из множества $\tilde{J}_i$ операций, каждая из которых выполняется на некоторой определенной машине. Порядок выполнения всех операций каждого из заданий может быть произвольным. В дальнейшем каждая операция (обозначим ее O_{ijk}, $i = 1, ..., N$, $j = 1, \ldots, \bar{J}_i$, $k = 1, ..., K$) однозначно определяется мультииндексом $a = (i, j, k)$. Времена выполнения операций суть целочисленные величины. Не допускается прерываний времен выполнения операций на машине. Одновременно на каждой машине не может выполняться более одной операции какого-либо задания. Каждая операция одного и того же задания выполняется на конкретной, определенной для нее технологией обработки машине. Схема производства многостадийных open-shop-problem представлена на рис. 1.

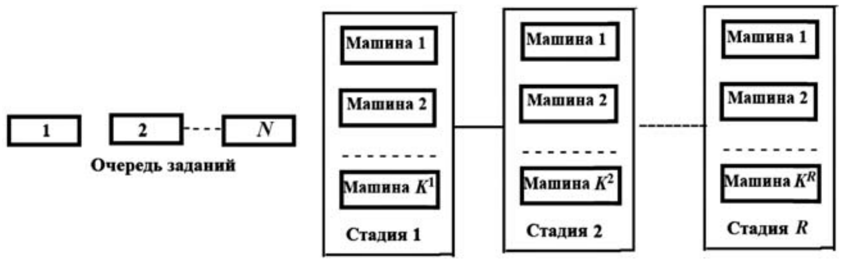

Схема производства многостадийных open-shop-problem

В случае многостадийных open-shop-problem приняты следующие обозначения переменных:

$r = 1, ..., R$ — индексы стадий производственного процесса;

$t^r(i, j, k)$ — время выполнения j-й операции i-го задания на k-й машине на r-й стадии обработки, $r = 1, ..., R$;

$x^r(i, j; k, l)$, $\theta^r(i, j; k, l) = [x^r(i, j; k, l), + + t^r(i, j; k, l) - 1\ ]$ — соответственно фактическое (установленное построенным расписанием выполнения работ) время начала и время завершения выполнения операции O_{ijk}, стоящей в очередности l на k-й машине на r-й стадии обработки, $r = 1, ..., R$;

$T^r(i)$ — время завершения выполнения i-го задания на r-й стадии обработки, $r = 1, ..., R$;

K^r — число машин на r-й стадии обработки;

$h^r(k)$, $H^r(k)$ — соответственно допустимые наиболее ранние сроки начала и наиболее поздние сроки окончания работы k-й машины, $k = 1, ..., K$, на r-й стадии обработки, $r = 1, ..., R$;

$d(i)$, $i = 1, ..., N$, — наиболее ранний срок начала выполнения i-го задания.

Значения возможных наиболее ранних и допустимых самых поздних времен начала выполнения всех этих операций на каждой k-й машине могут быть определены из следующих соотношений:

$$\begin{gathered} x^1(i,1;k,1) = \max[d(i), h^1(k)]; \\ x^1(i,1;k,l) = \max[d(i), \{\theta^1(p,j;k,l-1)+1\}], \\ i,p = 1,\ldots,N;\ j = 1,\ldots,m_i;\ k,g = 1,\ldots,K; \\ l = 1,\ldots,L_k; \end{gathered} \quad (1)$$

$$\begin{gathered} x^1(i,j;k,1) = \max[h^1(k), \{\theta^1(i,j-1;g,1)+1\}], \\ i = 1,\ldots,N;\ j = 1,\ldots,m_i; \\ k,g = 1,\ldots,K;\ l = 1,\ldots,L_k; \end{gathered} \quad (2)$$

$$\begin{gathered} x^1(i,j;k,l) = \\ = \max[h^1(k), \theta^1(i,j-1;g,q), \theta^1(p,q;k,l-1)] + 1, \\ i,p = 1,\ldots,N;\ j = 1,\ldots,m_i;\ k,g = 1,\ldots,N;\ l,q = 1,\ldots,L_k; \end{gathered} \quad (3)$$

$$x^r(i,1;k,1) =$$
$$= \max[h^r(k), \{\theta^{r-1}(i,m_i;g,l)+1\}], \quad (4)$$
$$i=1,\ldots,N;\ k,g=1,\ldots,K;$$
$$r=2,\ldots,R;\ l=1,\ldots,L_k;$$

$$x^r(i,1;k,l) =$$
$$= \max[\theta^{r-1}(i,m_i;g,q), \theta^r(p,j;k,l-1)]+1, \quad (5)$$
$$i=1,\ldots,N;\ j=1,\ldots,m_i;\ k,g=1,\ldots,K;$$
$$l,q=1,\ldots,L_k;\ r=2,\ldots,R;$$

$$x^r(i,j;k,1) =$$
$$= \max[h^r(k), \{\theta^r(i,j-1;g,l)+1\}], \quad (6)$$
$$i=1,\ldots,N;\ j=1,\ldots,m_i;\ k,g=1,\ldots,K;$$
$$l=1,\ldots,L_k; r=2,\ldots,R;$$

$$x^r(i,j;k,l) =$$
$$= \max[h^r(k), \theta^r(i,j-1;g,q), \theta^r(p,v;k,l-1)]+1, \quad (7)$$
$$i,p=1,\ldots,N;\ j,v=1,\ldots,m_i;\ k,g=1,\ldots,K;$$
$$l,q=1,\ldots,L_k; r=2,\ldots,R;$$

$$\theta^r(i,j;k,l) = x^r(i,j;k,l)+t(i,j;k)-1,$$
$$i=1,\ldots,N;\ j=1,\ldots,m_i;\ k=1,\ldots,K; \quad (8)$$
$$l=1,\ldots,L_k;\ r=1,\ldots,R.$$

Время завершения выполнения i-го задания на r-й стадии обработки определяется выражением

$$T^r(i) = \theta^r(i,m_i;k,l),$$
$$i=1,\ldots,N, r=1,\ldots,R \quad (9)$$

Время завершения выполнения всех операций на k-й машине на r-й стадии обработки равно

$$T^r(k) = \theta^r(i,j;k,L_k), k=1,\ldots,K, r=1,\ldots,R. \quad (10)$$

Допустимые времена выполнения всех операций каждого задания рассчитываются по формулам

$$T^R(i) = \theta^R(i,m_i;k,l) \leqslant D(i), i=1,\ldots,N; \quad (11)$$

$$T^r(k) = \theta^r(i,j;k,L_k) \leqslant H(k),$$
$$k=1,\ldots,K, r=1,\ldots,R. \quad (12)$$

Время завершения выполнения всех заданий определяется выражением

$$F = \max_{1\leqslant i\leqslant N} T(i). \quad (13)$$

2. Нижняя граница оптимального решения задачи

Рассмотрим нижнюю границу значения критерия оптимальности для одностадийных open-shop-problem.

Если $t(i,j;k,l)$ — время выполнения j-й операции i-го задания на k-й машине, то время завершения выполнения расписания не может быть меньше следующего значения:

$$\vartheta(T) =$$
$$= \max\left[\max_{1\leqslant i\leqslant N}\sum_{j=1}^{m_i} t(i,j;k,l), \max_{1\leqslant k\leqslant K}\sum_{l=1}^{L_k} t(i,j;k,l)\right]. \quad (14)$$

Если справедливо $\vartheta(T) = \max_{1\leqslant i\leqslant N}\sum_{j=1}^{m_i} t(i,j;k,l)$ для некоторого индекса $k = k_1$, то в job-shop-problem к данной величине еще необходимо добавить минимальное суммарное время выполнения всех операций на других машинах какого-либо из заданий, т. е. величину

$$\beta_{k1} = \min_{1\leqslant i\leqslant n}\sum_{k=1}^{K}\{t(i,j;k,l) \mid k \neq k_1\}. \quad (15)$$

Если справедливо $\vartheta(T) = \max_{1\leqslant k\leqslant K}\sum_{l=1}^{L_k} t(i,j;k,l)$ для некоторого индекса $i = i_1$, то к данной величине еще необходимо добавить минимальное суммарное время выполнения всех операций на какой-либо машине одного из заданий, т.е. величину

$$\beta_{k2} = \min_{1\leqslant k\leqslant K}\sum_{j=1}^{m_i}\{t(i,j;k,l) \mid i \neq i_1\}. \quad (16)$$

Следовательно,

$$\vartheta(T) = \max\left\{[\max_{1\leqslant i\leqslant N}\sum_{j=1}^{m_i} t(i,j;k,l)+\beta_{k1}],\right.$$
$$\left.\max_{1\leqslant k\leqslant K}[\sum_{l=1}^{L_k} t(i,j;k,l)+\beta_{k2}]\right\} \quad (17)$$

Пусть известны значения нижних границ длины выполнения всего комплекса работ на каждой стадии обработки $\vartheta(T^r)$, $r = 1, ..., R$, которые вычислены по формулам (14) или (15)—(17). Вычислим значения

$$B^r = \min_{1\leqslant i\leqslant N} B_i^r = \min_{1\leqslant i\leqslant N}\sum_{j=1}^{m_i} t^r(i,j;k), r=1,\ldots,R. \quad (18)$$

Так как перед началом и после завершения выполнения всех заданий на каждой стадии обработки необходимо, как минимум, завершение выполнений хотя бы данного задания на всех других стадиях обработки, значение нижней границы выполнения расписания многостадийной обработки вычисляется по формуле

$$\vartheta(F) = \max_{1 \leqslant r \leqslant R}\left[\vartheta(T^r) + \sum_{p=1}^{R}\{B^p \mid p \neq r\}\right]. \quad (19)$$

Если справедливо выражение $\vartheta(T) = \max_{1 \leqslant k \leqslant K}\sum_{l=1}^{L_k} t(i, j; k, l)$, и оно достигается для некоторого индекса $i = i_1$, то к данной величине необходимо добавить минимальное суммарное время выполнения всех операций на какой-либо машине одного из заданий, т. е. величину

$$\beta_{k2} = \min_{1 \leqslant k \leqslant K}\sum_{j=1}^{m_i}\{t(i, j; k, l) \mid i \neq i_1\}. \quad (20)$$

Следовательно,

$$\vartheta(T) = \max\left\{[\max_{1 \leqslant i \leqslant N}\sum_{j=1}^{m_i} t(i, j; k, l) + \beta_{k1}], \max_{1 \leqslant k \leqslant K}[\sum_{l=1}^{L_k} t(i, j; k, l) + \beta_{k2}]\right\}. \quad (21)$$

Пусть определены $\vartheta(T^r)$, $r = 1, ..., R$. Вычислим также величины

$$B^r = \min_{1 \leqslant i \leqslant N} B_i^r = \min_{1 \leqslant i \leqslant N}\sum_{j=1}^{m_i} t^r(i, j; k), r = 1, \ldots, R, \quad (22)$$

и другое, требующее меньшего объема вычислений выражение для определеня нижней границы длины расписания:

$$\vartheta(F) = \max_{1 \leqslant r \leqslant R}\left[\vartheta(T^r) + \sum_{p=1}^{R}\{B^p \mid p \neq r\}\right]. \quad (23)$$

3. Свойства допустимых и оптимальных расписаний

Утверждение 1. Если справедливо хотя бы одно из системы неравенств

$$d(i) + \sum_{r=1}^{R}\sum_{j=1}^{m_i} t^r(i, j, k) > D(i), i = 1, \ldots, N;$$
$$h(k) + \sum_{r=1}^{R}\sum_{i=1}^{N} t^r(i, j, k) > H(k), k = 1, \ldots, K, \quad (24)$$

то сформулированная задача не содержит допустимых решений.

Пусть в процессе решения задачи определены некоторые значения $x^r(i, j; k, l)$, и $\theta^r(i, j; k, l)$. Обозначим $\tilde{J}_i^{1r}, \tilde{J}_i^{1r}$ — соответственно подмножество выполненных и подлежащих выполнению операций i-го задания на r-й стадии обработки; $\tilde{Y}_k^{1r}, \tilde{Y}_k^{2r}$ — соответственно подмножество выполненных и подлежащих выполнению операций на k-й машине на r-й стадии обработки.

Утверждение 2. Если справедливо хотя бы одно из следующей ниже системы неравенств:

$$\theta^r(i, j; k, l) + \sum_{j \in \tilde{J}_i^{2k}} t^r(i, j; k, l) + \sum_{g=r+1}^{R}\sum_{j=1}^{m_i} t^g(i, j; k, l) > D(i), i = 1, \ldots, N; \quad (25)$$

$$\theta^r(i, j; k, l) + \sum_{l \in \tilde{Y}_k^{2r}} t^r(i, j; k, l) + \sum_{g=r+1}^{R}\sum_{l=1}^{L_k} t^g(i, j; k, l) > H(k), k = 1, \ldots, K, \quad (26)$$

то строящееся расписание не содержит допустимых решений и может быть исключено из рассмотрения.

Определим значение нижней границы длины выполняемого расписания.

Вычислим следующие величины:

$$\delta_1(F) = \max_{1 \leqslant i \leqslant N}\{\theta^r(i, j; k, l) + \sum_{j \in \tilde{J}_i^{2k}} t^r(i, j; k, l) + \sum_{g=r+1}^{R}\vartheta(T^r)\}; \quad (27)$$

$$\delta_2(F) = \max_{1 \leqslant k \leqslant K}\{\theta^r(i, j; k, l) + \sum_{l \in \tilde{Y}_k^{2r}} t^r(i, j; k, l) + \sum_{g=r+1}^{R}\vartheta(T^r)\}. \quad (28)$$

Тогда значение нижней границы определяется по формуле

$$\vartheta(F) = \max[\delta_1(F), \delta_2(F)]. \quad (29)$$

4. Алгоритм построения допустимых решений задачи

В целом ряде практических приложений нахождение допустимого решения задачи является достаточно важным и связано с определенными трудностями. Поэтому в качестве главной цели построения расписания будем рассматривать построение последовательностей выполнения операций всех заданий на всех стадиях обработки, обеспечивающих получение допустимого решения, удовлетворяющего всем ограничениям задачи. Эта задача также относится к классу *NP*-сложных проблем. Рассмотрим алгоритм приближенного решения этой задачи.

Алгоритм решения задачи состоит из некоторого числа итераций $s = 1, \ldots, S,$ на каждой из которых строится последовательность выполнения операций некоторых заданий на различных машинах определенных стадий обработки. Проводится расчет времен начала и завершения выполнения этих операций по формулам (1)—(8). Проверяется возможность выполнения системы ограничений, связанная с таким назначением, в соответствии с формулами (19)—(22).

Упорядочим все выполняемые задания в последовательность

$$\tilde{W} = \{i_1, i_2, \ldots, i_\gamma, i_{\gamma+1}, \ldots, i_N \mid D(i_\gamma) \leqslant D(i_{\gamma+1})\}. \quad (30)$$

Такая последовательность на s-м шаге итеративного процесса имеет вид

$$\tilde{W}^s = \{i_1^s, \ldots, i_\gamma^s, i_{\gamma+1}^s, \ldots, i_n^s \mid D(i_\gamma^s) \leqslant D(i_{\gamma+1}^s)\}, n \leqslant N. \quad (31)$$

Пусть $\tilde{I}_1^s, \tilde{I}_2^s$ — соответственно подмножества индексов заданий, для которых на s-м шаге алгоритма определены и еще не определены времена завершения их выполнения

$$\tilde{I}_1^s \cup \tilde{I}_2^s = \tilde{I} = \{i = 1, 2, \ldots, n \leqslant N\},$$
$$\tilde{I}_1^s \cap \tilde{I}_2^s = \varnothing, s = 1, \ldots, S.$$

В начале процесса положим $\tilde{I}_1^s = \varnothing, \tilde{I}_2^s = \tilde{I}.$ Вычислим значения следующих величин:

$$Z(i) = d(i) + \sum_{r=1}^{R} \sum_{j=1}^{m_i} t^r(i, j; k, l);$$
$$\Delta(i) = D(i) - Z(i), i = 1, \ldots, N; \quad (32)$$

$$\theta^s(i, p; k, l) = \max_{j \in \tilde{J}_i^{1s}} \theta^s(i, j; k, l); \quad (33)$$

$$Z^s(i) = \theta^s(i, p; k, l) + \sum_{r=\rho}^{R} \sum_{j \in \tilde{J}_i^{2r}} t^r(i, j; k, l),$$
$$\Delta^s(i) = D(i) - Z^s(i), i \in \tilde{I}_2^s, \rho \geqslant 1. \quad (34)$$

Значения $\Delta(i)$ и $\Delta^s(i)$ могут рассматриваться как граничные значения возможности выполнения ограничений на время завершения выполнения i-го задания в установленные сроки. Рассмотрим последовательности

$$\tilde{U} = \{i_1, i_2, \ldots, i_\gamma, i_{\gamma+1}, \ldots, i_N \mid \Delta(i_\gamma) \leqslant \Delta(i_{\gamma+1})\}, \quad (35)$$

а на s-м шаге итеративного процесса —

$$\tilde{U}^s = \{i_1^s, \ldots, i_\gamma^s, i_{\gamma+1}^s, \ldots, i_n^s \mid \Delta(i_\gamma^s) \leqslant \Delta(i_{\gamma+1}^s)\}, n \leqslant N. \quad (36)$$

Алгоритм предусматривает выполнение следующих шагов.

Шаг 1. Если $\tilde{I}_1^s = \tilde{I}, \tilde{I}_2^s = \varnothing,$ и выполняются все ограничения задачи (11) и (12), то получено допустимое решение задачи. Если какое-либо из этих ограничений не выполняется, то алгоритм завершает свою работу с сообщением, что не найдено допустимых решений задачи. В противном случае вычислим значения $Z(i)$ и $\Delta(i)$ по формулам (32). Если хотя бы для одного из индексов i = 1, ..., N выполняется неравенство $\Delta(i) < 0$, то не существует допустимых решений, и алгоритм завершает работу. В противном случае выполняем следующие вычисления.

Пусть $\tilde{K}^r$ — множество машин на r-й стадии обработки. Тогда

1) выбираем первые в множестве (35) $\gamma = \min(K^1, N)$ заданий — $(i_1, \ldots, i_\gamma)$;

2) определяем номер операции и номер машины, на которой она выполняется:

$$(i_1, k_1) =$$
$$= \arg \min_{1 \leqslant k \leqslant K^1} [\max\{h(k), d(i_1)\} + t(i_1, j; k, l)], \quad (37)$$

$$\tilde{K}_2^1 = (\tilde{K}^1 / k_1),\ \tilde{K}_1^1 = \{k_1\}; \quad (38)$$

3) вычисляем также последовательно для всех остальных заданий

$$\bar{j}(i_v, k_v) =$$
$$= \arg \min_{k \in \tilde{K}_{v-1}^1} [\max\{h(k), d(i_v)\} + t(i_v, j; k, l)], \quad (39)$$
$$\tilde{K}_v^1 \in (\tilde{K}_{v-1}^1 / k_v), v = 2, \ldots, \gamma.$$

Здесь $\bar{j}(i_v, k_v)$ — индексы выбранных операций и номера машины i_v-го задания на данном этапе обработки;

4) рассчитываем значения $x^1(i_v, 1; k, 1),$ $\theta^1(i_v, 1; k, 1),$ $v = 1, 2, \ldots, \gamma,$ по формулам (1);

5) определяем

$$\tilde{J}_{i_v}^{11} = \bar{j}(i_v, k_v),\ \tilde{J}_{i_v}^{21} = \{\tilde{J}_{i_v} / \tilde{J}(i_v, k_v)\};\ \tilde{I}_1^1 = \varnothing,$$
$$\tilde{I}_2^1 = \tilde{I};\ \tilde{K}_1^1 = \left\{\bigcup_{v=1}^{\mu} k_v\right\},\ \tilde{K}_2^1 = \{\tilde{K}_1 / \tilde{K}_1^1\}. \quad (40)$$

Полагаем $r = 2$ и переходим к выполнению шагов $s = 2,\dots,S$.

Шаг s.

1) определяем подмножество заданий $\tilde{U}^s$ по формулам (36). Выбираем стоящие первыми $\gamma^s = \min(K^s, N^s)$ заданий — $(i_1^s,\dots i_\nu^s,\dots,i_\gamma^s)$. Здесь $N^s \leqslant N$ — число заданий, время завершения выполнения которых на всех стадиях обработки еще не определено на данном шаге работы алгоритма;

2) рассчитываем время завершения выполнения этих заданий $\theta^{s,r}(i_\nu^s, j; k, l)$, $r = 1,\dots,R$, по формулам (1)—(8);

3) выбираем индекс задания с минимальным значением $\Delta(D_i)$ по формуле

$$\bar{i}^s = \min_{i \in \tilde{I}_2^s} \Delta(D_i).$$

4) полагаем $\tilde{I}_1^s = \{\tilde{I}_1^s \cup \bar{i}^s\}$, $\tilde{I}_2^s = \{\tilde{I}_2^s / \bar{i}^s\}$, $s := (s+1)$ и вновь переходим к выполнению шага 1.

Через некоторое число итераций либо будет получено допустимое решение задачи, либо установлено, что задача не имеет допустимых решений.

5. Алгоритм решения построения многостадийных расписаний в условиях отсутствия ограничений

Рассмотрим две стратегии выбора последовательности машин, на которых осуществляется выполнение задания u_i на каждой стадии обработки. На всех стадиях обработки $r = 1, 2, \dots, (R-1)$, кроме последней R-й стадии, выбор осуществляется согласно стратегии 1, а на последней — согласно стратегии 2. Пусть построена некоторая подпоследовательность выполнения заданий $\tilde{U} = \{u_1, u_2, \dots, u_{i-1}\}$. Рассмотрим последовательность выбора машин, на которых будет выполняться установленное на последнее место в $\tilde{U}$ задание u_i.

Стратегия 1. В этой стратегии на каждом шаге осуществляется выбор той машины, на которой будет достигнуто минимальное время завершения выполнения этого задания. Пусть определены следующие подмножества: $\tilde{K}_1^r(u_i) = \{k_1^r, k_2^r, \dots, k_{p-1}^r\}$, $\tilde{K}_2^r(u_i) = \{k_p^r, k_{p+1}^r, \dots, k_{K^r}^r\}$ — подмножества машин r-й стадии обработки, на которых, соответственно, выполнены операции u_i-го задания и других машин, на которых еще должны быть выполнены операции этого u_i-го задания.

Пусть K^r — число машин на r-й стадии обработки; $T^r(u_i)$ — время завершения выполнения задания на всех машинах r-й стадии обработки.

Если $\tilde{K}_2^r(u_i) = \varnothing$, то выбор машины, стоящей на первом месте в последовательности $k_1 \in \tilde{K}_2^r(u_i)$ осуществляется из условий

$$g = \arg\min_{k_1 \in \tilde{K}_2^r(u_i)} \{\max[d_{u_i}, h_{k_1}^1] + t_{u_i,k_1}^1\}, \tag{41}$$

$$g = \arg\min_{k_p \in \tilde{K}_2^r(u_i)} \{\max[\theta_{u_1,k_{p-1}}^r, h_{k_p}^r] + t_{u_i,k_p}^r\}, \quad r = 2, \dots, R-1. \tag{42}$$

Если $\tilde{K}_2^r(u_i) \neq \varnothing$, то для всех остальных машин $k_p \in \tilde{K}_2^r(u_i)$ этот выбор происходит из условий

$$g = \arg\min_{k_1 \in \tilde{K}_2^r(u_i)} \{\max[\theta_{u_i,k_{p-1}}^r, \theta_{u_{i-1},k_p}^r] + t_{u_i,k_1}^r\}. \tag{43}$$

Если $\tilde{K}_2^{r-1}(u_i) \neq \varnothing$ и $\tilde{K}_2^r(u_i) \neq \varnothing$, то выбор первой машины в последовательности $k_p \in \tilde{K}_2^r(u_i)$ осуществляем по правилам

$$g = \arg\min_{k_p \in \tilde{K}_2^r(u_i)} \{\max[\theta_{u_{i-1},k_p}^r, T^{r-1}(u_i)] + t_{u_i,k_p}^r\}, \quad r = 2, \dots, R-1. \tag{44}$$

Каждый раз после выбора g-й машины выполняем преобразование

$$\tilde{K}_1^r(u_i) = \{\tilde{K}_1^r(u_i) \cup g\}, \tilde{K}_2^r(u_i) = \{\tilde{K}_2^r(u_i) / g\}. \tag{45}$$

Если $\tilde{K}_2^{R-1}(u_i) = \varnothing$, то алгоритм реализации стратегии 1 определения последовательности машин, на которых выполняется задание u_i на первых $(R-1)$ стадиях обработки, завершает свою работу.

Стратегия 2. На последней стадии обработки последовательность машин выбирается из условий минимизации значения нижней границы критерия оптимальности. Для выбранного решения значение нижней границы вычисляется по упрощенной формуле (23).

Пусть построена $\tilde{K}^R(u_i) = \{k_1^R, k_2^R, \dots, k_{K^R}^R\}$ — последовательность машин, на которых осуществлено выполнение u_i-го задания на последней R-й стадии обработки. Известны $\tilde{I}_1(u_i)$ — подмножество заданий, стоящих в строящейся последовательности их выполнения перед заданием u_i; $\tilde{I}_2(u_i) = \{\tilde{I} / \tilde{I}_1(u_i)\}$ — подмножество заданий, подлежащих выполнению.

Времена начала и завершения выполнения заданий на каждой машине для этой последовательности вычисляются по формулам (41)—

(44) при $r := R$. Выбор последовательности машин $\tilde{K}^R(u_i)$ проводится из условий

$$\vartheta[F(u_i)] = \min \max_{k_p \in \tilde{K}^R} \left\{ (\theta^R_{u_i,k_p} + t^R_{u_i,k_p}) + \sum_{j \in \tilde{I}_2(u_i)} t^R_{j,k_p} \right\}. \quad (46)$$

Решение этой задачи часто тривиально, выполняется эвристическим алгоритмом, а при небольшом числе машин — полным или частичным перебором.

Алгоритм 3 — приближенное решение задачи.

Алгоритм построения последовательности выполнения заданий состоит из N шагов $s = 1, 2, \ldots, N-1$. На каждом шаге на последнее место в строящейся последовательности $\tilde{U}$ устанавливается одно задание из подмножества $\tilde{I}_2^s$. В начале процесса $s = 0$ положим $\tilde{I}_1^s = \tilde{U}^s = \varnothing$, $\tilde{I}_2^s = \tilde{I}$.

Шаг 1. Проверяем после установки каждого из заданий на 1-е место в строящейся последовательности $\tilde{U}^1$ эффективность каждой из альтернатив. Для этого рассчитываем для каждого из заданий по формулам (1)—(9) значения величин

$$x^1(i,1;k,1), x^r(i,1;k,l), x^r(i,j;k,l),$$
$$r = 2,\ldots,R;$$
$$\theta^r(i,j;k,l) = x^r(i,j;k,l) + t(i,j;k) - 1,$$
$$r = 1,\ldots,R.$$

Вычисляем индекс задания, для которого достигается минимальное значение нижней границы оптимального решения задачи по формуле

$$u_1 = \arg\min_{i \in \tilde{I}} \left[\theta^R(i,j;k) + \sum_{p \in \{\tilde{I}/i\}} t^R(p,j;k) \right]. \quad (47)$$

Задание с индексом $\bar{i}$ устанавливаем на 1-е место в строящейся последовательности $\tilde{U}^1 = \{u_1\}$, $\tilde{I}_1^1 = \{u_1\}$, $\tilde{I}_2^1 = \{\tilde{I}/u_1\}$. Переходим к выполнению шага *s*_1. Для каждого из заданий $s = 2,\ldots,N-1$ подмножества $\tilde{I}_2^s$ выполняем последовательно вычисления, описанные в шаге *s*_1 и шаге *s*_2. После выбора соответствующего задания на последнее место в строящейся последовательности проводим корректировку последовательности $\tilde{U}^s$ и корректировку подмножеств $\tilde{I}_1^s$ и $\tilde{I}_2^s$.

Шаг s_1. Если $\tilde{I}_2^s = \varnothing$, то построено расписание выполнения заданий, и алгоритм построения последовательности завершает работу. Переходим к выполнению последнего шага *S*. В противном случае для каждого из заданий $i \in \tilde{I}_2^s$ для стадий обработки $r = 1, 2, \ldots, R-1$ проводим вычисления в соответствии со стратегией 1. Переходим к выполнению шага *s*_2.

Шаг s_2. На последней стадии обработки проводим вычисления в соответствии со стратегией 2. В результате этих вычислений будет осуществлен выбор задания $u_i \in \tilde{I}_2^s$, которое устанавливается на последнее место в строящейся последовательности $\tilde{U}^s$. Выполняем корректировку подмножеств $\tilde{I}_1^s$ и $\tilde{I}_2^s$, полагаем $s := (s+1)$, и переходим к выполнению шага *s*_1.

Шаг S. Построена оптимальная последовательность выполнения заданий $\tilde{U}^N$. Времена начала и завершения выполнения всех заданий, следующих друг за другом в последовательности $\tilde{U}^N$, вычислим по формулам (1)—(9). Время завершения расписания выполнения всех заданий (значение критерия оптимальности), которое для построенного расписания равно нижней границе оптимального решения, определяется выражением $F = \max_{i \in \tilde{I}} T_i^R$. На этом алгоритм завершает свою работу.

6. Иллюстративный пример

Решим с использованием алгоритма 1 задачу построения многостадийного расписания open-shop-problem в условиях отсутствия ограничений на времена начала и завершения выполнения заданий. Времена выполнения заданий машинами трех стадий обработки приведены в табл. 1.

В табл. 2 представлены суммарные времена выполнения заданий всеми машинами каждой стадии обработки.

Таблица 1

Времена выполнения заданий трех стадий обработки

№ заданий	Времена выполнения заданий машинами трех стадий обработки								
	Машины 1-й стадии			Машины 2-й стадии			Машины 3-й стадии		
	1	2	3	1	2	3	1	2	3
1	7	6	9	3	8	10	5	12	10
2	8	7	3	4	12	8	3	15	6
3	10	4	7	9	7	6	8	9	4
4	5	10	5	3	11	5	6	10	8
5	3	6	12	9	7	8	7	7	10
Сумма	33	33	36	28	45	37	31	53	38

Таблица 2

Суммарные времена выполнения заданий

№ заданий	Суммарные времена выполнения заданий всех стадий обработки			
	1	2	3	Суммы всех трех стадий
1	22	21	27	70
2	18	24	24	66
3	21	21	21	63
4	20	19	24	63
5	21	24	24	69

Вычислим по формулам (22), (23) нижнюю границу длины многостадийного расписания.

$$\vartheta(T_1) = \max\{(22 + 3 + 7 + 5 + 12), (18 + 9 + 7 + 5 + 12), (21 + 9 + 3 + 5 + 12), (20 + 9 + 3 + 7 + 12), (21 + 9 + 3 + 7 + 5), (33 + 6 + 5), (33 + 10), (36 + 9)\} = 51;$$

$$\vartheta(T_2) = \max\{(21 + 8 + 6 + 5 + 8), (24 + 10 + 6 + 5 + 8), (21 + 10 + 8 + 5 + 8), (19 + 10 + 8 + 5 + 8), (24 + 10 + 8 + 6 + 5), (28 + 7 + 6), (45 + 3 + 5), (37 + 3 + 8)\} = 53;$$

$$\vartheta(T_3) = \max\{(27 + 6 + 4 + 8 + 10), (24 + 10 + 4 + 8 + 10), (21 + 10 + 6 + 8 + 10), (24 + 10 + 6 + 4 + 10), (24 + 10 + 6 + 4 + 8), (31 + 9 + 4), (53 + 3 + 6), (38 + 7 + 7)\} = 62;$$

$$\vartheta(F) = \max\{(51 + 21 + 21), (53 + 21 + 21), (62 + 21 + 21)\} = 104.$$

Решение задачи алгоритмом 1, включающим пять шагов, приводится ниже.

Шаг 1. Рассчитываем наиболее ранние времена выполнения каждого из заданий и соответствующие значения нижней границы. Результаты вычислений приведены в табл. 3. На первое место в строящейся последовательности выбираем четвертое задание.

Шаг 2. Выбор задания на второе место в последовательности. Времена выполнения заданий на втором шаге алгоритма приведены в табл. 4. Выбираем задание 1 и устанавливаем его на второе место в строящейся последовательности.

Шаг 3. Выбор задания на третье место в последовательности. Времена выполнения заданий на третьем шаге алгоритма приведены в табл. 5. Выбираем пятое задание на третье место в строящейся последовательности.

Таблица 3

Времена выполнения заданий на первом шаге алгоритма

№ заданий	Времена завершения выполнения заданий машинами трех стадий обработки									Значение нижней границы
	Машины 1-й стадии			Машины 2-й стадии			Машины 3-й стадии			
	1	2	3	1	2	3	1	2	3	
1	13	6	22	25	33	43	60	55	70	98
2	18	10	3	22	42	30	60	57	66	98
3	21	4	11	43	34	27	64	52	56	96
4	5	20	10	23	39	28	45	55	65	96
5	3	9	21	45	28	36	79	69	62	115

Таблица 4

Времена завершения выполнения заданий на втором шаге алгоритма

№ заданий	Времена завершения выполнения заданий машинами трех стадий обработки									Значение нижней границы
	Машины 1-й стадии			Машины 2-й стадии			Машины 3-й стадии			
	1	2	3	1	2	3	1	2	3	
1	12	27	21	30	46	38	51	67	77	98
2	13	27	16	31	51	39	54	70	83	107
3	15	26	22	43	50	34	58	67	71	107
5	8	28	22	37	52	45	57	64	75	100

Таблица 5

Времена завершения выполнения заданий на третьем шаге алгоритма

№ заданий	Времена завершения выполнения заданий машинами трех стадий обработки									Значение нижней границы
	Машины 1-й стадии			Машины 2-й стадии			Машины 3-й стадии			
	1	2	3	1	2	3	1	2	3	
2	20	34	24	38	58	46	61	82	88	102
3	22	32	28	41	54	47	62	76	81	104
5	15	39	33	48	55	63	70	77	87	101

Шаг 4. Выбор задания на четвертое и пятое места в последовательности. Времена выполнения заданий на четвертом шаге алгоритма приведены в табл. 6.

Следовательно, на четвертое место в последовательности ставится третье задание, а на пятое — задание 2.

Времена начала и завершения выполнения заданий в построенной последовательности

Таблица 6

Времена завершения выполнения заданий на четвертом шаге алгоритма

№ заданий	Времена завершения выполнения заданий машинами трех стадий обработки									Значение нижней границы
	Машины 1-й стадии			Машины 2-й стадии			Машины 3-й стадии			
	1	2	3	1	2	3	1	2	3	
2	23	43	36	52	67	75	78	93	99	103
3	25	44	40	53	62	68	76	86	91	101

Таблица 7

Результаты решения задачи open-shop-problem

№ заданий	Времена завершения выполнения заданий машинами трех стадий обработки									Значение нижней границы
	Машины 1-й стадии			Машины 2-й стадии			Машины 3-й стадии			
	1	2	3	1	2	3	1	2	3	
4	5	20	10	23	39	28	45	55	65	96
1	12	27	21	30	46	38	51	67	77	98
5	16	39	33	48	55	63	70	77	87	101
3	25	44	40	53	62	68	76	86	91	101
2	33	51	43	55	67	75	79	100	106	106

$\tilde{U} = \{4, 1, 5, 3, 2\}$ на машинах всех стадий обработки приведены в табл. 7.

Для сравнения в табл. 8 приведено решение задачи flow-shop-problem в условиях строго фиксированных и одинаковых для всех заданий на всех стадиях обработки последовательностей выполнения всех операций.

Таблица 8

Результаты решения задачи многостадийной job-shop-problem

№ заданий	Времена завершения выполнения заданий машинами трех стадий обработки								
	Машины 1-й стадии			Машины 2-й стадии			Машины 3-й стадии		
	1	2	3	1	2	3	1	2	3
4	5	15	20	23	34	39	45	55	63
3	15	19	27	36	43	49	57	66	70
2	23	30	33	40	55	63	66	81	87
5	26	36	48	57	64	72	79	86	96
1	33	42	57	60	68	82	87	98	108

Заключение

Рассматривается постановка задачи теории расписаний для дискретных производств в условиях, когда множество изделий должно пройти последовательную обработку в R структурных подразделениях предприятия (например, последовательная цепочка, несколько участков и (или) цехов с различным числом и отличающимся своими техническими характеристиками оборудованием). Каждое такое структурное подразделение рассматривается как отдельная стадия производства. Такими стадиями обработки могут быть группы станков, многопроцессорные, многопоточные или автоматические линии. На каждой стадии производства (обработки изделий) очередность выполнения технологических операций может осуществляться в произвольной последовательности (open-shop-problem). Необходимо найти последовательность запуска в производство этого множества изделий, обеспечивающую выполнение всех ресурсных ограничений и оптимальное значение некоторого критерия оптимальности.

Предложены математические модели, установлены свойства допустимых и оптимальных расписаний, для выбранной последовательности запуска изделий приведены формулы расчета времен начала и завершения выполнения операций для всех структурных подразделений производства, а также значение нижней границы критерия оптимального решения для анализируемых подпоследовательностей на всех этапах решения.

Сформулированные задачи относятся к классу NP-трудных проблем экспоненциальной сложности. Разработаны алгоритмы приближенного решения этих задач полиномиальной сложности, использующие разные эвристики на различных этапах решения, которые проиллюстрированы на числовом примере.

Эффективность предложенных алгоритмов приемлема для практических приложений. Полученные в работе результаты могут использоваться при проектировании систем календарного планирования и организации работы единичного, серийного и массового производства в различных отраслях промышленности.

Список литературы

1. **Конвей Р. В., Максвелл В. Л., Миллер Л. В.** Теория расписаний. М.: Физматгиз, Наука. 1975. 359 с.
2. **Танаев В. С., Сотсков Ю. Н., Струсевич В. А.** Теория расписаний. Многостадийные системы. М.: URSS, 1989. 328 с.
3. **Зак Ю. А.** Прикладные задачи теории расписаний и маршрутизации перевозок. М.: URSS, 2012. 394 с.

4. **Лазарев А. А.** Теория расписаний. Методы и алгоритмы. М.: ИПУ РАН, 2019. 408 с.
5. **Лазарев А. А., Графов Е. Р.** Теория расписаний. Задачи и алгоритмы. М.: Изд-во МГУ, 2011. 224 с.
6. **Пяткин А. В., Черных И. Д.** Задача open-shop с маршрутизацией двухвершинной сети и разрешением прерываний // Дискретный анализ и исследование операций. 2012. Т. 19, № 3. С. 65—78.
7. **Brucker P.** Scheduling Algorithms. Berlin, Heidelberg und New York. Springer-Verlag, 1998.
8. **Domschke W., Scholl A., Voß S.** Produktionsplanung. Ablauforganisatorische Aspekte. Berlin, Heidelberg, Springer Verlag, 2005. 456 p.
9. **Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G.** Minimizing maximum lateness in two-machine open shop // Mathematics of Operations Research. 1981. N. 6. P. 153—158.
10. **Achubgue J. O., Chin F. Y.** Scheduling the open-shop to minimize mean flow time // SIAM Journal on Computing. 1982. N. 11. P. 709—720.
11. **Du J., Long J.** Minimizing mean flow time in two machine open shops and flow shops // Journal of Algorithms. 1993. 14. P. 24—44.
12. **Averbakh I., Berman O., Chernykh I.** The routing open-shop problem ona network: complexity and approximation // Eur. J. Oper. Res. 2006. Vol. 173, N. 2. P. 531—539.
13. **Gonzalez T., Sahni S.** Open shop scheduling to minimize finish time // J. ACM. 1976. Vol. 23, N. 4. P. 665—679.
14. **Williamson D. P., Hall L. A., Hoogeveen J. A., Hurkens C. A. J.,Lenstra J. K., Sevastianov S. V., Shmoys D. B.** Short shop schedules // Oper. Res. 1997. Vol. 45, N. 2, P. 288—294.

Yu. A. Zack, D. Sc., e-mail: yuriy_zack@hotmail.com,
Aachen, Germany

Multistage Open-Shop-Problem: Schedules for N Tasks on K Machines with Different Technical Characteristics for an Arbitrary order of Tasks

The statements and mathematical models of the open — shop-problem are considered: performing N tasks on K machines with an arbitrary order of processing each task on all machines. Unlike most publications on solving this problem, the task posed is solved in a multi-stage formulation under the conditions of a sequential chain of sections and workshops of the enterprise, as well as in the presence of restrictions on the start and end times of tasks and the permissible operating times of machines. In addition, for scheduling systems of enterprises, it is extremely important to consider this task in a multi-stage setting, i.e. in a sequential chain of plots and workshops. The properties of admissible and optimal schedules, as well as algorithms of exact and approximate methods for solving these problems by successive optimization algorithms are investigated. At the early stages of solving the problem, the fact of incompatibility of the system of constraints of the task and the definition of a subset of tasks or resources, the time range of which should be expanded, is established. The described algorithms for solving the problem are illustrated by numerical examples.

Keywords: *multi-stage schedules, open — shop-problem, restrictions on the execution time of tasks, lower bound on the length of the schedule, sequential optimization algorithms, heuristic algorithm*

DOI: 10.17587/it.27.249-258

References

1. **Konwej R. W., Maxwell W. L., Müller W. L.** Scheduling theory, Moscow, Fismatgis, Nauka, 1975, 359 p. (in Russian).
2. **Tanajev V. S., Sotskov Yu. N., Strusevitch W. A.** Schedule theory. Multistage systems, Moscow, URSS, 1989, 328 p. (in Russian).
3. **Zack Yu. A.** Applied problems of the theory of schedules and traffic routing, Moscow, URSS, 2012, 394 p. (in Russian).
4. **Lazarev A. A.** Schedule theory. Methods and Algorithms, Moscow, IPU RAN, 2019, 408 p. (in Russian).
5. **Lazarev A. A., Grafov E. R.** Schedule theory. Tasks and algorithms, Moscow, Publishing house of MGU, 2011, 224 p. (in Russian).
6. **Pyatkin A. V., Chernich I. D.** Open-shop task with two-node routing and interrupt resolution, *Discrete Analysis and Operations Research,* 2012, vol. 19, no. 3, pp. 65—78 (in Russian).
7. **Brucker P.** Scheduling Algorithms, Berlin, Heidelberg und New York, Springer-Verlag, 1998.
8. **Domschke W., Scholl A., Vo@ S.** Produktionsplanung. Ablauforganisatorische Aspekte, Berlin, Heidelberg, Springer Verlag, 2005, 456 p.
9. **Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G.** Minimizing maximum lateness in two-machine open shop, *Mathematics of Operations Research*, 1981, no. 6, pp. 153—158.
10. **Achubgue J. O., Chin F. Y.** Scheduling the open-shop to minimize mean flow time, *SIAM Journal on Computing*, 1982, no. 11, pp. 709—720.
11. **Du J., Long J.** Minimizing mean flow time in two machine open shops and flow shops, *Journal of Algorithms*, 1993, vol. 14, pp. 24—44.
12. **Averbakh I., Berman O., Chernykh I.** The routing open-shop problem ona network: complexity and approximation, *Eur. J. Oper. Res.*, 2006, vol. 173, no. 2, pp. 531—539.
13. **Gonzalez T., Sahni S.** Open shop scheduling to minimize finish time, *J. ACM,* 1976, vol. 23, no. 4, pp. 665—679.
14. **Williamson D. P., Hall L. A., Hoogeveen J. A., Hurkens C. A. J., Lenstra J. K., Sevastianov S. V., Shmoys D. B.** Short shop schedules, *Oper. Res.*, 1997, vol. 45, no. 2, pp. 288—294.