УДК 519.2

Ю.А. ЗАК

ПОСТРОЕНИЕ ДОПУСТИМЫХ И ОПТИМАЛЬНЫХ РАСПИСАНИЙ ВЫПОЛНЕНИЯ РАБОТ НА ОДНОЙ МАШИНЕ

Ключевые слова: *оптимальное расписание, последовательности выполнения заданий, ограничения на время выполнения работ, последовательные алгоритмы оптимизации.*

ВВЕДЕНИЕ

В монографиях и периодических изданиях по теории расписаний достаточно много внимания уделяется задаче определения оптимальной последовательности выполнения n заданий на одной машине. В рассматриваемой задаче заданы время выполнения заданий t_i, ограничения на начальные сроки их выполнения a_i, а также заключительные времена g_i, необходимые для завершения выполнения каждого задания после окончания его обработки на данной машине. Все значения t_i, a_i, g_i, $i=1,\ldots,n$, — целые числа. Необходимо найти оптимальную последовательность L обработки всех заданий на этой машине, а также время начала x_i и завершения σ_i выполнения каждого из заданий. Оптимальное расписание должно обеспечить выполнение всех сформулированных выше ограничений и минимизировать сроки завершения наиболее позднего из всех заданий с учетом добавления заключительного времени обработки каждого из этих заданий на других машинах — g_i. Для сформулированной, как показано в [1], NP-сложной проблемы предложены достаточно эффективные алгоритмы решения (см., например, [2–10]), позволяющие находить точные и приближенные решения практических задач достаточно большой размерности. Эффективные алгоритмы точного решения задачи с помощью Schrage-algorithms, и его модификации впервые были предложены J. Carlier (1982) [5] и развиты в работах [6, 7]. Эти алгоритмы наиболее часто применяются в настоящее время для получения точных решений практических задач большой размерности. Однако для практических приложений планирования производства, организации обслуживания и вычислительного процесса чрезвычайно актуальна постановка задачи построения допустимого расписания (последовательности) выполнения работ на одной машине, в которой учитываются ограничения на сроки завершения каждого из заданий d_i, $i=1,\ldots,n$, а также некоторые фиксированные порядки, заданные условиями предшествования отдельных подмножеств заданий перед другими. При этом предполагается выполнение дополнительных условий:

а) ни одно из заданий не может прерываться в процессе своего выполнения и одновременно на машине не может выполняться больше одного задания.

Рассматриваемая задача, как и известная в литературе «One-machine sequencing problem» [1–10], также относится к классу NP-сложных проблем. В условиях этих дополнительных ограничений в настоящее время автору не известны эффективные алгоритмы решения сформулированных задач, позволяющие с таким же объемом вычислений решать задачи большой размерности, как и методы [5–7].

В настоящей работе предлагается другой, отличный от рассматриваемого в [5–7] подход к решению сформулированной задачи и показано, что наличие ограничений в ряде случаев позволяет быстро отсечь подмножество расписаний, не удовлетворяющих установленной системе ограничений. Оптимальное реше-

ние задачи как при наличии этих дополнительных ограничений, так и при отсутствии их можно получить предлагаемым методом с эффективностью, сравнимой с эффективностью методов J. Carlier [5].

При отсутствии допустимых расписаний на начальных этапах решения задачи устанавливается факт несовместности заданной системы ограничений и определяется одно из условий невязки.

1. ПОСТАНОВКА ЗАДАЧИ

Пусть заданы условия предшествования выполнения заданий в виде некоторого графа или совокупности деревьев, которые могут быть описаны, например, соотношениями

$$i_l^{\xi} \Rightarrow \{j_p^{\xi}, j_{p+1}^{\xi}, \ldots, j_P^{\xi}\},\ \xi = 1, \ldots, \Xi. \tag{1}$$

Это означает, что задание i_l^{ξ} должно быть выполнено раньше и стоять в допустимой последовательности впереди каждого из заданий подмножества $\{j_p^{\xi}, j_{p+1}^{\xi}, \ldots, j_P^{\xi}\}$. Количество таких соотношений равно Ξ.

Тогда поставленная задача математически формулируется следующим образом. Необходимо найти оптимальную последовательность L обработки всех заданий на машине, удовлетворяющую условиям предшествования (1), а также время начала x_i и завершения выполнения каждого из заданий σ_i, $i = 1, \ldots, n$, обеспечивающие выполнение всех директивных сроков на время начала a_i и завершения выполнения d_i, т.е. необходимо при выполнении условий (1) и а) найти решение следующей системы неравенств:

$$a_i \le x_i,\ d_i \ge x_i + t_i,\ \sigma_i - x_i = t_i,\ i = 1, \ldots, n; \tag{2}$$

$$x_i + t_i > x_j \ \text{ или } \ x_j + t_j > x_i,\ i \ne j,\ i = 1, \ldots, n. \tag{3}$$

В качестве критерия оптимальности расписания, как и в работах [1–10], может рассматриваться функционал вида

$$\max_i (\sigma_i + g_i) \to \min. \tag{4}$$

Даже при отсутствии условий непрерывности выполнения каждого из заданий наличие условий (1) и альтернативных ограничений (3), которые могут быть математически строго сформулированы только введением булевых переменных, задача поиска допустимого расписания носит комбинаторный характер и требует специальных методов решения.

В работе на основе установленных свойств допустимых расписаний рассматривается алгоритм построения допустимых расписаний, удовлетворяющих условиям (1), (2) и а). На основе введенных возможных наиболее ранних λ_i и допустимых самых поздних сроков τ_i начала выполнения каждого из заданий устанавливаются свойства допустимых расписаний и строятся алгоритмы выделения подмножеств расписаний, не удовлетворяющих установленной системе ограничений. В случае несовместности системы ограничений задачи (1)–(3) на начальных шагах работы алгоритма устанавливается факт несовместности и определяется одно из условий невязки. На основе установленных свойств строится полиномиальный алгоритм наиболее эффективного (в смысле выполнения всех ограничений задачи) расписания R, допускающего разрывы во времени выполнения заданий. Если в процессе построения устанавливается факт нарушения директивного срока завершения какого-либо задания $\sigma_i > d_i$, то из этого следует, что система ограничений задачи несовместна. Величина $(\sigma_i - d_i)$ определяет количество интервалов, на которое необходимо увеличить срок завершения выполнения i-го задания, чтобы система ограничений задачи могла стать совместной.

Рассматриваемый вначале алгоритм (алгоритм D) определения допустимого решения задачи (задачи (2), (3) без учета ограничений на частичный порядок выполнения заданий (1)) с учетом условий а) предусматривает ветвящийся процесс использования полиномиальных алгоритмов построения наиболее эффективных расписаний, не допускающих разрывов выполнения заданий. Если в процессе построения для какого-либо задания установится факт $\sigma_i > d_i$, то построение этого расписания прерывается и осуществляется переход к следующей наиболее эффективной из оставшихся альтернатив. Первое построенное до конца расписание является решением задачи. Рассмотрение и исключение всех возможных альтернатив без построения хотя бы одного расписания до конца служит доказательством несовместности системы ограничений. При этом, если $d_i < x_i + t_i$, определяется то минимальное количество интервалов, на которое необходимо увеличить срок завершения выполнения i-го задания, чтобы система ограничений задачи могла стать совместной.

Алгоритм начинает построение с наиболее перспективных альтернатив и проводит отсев всех неперспективных продолжений. Благодаря этому, как показали обширные вычислительные эксперименты, в большинстве рассматриваемых практических случаев уже развитие первой или нескольких первых альтернатив приводит к построению допустимого расписания, и это достигается за полиномиальное время при относительно небольшом объеме вычислений на каждой итерации.

Показано, как на основе соотношений предшествования (1) скорректировать граничные значения a_i и d_i для отдельных заданий; следовательно, и задача (1)–(3) в условиях ограничений на частичный порядок выполнения заданий (1) также эффективно решена алгоритмом D. Работа и эффективность предложенных алгоритмов иллюстрируется на числовых примерах.

2. СВОЙСТВА ДОПУСТИМЫХ РАСПИСАНИЙ

2.1. Оценки возможности выполнения системы ограничений на сроки выполнения заданий. Без ограничения общности будем полагать, что для всех заданий выполняются условия $a_i + t_i \le d_i,\ i \in I$.

Упорядочим все задания в последовательность J по неубыванию значений d_i: $d_{i_1} \le d_{i_2} \le \ldots d_{i_l} \le \ldots \le d_{i_{n-1}} \le d_{i_n}$. Пусть $\theta_0 = \min\limits_{1 \le i \le n} a_i$ — наиболее ранний срок начала выполнения расписания. Рассмотрим все задания в порядке их расположения в последовательности

$$J = \{i_1, i_2, \ldots, i_k, i_{k+1}, \ldots, i_n : d_{i_k} \le d_{i_{k+1}}\}. \tag{5}$$

Утверждение 1. Если хотя бы для одного из заданий $i_p \in J$ несправедливы условия неравенства

$$\max\left(\min_{1 \le l \le p} a_{i_l}, \theta_0\right) + \sum_{l=1}^{p} t_{i_l} \le d_{i_p},\ p = 1, \ldots, n, \tag{6}$$

то не существует расписаний, удовлетворяющих системе ограничений задачи.

В работе [11] приведено доказательство отсутствия допустимых расписаний в случае невыполнения хотя бы одного из неравенств

$$\theta_0 + \sum_{l=1}^{p} t_{i_l} \le d_{i_p},\ p = 1, \ldots, n, \tag{7}$$

т.е. для случая равных граничных сроков начала выполнения всех заданий.

Система неравенств (6) может быть представлена в виде (7) и дополнительных условий

$$a_{i_p} + t_{i_p} \le d_{i_p}, \ \ p = 1, \ldots, n. \tag{8}$$

В случае выполнения условия (7) и невыполнения (8) для некоторого индекса i_p следует, что для этого задания выполнение ограничений (6) возможно только в том случае, если начало выполнения его будет раньше момента времени a_{i_p}, что противоречит условиям задачи.

Рассмотрим итеративный процесс последовательного включения в последовательность некоторого из заданий в условиях завершения выполнения его до конца без прерываний. Пусть построен некоторый частичный план P^s, в котором определены подпоследовательность и сроки начала и окончания выпонения некоторого подмножества I_1^s, включающего b^s выполненных заданий. Пусть $\overline{\sigma}^s$ — срок завершения выполнения частичного плана P^s. Второе подмножество I_2^s включает те задания, выполнение которых еще не начато к моменту времени $\overline{\sigma}^s$. Определим

$$\overline{\theta}^s = \max \Big(\min_{\substack{1 \le l \le n - b^s \\ i_l \in I_2^s}} a_{i_l}, \overline{\sigma}^s \Big). \tag{9}$$

Упорядочим все задания подмножества I_2^s в порядке невозрастания величин d_i: $d_{i_1} \le d_{i_2} \le \ldots d_{i_l} \le \ldots \le d_{i_{n-b^s}}$.

Утверждение 2. Если хотя бы для одного из заданий $i_l \in I_2^s$ справедливы условия неравенства

$$\overline{\theta}^s + \sum_{l=1}^{p} t_{i_l} > d_{i_p}, \ \ p = 1, \ldots, n - b^s, \tag{10}$$

то частичный план P^s не содержит расписаний, удовлетворяющих системе ограничений задачи.

Доказательство утверждения 2 непосредственно следует из справедливости утверждения 1 и определяет оптимальную стратегию на каждом шаге построения допустимого расписания, разрешающего разрывы выполнения заданий. Эта стратегия заключается в выполнении в каждый момент времени t такого задания $i \in I_2^s$, у которого допустимое время начала $a_i \ge t$ и граничный срок завершения задания d_i минимальный среди всех, разрешенных для выполнения по первым двум признакам заданий. Если такой выбор неоднозначен, то из наиболее перспективных заданий выбирается задание с наименьшей длительностью t_i, а затем — с наименьшим индексом. Эта стратегия реализуется в приведенном ниже алгоритме D.

Построение расписаний, не допускающих разрывов в выполнении заданий, фактически сводится к определению некоторого полного порядка (последовательности $\widetilde{W} = \{i_1, i_2, \ldots, i_p, \ldots, i_n\}$) выполнения заданий, для которой сроки начала и окончания выполнения легко вычисляются по рекуррентным формулам:

$$\sigma_0 = 0, \ x_{i_p} = \max (a_{i_p}, \sigma_{i_{p-1}}), \ \sigma_{i_p} = x_{i_p} + t_{i_p}, \ \ p = 1, \ldots, n. \tag{11}$$

Далее рассматриваются некоторые особенности алгоритмов построения допустимых последовательностей выполнения заданий с учетом условий а), т.е. не допускающих разрывов в процессе их выполнения.

Рассмотрим некоторый шаг алгоритма, выполняемый на τ-м временном интервале, т.е. определение k-го члена в последовательности. Рассмотрим частич-

ный план (развиваемую альтернативу) P^s. Обозначим $\tau = \overline{\theta}^s$. Если в подпоследовательность не включено ни одно задание, то $\tau = \theta_0$.

Пусть все $m^s = n - b^s$ заданий подмножества I_2^s упорядочены в последовательность

$$J^s = \{j_1^s, j_2^s, ..., j_l^s, ..., j_m^s \mid d_{j_1^s} \le d_{j_2^s} \le .... \le d_{j_m^s}\}. \tag{12}$$

Разделим I_2^s на два подмножества:

$$I_{21}^s = \{i \in I_2^s : a_i \le \tau\} \text{ и } I_{22}^s = \{i \in I_2^s : a_i > \tau\}. \tag{13}$$

Выберем какое-либо задание $i^* \in I_{21}^s$, например, следующим образом:

$$i^* = \arg\min\,(d_i : i \in I_{21}^s) \tag{14}$$

или

$$i^* = \arg\min(t_i : i \in I_{21}^s). \tag{15}$$

Если существует целое множество значений, удовлетворяющих (14), то выбираем среди них задание с наименьшей длительностью (или в случае (15) — с наименьшим d_i), а и при этих равных условиях — задание с наименьшим индексом.

Выделим в $I_2^s(i^*) = I_2^s / i^*$ следующие четыре подмножества:

$$I_{21}^s(i^*) = \{i \in I_2^s(i^*) : a_i < \tau(k) + t_{i^*},\ d_i < d_{i^*}\}, \tag{16}$$

$$I_{22}^s(i^*) = \{i \in I_2^s(i^*) : a_i < \tau(k) + t_{i^*},\ d_i \ge d_{i^*}\}, \tag{17}$$

$$I_{23}^s(i^*) = \{i \in I_2^s(i^*) :\ a_i \ge \tau(k) + t_{i^*},\ d_i \ge d_{i^*}\}, \tag{18}$$

$$I_{24}^s(i^*) = \{i \in I_2^s(i^*) :\ a_i \ge \tau(k) + t_{i^*},\ d_i < d_{i^*}\}. \tag{19}$$

Пусть g^s — количество членов подмножества I_1^s, а v^s— количество членов последовательности $J_3^s(i^*) = \{i \in I_2^s : i \ne i^*,\ d_i < d_{i^*}\}$. Все задания последовательности $J_3^s(i^*)$ суть первые v^s задания последовательности (12). Последовательность $J^s(i^*)$ подмножества $I_2^s(i^*)$ содержит $(m^s - 1)$ заданий и является последовательностью (12), в которой отсутствует задание i^*.

Утверждение 3. Если не выполняется ни одно из системы неравенств (8), справедливы системы неравенств

$$\tau + t_{i^*} \le d_{i^*};\ \tau + t_{i^*} + \sum_{l=1,\, l \ne i^*}^{p} t_{i_l} \le d_{i_p},\ i_l \in J_3^s(i^*),\ p = 1, ..., v^s, \tag{20}$$

где все задания упорядочены в последовательность $i_l \in J_3^s(i^*)$, то назначение в τ-й временной интервал начала выполнения задания i^* и выполнение его до окончания без прерываний не приведет в следующий момент времени $\overline{\tau} = \tau + t_{i^*}$ к выполнению хотя бы одного из системы неравенств (10), т.е. будет выполняться система неравенств

$$\tau + t_{i^*} + \sum_{l=1^*}^{p} t_{i_l} \le d_{i_p},\ i_l \in J^s(i^*). \tag{21}$$

Доказательство. Для первых v^s членов последовательности $J^s(i^*)$ справедливость (21) следует из (20). Для каждого из индексов второй подпоследователь-

ности, включающей $(m^s - v^s - 1)$ членов, $i_l \in J^s(i^*)$, $l = v^s + 1, v^s + 2, \ldots, m^s - 1$, справедливость утверждения 3 доказывает равенство сумм, в которых изменено место лишь одного слагаемого.

Система неравенств (21) рассматривается как необходимое условие допустимости решения задачи (2), (3) с учетом ограничений а) для развиваемого частичного плана P^s.

Следствия утверждения 3. 1. Проверяемая в момент времени τ система неравенств может не включать задания подмножества $I^s_{22}(i^*) \cup I^s_{23}(i^*)$, для которых $d_{i_l} > d_{i^*}$, и содержать меньшее число проверок. Если $I^s_{21}(i^*) \cup I^s_{24}(i^*) = \varnothing$, то при назначении задания i^*, выбираемого согласно условиям (21), соответствующее место последовательности не требует никаких дополнительных проверок.

2. Так как для подмножества индексов $j \in I^s_{24}(i^*)$ справедливы неравенства $a_i \geq \tau + t_{i'}$ и $i \in I^s_{24}(i^*)$, то проверка системы неравенств для данного подмножества индексов в момент времени τ также не требуется. Однако выполнение неравенств (10) для некоторого из подмножества индексов может достигаться на каком-то другом последующем шаге $q > k + 1$. Это означает, что выбор i^* на предыдущем k-м или более ранних шагах недопустим. Поэтому выбору индекса i^* на данном шаге k должна предшествовать некоторая альтернатива (ветвление).

3. Если $I^s_{24}(i^*) = \varnothing$, то если частичный план P^s содержит некоторое подмножество допустимых решений, назначение на k-е место в последовательности задания i^*, выбор которого производится в соответствии с условиями (21), не сделает это подмножество пустым. Следовательно, при выборе в момент времени τ на соответствующее место в последовательности задания i^* в этих условиях никакие другие альтернативы могут не рассматриваться.

Утверждение 4. Если в частичном плане P^s в момент времени τ включение некоторого i_p-го задания, стоящего на p-м месте в последовательности (12), не обеспечивает выполнение всей системы неравенств (21), то включение на это место последовательности любого другого задания i_r, стоящего в (12) правее его $(r > p)$, для которого $t_{i_r} > t_{i_p}$, и выполнение его без прерываний до завершения также не обеспечит выполнение всех ограничений (21).

Доказательство утверждения следует из таких соотношений:

$$\tau + t_{i_p} + \sum_{l=1}^{q-1} t_{i_l} < \tau + t_{i_r} + \sum_{\substack{l=1 \\ l \neq p}}^{q} t_{i_l},\ q = 1, 2, \ldots, (r-1),\ \text{так как}\ t_{i_r} > t_{i_p},$$

$$\tau + t_{i_p} + \sum_{\substack{l=1 \\ l \neq pl}}^{r} t_{i_l} + \sum_{l=r+1}^{q} t_{i_l} = \tau + t_{i_r} + \sum_{l=1}^{r-1} t_{i_l} + \sum_{l=r+1}^{q} t_{i_l},\ \text{так как}\ q = r, (r+1), \ldots, m^s.$$

Следствие утверждения 4. Если в момент времени τ включение на соответствующее место строящейся последовательности задания i_p не обеспечивает выполнение всех ограничений системы (21), то в качестве возможных альтернатив могут рассматриваться только те задания i_r $(r > p)$ из (12), для которых $t_{i_r} < t_{i_p}$.

Отметим, что в подмножестве индексов $I^s_{21}(i^*)$ может появиться такой индекс $\xi \in I^s_{21}(i^*)$, для которого справедливы условия

$$a_\xi < \tau(k) + t_{i^*},\ d_\xi < d_{i^*}\ \text{ и }\ \tau(k) + t_{i^*} + \sum_{l=1}^{\xi} t_l > d_\xi. \qquad (22)$$

Выбор $\bar{j}$ из условий (15) обеспечивает минимальное значение t_i, т.е. $t_{\bar{j}} < t_{i^*}$. Выбор задания $i \in I_{21}^s(i^*)$ из условий

$$\bar{I}_{23}^s = \{i \in I_2^s : t_{i^*} > t_i \geq t_{\bar{j}},\ d_{i^*} \leq d_i \leq d_{\bar{j}}\} \tag{23}$$

может обеспечить выполнение условий $\tau(k) + t_{\bar{j}} + \sum_{l=1}^{\xi} t_l \leq d_\xi$.

Если такое задание не может быть найдено, то в τ-й момент времени для альтернативы P^s не может быть начато выполнение никакого задания, и следующий рассматриваемый временной интервал должен выбираться из условий

$$\bar{\tau} = \min_{i \in I_2^s(k)} \{a_i : a_i > \tau\}. \tag{24}$$

2.2. Ветвление. В качестве анализируемых альтернатив (ветвлений) могут рассматриваться назначения различных заданий на каждое из мест последовательности. На основе анализа свойств допустимых последовательностей удается существенно сократить число рассматриваемых альтернатив.

Утверждение 5. Если в *s*-м частичном плане существует некоторая подпоследовательность (21), включающая g^s заданий и совпадающая с первыми g^s членами этой же подпоследовательности (5), т.е. для которых $d_{l-1}^s \leq d_l^s$, $l = 2, \ldots, g^s$, и в промежутке времени $t \in [\theta_0, \tau = \bar{\theta}^s]$ не существует ни одного временного интервала, в котором не выполнялось бы ни одно задание, то никакие другие альтернативы назначения на первые g^s мест в допустимой последовательности выполнения заданий для данного частичного плана могут не рассматриваться.

Утверждение 6. Если на *k*-м шаге анализа *s*-го частичного плана выбор задания i^* допустимый, а подмножество $I_{24}^s(i^*) = \varnothing$, то в этом частичном плане на данном месте последовательности может не рассматриваться никакая другая альтернатива. Если $I_{24}^s(i^*) \neq \varnothing$, то в качестве альтернатив на это место в последовательности должны рассматриваться как любое другое задание $\xi \in I_2^s$, удовлетворяющее условиям утверждения 3, так и возможность невыполнения во временном интервале τ никакого другого задания, а также выбор следующего значения $\bar{\tau} = \min_{i \in I_{23}^s(i^*) \cup I_{24}^s(i^*)} a_i$ и расширение подмножества I_2^s.

Если на каком-то *k*-м шаге построения допустимой последовательности для *s*-го частичного плана условия утверждения 3 не выполняются для задания i^*, выбранного как в соответствии с условиями (14), так и согласно условиям (15), то выбор задания на каком-то предыдущем шаге (q-го члена последовательности, $q = k-1, \ldots, 1$) был недопустимым, и на это место должно быть поставлено какое-то другое задание. Если в этом случае для всех предыдущих членов последовательности $q = k-1, \ldots, 1$ нет других допустимых альтернатив, то система ограничений задачи несовместна. При наличии альтернатив анализ их целесообразно начинать с наиболее дальнего члена строящейся последовательности.

2.3. Допустимые времена начала и завершения выполнения заданий. Пусть $h_i^1 = \bar{a}_i$ и $h_i^2 = (\bar{d}_i - \bar{a}_i - t_i + 1)$ — соответственно допустимые наиболее раннее и наиболее позднее время начала выполнения задания i.

Утверждение 7. Если хотя бы для одного задания $i = 1, \ldots, n$ справедливо неравенство $h_i^2 > h_i^1$, то не существует расписаний, обеспечивающих выполнение всех ограничений задачи.

Рассмотрим два задания: i и j, для которых вычислены значения допустимых сроков начала выполнения, а также задано время завершения каждого из них.

Утверждение 8. Для каждой рассматриваемой пары i и j заданий справедливы следующие условия:

1) если $h_i^1 + t_i > h_j^2$ или $h_j^1 + t_j > h_i^2$, то не существует расписаний, обеспечивающих выполнение всех ограничений задачи;

2) если $h_j^1 < h_i^1 + t_i \le h_j^2$, то в допустимом расписании значения h_j^2 и $\bar{d}_j$ должны быть скорректированы следующим образом:

$$h_j^2 = \min\{h_j^2;\ h_i^1 + t_i\},\ \bar{d}_j = \min\{h_j^2;\ h_i^1 + t_i\} + t_j - 1; \tag{25}$$

3) если $h_j^1 < h_i^1 + t_i \le h_j^2$, а $h_j^1 + t_j > h_i^2$, то в допустимом расписании задание i должно выполняться раньше, чем задание j, и сроки начала и завершения должны быть скорректированы следующим образом:

$$h_j^1 = \max\{h_j^1, (h_i^1 + t_i)\},\ h_i^2 = \min\{h_i^2, (h_j^2 - t_j)\},\ \bar{d}_i = h_i^2 + t_i - 1. \tag{26}$$

Для скорректированных граничных сроков выполнения заданий полагаем $\bar{a}_i = h_i^1$.

Может быть построен итеративный процесс проверки для каждой пары заданий выполнения всех условий утверждений 8, 9, в результате которого будут скорректированы сроки начала и завершения всех заданий, либо еще до начала построения расписаний установлен факт несовместности системы ограничений задачи.

2.4. Некоторые свойства расписаний с ограничениями на частичные порядки выполнения заданий. Пусть ограничения на частичный порядок выполнения заданий заданы некоторым графом с матрицей смежности B, элементы которой $\beta(i, j) = 1$, если задание i в допустимой последовательности должно предшествовать заданию j и $\beta(i, j) = 0$ в противном случае. Обозначим $\tilde{\Pi}(j)$ и $\tilde{Q}(j)$ соответственно подмножество всех заданий, которые в соответствии с ограничениями на частичный порядок следования заданий должны предшествовать заданию j и следовать за заданием j (непосредственно или косвенно). Обозначим $H_1(j)$ и $H_2(j)$ соответственно количество всех предшественников и последователей задания j (количество элементов в подмножествах $\tilde{\Pi}(j)$ и $\tilde{Q}(j)$).

Утверждение 9. Если $i \Rightarrow j$, т.е. задание i должно предшествовать заданию j, то в допустимом расписании должны выполняться неравенства

$$x_j \ge x_i + t_i + 1,\ \sigma_j \ge \sigma_i + t_i.$$

Следствие утверждения 9. При построении допустимых расписаний с ограничениями на условия предшествования заданий значения граничных сроков начала a_j и завершения d_j j-го задания должны быть скорректированы:

$$\bar{a}_j = \max\{a_j,\ \max_{i \in \tilde{\Pi}(j)} [\bar{a}_i + t_i] + 1\},\quad \bar{d}_j = \min\{d_j,\ \min_{i \in \tilde{Q}(j)} [\bar{d}_i - t_j]\}. \tag{27}$$

Таким образом, начальные и конечные сроки выполнения всех связанных графами предшествования заданий должны быть скорректированы в соответствии с выражением (27).

2.5. Нижняя граница длительности выполнения расписания. Грубая нижняя граница длительности выполнения расписания вычисляется по формуле

$$\zeta^0(F) = \min\left\{\min_{1\le i\le n}\bar{a}_i + \sum_{i=1}^{n} t_i + \min_{1\le i\le n} g_i;\ \min_{1\le i\le n}\sum_{i=1}^{n}(\bar{a}_i + t_i + g_i)\right\}. \qquad (28)$$

Для каждого из заданий расписания может быть вычислено значение D_i наиболее позднего времени завершения его выполнения, обеспечивающее завершение выполнения всего расписания выполнения работ за время, не превышающее $\zeta^0(F)$ и обеспечивающее выполнение всех ограничений на сроки выполнения и частичные порядки следования заданий:

$$D_i^0 = \min\{\bar{d}_i;\ \zeta^0(F) - g_i\},\ i = 1, \ldots, n. \qquad (29)$$

Утверждение 10. Если среди множества заданий $i = 1, \ldots, n$ найдется хотя бы одно задание j, для которого справедливо

$$D_j^0 = \min\{\bar{d}_j;\ \zeta^0(F) - g_j\} = \zeta^0(F) - g_j > \bar{d}_j, \qquad (30)$$

то не существует допустимых расписаний, обеспечивающих завершение всех работ за время, меньшее или равное $\zeta^0(F)$, и обеспечивающих выполнение всех ограничений задачи. Нижняя граница длины расписания не может быть меньше величины

$$\Delta_j^0 = \zeta^0(F) + [\bar{d}_j - (\zeta^0(F) - g_j)]. \qquad (31)$$

Следствие утверждения 10. Нижняя граница длины расписания задачи, обеспечивающего выполнение всех ограничений, определяется выражением

$$\zeta^1(F) = \zeta^0(F) + \Delta^1 = \zeta^0(F) + \max_{1\le i\le n}\{\min[0;\ \bar{d}_i - (\zeta^0(F) - g_i)]\}. \qquad (32)$$

Для значения нижней границы $\zeta^1(F)$, аналогично выражению (29), могут быть вычислены скорректированные значения D_i^1. Принимая их в качестве значений, построим расписание, допускающее разрывы в выполнении заданий. Поскольку в этом расписании могут быть временные интервалы, в течение которых не выполняется ни одно из заданий, то время завершения этого расписания $T = \zeta^2(F)$ может быть больше величины $\zeta^1(F)$. Значение $\zeta^2(F)$ может быть принято за более точную нижнюю границу допустимого расписания, не допускающего разрывы во времени выполнения заданий. После этого вычисляются новые значения граничных времен завершения отдельных заданий D_i^2 аналогично (29).

Ясно, что при отсутствии ограничений на сроки завершения заданий и частичные порядки их выполнения нижняя граница длины расписания вычисляется по формуле (28), в которой вместо значений $\bar{a}_i$ подставляются значения a_i, и затем может быть также уточнена построением расписания, допускающего разрывы в выполнении заданий.

3. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ

Установленные выше свойства допустимых решений и правила отсева неперспективных продолжений используются ниже при конструировании алгоритмов решения задачи. В описанных ниже алгоритмах рассматриваются последовательности выполнения заданий (5), (12), ограничения (10), (16)–(20), в которых значения d_i могут быть заменены на значения D_i^{μ}, $\mu = 0, 1, 2, \ldots$

3.1. Алгоритм В — построение допустимого расписания, допускающего разрывы выполнения заданий. Пусть t — текущий номер анализируемого вре-

менного интервала, $z(t)$ — индекс задания, выполняемого во временном интервале t. Если в момент времени t не выполняется ни одно задание, то $z(t) = -1$; $l_i(t)$ — количество временных интервалов, необходимых для завершения выполнения задания i в данный момент времени t; $I_1(t)$ — подмножество выполненных в момент времени t заданий; $I_2(t)$ — подмножество заданий, выполнение которых начато, но не закончено в момент времени t; $I_3(t)$ — подмножество заданий, выполнение которых еще не начато в момент времени t; λ_{is} и σ_{is} — s-е назначенные времена начала и завершения выполнения задания i, которое в процессе выполнения может прерываться. Заметим, что при наличии разрывов во времени выполнения заданий каждое из них может начинаться и прерываться несколько раз; T — максимальная длительность расписания; $B(t)$, $t = 1, \ldots, T$, — вектор номеров заданий, выполнение которых уже начато, но не закончено в соответствующий временной интервал. Если в некотором временном интервале t_1 таких заданий не существует, то $B(t_1) = -1$.

Итерационный алгоритм В состоит из нескольких шагов.

Шаг 0 (предварительный). Полагаем $t = 0$; $z(t) = \varnothing$; $l_i(t) = t_i$, $i = 1, \ldots, n$; $I_1(t) = I_2(t) = \varnothing$, $I_3(t) = \{1, \ldots, n\}$, индексы s номеров времен начал и окончаний выполнения всех заданий равными 1, а также $B(t) = -1$, $t = 1, \ldots, T$. Переходим к шагу 1.

Шаг 1. Если $I_1(t) = \{1, \ldots, n\}$, $I_2(t) = \varnothing$ и $I_3(t) = \varnothing$, то алгоритм завершает свою работу. Если $I_1(t) \neq \{1, \ldots, n\}$, $I_2(t) \neq \varnothing$ и (или) $I_3(t) \neq \varnothing$, в момент времени t выполняется задание j и $t < \min\{a_i : i \in I_3(t)\}$, то находим новое значение t_1 из условий

$$t_1 := \min \min \{a_i \mid i \in I_3(t), d_i < d_j, a_i > t\}. \tag{33}$$

Если $t_1 = a_i$ и существуют несколько индексов i, для которых справедливо выражение (33), то выбираем номер того задания, у которого значение d_i минимально. В случае нескольких заданий с минимальным значением d_i выбираем задание с наименьшей длительностью. При равных и этих условиях выбираем задание с наименьшим индексом. Пусть выбрано некоторое задание ξ.

Если $t_1 = a_\xi = \min\{a_i : i \in I_3(t)\}$ и $\lambda_\xi - t < l_j(t)$, то полагаем

$$\lambda_{\xi,1} = t_1, \; z(t_1) = \xi. \tag{34}$$

Если $I_3(t) = \varnothing$ или $z(t) = j$ и при этом $l_j(t) \le \min\{a_i - t : i \in I_3(t), a_i > t\}$, то полагаем

$$t_1 := t + l_j(t), \; \sigma_j = t_1. \tag{35}$$

Определяем $B(\bar{t}) = j$ для значений $\bar{t} = t, t+1, \ldots, \sigma_j - 1$.

Если $z(t) = j$, то $l_j(t_1) = l_j(t) - (t_1 - t)$, $\sigma_{js} = t_1$.

Увеличиваем индексы s номеров начала выполнения задания ξ и окончания выполнения задания j на 1. Переходим к шагу 2.

Если $I_2(t) \neq \varnothing$ и $t_1 < a_i$, то находим $\rho = \arg\min\{d_i \mid i \in I_2(t)\}$.

Если $I_3(t) \neq \varnothing$, то определяем $l_\rho = 0$. Если $I_3(t) \neq \varnothing$ и $t_1 = t + l_j(t) < a_i$, а $I_2(t) \neq \varnothing$, то полагаем $l_\rho = \min(0, l_\rho - a_i + t_1)$, $z(t) = \rho$, $t = t, t+1, \ldots, t + l_\rho - 1$. Если $l_\rho = 0$, то определяем $t_2 = t + l_\rho$, $I_2(t_2) = I_2(t) / \rho$ и переходим к шагу 2.

Шаг 2. Если $l_j(t_1) = 0$, то полагаем

$$I_1(t_1) = I_1(t) \cup j, \; I_2(t_1) = I_2(t) / j. \tag{36}$$

Если $z(t_1) = \xi$ и $l_\xi = t_\xi$, то полагаем

$$I_3(t_1) = I_3(t) / \xi, \; I_2(t_1) = I_2(t) \cup \xi. \tag{37}$$

Полагаем $t := t_1$ или $t := t_2$ и переходим к шагу 1.

Через конечное количество итераций будет построено некоторое расписание выполнения заданий, допускающее разрывы во времени их выполнения. В каждый момент времени k из разрешенных для выполнения, но еще не завершенных заданий, выполняется задание, имеющее наименьший директивный срок завершения. Описанное выше построение в соответствии с утверждениями 1–3 обеспечивает оптимальный порядок и сроки начала и завершения выполнения всех заданий с точки зрения обеспечения выполнения всей системы ограничений.

3.2. Алгоритм D — построение допустимых расписаний, не допускающих разрывов во времени выполнения заданий. Алгоритм осуществляет построение последовательности выполнения заданий $\widetilde{W}$.

Пусть построена последовательность выполнения заданий J в соответствии с выражением (5), а также сформирован некоторый вектор Y, в котором для каждого задания определен y_i^1 номер первого стоящего в последовательности J перед заданием i задания и не включенного в строящуюся последовательность $\widetilde{W}$. Обозначим: τ — текущий номер анализируемого временного интервала, $\overline{B}(\tau)$, $\tau = 1, \ldots, T$, — вектор признаков выполненных в момент времени τ заданий, где $\overline{b}_i(\tau) = 1$, если в момент времени τ задание i выполнено, и $\overline{b}_i(\tau) = 0$ в противном случае; $\overline{i}(\tau)$ — наименьший номер в последовательности J задания, для которого $\overline{b}_i(\tau) = 0$; M — трехмерный массив, в котором указаны $m_{\mu 1}$ — номер временного интервала, где произошло изменение порядка следования заданий; $m_{\mu 2}$ — первый номер задания в последовательности J, для которого $\overline{b}_i(\tau) = 0$; $m_{\mu 3}$ — номер задания, стоящего в последовательности J после задания $m_{\mu 2}$ и выполняемого в момент времени τ в расписании $\widetilde{W}$.

Пусть $\overline{\mu}$ — количество строк массива M.

На начальном этапе построения полагаем $\tau = 0$, $\tau = 1, \ldots, T$, $i = 1, 2, \ldots, n$. Массив M не содержит ни одной строки и $\overline{\mu} = 0$.

Алгоритм построения предусматривает выполнение следующих шагов.

Шаг 1. Если $b_i(\tau) = 1$, $i = 1, 2, \ldots, n$, то допустимое расписание построено и алгоритм заканчивает работу. В противном случае для временного интервала τ в последовательности J находим первое среди стоящих слева заданий — $\overline{j}$, удовлетворяющее условиям

$$a_{\overline{j}} \le \tau, \ \overline{b}_{\overline{j}}(\tau) = 0. \tag{38}$$

Если такого задания не существует, то переходим к шагу 2. Если $\overline{j} = \overline{i}(\tau)$ — задание, стоящее первым слева в последовательности J среди заданий, для которых $b_j(\tau) = 0$, то переходим к шагу 3. Если $\overline{j}$ — не первое задание в последовательности J, для которых выполняются условия $b_j(\tau) = 0$, то переходим к шагу 4.

Шаг 2. Находим новое значение τ_1 из условий

$$\tau_1 = \min \{a_{l_i} : l_i \in J; \overline{b}_{l_i}(\tau) = 0, a_{l_i} \ge \tau\}. \tag{39}$$

Если значений τ_1 не существует, то переходим к шагу 6. В противном случае полагаем $\tau = \tau_1$ и переходим к шагу 1.

Шаг 3. Включаем задание $\overline{j} = \overline{i}(\tau)$ в строящуюся последовательность $\widetilde{W}$, выполняем его $t_{\overline{j}}$ временных интервалов. Полагаем $\overline{b}_j(\tau) = 1$, $\tau = \tau + t_{\overline{j}}$, и переходим к шагу 1.

Шаг 4. Для данного задания $\overline{j} = \overline{i}(\tau)$ определяем его номер p в последовательности J. Определяем также номер первого задания в последовательности J, для которого $\overline{b}_j(\tau) = 0$. Пусть номером этого задания в J будет r.

Для всех заданий, стоящих в последовательности J, начиная с задания j_p, удовлетворяющих условиям $\bar{b}_{j_s}(\tau)=0$, $s=r, r+1,\ldots, p-1$, проверяем выполнение неравенств

$$\tau+t_{\bar{j}}\le d_{\bar{j}},\ \tau+t_{\bar{j}}+\sum_{\substack{s=r\\ \bar{b}_{j_s}=0}}^{p-1} t_{j_s}\le d_{j_s},\ s=r, r+1,\ldots, p-1. \tag{40}$$

Если не выполняется хотя бы одно из системы неравенств (40), то эта подпоследовательность не имеет допустимых решений. Переходим к шагу 5. В противном случае выполняем следующие действия:

а) формируем $(\mu+1)$-ю строку массива M, положив $m_{\mu 1}=\tau$, $m_{\mu 2}=r$, $m_{\mu 3}=p$, увеличиваем μ на 1, $\mu=(\mu+1)$;

б) полагаем $\bar{b}_{j_p}(\tau)=1$, $\tau=\tau+t_{j_p}$.

Для включения задания $\bar{j}=j_p$ в строящуюся последовательность переходим к шагу 3.

Шаг 5. В последовательности J находим первое задание j_ω, стоящее в J правее задания j_p с номером p, и удовлетворяющее условиям $\omega>p$, $a_{j_\omega}\le\tau$, $\bar{b}_{j_\omega}(\tau)=0$, $t_{j_\omega}<t_{j_p}$. Если таких заданий не существует, то переходим к шагу 2. В противном случае полагаем $p=\omega$ и переходим к шагу 4.

Шаг 6. Находим последнюю $\bar{\mu}$-ю запись (строку) в массиве M. Если $\bar{\mu}=0$, т.е. таких записей не существует, то задача не содержит допустимых решений и алгоритм завершает свою работу. В противном случае:

а) определяем временной интервал $\tau=m_{\bar{\mu}1}$;

б) номера заданий $\bar{i}(\tau)=m_{\bar{\mu}2}$, $j_p=m_{\bar{\mu}3}$;

в) для всех заданий, которые в последовательности $\widetilde{W}$ стоят после задания j_p, полагаем $\bar{b}_{\bar{i}}(\tau)=0$;

г) формируем новую подпоследовательность, которая содержит только задания рассматриваемой ранее последовательности $\widetilde{W}$, выполненные до момента времени τ;

д) находим в последовательности J первое задание j_η, стоящее в J правее задания j_p и удовлетворяющее одному из пары условий:

$$a_{j_\eta}\le\tau,\ t_{j_\eta}<t_{j_p}; \tag{41}$$

$$a_{j_\eta}>\tau,\ t_{j_\eta}+(a_{j_\eta}-\tau)<t_{j_p}. \tag{42}$$

Если выполняются условия (42), то во вновь строящейся последовательности полагаем $a_{j_\eta}=\tau$. Если таких заданий не существует, то исключим строку $\bar{\mu}$ из массива M $(\bar{\mu}=\bar{\mu}-1)$. Снова возвращаемся к выполнению шага 6. В противном случае полагаем $j_p=j_\eta$, $m_{\bar{\mu}1}=\tau$, $m_{\bar{\mu}2}=\bar{i}(\tau)$, $m_{\bar{\mu}3}=j_\eta$ и переходим к шагу 4.

Через конечное число шагов либо будет построено допустимое расписание выполнения заданий, не допускающее разрывов во времени их выполнения, либо установлен факт несовместности ограничений задачи.

Конечность алгоритма D следует из следующих соображений. На каждой итерации алгоритма либо на какое-то место в последовательности устанавливается некоторое ранее не установленное на это место задание и меняется вид построенной ранее последовательности, либо алгоритм заканчивает свою работу с сообщением, что получено решение или допустимых решений не существует. Так как число различных последовательностей конечно, то даже в предельном случае алгоритм завершит работу за конечное количество итераций.

Аналогичным алгоритмом может быть преобразовано расписание, содержащее разрывы в выполнении заданий, в расписание, не допускающее разрывов. В качестве рассматриваемого момента времени τ в данном случае выбирается время первого выполняемого в строящейся последовательности задания, допускающего его прерывание. В качестве одной из альтернатив при этом, требующей проверки выполнения условий (40), является завершение выполнения этого задания до конца без прерываний.

3.3. Алгоритм R — построение подмножеств $\widetilde{\Pi}(i)$ и $\widetilde{Q}(i)$ (предшественники и последователи задания i) и коррекция граничных сроков начала и завершения выполнения заданий. Рассмотрим подмножества заданий-предшественников $Y_1(k)$, выделенных на каждом k-м шаге алгоритма, а также подмножества $Y_2(k) = I / Y_1(k)$ заданий, не связанных на k-м шаге отношениями предшествования. Пусть $Z_1 = \{\zeta_1, ..., \zeta_k, \ldots, \zeta_n\}$ и $Z_2 = \{\varphi_1, ..., \varphi_k, \ldots, \varphi_n\}$ определяют соответственно последовательности корректировки допустимых времен начала и завершения выполнения заданий, связанных отношениями предшествования. Положим в начале итеративного процесса для $(k = 0)$ $Y_1(k) = Z_1 = Z_2 = \varnothing$, $Y_2(k) = I$, а также $\widetilde{\Pi}(i) = \varnothing$, $\widetilde{Q}(i) = \varnothing$, $i = 1, \ldots, n$.

Первая итерация алгоритма (корректировка ограничений на начало выполнения заданий). На каждом k-м шаге алгоритма, $k = 1, 2, \ldots$, выполняем следующее:

1) находим такое задание $q \in Y_2(k)$, для которого $\beta(j, q) = 0$, $j \in Y_2(k)$. Если таких заданий несколько, то в качестве q на данном шаге выбираем задание с наименьшим индексом;

2) включаем индекс выбранного задания q на k-е место в последовательность Z_1;

3) для всех $i = 1, \ldots, n$, если $\beta(i, q) = 1$ и $i \in Y_1(k)$, дополняем подмножества предшественников этих заданий: $\widetilde{\Pi}(q) = \widetilde{\Pi}(q) \cup i \cup \widetilde{\Pi}(i)$;

4) при переходе к следующему $(k+1)$-му шагу корректируем подмножества $Y_1(k+1) = Y_1(k) \cup q$, $Y_2(k+1) = Y_2(k) / q$ и полагаем $k = k+1$. Если $Y_2(k) \neq \varnothing$, то возвращаемся к п. 1.

(В результате работы алгоритма R будут сформированы подмножества заданий $\widetilde{\Pi}(i)$, $i = 1, \ldots, n$ ($H_1(i)$ — количество элементов в подмножестве $\widetilde{\Pi}(i)$), и построена последовательность заданий Z_1.)

5) для всех заданий в последовательности Z_1 корректируем значения a_i в соответствии с формулами (27).

Вторая итерация алгоритма (корректировка граничных значений завершения выполнения заданий). Полагаем в начале итеративного процесса для $(k = 0)$ $Y_1(k) = Z_2 = \varnothing$, $Y_2(k) = I$. На каждом k-м шаге алгоритма, $k = 1, 2, \ldots, n$, выполняем следующее:

1) находим такое задание $r \in Y_2(k)$, для которого $\beta(r, j) = 0$, $j \in Y_2(k)$, если таких заданий несколько, то в качестве r на данном шаге выбираем задание с наименьшим индексом;

2) включаем индекс выбранного задания на k-е место в последовательность Z_2;

3) для всех $i = 1, \ldots, n$, если $\beta(r, i) = 1$ и $i \in Y_1(k)$, то дополняем подмножества последователей этих заданий: $\widetilde{Q}(r) = \widetilde{Q}(r) \cup i \cup \widetilde{Q}(i)$;

4) при переходе к следующему $(k+1)$-му шагу корректируем подмножества $Y_1(k+1) = Y_1(k) \cup r$, $Y_2(k+1) = Y_2(k) / r$ и полагаем $k = k+1$, если $Y_2(k) \neq \varnothing$, то возвращаемся к п. 1;

(В результате работы алгоритма R будут сформированы подмножества заданий $\widetilde{Q}(i)$, $i=1,\ldots,n$, и построена отличная от полученной в первой итерации последовательность заданий Z_2.)

5) затем для всех заданий в последовательности Z_2 корректируем значения d_i в соответствии с формулами (27).

Заметим, что невозможность выбора индекса q (первая итерация) или индекса r (вторая итерация) на некотором шаге алгоритма R свидетельствует о наличии циклов, т.е. некорректности задания ограничений на частичный порядок выполнения заданий.

Если найдется хотя бы одно такое задание, для которого $\overline{a}_i + t_i \geq \overline{d}_i$, то система ограничений задачи несовместна, т.е. при заданных ограничениях на очередности выполнения заданий не существует расписаний, обеспечивающих выполнение директивных всей системы ограничений на начало и сроки завершения заданий.

Следует отметить, что первая итерация, а также пп. 1–4 второй итерации алгоритма R выполяются только один раз в процессе решения всех задач. В процессе решения оптимизационных задач в условиях ограничений на частичные порядки следования заданий на основе один раз построенных в пп. 1–4 второй итерации последовательности Z_2 и подмножеств $\widetilde{Q}(i)$, $i=1,\ldots,n$, п. 5 второй итерации могут многократно корректироваться только граничные сроки завершения выполнения заданий.

Заключительный этап алгоритма *R* включает итеративный процесс проверки и корректировки для каждой пары заданий значений $\overline{a}_i$ и $\overline{d}_i$ в соответствии с выражениями (25), (26), который осуществляется до тех пор, пока в результате проведенных вычислений не может быть скорректировано ни одно из допустимых времен начала или завершения какого-либо из заданий.

3.4. Алгоритм C — **построение допустимых расписаний в условиях ограничений на частичные порядки и сроки выполнения заданий.** Алгоритм C состоит из двух шагов. На первом шаге алгоритмом R строятся подмножества предшественников $\widetilde{\Pi}(i)$ и последователей $\widetilde{Q}(i)$ для каждого из заданий, а также последовательности выполнения расчетов Z_1 и Z_2, на основе которых пересчитываются в граничные значения $\overline{a}_i$ и $\overline{d}_i$, $i=1,\ldots,n$. На втором шаге алгоритмом D определяется допустимое расписание, удовлетворяющее всем ограничениям сформулированной задачи.

Если алгоритм D завершает работу сообщением о несовместности ограничений задачи, то согласно выражениям (30), (31) вычисляются величины невязок для каждого из заданий, а также по формулам (32) — минимальное количество временных интервалов, на которое должна быть увеличена нижняя граница длины расписания. На основе этих данных устанавливается один из следующих фактов:

а) либо при данных ограничениях на сроки начала и завершения заданий установленный частичный порядок их выполнения является недопустимым;

б) либо при данной очередности выполнения заданий директивный срок завершения некоторого задания должен быть увеличен на рассчитанное количество временных интервалов.

4. ОБСУЖДЕНИЕ АЛГОРИТМОВ ПОСТРОЕНИЯ ОПТИМАЛЬНЫХ РАСПИСАНИЙ

4.1. На основе предложенных методов построения допустимых последовательностей выполнения заданий могут быть сконструированы эффективные алгоритмы решения оптимизационной задачи (1)–(4). Кроме того, разработаны качественно новые методы решения хорошо изученной в литературе и рассматриваe-

мой в качестве частного случая этой проблемы задачи (2)–(4), в которой не учитываются ограничения на директивные сроки завершения заданий и на частичные порядки расположения заданий в последовательности их выполнения.

Вычислим грубое значение нижней границы длины расписания $\zeta^0(F)$ по формуле (28) и затем значения D_i^0, $i=1,\ldots,n$, по формулам (29). Для вычисления более точной нижней границы $\zeta^1(F)$ может использоваться алгоритм B. Как и Schrage-algorithms [3, 4], для решения задачи без учета ограничений d_i и разрешением прерываний выполнения заданий этот алгоритм использует динамические решающие правил вида: готовые к выполнению задания выполняются в порядке упорядочения по возрастанию граничных сроков завершения (назовем его алгоритм S). Алгоритм обеспечивает получение решения в [3] за $O(n\log n)$ вычислений, и в [4] — за $[O(n^2)]$.

4.2. Пусть при наличии ограничений на частичные порядки сроки начала и завершения заданий алгоритмом C построено допустимое расписание задачи (1)–(3).

Процесс поиска оптимального решения осуществляется методом последовательных приближений снизу. На основе рассчитанной по формуле (28) для вычисленных значений $\bar{a}_i$ и $\bar{d}_i$, $i=1,\ldots,n$, нижней границы $\zeta^0(F)$ корректируются значения директивных сроков завершения выполнения некоторых заданий D_i^0, $i=1,\ldots,n$, в сторону их уменьшения. Второй итерацией алгоритма R корректируются граничные сроки завершения заданий $\bar{d}_i$. В этих новых условиях решается задача получения допустимого решения алгоритмом D. Если такое решение будет получено, то оно будет оптимальным решением исходной задачи (1)–(4). В противном случае по формуле (32) вычисляется та минимальная невязка, на которую должно быть увеличено значение $\zeta^w(F)$ по отношению к $\zeta^{w-1}(F)$, $w=1,2,..,W$, чтобы условия ограничений задачи могли стать совместными, и вновь вычисляются значения

$$D_i^w = \min\,[\bar{d}_i, \zeta^w(F) - g_i],\ i=1,\ldots,n. \tag{43}$$

Эти значения вновь корректируются, выполняя п. 5 второй итерации алгоритма R, затем снова выполняется алгоритм D. Первое полученное алгоритмом D допустимое решение задачи (1)–(3) с данными значениями D_i^w и является оптимальным решением задачи. Если решение исходной задачи (1)–(3) алгоритмом D не может быть получено, то не существует допустимых решений и задачи (1)–(4). В противном случае предлагаемыми алгоритмами гарантировано получение точного решения задачи (1)–(4).

4.3. Если ограничения d_i на сроки завершения заданий отсутствуют и необходимо учитывать только ограничения на частичные порядки и времена начала выполнения заданий, то задача построения оптимального решается в такой последовательности:

1) выполнив первую итерацию алгоритма R, вычисляем скорректированные допустимые сроки начала заданий $\bar{a}_i$, $i=1,\ldots,n$;

2) алгоритмом S вычисляем нижнюю границу значения критерия в оптимальном решении $\zeta^w(F)$, где $w=0$, вычисляем значения $D_i^w = \zeta^w(F) - g_i$, $i=1,\ldots,n$;

3) выполнив вторую итерацию алгоритма R, вычисляем скорректированные допустимые сроки завершения выполнения заданий $\bar{d}_i$, $i=1,\ldots,n$, принимая в качестве $\bar{d}_i$ значения D_i^w;

4) для вычисленных значений $\bar{a}_i$ и D_i^w проверяем выполнение системы неравенств (6), если она не выполняется, то вычисляем максимальное значение невязки

$$\Delta^w = \max_{1\le p\le n}\left\{0;\left[d_{i_p} - \left(\max\left(a_{i_p},\, \theta_0 + \sum_{l=1}^{p-1} t_{i_l}\right) + t_{i_p}\right)\right]\right\}. \qquad (44)$$

(Корректируем значения $\zeta^{w+1}(F) = \zeta^w(F) + \Delta^w$, увеличиваем w на единицу и возвращаемся к п. 3, в котором выполняется только п. 5 второй итерации алгоритма R, в противном случае алгоритмом D решается задача в новых условиях.)

5) если получено допустимое решение задачи, то построенное расписание является оптимальным, и алгоритм завершает работу, в противном случае вычисляется минимальная невязка среди всех рассмотренных альтернатив в соответствии с (30), (31). Корректируется значение согласно (32), увеличиваем w на единицу и возвращаемся к п. 3, производя в нем во второй итерации алгоритма R только вычисления п. 5.

5. ИЛЛЮСТРАТИВНЫЕ ПРИМЕРЫ

Пример 1. Условия примера 1 сведены в табл. 1.

Таблица 1

Значения	Исходные данные для заданий $i = 1, \ldots, 10$									
	1	2	3	4	5	6	7	8	9	10
a_i	0	0	2	3	5	7	7	10	12	17
t_i	4	2	3	4	2	5	3	2	4	3
d_i	21	17	14	25	36	16	35	18	22	30
g_i	1	9	15	4	3	6	20	5	2	12

Ограничения на последовательность выполнения заданий представлены графом на рис. 1.

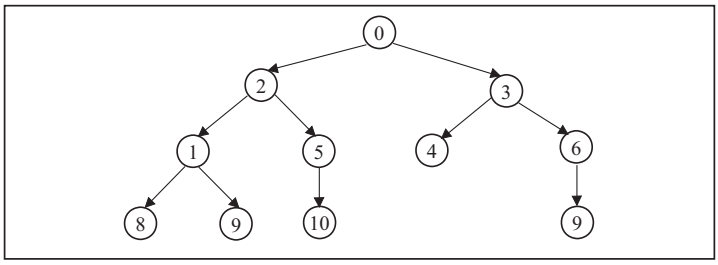

Рис. 1. Граф частичного порядка выполнения задания

На первой итерации алгоритма R получены последовательность применения формул (27) для вычисления значений $\bar{a}_i$ — $Z_1 = \{2, 3, 7, 1, 4, 5, 6, 8, 9, 10\}$, а также определены подмножества заданий $\tilde{\Pi}(2) = \tilde{\Pi}(3) = \tilde{\Pi}(7) = \varnothing$, $\tilde{\Pi}(1) = \{2\}$, $\tilde{\Pi}(4) = \{3\}$, $\tilde{\Pi}(5) = \{2\}$, $\tilde{\Pi}(6) = \{3\}$, $\tilde{\Pi}(8) = \{1, 2\}$, $\tilde{\Pi}(9) = \{1, 2, 3, 6\}$, $\tilde{\Pi}(10) = \{2, 5\}$. В последовательности Z_1 по формулам (27) вычислены значения $\bar{a}_i$, которые сведены в табл. 2.

На второй итерации алгоритма R получены последовательность применения формул (27) для вычисления $\bar{d}_i$ — $Z_2 = \{4, 7, 8, 9, 10, 1, 5, 6, 3, 2\}$, подмножества заданий $\tilde{Q}(4) = \tilde{Q}(7) = \tilde{Q}(8) = \tilde{Q}(9) = \tilde{Q}(10) = \varnothing$, $\tilde{Q}(1) = \{8, 9\}$, $\tilde{Q}(5) = \{10\}$, $\tilde{Q}(6) = \{9\}$, $\tilde{Q}(3) = \{4, 6, 9\}$, $\tilde{Q}(2) = \{1, 5, 8, 9\}$. В последовательности Z_2 по формулам (27) вычислены значения $\bar{d}_i$, которые также сведены в табл. 2.

Таблица 2

Значения	Вычисленные алгоритмом R значения $\bar{a}_i$ и $\bar{d}_i$ для заданий $i = 1, \ldots, 10$									
	1	2	3	4	5	6	7	8	9	10
$\bar{a}_i$	2	0	2	5	5	7	7	10	12	17
$\bar{d}_i$	14	12	14	25	28	16	35	18	21	30

Допустимое решение примера 1 алгоритмом D (расписание 1) получено в результате развития первой альтернативы и приведено во втором и пятом столбцах табл. 3. В третьем и шестом столбцах табл. 3 приведено полученное с помощью Schrage-algorithms оптимальное решение задачи без учета ограничений на d_i и допускающее разрывы в выполнении заданий (расписание 2), на основе которого определена нижняя граница решения $\zeta^0(F) = 33$.

На основе вычисленного значения $\zeta^0(F)$ пересчитаны значения D_i^0 и второй итерацией алгоритма R вычислены значения $\bar{d}_i^0$. В процессе выполнения алгоритма D уже в процессе проверки выполнения системы неравенств (6) была устанавлена несовместность системы ограничений, вычислена максимальная невязка Δ^w, на величину которой скорректированы значения $\zeta^w(F)$ и $\bar{d}_i^w$. Результаты этих вычислений сведены в табл. 4.

Для значений $\bar{d}_i[\zeta^7(F) = 52]$, приведенных в последней строке табл. 4, алгоритмом получено допустимое расписание (расписание 1), которое в точности совпадает с расписанием решения задачи 1 и приведенном во втором и пятом столбцах табл. 3. Следовательно, это расписание, предусматривающее следующий порядок выполнения заданий — $\widetilde{W} = \{2, 3, 1, 6, 8, 9, 4, 5, 10, 7\}$ — со сроками начала и завершения их выполнения, приведенными в табл. 3, является оптимальным.

Таблица 3

Номер временного интервала	Номер задания		Номер временного интервала	Номер задания	
	(расписание 1)	(расписание 2)		(расписание 1)	(расписание 2)
0	2	2	16	9	8
1	2	2	17	9	10
2	3	3	18	9	10
3	3	3	19	9	10
4	3	3	20	4	4
5	1	4	21	4	4
6	1	4	22	4	5
7	1	7	23	4	5
8	1	7	24	5	9
9	6	7	25	5	9
10	6	6	26	10	9
11	6	6	27	10	9
12	6	6	28	10	1
13	6	6	29	7	1
14	8	6	30	7	1
15	8	8	31	7	**1**

Таблица 4

Значения $\bar{d}_i[\zeta^w(F)]$	Пересчитанные граничные значения завершения заданий $i=1,\dots,10$ для различных значений $\bar{d}_i[\zeta^w(F)]$										Величина невязки Δ^w
	1	2	3	4	5	6	7	8	9	10	
$\bar{d}_i[\zeta^0(F)=33]$	14	12	14	25	28	16	13	18	21	21	5
$\bar{d}_i[\zeta^1(F)=37]$	14	12	14	25	28	16	17	18	21	25	5
$\bar{d}_i[\zeta^2(F)=42]$	14	12	14	25	28	16	22	18	21	30	2
$\bar{d}_i[\zeta^3(F)=44]$	14	12	14	25	28	16	24	18	21	30	2
$\bar{d}_i[\zeta^4(F)=46]$	14	12	14	25	28	16	26	18	21	30	2
$\bar{d}_i[\zeta^5(F)=48]$	14	12	14	25	28	16	28	18	21	30	2
$\bar{d}_i[\zeta^6(F)=50]$	14	12	14	25	28	16	30	18	21	30	2
$\bar{d}_i[\zeta^7(F)=52]$	14	12	14	25	28	16	32	18	21	30	—

В табл. 5 приведены результаты решения частных оптимизационных задач: расписание 3, в котором отсутствуют ограничения на сроки завершения заданий и порядки выполнения заданий ($\zeta^w(F)=33$); расписание 4, в котором учтены только ограничения на сроки завершения заданий и является допустимым любой порядок их выполнения ($\zeta^w(F)=46$); расписание 5, в котором отсутствуют ограничения на d_i, но частичный порядок выполнения заданий определен графом, представленном на рис. 1 ($\zeta^w(F)=34$).

Следует отметить, что решение задачи 3 алгоритма D получено в результате развития первой альтернативы. В процессе решения задач 4, 5 после пересчета во второй итерации алгоритма R значений $\bar{d}_i[\zeta^w(F)]$ величины невязок определялись в результате проверки системы неравенств (6), и в случае выполнения этой системы неравенств также в результате развития только первой альтернативы сразу находились оптимальные решения.

Таблица 5

Номер временного интервала	Номер задания			Номер временного интервала	Номер задания		
	(расписание 3)	(расписание 4)	расписание 5)		(расписание 3)	(расписание 4)	расписание 5)
0	2	2	2	16	6	9	6
1	2	2	2	17	10	9	6
2	3	3	3	18	10	9	6
3	3	3	3	19	10	9	10
4	3	3	3	20	8	4	10
5	4	1	5	21	8	4	10
6	4	1	5	22	5	4	8
7	4	1	7	23	5	4	8
8	4	1	7	24	9	7	4
9	7	6	7	25	9	7	4
10	7	6	1	26	9	**7**	4
11	7	6	1	27	9	10	4
12	6	6	1	28	1	10	9
13	6	6	1	29	1	10	9
14	6	8	6	30	1	5	9
15	6	8	6	31	**1**	5	**9**

Пример 2. Требуется найти оптимальное расписание выполнения заданий. Ограничения на очередность выполнения приведены на рис. 2, а значения a_i, t_i, g_i приведены в табл. 6. Ограничения на сроки завершения заданий d_i отсутствуют.

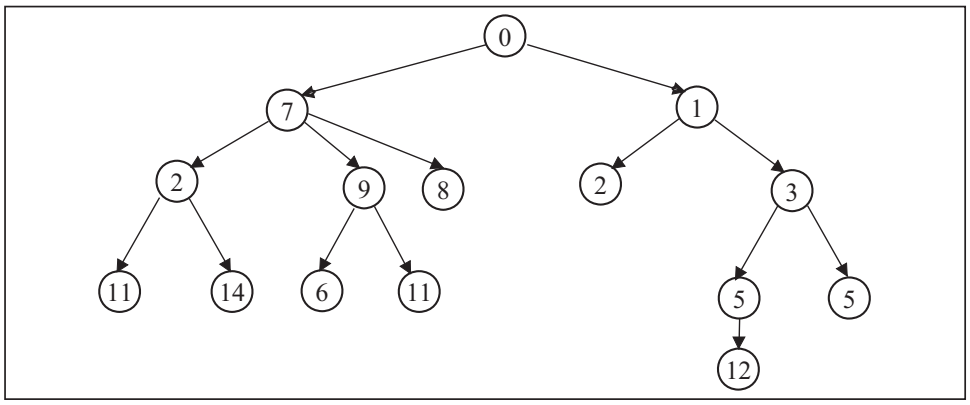

Рис. 2. Граф частичного порядка выполнения задания примера 2

Рассчитанные первой итерацией алгоритма R значения $\bar{a}_i$ сведены в первой строке табл. 7. Полученное с помощью Schrage-algorithms расписание с разрывами выполнения заданий (расписание 6), определяющее нижнюю границу оптимального решения $\zeta^0(F) = 47$, приведено во втором, пятом и восьмом столбцах табл. 8. На основе рассчитанной нижней границы второй итерацией алгоритма R вычислены значения $\bar{d}_i[\zeta^0(F) = 47]$, которые приведены в третьей строке табл. 7. На основе вычисленной в результате проверки системы неравенств (6) невязки, равной 2, скорректированы значения $\zeta^1(F) = 49$ и $\bar{d}_i[\zeta^1(F) = 49]$, приведенные в четвертой строке табл. 7. Допустимое решение для новой системы ограничений, которое является оптимальным решением примера 2 (расписание 7), получено алгоритмом D в процессе развития первой альтернативы и приведено в третьем, шестом и девятом столбцах табл. 8.

Таблица 6

Значения	**Исходные данные для заданий $i = 1, \ldots, 15$**														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a_i	0	2	4	4	6	6	8	9	9	12	14	16	18	20	20
t_i	3	2	4	2	5	1	2	3	4	2	3	2	4	5	3
g_i	7	1	10	2	15	4	3	8	12	14	3	5	6	20	10

Таблица 7

Значения	**Вычисленные алгоритмом R значения $\hat{a}_i$, $\hat{d}_i$ для заданий $i = 1, \ldots, 15$**														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\hat{a}_i$	0	10	4	8	8	14	8	10	10	12	14	16	18	20	20
$\hat{d}_i$	40	46	37	45	32	43	44	39	35	33	44	42	41	27	37
$\bar{d}_i[\zeta^0(F) = 47]$	36	26	37	41	32	43	25	39	35	33	44	42	41	27	37
$\bar{\hat{d}}_i[\zeta^1(F) = 49]$	38	28	39	43	34	45	27	41	37	35	46	44	43	29	39

Таблица 8

Номер временного интервала	Номер задания		Номер временного интервала	Номер задания		Номер временного интервал	Номер задания	
	(расписание 6)	(расписание 7)		(расписание 6)	(расписание 7)		(расписание 6)	расписание 7)
0	1	1	16	9	2	32	13	9
1	1	1	17	9	10	33	13	9
2	1	1	18	9	10	34	12	4
3	—	—	19	8	8	35	12	4
4	3	3	20	14	8	36	7	13
5	3	3	21	14	8	37	7	13
6	3	3	22	14	14	38	6	13
7	3	3	23	14	14	39	11	13
8	5	5	24	14	14	40	11	12
9	5	5	25	15	14	41	11	12
10	5	5	26	15	14	42	4	6
11	5	5	27	15	15	43	4	11
12	5	5	28	8	15	44	2	11
13	10	7	29	8	15	45	**2**	**11**
14	10	7	30	13	9			
15	9	2	31	13	9			

В расписаниях 1–7 жирными шрифтом выделены номера заданий, определяющих значение критерия оптимальности задачи.

На основе описанных выше алгоритмов автором разработано программное обеспечение, ориентированное на решение практических задач теории расписаний большой размерности. Как правило, в случае совместности системы ограничений существует не одно, а целое множество допустимых и оптимальных решений. Как показали проведенные нами вычислительные эксперименты, одно из этих оптимальных решений может быть найдено при небольшом количестве внешних итераций и в результате развития ограниченного числа альтернатив.

ЗАКЛЮЧЕНИЕ

Установлены свойства допустимых решений задачи построения расписаний выполнения заданий на одной машине в условиях ограничений на частичные порядки и сроки завершения заданий. На основе этих свойств построены алгоритмы отсечения подмножества решений, не содержащих допустимых планов и уточнены оценки нижней границы длины расписания. Предложенные алгоритмы позволяют в целом ряде случаев уже на начальных этапах решения установить факт несовместности исходной системы ограничений и перечень заданий, временной диапазон выполнения которых должен быть расширен.

Разработаны новые, не рассматриваемые в литературе точные и приближенные методы и алгоритмы решения сформулированной задачи, которые осуществляют приближение к оптимальному расписанию снизу.

Предложенные алгоритмы решения могут применяться и для решения задач построения расписаний выполнения заданий на одной машине без ограничений на частичные порядки и сроки завершения заданий. Как показали результаты вычислительных экспериментов, при решении этих задач, благодаря простоте вычислений на каждой итерации и небольшому количеству требуемого количества

итераций, объем вычислений в предлагаемых алгоритмах такого же порядка, как и в алгоритмах J. Carlier [5]. Предложенные алгоритмы могут быть рекомендованы как для решения прикладных задач, предусматривающих выше описанную математическую формулировку проблемы, так и в алгоритмах решения общей job-shop-problem [12, 13].

СПИСОК ЛИТЕРАТУРЫ

1. Lenstra J.K., Rinnooy Kan A.H.G. , Brucker P. Complexity of maschine scheduling problema // Annals of Diskrete Mathematics. — 1977. — **1.** — P. 343–362.
2. Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmoys D.B. Sequencing and scheduling algorithms and complexity // Graves S.C., Rinnooy Kan A.H.G., Zipkin (Hrsg) P.H. // Handbooks in Operations Research and Management Science. — Vol. 4. Logistics of production and inventory. — Amsterdam: Elsiver, 1993. — 212 p.
3. Baker K.R., Su Z.-S. Sequencing with due-dates and early start times to minimize maximum tardiness // Naval Research Logistics Quarterly. — 1974. — **21.** — P. 171–176.
4. MhMahon G., Florian M. On scheduling with ready times and du dates to minimize maximum lateness // Operations Research. — 1975. — **23.** — P. 475–482.
5. Carlier J. The one-machine sequencing problem // European Journ. of Oper. Res. — 1982. — **11.** — P. 42–47.
6. Nowicki E., Zdrzalka S. A note on minimizing maximum lateness in one-machine sequencing problem with release dates // Ibid. — 1986. — **23.** — P. 266–267.
7. Grabowski J., Nowicki E., Zdrzalka S. A block approach for single machine scheduling with release dates and due dates // Ibid. — 1986. — **26.** — P. 278–285.
8. Domschke W., Schol A., Voβ S. Produktionsplanung. Ablauforganisatorische Aspekte. — Berlin: Springer Verlag, 1997. — 455 s.
9. Lageweg B.J., Lenstra J.K., Lawler E.L, Rinnooy Kan A.H.G. Computer-aided complexity classification of combinatorial problems // Communications of the ACM. — 1982. — **25.** — P. 817–822.
10. Blazewicz J., Cellary W., Slowinski R. Scheduling under resource constraints — deterministic models // Annals of Oper. Res. — 1986. — **7**. — P. 383–300.
11. Танаев В.С., Шкурба В.В. Введение в теорию расписаний. — М.: Наука, 1975. — 256 с.
12. Zack Yu.A. Certain properties of scheduling theory problems // Automat. and Remote Contr. — 1978. — **39.** — P. 99–107.
13. Zack Yu., Rotin S. Mathematisches Modell und Algorithmen der Termin und Kapazitätsplanung. — 2003. — 29 p.— http://www.optimorum.de/projekt5.htm

Поступила 19.11.2009