

## МЕТОДОЛОГИЧЕСКИЕ И ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПДС-АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ МИНИМИЗАЦИИ СУММАРНОГО ВЗВЕШЕННОГО ЗАПАЗДЫВАНИЯ ПРИ ВЫПОЛНЕНИИ ЗАДАНИЙ ОДНИМ ПРИБОРОМ

ПДС-алгоритм решения задачи минимизации суммарного взвешенного запаздывания основан на направленных перестановках, реализующих использование запаздывающими заданиями резервов незапаздывающих заданий. В данной статье изложены свойства и признаки оптимальности решений, получаемых в процессе выполнения алгоритма. Описаны типы перестановок и обоснованы правила их выполнения, позволяющие резко сократить область поиска оптимального решения и исключить бесперспективные варианты решений.

The PDC-algorithm for solving the problem of minimizing total weighted tardiness of jobs is based on directed permutations implementing the use of reserves of nontardy jobs by tardy jobs. In this article the properties and signs of optimality are given for the solutions that are obtained during execution of the algorithm. The types of permutations are described and the rules of their execution are justified that allow to highly shrink the area of search for the optimal solution and to exclude the unpromising solutions.

### Введение

ПДС-алгоритмом называется алгоритм [1, 2, 3], содержащий полиномиальную составляющую, которая строго реализует оптимальное решение, и точный экспоненциальный подалгоритм либо его полиномиальную аппроксимацию (приближенный либо эвристический алгоритм). Полиномиальная составляющая ПДС-алгоритма порождается теоретически обоснованными логико-аналитическими условиями ( $p$ -условиями) – достаточными признаками оптимальности допустимого решения. В отличие от известных методов, проверка  $p$ -условий реализуется лишь при построении допустимого решения. Эффективность получения точного решения полиномиальной составляющей ПДС-алгоритма определяется двумя факторами:

1) статистически значимый вычислительный эксперимент, который должен состоять из двух частей:

а) моделируются индивидуальные задачи, каждая из которых имеет допустимое решение, удовлетворяющее  $p$ -условиям (достаточный признак оптимальности); полиномиальная составляющая должна практически всегда находить такое допустимое решение; в противном случае, полиномиальная составляющая является неэффективной;

б) моделируется произвольное множество индивидуальных задач и определяется процент получения оптимальных решений, который должен быть статистически значимым;

2) вычислительная простота полиномиальной составляющей: если в процессе реализации полиномиальной составляющей  $p$ -условия не выполняются, то решение задачи осуществляется либо точным экспоненциальным алгоритмом, либо его полиномиальной аппроксимацией. Таким образом, интегральный эффект ПДС-алгоритма определяется как эффективностью полиномиальной составляющей, реализующей  $p$ -условия оптимальности допустимого решения, так и эффективностью его приближенного или эвристического алгоритма либо экспоненциального точного алгоритма.

### Постановка задачи

Задано множество независимых заданий  $J = \{j_1, j_2, \dots, j_n\}$ , каждое из которых состоит из одной операции. Для каждого задания известны  $l_j$  – длительность выполнения,  $\omega_j$  – весовой коэффициент и  $d_j$  – директивный срок выполнения. Задания поступают в систему одновременно в момент времени  $d_j = 0, j = \overline{1, n}$ . Прерывания не допускаются. Необходимо построить расписание выполнения заданий для одного прибора, минимизирующее суммарное взвешенное опоздание при выполнении заданий:

$$f = \sum_{j=1}^n \omega_j \max(0, C_j - d_j), \quad (1)$$

где  $C_j$  – момент завершения выполнения задания  $j$ .

Данная задача относится к NP-трудным в сильном смысле [4]. ПДС-алгоритм решения рассматриваемой задачи основан на результатах, опубликованных в [5] и ПДС-алгоритме решения задачи минимизации суммарного запаздывания при выполнении заданий одним прибором, изложенном в [3].

При реализации алгоритма используются следующие типы перестановок [5]:

– EFSR (extraction and forward-shifted reinsertion – извлечение и повторная вставка со сдвигом вперед – определение 3 ниже);

– EBSR (extraction and backward-shifted reinsertion – извлечение и повторная вставка со сдвигом назад – определение 5 ниже).

### Основные теоретические положения

Введем следующие обозначения и определения [5].

Пусть  $j$  и  $j_i$  обозначает номер задания в соответствии с индексацией, заданной функционалом;  $j_{[g]}$  – номер задания, стоящего в допустимом расписании на позиции  $g$ .

**Определение 1.** Резервом времени задания  $j_{[g]}$  называется величина  $r_{j_{[g]}} = d_{j_{[g]}} - C_{j_{[g]}} > 0$ .

**Определение 2.** Приоритетом  $P_{j_{[i]}}$  задания  $j_{[i]}$  называется величина  $\omega_{j_{[i]}}/l_{j_{[i]}}$ .

**Определение 3.** Перестановкой называется процедура переноса задания  $j_{[g]}$  на позицию  $k$  ( $k > g$ ) и, одновременно, заданий, занимающих позиции  $g + 1, g + 2, \dots, k - 1, k$  на позиции  $g, g + 1, \dots, k - 2, k - 1$ , соответственно.

**Определение 4.** Интервалом перестановки задания  $j_{[g]}$  на позицию  $k$  в последовательности  $\square$  называется интервал, определяемый суммой длительностей заданий, занимающих в этой последовательности позиции  $g+1, g+2, \dots, k-1, k$ .

**Определение 5.** Встраиванием называется процедура переноса задания  $j_{[g]}$  на позицию  $p$  ( $g > p$ ) и, одновременно, заданий  $p, p + 1, \dots, g - 2, g - 1$  на позиции  $p + 1, p + 2, \dots, g - 1, g$ , соответственно.

**Определение 6.** Интервалом встраивания  $I_{j_{[g]}}$  задания  $j_{[g]}$  на позицию  $p$  ( $g > p$ ) в последовательности  $\sigma$  называется интервал, определяемый суммой длительностей заданий, занимающих в этой последовательности позиции  $p, p + 1, \dots, g - 1$ , где  $p$  определяется из условия

$$\sum_{i=p-1}^{g-1} l_{j_{[i]}} < C_{j_{[g]}} - d_{j_{[g]}} \leq \sum_{i=p}^{g-1} l_{j_{[i]}} \quad (1)$$

Если же условие (1) не выполняется ни для одной позиции, то  $p = 1$ . Таким образом, запаздывание по заданию  $j$  на позиции  $p$  должно быть равно нулю или минимально.

**Определение 7.** Задание  $j_{[g]}$  называется запаздывающим в последовательности  $\sigma$ , если для него выполняется условие  $d_{j_{[g]}} < C_{j_{[g]}}$ .

**Определение 8.** Последовательностью  $\sigma^{yn}$  (сигма упорядоченная) называется последовательность заданий множества  $J, j = \overline{1, n}$ , в которой задания упорядочены по невозрастанию приоритетов  $P_j$ , т.е.  $\forall j, i, j < i: P_j \geq P_i$ , а при  $P_j = P_i, d_j \leq d_i$ .

**Определение 9.** Процедурой свободной перестановки называется процедура перестановки задания  $j_{[k]}$  на позицию  $q$  ( $k < q$ ) такую, что  $d_{j_{[k]}} \geq C_{j_{[q]}}$ ,  $d_{j_{[k]}} < C_{j_{[q+1]}}$ , если хотя бы для одного задания на интервале  $\overline{k+1, q}$  выполняется:

$$d_{j_{[i]}} < C_{j_{[i]}}, \quad i = \overline{k+1, q}.$$

**Определение 10.** Последовательность заданий, полученную в результате выполнения всех свободных перестановок в последовательности  $\sigma^{yn}$ , назовем  $\sigma^{cn}$ .

**Определение 11.** Запаздывающее задание  $j_{[g]}$  в последовательности  $\square^{cn}$  называется конкурирующим, если в этой последовательности найдется хотя бы одно предшествующее задание  $j_{[l]}$ , для которого выполняются условия

$$d_{j_{[l]}} > d_{j_{[g]}} - l_{j_{[g]}} \quad \text{и} \quad d_{j_{[l]}} - C_{j_{[l]}} > 0.$$

**Определение 12.** Текущей итерацией оптимизации называется итерация, выполняемая для очередного конкурирующего задания, определенного в последовательности  $\sigma^{cn}$ .

**Определение 13.** Текущей подпоследовательностью  $\sigma^k$  называется подпоследовательность, в которой выполняется текущая итерация оптимизации. Эта подпоследовательность ограничена позициями  $\overline{1, g_k}$ , где  $j_{[g_k]}$  – очередное конкурирующее запаздывающее задание в  $\sigma^{cn}$ , и получена из  $\sigma^{cn}$  в результате выполнения ряда перестановок и встраиваний на предыдущих итерациях.

**Определение 14.** Задания  $j_{[i]}$ , для которых в последовательности  $\sigma^{cn}$  выполнялось  $d_{j_{[i]}} - C_{j_{[i]}} \geq 0$ , и которые в результате выполнения процедур встраивания запаздывающих заданий на более ранние позиции в свою очередь стали запаздывающими, называются порожденными запаздывающими заданиями.

В процессе решения задачи в результате выполнения направленных перестановок осуществляется последовательный переход от последовательности  $\sigma^{yn}$  к последовательности  $\sigma^{cp}$  и затем к текущим последовательностям.

В следующих утверждениях сформулированы и обоснованы свойства и признаки оптимальности каждой из рассматриваемых последовательностей, обоснованы правила выполняемых перестановок, доказательства утверждений приведены в [5].

*Последовательность  $\sigma^{yn}$ :*

*Утверждение 1.* Если в последовательности  $\sigma^{yn}$  запаздывающим заданиям не предшествуют задания с резервом времени, то не существует переносов заданий, приводящих к улучшению целевой функции.

*Утверждение 2.* Встраивание запаздывающего задания  $j_{[g]}$  на позицию  $f < p$  не может привести к улучшению целевой функции.

*Утверждение 3.* Пусть в последовательности  $\sigma^{yn} \forall j_{[i]}, i = \overline{1, p-1} R_{j_{[i]}} \leq 0$ . Обозначим ее  $\sigma^{yn1}$ . Запаздывающее задание  $j_{[g]}$  в последовательности  $\sigma^{yn1}$  в результате выполнения встраивания ( $I_{j_{[g]}} = p, g-1$ ) может занять более раннюю позицию, что приведет к улучшению целевой функции только в том случае, если в последовательности  $\sigma^{yn1}$  хотя бы у одного из заданий на интервале встраивания  $I_{j_{[g]}}$  есть резерв времени, больший нуля.

*Утверждение 4.* Если в последовательности  $\sigma^{yn}$  ни для одного из запаздывающих заданий  $j_{[g]}$  нет предшествующих заданий  $j_{[i]}, i = \overline{1, g-1}$ , для которых  $d_{j_{[i]}} - C_{j_{[i]}} > 0, d_{j_{[i]}} > d_{j_{[g]}} - l_{j_{[g]}}$ , то не существует перестановок и встраиваний, приводящих к улучшению целевой функции.

*Утверждение 5.* Пусть в последовательности  $\sigma^{yn} j_{[g]}$  – запаздывающее задание. Уменьшение значения функционала при перемещении  $j_{[g]}$  на более ранние позиции в результате перестановок и встраиваний возможно только при выполнении одного из следующих условий:

1.  $\exists j_{[i]}, P_{min} \leq i \leq g \mid r_{j_{[i]}} > 0$ . На интервале переноса задания  $j_{[g]}$  есть задания с резервами времени, где  $P_{min}$  – позиция, на которой штраф по заданию  $j_{[g]}$  минимален (или равен нулю).

2.  $\exists j_{[q]}, q < g, d_{j_{[q]}} > C_{j_{[g]}}$ . В последовательности  $\sigma^{yn}$  на позиции  $q$ , предшествующей позиции  $g$ , есть задание с резервом времени, перестановка которого после задания  $j_{[g]}$  уменьшает

штраф по заданию  $j_{[g]}$ . Задание  $j_{[q]}$  остается незапаздывающим.

3.  $\exists j_{[q]}, q < g, C_{j_{[g]}} - l_{j_{[g]}} < d_{j_{[q]}} < C_{j_{[g]}}$ . Существует незапаздывающее задание  $j_{[q]}$ , директивный срок которого больше момента начала выполнения задания  $j_{[g]}$ , но меньше момента окончания задания  $j_{[g]}$ . При этом выполняется:

$$\min(C_{j_{[g]}} - d_{j_{[g]}}, l_{j_{[q]}}) \omega_{j_{[g]}} - (C_{j_{[g]}} - d_{j_{[q]}}) \omega_{j_{[q]}} > 0.$$

Следовательно, перестановка задания  $j_{[q]}$  после задания  $j_{[g]}$  приведет к уменьшению значения функционала, за счет использования резерва задания  $j_{[q]}$ .

4.  $\forall i \mid P_{min} \leq i \leq g \mid r_{j_{[i]}} \leq 0$ , но  $\exists j_{[k]}, k < P_{min} \mid d_{j_{[k]}} > C_{j_{[P_{min}]}}$ . На интервале перестановки задания  $j_{[g]}$  резервы отсутствуют, но существует задание  $j_{[k]}, k < P_{min}$ , директивный срок которого больше  $C_{j_{[P_{min}]}}$ , следовательно, при перестановке задания  $j_{[k]}$  на позицию  $P_{min}$  образуется резерв на интервале переноса задания  $j_{[g]}$ .

5. Условие 4 не выполняется и  $\forall i \mid P_{min} \leq i \leq g \mid r_{j_{[i]}} \leq 0$ , но  $\exists j_{[k]}, k < P_{min} \mid d_{j_{[k]}} > C_{j_{[k]}}$ ,  $d_{j_{[k]}} > C_{j_{[P_{min}]}} - l_{j_{[P_{min}]}}$ ,  $d_{j_{[k]}} > d_{j_{[g]}} - l_{j_{[g]}}$ . На интервале переноса задания  $j_{[g]}$  резервы отсутствуют, но существует незапаздывающее задание  $j_{[k]}, k < P_{min}$  с директивным сроком, большим самого позднего момента начала выполнения задания  $j_{[g]}$ , при котором задание  $j_{[g]}$  станет незапаздывающим, и, следовательно, существуют перестановки и встраивания, которые могут привести к образованию резервов на интервале переноса задания  $j_{[g]}$ .

Пусть в последовательности  $\sigma^{yn} = \{j_{[1]}, j_{[2]}, \dots, j_{[g]}\}$  запаздывающее задание  $j_{[g]}$  в результате встраивания заняло позицию  $p = m$ . Пометим его  $j_{[m]}^*$ , а полученную последовательность  $\sigma(j_{[g]})$ .

*Утверждение 6.* Запаздывающее задание  $j_{[k]}, k = \overline{m+1, g}$ , в последовательности  $\sigma(j_{[g]})$ , может быть встроено на более раннюю позицию, что приведет к улучшению целевой функции, только в том случае, если хотя бы у одного предшествующего задания есть резерв времени, либо (в случае отсутствия резервов) только в результате перестановки  $j_{[m]}^*$  после задания  $j_{[k]}$ .

*Утверждение 7.* Пусть в последовательности  $\sigma(j_{[g]})$  задания  $j_{[l]}, l = \overline{g+1, n}$ , – запаздывающие, и выполняется  $\forall j_{[l]} : l_{j_{[l]}} \leq l_{j_{[g]}}$ ,  $l = \overline{g+1, n}$ , и у всех предшествующих заданий  $j_{[i]}, i = \overline{1, g}$ ,  $r_{j_{[i]}} \leq 0$ . В этом случае в оптимальной последо-

вательности задания  $j_{[l]}$  останутся на занимаемых позициях на интервале  $\overline{g+1, n}$ .

*Последовательность*  $\sigma^{\text{сп}}$ :

*Утверждение 8.* Утверждения 1–5 справедливы для последовательности  $\sigma^{\text{сп}}$ .

*Утверждение 9.* Для последовательности  $\sigma^{\text{сп}}$  не существует перестановок и встраиваний, приводящих к улучшению целевой функции, если выполняется хотя бы одно из условий:

$$\text{а) } d_{j_{[i]}} - C_{j_{[i]}} \geq 0, \forall i = \overline{1, n};$$

$$\text{б) } d_{j_{[i]}} - C_{j_{[i]}} \leq 0, \forall i = \overline{1, l};$$

$$d_{j_{[i]}} - C_{j_{[i]}} \geq 0, \forall i = \overline{l+1, n};$$

$$\text{в) } d_{j_{[i]}} - C_{j_{[i]}} \leq 0, \forall i = \overline{1, n};$$

$$\text{г) } d_{j_{[i]}} - C_{j_{[i]}} \geq 0, \forall i = \overline{1, l};$$

$$d_{j_{[i]}} - C_{j_{[i]}} \leq 0, \forall i = \overline{l+1, n}.$$

*Текущая последовательность*  $\sigma^k$ :

В текущей подпоследовательности  $\sigma^k$  звездочкой «\*» обозначены задания, которые были запаздывающими в последовательности  $\sigma^{\text{сп}}$  и в результате выполнения встраиваний переместились на более ранние позиции, используя резервы времени предшествующих заданий. Двумя звездочками «\*\*» помечены задания, ранее помеченные «\*», которые в результате последующих перестановок переместились на более поздние позиции, освободив часть использованных резервов.

Число *текущих итераций* предлагаемого алгоритма определяется количеством конкурирующих заданий в последовательности  $\sigma^{\text{сп}}$ . На каждой итерации определяется возможность использования резервов времени предшествующих заданий очередным конкурирующим заданием последовательности  $\sigma^{\text{сп}}$  и строится оптимальное расписание рассматриваемой подпоследовательности. На первой итерации строится оптимальное расписание для заданий на интервале  $\overline{1, g_1}$  последовательности  $\sigma^{\text{сп}}$ , где  $j_{[g_1]}$  – первое запаздывающее конкурирующее задание. На следующей итерации рассматривается подпоследовательность заданий на интервале  $\overline{1, g_2}$ , в которой позиции  $\overline{1, g_1}$  занимает оптимальная подпоследовательность, полученная на первой итерации; позиции  $\overline{g_1+1, g_2}$  занимают задания, стоящие на этих позициях в последовательности  $\sigma^{\text{сп}}$ ;  $j_{[g_2]}$  – следующее запаздывающее конкурирующее задание в последовательности  $\sigma^{\text{сп}}$ .

Пусть уже выполнена  $k-1$  итерация и построено оптимальное расписание для подпоследовательности заданий на интервале  $\overline{1, g_{k-1}}$ , где  $j_{[g_{k-1}]}$  – конкурирующее задание в последовательности  $\sigma^{\text{сп}}$ . Переходим к очередному конкурирующему заданию  $j_{[g_k]}$ , определенному в последовательности  $\sigma^{\text{сп}}$ , и по аналогичным правилам строим последовательность  $\sigma^k$ , включающую задания на интервале  $\overline{1, g_k}$ . Проверяется возможность уменьшения функционала за счет использования заданием  $j_{[g_k]}$  существующих резервов времени или резервов, полученных в результате перемещения на более поздние позиции заданий, ранее использовавших эти резервы на предыдущих итерациях. На каждой итерации значение функционала уменьшается или остается неизменным.

*Утверждение 10.* Если при выполнении текущей итерации запаздывающему заданию  $j_{[k]}$  в текущей подпоследовательности предшествует задание  $j_{[l]}$  меньшего приоритета, то уменьшение взвешенного запаздывания по заданию  $j_{[k]}$  возможно в результате переноса задания  $j_{[l]}$  на позицию  $k$ . При этом необходимо выполнение следующего условия:

$$\omega_{j_{[k]}} (C_{j_{[k]}} - d_{j_{[k]}}) > \omega_{j_{[l]}} \max(0, C_{j_{[k]}} - d_{j_{[l]}}).$$

Задание  $j_{[k]}$  – это задание, по которому на предыдущих шагах алгоритма была выполнена свободная перестановка, а затем в результате выполнения процедуры встраивания очередного запаздывающего задания появилось запаздывание по заданию  $j_{[k]}$ .

*Утверждение 11.* Если на итерации  $k$  в последовательности  $\sigma^k$  для запаздывающего задания  $j_{[g]}$  существует предшествующее задание  $j_{[l]}$  такое, что выполняются условия

$$d_{j_{[l]}} > C_{j_{[g]}} - l_{j_{[g]}},$$

$$\omega_{j_{[g]}} \min(l_{j_{[l]}}, C_{j_{[g]}} - d_{j_{[g]}}) > \omega_{j_{[l]}} (C_{j_{[g]}} - d_{j_{[l]}}), \quad (2)$$
 то перестановка задания  $j_{[l]}$  на позицию  $g$  приводит к уменьшению целевой функции.

*Утверждение 12.* Пусть в последовательности  $\sigma^k$   $j_{[g]}$  – запаздывающее задание. Уменьшение значения функционала при перемещении  $j_{[g]}$  на более раннюю позицию возможно только при выполнении одного из следующих условий:

1.  $\exists j_{[i]}, P_{\min} \leq i \leq g \mid r_{j_{[i]}} > 0$ . На интервале переноса задания  $j_{[g]}$  есть задания с резервами времени, где  $P_{\min}$  – позиция, на которой штраф по заданию  $j_{[g]}$  минимален (или равен нулю).

2.  $\exists j_{[l]}, l < g, d_{j_{[l]}} > C_{j_{[g]}} - l_{j_{[g]}}$  и выполняется условие (2): существует незапаздывающее задание  $j_{[l]}$ , директивный срок которого больше момента начала выполнения задания  $j_{[g]}$ . При этом выполняется условие (2).

3.  $\forall i | P_{\min} \leq i \leq g | r_{j_{[i]}} \leq 0$ , но  $\exists j_{[k]}, k < P_{\min} | d_{j_{[k]}} > C_{j_{[P_{\min}]}}$ .

4.  $\forall i | P_{\min} \leq i \leq g | r_{j_{[i]}} \leq 0$ , но  $\exists j_{[k]}, k < P_{\min}, d_{j_{[k]}} > C_{j_{[k]}}, d_{j_{[k]}} > C_{j_{[P_{\min}]}} - l_{j_{[P_{\min}]}}$ ,  $d_{j_{[k]}} > d_{j_{[g]}} - l_{j_{[g]}}$ .

5.  $\forall i | i = \overline{1, g-1}, r_{j_{[i]}} \leq 0$ , но  $\exists j_{[l]}, l < g, P_{j_{[i]}} \leq P_{j_{[g]}}$ .

6.  $\forall i | i = \overline{1, g-1}, r_{j_{[i]}} \leq 0$ , но  $\exists j_{[m]^*} (j_{[m]^{**}}), m < g$ .

*Утверждение 13.* Запаздывающее задание  $j_{[g]}$  на  $k$ -й итерации может занять более раннюю позицию, что приведет к улучшению целевой функции, только в том случае, если хотя бы у одного задания  $j_{[k]}$  на интервале  $\overline{1, g-1}$  есть резерв и выполняется:  $d_{j_{[k]}} > d_{j_{[g]}} - l_{j_{[g]}}$ , либо (в случае отсутствия резервов) на интервале  $\overline{1, g-1}$  есть задания с метками или задания с меньшим (или равным) значением приоритета.

*Утверждение 14.* Пусть в последовательности  $\sigma^{\text{сп}}$  для конкурирующих заданий  $j_{[k]}, j_{[r]} \in J$  выполняется:  $l_{j_{[k]}} \leq l_{j_{[r]}}$ ,  $\omega_{j_{[k]}} \geq \omega_{j_{[r]}}$ ,  $d_{j_{[k]}} \leq d_{j_{[r]}}$ . Тогда в оптимальном расписании задание  $j_{[k]}$  будет предшествовать заданию  $j_{[r]}$ .

*Утверждение 15.* Неконкурирующие задания в последовательности  $\sigma^k$  не могут занять более ранние позиции, чем позиции, занимаемые ими в  $\sigma^{\text{сп}}$ .

*Утверждение 16.* Пусть уже построена оптимальная подпоследовательность на интервале  $\overline{1, g-1}$ . Запаздывающее конкурирующее задание  $j_{[g]}$  на  $k$ -й итерации не может быть перемещено на более раннюю позицию и исключается из числа конкурентоспособных, если на интервале  $\overline{1, g-1}$  ни у одного из заданий нет резервов, ни одному из запаздывающих заданий не предшествуют задания с меньшим или равным значением приоритета, и для всех помеченных заданий выполняется:

$$\omega_{j_{[i]}}^* \geq \omega_{j_{[g]}}^*, l_{j_{[i]}}^* \leq l_{j_{[g]}}^*, i = \overline{1, g-1}.$$

*Утверждение 17.* Если в последовательности  $\sigma^k$  конкурирующее задание  $j_{[g]}$  в результате выполнения перестановок и встраиваний не использовало существующие резервы, минимальное значение функционала соответствует позиции  $g$ , занимае-

мой этим заданием; на интервале  $\overline{1, g-1}$  отсутствуют непомеченные задания с меньшим или равным значением приоритета и нет заданий, помеченных \*, \*\*, то задание  $j_{[g]}$  исключается из множества конкурирующих заданий.

Следующие утверждения позволяют определить и исключить бесперспективные перестановки и встраивания.

*Утверждение 18.* Пусть уже выполнено  $k$  итераций. Переходим к очередному конкурирующему заданию  $j_{[g]}$ . Если при выполнении итерации оптимизации для конкурирующего задания  $j_{[g]}$  для всех предшествующих запаздывающих заданий  $j_{[i]}$  выполняется:  $d_{j_{[i]}} \leq d_{j_{[g]}} - l_{j_{[g]}}$ , или на интервале  $\overline{1, g-1}$  резервы отсутствуют, а для заданий, помеченных \*, \*\*, выполняется  $l_{j_{[i]}}^* \leq l_{j_{[g]}}^*$ , то задание  $j_{[g]}$  остается на занимаемой позиции и в оптимальном расписании не может занять позицию меньше  $g$ .

*Утверждение 19.* Если в последовательности  $\sigma^k$  при встраивании запаздывающего задания  $j_{[g]}$  на интервале  $\overline{1, g-1}$  нет предшествующих заданий, для которых  $d_{j_{[i]}} > d_{j_{[g]}} - l_{j_{[g]}}$  и  $d_{j_{[i]}} > C_{j_{[i]}}$ , но есть задания  $j_{[m]^*} (j_{[m]^{**}})$ , для которых  $l_{j_{[m]}^*} (l_{j_{[m]}^{**}}) \geq l_{j_{[g]}}$ , то оптимизация осуществляется за счет резервов, освобожденных заданиями  $j_{[m]^*} (j_{[m]^{**}})$  при встраивании их после задания  $j_{[g]}$ . При этом необходимо выполнение следующего условия:

$$C_{j_{[g]}} - d_{j_{[m]}^*} < \sum_{i=m}^g \omega_{j_{[i]}} \max(0, C_{j_{[i]}} - d_{j_{[i]}}). \quad (3)$$

*Утверждение 20.* Если очередное конкурирующее задание  $j_{[g]}$  в результате выполнения для него итерации оптимизации, определяющей возможность использования резервов этим заданием, заняло позицию  $k > p$ , то следующие за ним конкурирующие задания  $j_{[l]}$ , для которых  $d_{j_{[l]}} \geq d_{j_{[g]}}$ ,  $l_{j_{[l]}} \geq l_{j_{[g]}}$ , в оптимальном расписании не смогут занять позицию меньшую, чем  $k+1$ .

*Утверждение 21.* Пусть в последовательности  $\sigma^k$  для запаздывающих заданий  $j_{[k]}, j_{[r]} \in J$  выполняется:  $l_{j_{[k]}} \leq l_{j_{[r]}}$ ,  $d_{j_{[k]}} \leq d_{j_{[r]}}$ ,  $\omega_{j_{[k]}} \geq \omega_{j_{[r]}}$ . Тогда в оптимальном расписании задание  $j_{[k]}$  будет предшествовать заданию  $j_{[r]}$  [6].

*Утверждение 22.* Если конкурирующее задание  $j_{[g]}$  в результате выполнения для него итерации оптимизации осталось на исходной позиции, и все задания на позициях  $\overline{g+1, n}$  запаз-

дывающие, то задания, для которых  $d_{j_{[i]}} \geq d_{j_{[g]}}$ ,  $l_{j_{[i]}} \geq l_{j_{[g]}}$ ,  $i \in \overline{g+1, n}$ , исключаются из числа конкурирующих за резервы предшествующих заданий и в оптимальном расписании не смогут занимать более ранние позиции.

**Утверждение 23.** Запаздывающее задание  $j_{[g]}$  на  $k$ -й итерации может занять более раннюю позицию, что приведет к улучшению целевой функции, только в том случае, если хотя бы у одного задания  $j_{[k]}$  на интервале  $\overline{1, g-1}$  есть резерв и выполняется:  $d_{j_{[k]}} > d_{j_{[g]}} - l_{j_{[g]}}$ , либо (в случае отсутствия резервов) на интервале  $\overline{1, g-1}$  есть задания с метками или у некоторого задания  $j_{[i]}$  на этом интервале  $P_{j_{[i]}} \leq P_{j_{[g]}}$ .

В утверждениях 24, 25 сформулированы признаки оптимальности полученного решения.

**Утверждение 24.** Если в последовательности  $\sigma^k$  конкурирующее задание  $j_{[g]}$  в результате выполнения для него итерации оптимизации не использовало существующие резервы, минимальное значение функционала соответствует позиции  $g$ , занимаемой этим заданием, и  $\forall j_{[r]}$ ,  $r = \overline{g+1, n}$ ,  $d_{j_{[r]}} \geq d_{j_{[g]}}$ ,  $l_{j_{[r]}} \geq l_{j_{[g]}}$  и  $C_{j_{[r]}} \geq d_{j_{[r]}}$ , то задания  $j_{[r]}$  исключаются из множества конкурирующих, а последовательность  $\sigma^k$  оптимальна.

**Утверждение 25.** Пусть на итерации  $k$ , выполняющейся для очередного конкурирующего запаздывающего задания  $j_{[g]}$ , для всех  $j_{[l]}$ ,  $l = \overline{1, g-1}$ , справедливо  $d_{j_{[l]}} \leq C_{j_{[l]}}$ , для всех  $j_{[l]}^*$  ( $j_{[l]}^{**}$ ) выполняется  $l_{j_{[l]}}^* \left( l_{j_{[l]}}^{**} \right) \leq l_{j_{[g]}}$ , и на интервале  $\overline{g, n}$  резервы отсутствуют. Тогда запаздывающие задания, занимающие позиции  $\overline{g, n}$ , исключаются из множества конкурирующих, а последовательности  $\sigma^k$  отвечает оптимальное значение функционала.

**Утверждение 26.** Пусть на итерации  $k$ , выполняемой для очередного конкурирующего задания  $j_{[g]}$ , на интервале его встраивания есть задание  $j_{[r]}$ , для которого  $d_{j_{[r]}} < d_{j_{[g]}}$ ,  $\omega_{j_{[r]}} > \omega_{j_{[g]}}$  и  $l_{j_{[r]}} < l_{j_{[g]}}$ , и задание  $j_{[r]}$  запаздывает, тогда осуществляется декомпозиция рассматриваемой подпоследовательности, и при выполнении итерации оптимизации для задания  $j_{[g]}$  задания на интервале  $\overline{1, r}$  не рассматриваются, а оптимизация осуществляется на интервале  $\overline{r+1, g-1}$ .

**Утверждение 27.** Алгоритм решения задачи является конечным и точным.

Алгоритм основан на направленных перестановках, реализующих использование запаздывающими заданиями резервов незапаздывающих заданий

### Характеристика полиномиальной составляющей ПДС-алгоритма

Если в процессе решения произвольной индивидуальной задачи выполняются строго определенные для полиномиальной составляющей алгоритма логико-аналитические условия, полученные в результате исследования свойств задачи, то данная произвольная индивидуальная задача решается этим подалгоритмом точно (сложность решения – полином от размерности произвольной индивидуальной задачи). В этом случае точное решение достигается выполнением только тех блоков алгоритма, которые имеют полиномиальную трудоемкость.

В следующих утверждениях [3] рассматриваются отдельные случаи, когда оптимальное решение задачи достигается полиномиальной составляющей алгоритма.

**Утверждение 28.** Пусть в последовательности  $\sigma^{yn}$   $j_{[g]}$  – первое запаздывающее задание, и выполняется:  $\max r(\sigma^{yn}) \leq l_{j_{[g]}}$ , задания на интервале  $\overline{g+1, n}$  запаздывают, и  $\forall j_{[r]}$ ,  $r = \overline{g+1, n}$ ,  $l_{j_{[r]}} \geq l_{j_{[g]}}$ . В этом случае оптимальное значение функционала достигается за полиномиальное время с трудоемкостью, определяемой функцией  $O(n \log n)$ .

**Утверждение 29.** Пусть  $j_{[g]}$  – первое запаздывающее задание в последовательности  $\sigma^{cn}$ . Если в этой последовательности на интервале  $\overline{1, p-1}$ , где  $p$  – позиция встраивания задания  $j_{[g]}$ , резервы отсутствуют, задание  $j_{[g]}$  на позиции  $p$  остается запаздывающим, на интервале  $\overline{p, g-1}$  для каждой пары заданий  $j_{[s]}$ ,  $j_{[t]}$ ,  $s < t$ , выполняется  $l_{j_{[s]}} \leq l_{j_{[t]}}$ ,  $d_{j_{[s]}} \leq d_{j_{[t]}}$ ,  $\omega_{j_{[s]}} > \omega_{j_{[t]}}$ , и  $\forall j_{[r]}$ ,  $r = \overline{g+1, n}$ , выполняется  $d_{j_{[r]}} \leq C_{j_{[r]}}$ ,  $d_{j_{[g]}} \geq d_{j_{[r]}}$ , то задача решается за полиномиальное время с трудоемкостью, не превышающей  $O(n^2)$ .

**Утверждение 30.** Пусть  $j_{[g]}$  – первое запаздывающее задание в последовательности  $\sigma^{cn}$ . Если на интервале  $\overline{1, g-1}$  все задания упорядочены по невозрастанию их приоритетов;  $\max r_i \leq l_{j_{[g]}}$ ,  $i = \overline{1, g-1}$  и  $\forall j_{[s]}$ ,  $s = \overline{g+1, n}$ ,  $d_{j_{[s]}} \leq C_{j_{[s]}}$ ,

$l_{j_{s1}} \geq l_{j_{g1}}$ , то задача решается за полиномиальное время с трудоемкостью  $O(n^2)$ .

**Утверждение 31.** Если условия утверждений 5 или 12 (для всех конкурирующих заданий) не выполняются, или выполняются условия хотя бы одного из утверждений 9 или 28–30, то задача решается посредством полиномиальной составляющей предложенного алгоритма.

Общая трудоемкость выполнения полиномиальной составляющей ПДС-алгоритма не превышает  $O(n^3)$ .

### Характеристика экспоненциальной составляющей ПДС-алгоритма

Если не выполняются  $p$ -условия, то полиномиальная составляющая алгоритма не реализована, и решение задачи осуществляется точным экспоненциальным подалгоритмом, в процессе выполнения которого исключаются заведомо бесперспективные варианты решений, что позволяет резко сократить область поиска оптимального решения.

Эффективность экспоненциальной составляющей алгоритма в целом определяется следующими факторами:

1. На этапе оптимизации последовательность  $\sigma^{сп}$  декомпозируется на подпоследовательности меньшего размера. Оптимизация осуществляется на подпоследовательности, ограниченной позицией встраивания очередного конкурирующего задания и позицией, занимаемой этим за-

данием в последовательности  $\sigma^{сп}$ . В эту подпоследовательность в процессе решения могут быть включены только задания, образующие резервы на интервале встраивания рассматриваемого конкурирующего задания. Таким образом, реализуется декомпозиция задачи на подзадачи меньшего размера.

2. В процессе решения задачи проверяются условия утверждений 22, 24, 25, позволяющие часть заданий исключить из множества конкурирующих. В утверждениях 24, 25 сформулированы признаки оптимальности полученных решений, что существенно сокращает число выполняемых итераций оптимизации, а следовательно, и время решения задачи. Только для специальных случаев параметров задачи можно выйти на полный перебор вариантов для отдельных подпоследовательностей.

### Выводы

Определены теоретически обоснованные признаки оптимальности решений, получаемых в процессе выполнения алгоритма ( $p$ -условия), на основании которых построена полиномиальная составляющая алгоритма. Если не выполняются  $p$ -условия, то решение задачи осуществляется точным экспоненциальным подалгоритмом, в процессе выполнения которого исключаются заведомо бесперспективные варианты решений, что позволяет резко сократить область поиска оптимального решения.

### Список литературы

1. Конструктивные полиномиальные алгоритмы решения индивидуальных задач из класса NP. / А.А.Павлов, А.Б.Литвин, Е.Б.Мисюра, Л.А.Павлова, В.И.Родионов, под редакцией А.А.Павлова.– К.: Техника, 1993.– 126 с.
2. Pavlov A., Pavlova L. PDC-algorithms for intractable combinatorial problems. Theory and methodology of design.– Uzhhorod, «Karpatskij region» shelf №15, 1998.– 320 pp.
3. Згуровский М.З., Павлов А.А. Принятие решений в сетевых системах с ограниченными ресурсами: Монография. – К.: Наукова думка. – 2010. – 573 с.
4. Lawler E. L. A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness // Annual Discrete Math. – 1977, №1. – pp.331-342.
5. Павлов А.А., Мисюра Е.Б. Новый подход к решению задачи «Минимизация суммарного взвешенного опоздания при выполнении независимых заданий с директивными сроками одним прибором» // Системні дослідження та інформаційні технології. – 2002. – №2. – С.3-32.
6. C.N.Potts and L.N.Van Wassenhove, A decomposition algorithm for the single machine total tardiness problem / Oper. Res. Lett. 1, 177–181 (1982).