УДК 519.854.2

ЗГУРОВСКИЙ М. З.,
ПАВЛОВ А. А.

Теоретические свойства ПДС-алгоритма для задачи минимизации суммарного взвешенного запаздывания на одном приборе

В статье исследуются свойства одной из наиболее известных *NP*-трудных в сильном смысле задач комбинаторной оптимизации, формулируются и обосновываются утверждения, необходимые для построения ПДС-алгоритма ее решения: достаточные признаки оптимальности получаемых решений, условия исключения конкурирующих заданий, правила отсечения бесперспективных перестановок и встраиваний. Показаны свойства полиномиальной и экспоненциальной составляющих ПДС-алгоритма, доказывается его конечность и оптимальность.

In this paper we research the properties of one of the most well-known *NP*-hard in the strong sense problems of combinatorial optimization. We formulate and substantiate the statements to construct the PSC-algorithm for its solving: sufficient signs of optimality of the obtained solutions, conditions for excluding competing tasks, and rules for cutting off unpromising permutations. We show properties of the polynomial and exponential components of the PSC-algorithm, prove its finiteness and optimality.

Ключевые слова: Составление расписаний, ПДС-алгоритмы, NP-трудные задачи

Введение

Рассматривается задача построения расписания для одного прибора с разными директивными сроками с фиксированным моментом запуска прибора, которое минимизирует суммарное взвешенное запаздывание заданий относительно их директивных сроков (МВЗ). Задача является *NP*-трудной в сильном смысле и одной из наиболее известных труднорешаемых задач комбинаторной оптимизации. Построен ПДС-алгоритм решения задачи, теоретическое обоснование и структура которого является обобщением результатов, приведенных в главе 1 [1] для аналогичной задачи с одинаковыми весами (минимизация суммарного запаздывания при выполнении заданий одним прибором – МСЗ). Помимо достаточных признаков оптимальности допустимого расписания, проверяемых полиномиальным подалгоритмом, ПДС-алгоритм содержит оригинальный точный экспоненциальный подалгоритм. Этим же свойством обладает ПДС-алгоритм для задачи МСЗ.

Определены теоретически обоснованные достаточные 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.

Постановка задачи

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

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

где C_j – момент завершения задания j.

Наиболее эффективным из известных точных алгоритмов для задачи МВЗ является алгоритм динамического программирования Танака и др. [2], с его помощью решаются индивидуальные задачи с размерностью до 300 заданий в среднем за 350 секунд на ПК с 2.4 ГГц процессором Pentium 4.

Разработанный авторами ПДС-алгоритм решения рассматриваемой задачи основан на результатах, опубликованных в [3], и ПДС-алгоритме решения задачи МСЗ [1].

Основные теоретические положения

Введем следующие обозначения и определения. 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 (EFSR – extraction and forward-shifted reinsertion – извлечение и повторная вставка со сдвигом вперед [3]) называется процедура переноса задания $j_{[g]}$ на позицию k $(k > g)$ и, одновременно, заданий, занимающих позиции $g + 1$, $g + 2$,..., $k - 1$, k на позиции g, $g + 1$,..., $k - 2$, $k - 1$, соответственно.

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

Определение 5. Встраиванием (EBSR – extraction and backward-shifted reinsertion – извлечение и повторная вставка со сдвигом назад [3]) называется процедура переноса задания $j_{[g]}$ на позицию p $(g > p)$ и, одновременно, заданий p, $p + 1$,..., $g - 2$, $g - 1$ на позиции $p + 1$, $p + 2$, ..., $g - 1$, g, соответственно.

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

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

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 σ, если для него выполняется условие $d_{j_{[g]}} < C_{j_{[g]}}$.

Определение 8. Последовательностью $\sigma^{\text{уп}}$ (сигма упорядоченная) называется последовательность заданий множества J, $j = \overline{1, n}$, в которой задания упорядочены по невозрастанию приоритетов p_j, т. е. $\forall j,\ i,\ j < i$: $p_j \ge p_i$, а при $p_j = p_i$, $d_j \le d_i$.

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

Таким образом, свободная перестановка задания k выполняется на позицию с максимальным номером, на которой задание k не будет запаздывать. Очевидно, что в результате выполнения всех свободных перестановок в последовательности $\sigma^{\text{уп}}$ значение целевой функции уменьшается.

При выполнении одной свободной перестановки значение целевой функции уменьшается на величину $\sum_{i \in Z} \omega_{j_{[i]}} \min\left(l_{j_{[k]}}, C_{j_{[i]}} - d_{j_{[i]}}\right)$, где

$$Z = \left\{ j_{[i]} \middle| C_{j_{[i]}} > d_{j_{[i]}},\ i = \overline{k+1, q} \right\}.$$

Примечание к определению 9. При выполнении нескольких свободных перестановок, во избежание зацикливания, начинать следует с задания, имеющего максимальный директивный срок, т. е. просматривать незапаздывающие задания по убыванию их директивных сроков.

Определение 10. Последовательность заданий, полученную в результате выполнения всех свободных перестановок в последовательности $\sigma^{\text{уп}}$, назовем $\sigma^{\text{сп}}$.

Теорема 1 (гл. 3 [4]). В оптимальном (по критерию минимизации суммарного запаздывания) расписании суммарное запаздывание равно нулю тогда и только тогда, когда в расписании, в котором задания упорядочены по неубыванию их директивных сроков, суммарное запаздывание равно нулю.

Следствие 1. *Достаточный признак оптимальности №1.* Если в расписании, в котором задания упорядочены по неубыванию директивных сроков, нет запаздывающих заданий, то в нем также суммарное взвешенное запаздывание равно нулю, и такое расписание оптимально.

Следствие 2. Если достаточный признак оптимальности №1 не выполняется, то в оптимальном расписании суммарное взвешенное запаздывание больше нуля.

В процессе решения задачи в результате выполнения направленных перестановок и встраиваний осуществляется последовательный переход от последовательности $\sigma^{\text{уп}}$ к последовательности $\sigma^{\text{сп}}$ и затем к текущим подпоследовательностям σ^k. В результате выполнения итераций оптимизации строится оптимальная последовательность на всем множестве заданий.

В следующих утверждениях сформулированы свойства и доказаны достаточные признаки оптимальности каждой из рассматриваемых допустимых последовательностей, обоснованы правила выполняемых перестановок.

Последовательность $\sigma^{уп}$

Достаточный признак оптимальности №2.

Утверждение 1. Если в последовательности $\sigma^{уп}$ запаздывающим заданиям не предшествуют задания с резервом времени, то не существует переносов (перестановок и встраиваний) заданий, приводящих к уменьшению функционала, и последовательность $\sigma^{уп}$ оптимальна на всем множестве заданий.

Достаточный признак оптимальности №3.

Утверждение 2. Если в последовательности $\sigma^{уп}$ 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 уменьшению функционала, и последовательность $\sigma^{уп}$ оптимальна на всем множестве заданий.

Утверждение 3. Пусть в последовательности $\sigma^{уп}$ интервал встраивания задания $j_{[g]}$ составляет $I_{j_{[g]}}=\overline{p,g-1}$. Дополнительные резервы на интервале встраивания для задания $j_{[g]}$ могут образовать только те задания $j_{[i]}$, $i\in\overline{1,p-1}$, для которых справедливо:

$$d_{j_{[i]}}>d_{j_{[g]}}-l_{j_{[g]}},\quad d_{j_{[i]}}>C_{j_{[i]}},\quad d_{j_{[i]}}>C_{j_{[p-1]}}.$$

Утверждение 4. Пусть в последовательности $\sigma^{уп}$ $j_{[g]}$ – 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. $\exists j_{[i]},\ p\le i\le g\ |\ R_{j_{[i]}}>0$. На интервале встраивания задания $j_{[g]}$ есть задания с резервами времени, где p – позиция, на которой взвешенное запаздывание по заданию $j_{[g]}$ минимально (или равно нулю).

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

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

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

4. $\forall i\ |\ p\le i\le g\ |\ R_{j_{[i]}}\le 0$, но $\exists j_{[k]}$, $k<p\ |\ d_{j_{[k]}}>C_{j_{[p]}}$. На интервале встраивания задания $j_{[g]}$ резервы отсутствуют, но существует задание $j_{[k]}$, $k<p$, директивный срок которого больше $C_{j_{[p]}}$, следовательно, при перестановке задания $j_{[k]}$ на позицию p образуется резерв на интервале встраивания задания $j_{[g]}$.

5. Условие 4 не выполняется и $\forall i\ |\ p\le i\le g\ |\ R_{j_{[i]}}\le 0$, но $\exists j_{[k]}$, $k<p\ |\ d_{j_{[k]}}>C_{j_{[k]}}$, $d_{j_{[k]}}>C_{j_{[p]}}-l_{j_{[p]}}$, $d_{j_{[k]}}>d_{j_{[g]}}-l_{j_{[g]}}$. На интервале встраивания задания $j_{[g]}$ резервы отсутствуют, но существует незапаздывающее задание $j_{[k]}$, $k<p$, 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 встраивания задания $j_{[g]}$.

Следствие. Достаточный признак оптимальности №4. Пусть в последовательности $\sigma^{уп}$ число запаздывающих заданий $n_з\ge 1$. Тогда, если в последовательности $\sigma^{уп}$ ни для одного из запаздывающих заданий $j_{[g]}$ нет предшествующих заданий $j_{[s]}$, $s<g$, для которых выполняется хотя бы одно из условий 1–5 утверждения 4, то последовательности $\sigma^{уп}$ отвечает оптимальное значение функционала.

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

Последовательность $\sigma^{сп}$

Утверждение 5. Утверждения 1–4 справедливы для последовательности $\sigma^{сп}$.

Достаточный признак оптимальности №5.

Утверждение 1. Для последовательности $\sigma^{\text{сп}}$ не существует перестановок и встраиваний, приводящих к уменьшению функционала, и последовательность $\sigma^{\text{сп}}$ оптимальна на всем множестве заданий, если выполняется хотя бы одно из условий:

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

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

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

г) $d_{j_{[i]}} - C_{j_{[i]}} \ge 0,\ \forall i = \overline{1,l}$ (множество *R*), $d_{j_{[i]}} - C_{j_{[i]}} \le 0,\ \forall k = \overline{l+1,n}$ (множество *Z*), и выполняется: $d_{j_{[k]}} \le d_{j_{[g]}} - l_{j_{[g]}}\ \forall k \in R, \forall g \in Z$

Следствие. При выполнении хотя бы одного из условий, сформулированных в утверждении 6, в последовательности $\sigma^{\text{сп}}$ достигается оптимальное значение функционала.

Введем следующие определения.

Определение 11. Запаздывающее задание $j_{[g]}$ в последовательности $\sigma^{\text{сп}}$ называется конкурирующим, если в этой последовательности найдется хотя бы одно предшествующее задание $j_{[l]}$, для которого выполняются условия $d_{j_{[l]}} > d_{j_{[g]}} - l_{j_{[g]}}$ и $d_{j_{[l]}} - C_{j_{[l]}} > 0$.

Конкурирующие задания в последовательности $\sigma^{\text{сп}}$ упорядочены по невозрастанию их приоритетов, так как последовательность $\sigma^{\text{сп}}$ получена из последовательности $\sigma^{\text{уп}}$ посредством выполнения свободных перестановок незапаздывающих заданий, которые не меняют порядок выполнения запаздывающих заданий.

Определение 12. Задания $j_{[i]}$, для которых в последовательности $\sigma^{\text{сп}}$ выполнялось $d_{j_{[i]}} - C_{j_{[i]}} \ge 0$, и которые в результате выполнения процедур встраивания запаздывающих заданий на более ранние позиции, в свою очередь, стали запаздывающими, называются порожденными запаздывающими заданиями.

Примечание. Конкурирующие запаздывающие задания определены в последовательности $\sigma^{\text{сп}}$, а у порожденных запаздывающих заданий запаздывание относительно их директивных сроков возникает в процессе выполнения процедур встраивания конкурирующих заданий.

Итерации оптимизации

Алгоритм состоит из ряда однотипных итераций. На каждой итерации определяется возможность использования резервов времени предшествующих заданий очередным конкурирующим заданием последовательности $\sigma^{\text{сп}}$ и строится оптимальное расписание для заданий 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.

В соответствии с определением 11, уменьшить взвешенное запаздывание по неконкурирующим заданиям невозможно, поэтому последовательность $\sigma^{\text{сп}}$ на интервале $\overline{1,g_1-1}$ оптимальна.

На второй 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 $\sigma^{\text{сп}}$.

Определение 13. На текущей итерации оптимизации *k* строится оптимальное расписание на текущей подпоследовательности σ^k, состоящей из оптимальной подпоследовательности, полученной на предыдущей итерации *k* – 1, выполняемой для конкурирующего задания $j_{[g_{k-1}]}$, очередного конкурирующего задания $j_{[g_k]}$ из последовательности $\sigma^{\text{сп}}$, приоритет которого не превышает приоритета заданий текущей подпоследовательности по построению последовательности $\sigma^{\text{сп}}$, и заданий, находящихся в последовательности $\sigma^{\text{сп}}$ на позициях между конкурирующими заданиями $j_{[g_{k-1}]}$ и $j_{[g_k]}$.

Примечание 1. На позициях между конкурирующими заданиями $j_{[g_{k-1}]}$ и $j_{[g_k]}$ могут находиться незапаздывающие задания и неконкурирующие запаздывающие задания. Подпоследовательность, включающая эти задания, также является оптимальной, так как, в соответствии с определением 11, уменьшить взвешенное запаздывание по неконкурирующим заданиям невозможно.

Примечание 2. Число итераций оптимизации определяется числом конкурирующих заданий в последовательности $\sigma^{\text{сп}}$, часть из которых, как будет показано далее, может быть исключена из множества конкурирующих в процессе выполнения итерации оптимизации. Таким образом, общее количество итераций является конечным (верхняя граница количества итераций – общее число заданий) в силу того что, по построению алгоритма, конкурирующие задания, для которых выполнена итерация оптимизации, исключаются из множества конкурирующих и повторно не включаются в него. На следующих итерациях они могут рассматриваться как порожденные запаздывающие задания.

Пусть уже выполнена k – 1 итерация и построена оптимальная подпоследовательность на множестве заданий на интервале $\overline{1, g_{k-1}}$, где $j_{[g_{k-1}]}$ – конкурирующее задание в последовательности $\sigma^{сп}$. Переходим по правилам, сформулированным в определении 13, к очередному конкурирующему заданию $j_{[g_k]}$, определенному в последовательности $\sigma^{сп}$, и по аналогичным правилам строим последовательность σ^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.

Лемма 1. Пусть уже выполнена k – 1 итерация и построена оптимальная подпоследовательность на множестве заданий на интервале $\overline{1, g-1}$. Тогда для получения оптимальной подпоследовательности на текущей итерации для очередного конкурирующего задания $j_{[g]}$ необходимо и достаточно получить подпоследовательность, в которой достигается минимально возможное увеличение суммарного взвешенного запаздывания по сравнению с оптимальным значением функционала на предыдущей итерации.

Следствие. Таким образом, для получения оптимальной подпоследовательности на текущей итерации необходимо и достаточно на этой итерации получить подпоследовательность, в которой увеличение суммарного взвешенного запаздывания по сравнению с оптимальным значением функционала, полученным на предыдущей итерации, минимально возможное.

С целью достижения минимально возможного увеличения суммарного взвешенного запаздывания, конкурирующее задание $j_{[g]}$ встраивается на позицию, на которой взвешенное запаздывание по нему будет минимальным или равным нулю, а затем выполняется пошаговая оптимизация для каждого запаздывающего задания на интервалах встраивания или перестановок, пока не будет получено минимально возможное увеличение суммарного взвешенного запаздывания.

Пусть уже выполнена k – 1 итерация. Переход к очередному конкурирующему заданию определяет начало новой итерации.

Текущая итерация

Справедливы следующие утверждения.

Утверждение 7. 1. Утверждения 1–4 верны и для подпоследовательности σ^k.

2. Если при выполнении текущей итерации после выполнения процедуры встраивания задания $j_{[g]}$ запаздывающему заданию $j_{[q]}$ в текущей подпоследовательности σ^k предшествует задание $j_{[l]}$ и выполняется $p_{j_{[l]}} \le p_{j_{[q]}}$, $p_{j_{[q]}} > p_{j_{[g]}}$, $q < g$, то уменьшение значения функционала возможно в результате переноса задания $j_{[l]}$ на позицию q. При этом необходимо выполнение условия:

$$\begin{aligned}\sum_{i=l}^{q} \omega_{j_{[i]}} \max\left(0, C_{j_{[i]}} - d_{j_{[i]}}\right) > \\ > \sum_{i=l+1}^{q} \omega_{j_{[i]}} \max\left(0, C_{j_{[i]}} - l_{j_{[l]}} - d_{j_{[i]}}\right) + \\ + \omega_{j_{[l]}} \max\left(0, C_{j_{[q]}} - d_{j_{[l]}}\right).\end{aligned} \quad (2)$$

Утверждение 8. Если на итерации k в текущей подпоследовательности σ^k для запаздывающего задания $j_{[g]}$ существует предшествующее незапаздывающее задание $j_{[l]}$ такое, что выполняются условия

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

$$\omega_{j_{[g]}} \min\left(l_{j_{[l]}}, C_{j_{[g]}} - d_{j_{[g]}}\right) > \omega_{j_{[l]}}\left(C_{j_{[g]}} - d_{j_{[l]}}\right), \quad (3)$$

то перестановка задания $j_{[l]}$ на позицию g приводит к уменьшению целевой функции. Задание $j_{[g]}$ после перестановки помечается звездочкой.

Утверждение 9. Пусть уже выполнена k – 1 итерация. Переходим по правилам, сформулированным в определении 13, к очередному конкурирующему заданию $j_{[g]}$ и к итерации k. Необходимые условия для встраивания конкурирующего задания $j_{[g]}$ в подпоследовательности σ^k на позицию p, где p – позиция, на которой запаздывание по заданию $j_{[g]}$ минимально или равно нулю:

а) наличие резервов на интервале встраивания задания $j_{[g]}$ у заданий $j_{[i]}$, для которых $d_{j_{[i]}} > d_{j_{[g]}} - l_{j_{[g]}}$, $j_{[i]} \in I_{j_{[g]}}$;

б) если резервы на интервале $\overline{p, g-1}$ отсутствуют, наличие резервов на позициях $\overline{1, p-1}$ у заданий $j_{[i]}$, для которых

$$d_{j_{[i]}} > d_{j_{[g]}} - l_{j_{[g]}},\ d_{j_{[i]}} > C_{j_{[i]}},\ d_{j_{[i]}} > C_{j_{[p-1]}};$$

в) если резервы на интервале $\overline{1, g-1}$ отсутствуют, наличие на этом интервале заданий $j_{[i]}$, помеченных «*» («**»).

Примечание. Если в случае (в) задания, помеченные звездочками, находятся на позиции, большей чем p, то определяется новая позиция встраивания $p^{н} > p$ для задания $j_{[g]}$, и аналогичным образом проверяются необходимые условия для встраивания задания $j_{[g]}$ на позицию $p^{н}$.

Сформулируем необходимые условия для перемещения в подпоследовательности σ^k запаздывающих заданий на более ранние позиции, при котором значение функционала уменьшается за счет существующих и освободившихся резервов.

Утверждение 10. Пусть в текущей подпоследовательности σ^k $j_{[g]}$ – запаздывающее задание (конкурирующее или порожденное). Уменьшение значения функционала при перемещении $j_{[g]}$ на более раннюю позицию возможно только при выполнении одного из следующих условий:

1. $\exists\, j_{[i]},\ p \le i \le g \mid R_{j_{[i]}} > 0,\ d_{j_{[i]}} > d_{j_{[g]}} - l_{j_{[g]}}$. На интервале встраивания задания $j_{[g]}$ есть задания $j_{[i]}$ с резервами, для которых $d_{j_{[i]}} > d_{j_{[g]}} - l_{j_{[g]}}$, где p – позиция, на которой взвешенное запаздывание по заданию $j_{[g]}$ минимально (или равно нулю) (утверждение 2).

2. $\exists\, j_{[l]},\ l < g,\ d_{j_{[l]}} > C_{j_{[g]}} - l_{j_{[g]}}$ и выполняется условие (3): существует незапаздывающее задание $j_{[l]}$, директивный срок которого больше момента начала выполнения задания $j_{[g]}$. При этом выполняется условие (3) (утверждение 8).

3. $\forall i \mid p \le i \le g \mid R_{j_{[i]}} \le 0$, но $\exists\, j_{[k]},\ k < p \mid d_{j_{[k]}} > C_{j_{[p]}}$ (возможна свободная перестановка задания $j_{[k]}$ после задания $j_{[p]}$).

4. $\forall i \mid p \le i \le g \mid R_{j_{[i]}} \le 0$, но $\exists\, j_{[k]},\ k < p$, $d_{j_{[k]}} > C_{j_{[k]}},\ d_{j_{[k]}} > C_{j_{[p]}} - l_{j_{[p]}},\ d_{j_{[k]}} > d_{j_{[g]}} - l_{j_{[g]}}$ (утверждение 2).

5. $\forall i \mid i = \overline{1, g-1},\ R_{j_{[i]}} \le 0$, но $\exists\, j_{[l]}, j_{[r]},\ l < r < g$, $p_{j_{[l]}} \le p_{j_{[r]}}$ и выполняется условие (2), где $j_{[r]}$ – порожденное запаздывающее задание на интервале встраивания конкурирующего задания $j_{[g]}$, для которого на предыдущих шагах алгоритма была выполнена свободная перестановка (утверждение 7).

6. $\forall i \mid i = \overline{1, g-1},\ R_{j_{[i]}} \le 0$, но $\exists\, j^*_{[m]}$ ($j^{**}_{[m]}$), $m < g$.

Следствие 1. *Достаточный признак оптимальности №6.* Пусть в результате выполнения k итераций алгоритма построена оптимальная подпоследовательность σ^k на множестве заданий на интервале $\overline{1, g}$. Если для каждого из запаздывающих заданий на интервале $\overline{g+1, n}$ не выполняется ни одно из условий 1–6 утверждения 10, то они остаются на занимаемых позициях, и текущая последовательность оптимальна на всем множестве заданий.

Следствие 2. *Достаточный признак оптимальности №7* (вытекает из следствия 1 при $k = 0$). Если в последовательности $\sigma^{сп}$ для каждого из запаздывающих заданий не выполняется ни одно из условий 1–6 утверждения 10, то последовательность $\sigma^{сп}$ оптимальна на всем множестве заданий.

Утверждение 11. При выполнении итераций оптимизации для текущей подпоследовательности необходимо выполнение для запаздывающих заданий следующих процедур:

а) определение и расширение интервала встраивания (условия 1–4 утверждения 10);

б) определение наличия резервов времени на расширенном интервале встраивания (определение 1);

в) в случае наличия резервов времени на расширенном интервале встраивания выполнение процедуры встраивания запаздывающего задания на позицию, определенную интервалом встраивания (определение 6). Все задания, занимающие позиции после встроенного запаздывающего задания, упорядочиваются по невозрастанию их приоритетов (утверждение 4) и выполняется оптимизация на расширенном интервале встраивания;

г) в случае отсутствия резервов на расширенном интервале встраивания или невозможности их использования, выполнение процедуры освобождения резервов посредством перестановки заданий, ранее их использовавших в результате предыдущих шагов итерации (утверждение 10, п. 5, 6, утверждение 7);

д) оптимизация на интервале перестановки за счет использования освобожденных резервов запаздывающими заданиями (утверждение 10, п. 5, 6).

Все процедуры а)–д) основываются на условиях перестановки заданий, приведенных в утверждении 10, и выполняются как для конкурирующих заданий, так и для запаздывающих заданий на интервале встраивания.

Утверждение 12. Пусть уже выполнена $k-1$ итерация и построена оптимальная подпоследовательность на множестве заданий на интервале $\overline{1, g-1}$. Очередное конкурирующее задание $j_{[g]}$ ($p_{j_{[g]}} \le p_{j_{[i]}}$, $i \in \overline{1, g-1}$) на k-й итерации может занять более раннюю позицию, что приведет к минимально возможному увеличению суммарного взвешенного запаздывания текущей подпоследовательности по сравнению со значением функционала на итерации $k-1$ и построению оптимальной подпоследовательности на k-й итерации, только в том случае, если хотя бы у одного задания $j_{[l]}$ на интервале $\overline{1, g-1}$ есть резерв и выполняется: $d_{j_{[l]}} > d_{j_{[g]}} - l_{j_{[g]}}$, либо (в случае отсутствия резервов) на интервале $\overline{1, g-1}$ есть задания с метками.

Утверждение 13. Пусть подпоследовательность σ^k на множестве заданий на интервале $\overline{1, g-1}$, построенная на предыдущей итерации, оптимальна. Если конкурирующее задание $j_{[g]}$ ($p_{j_{[g]}} \le p_{j_{[i]}}$, $i \in \overline{1, g-1}$) в результате выполнения перестановок и встраиваний не использовало существующие резервы, минимальное значение функционала соответствует позиции g, занимаемой этим заданием, и нет заданий, помеченных «*» («**»), то задание $j_{[g]}$ исключается из множества конкурирующих заданий, и текущая подпоследовательность на множестве заданий на интервале $\overline{1, g}$ оптимальна.

Следующие утверждения позволяют определить и исключить бесперспективные перестановки и встраивания.

Утверждение 14. Если в оптимальной подпоследовательности σ^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 $j^*_{[m]}$ ($j^{**}_{[m]}$), то минимально возможное увеличение суммарного взвешенного запаздывания текущей подпоследовательности по сравнению со значением функционала на предыдущей итерации и построение оптимальной подпоследовательности на текущей итерации осуществляется только за счет резервов, освобожденных заданиями $j^*_{[m]}$ ($j^{**}_{[m]}$) при встраивании их на позицию k, соответствующую их приоритету (если $p_{j^*_{[m]}}$ ($p_{j^{**}_{[m]}}$) $< p_{j_{[g]}}$, то $k = g$). При этом необходимо выполнение следующего условия:

$$\omega_{j^*_{[m]}}\left(C_{j_{[k]}} - d_{j^*_{[m]}}\right) < \sum_{i=m}^{g} \omega_{j_{[i]}} \max\left(0, C_{j_{[i]}} - d_{j_{[i]}}\right).$$

Утверждение 15. Пусть уже выполнена $k-1$ итерация и построена оптимальная подпоследовательность на множестве заданий на интервале $\overline{1, g-1}$. Запаздывающее задание $j_{[g]}$ на k-й итерации может занять более раннюю позицию, что приведет к достижению минимально возможного увеличения суммарного взвешенного запаздывания текущей подпоследовательности по сравнению со значением функционала на предыдущей итерации и построению оптимальной подпоследовательности на текущей итерации, только в том случае, если хотя бы у одного задания $j_{[l]}$ на интервале $\overline{1, g-1}$ есть резерв и выполняется: $d_{j_{[l]}} > d_{j_{[g]}} - l_{j_{[g]}}$, либо (в случае отсутствия резервов) на интервале $\overline{1, g-1}$ есть задания с метками или на интервале встраивания задания $j_{[g]}$ есть запаздывающее задание $j_{[r]}$, $p_{j_{[r]}} > p_{j_{[g]}}$, и заданию $j_{[r]}$ предшествует задание $j_{[s]}$ такое, что $p_{j_{[s]}} \le p_{j_{[r]}}$.

В утверждениях 16, 17 сформулированы достаточные признаки оптимальности полученного решения.

Достаточный признак оптимальности №8.

Утверждение 16. Пусть уже выполнена $k-1$ итерация и построена оптимальная подпоследовательность σ^k на множестве заданий на интервале $\overline{1, g-1}$. Если на итерации k в подпоследовательности σ^k конкурирующее задание $j_{[g]}$ ($p_{j_{[g]}} \le p_{j_{[i]}}$, $i \in \overline{1, g-1}$) после выполнения для него текущей итерации оптимизации не использовало существующие резервы, минимально возможное увеличение суммарного взвешенного запаздывания текущей подпоследовательности по сравнению со значением функционала на предыдущей итерации соответствует позиции g, занимаемой этим заданием, на интервале $\overline{1, g-1}$ отсутствуют помеченные задания или для помеченных заданий не выполняются условия утверждения 14, и $\forall j_{[r]}$, $r = \overline{g+1, n}$, $d_{j_{[r]}} \ge d_{j_{[g]}}$, $l_{j_{[r]}} \ge l_{j_{[g]}}$ и $C_{j_{[r]}} \ge d_{j_{[r]}}$, то задания $j_{[r]}$ исключаются из множества конкурирующих и остаются на своих позициях, а текущая последовательность оптимальна на всем множестве заданий.

Достаточный признак оптимальности №9 (проверяется непосредственно после добавления очередного конкурирующего запаздывающего задания).

Утверждение 17. Пусть уже выполнена $k-1$ итерация и построена оптимальная подпоследовательность σ^k на множестве заданий на интервале $\overline{1, g-1}$. Пусть на итерации k, выполняющейся для очередного конкурирующего запаздывающего задания $j_{[g]}$, для всех $j_{[l]}$, $l=\overline{1, g-1}$, справедливо $p_{j_{[g]}} \le p_{j_{[l]}}$, $d_{j_{[l]}} \le C_{j_{[l]}}$, для всех $j^*_{[l]}$ ($j^{**}_{[l]}$) выполняется $l_{j^*_{[l]}}(l_{j^{**}_{[l]}}) \le l_{j_{[i]}}, i=\overline{g,n}$, и на интервале $\overline{g, n}$ резервы отсутствуют. Тогда запаздывающие задания, занимающие позиции $\overline{g, n}$, исключаются из множества конкурирующих и остаются на своих позициях, соответствующих минимально возможному увеличению суммарного взвешенного запаздывания текущей подпоследовательности по сравнению со значением функционала на предыдущей итерации, а текущей последовательности на всем множестве заданий отвечает оптимальное значение функционала.

Следствие. В утверждениях 13, 16, 17 приведены условия, при выполнении которых (выполняются либо все условия утверждения 13, либо все условия утверждения 16, либо все условия утверждения 17) одно либо несколько конкурирующих заданий исключаются из множества конкурирующих и повторно не включаются в него. Для них не выполняются итерации оптимизации, на следующих итерациях они могут рассматриваться как порожденные запаздывающие задания.

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

Достаточный признак оптимальности №10.

Утверждение 18. Пусть в последовательности $\sigma^{\text{уп}}$ $j_{[g]}$ – первое запаздывающее задание, задания на интервале $\overline{g+1, n}$ запаздывают, и выполняется хотя бы одно из двух условий: а) $\max R(\sigma^{\text{уп}}) \le l_{j_{[g]}}$ и $\forall j_{[r]}$, $r=\overline{g+1, n}$, $l_{j_{[r]}} \ge l_{j_{[g]}}$; б) $\forall l, r \in \overline{g, n}$, $l<r$, $l_{j_{[l]}} < l_{j_{[r]}}, d_{j_{[l]}} < d_{j_{[r]}}$. В этом случае задача решается полиномиальным подалгоритмом, трудоемкость каждой итерации которого определяется функцией $O(n_k \log n_k)$, где n_k – количество заданий на текущей итерации, которое оптимизируется (см. примечание к утверждению 20).

Достаточный признак оптимальности №11.

Утверждение 19. Пусть $j_{[g]}$ – первое конкурирующее задание в последовательности $\sigma^{\text{сп}}$. 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 $l_{j_{[s]}} \le l_{j_{[t]}}, d_{j_{[s]}} \le d_{j_{[t]}}$, $\omega_{j_{[s]}} \ge \omega_{j_{[t]}}$, и $\forall j_{[r]}$, $r=\overline{g+1, n}$, выполняется $l_{j_{[r]}} > l_{j_{[r-1]}}$, $d_{j_{[r]}} > d_{j_{[r-1]}}$, $d_{j_{[r]}} < C_{j_{[r]}}$, то задача решается полиномиальным подалгоритмом, трудоемкость каждой итерации которого не превышает $O(n_k)$, где n_k – количество заданий на текущей итерации, которое оптимизируется (см. примечание к утверждению 20).

Достаточный признак оптимальности №12.

Утверждение 20. Пусть $j_{[g]}$ – первое конкурирующее задание в последовательности $\sigma^{\text{сп}}$. Если на интервале $\overline{1, g-1}$ все задания упорядочены по невозрастанию их приоритетов; $\max R_i \le l_{j_{[g]}}$, $i=\overline{1, g-1}$, $\forall j_{[s]}$, $s=\overline{g+1, k}$, $d_{j_{[s]}} < C_{j_{[s]}}$, $l_{j_{[s]}} \ge l_{j_{[g]}}$ и $\forall j_{[s]}$, $s=\overline{k+1, n}$, $d_{j_{[s]}} \ge C_{j_{[s]}}$, где $k<n$, то задача решается полиномиальным подалгоритмом, трудоемкость каждой итерации которого равна $O(n_k \log n_k)$, где n_k – количество заданий на текущей итерации, которое оптимизируется (см. примечание к утверждению 20).

Примечание. При выполнении любого из достаточных признаков оптимальности №10–12 задача решается соответствующим полиномиальным подалгоритмом, каждая итерация которого имеет соответствующую сложность – $O(n_k)$ или $O(n_k \log n_k)$, где n_k – оптимизируемое количество заданий на k-й итерации подалгоритма (описание полиномиальных подалгоритмов оставляем читателям в качестве упражнения). Трудоемкость решения задачи на всем множестве заданий определяется функцией $O(n^2)$ или $O(n^2 \log n)$.

Достаточный признак полиномиальной разрешимости текущей итерации (ДППР).

Утверждение 21. Пусть уже выполнена $k-1$ итерация и построена оптимальная подпоследовательность σ^k на множестве заданий на интервале $\overline{1, g-1}$. Пусть при выполнении текущей итерации оптимизации при встраивании очередного запаздывающего задания $j_{[g]}$ (конкурирующего или порожденного) выполняется: а) фактическая позиция встраивания $p^{н}$, определяемая в процессе выполнения алгоритма, больше позиции встраивания задания, встроенного последним на предыдущих шагах алгоритма; б) на интервале встраивания задания $j_{[g]}$ нет заданий, соответствующих условиям утверждения 7; в) на интервале $\overline{1, p^{н}-1}$ нет заданий $j_{[k]}$, для которых выполняются условия: $d_{j_{[k]}} > d_{j_{[g]}} - l_{j_{[g]}}$, $d_{j_{[k]}} > C_{j_{[k]}}$, $d_{j_{[k]}} > C_{j_{[p^{н}-1]}}$; г) на интервале $\overline{1, p^{н}-1}$ для заданий $j^{*}_{[m]}$ ($j^{**}_{[m]}$) не выполняется следующее условие, позволяющее сместить эти задания на более поздние позиции:

$$\omega_{j^{*}_{[m]}}\left(C_{j_{[k]}} - d_{j^{*}_{[m]}}\right) < \sum_{i=m}^{p^{н}} \omega_{j_{[i]}} \max\left(0, C_{j_{[i]}} - d_{j_{[i]}}\right),$$

где k – позиция, соответствующая приоритету $j^{*}_{[m]}$ ($j^{**}_{[m]}$) (если $p_{j^{*}_{[m]}}$ ($p_{j^{**}_{[m]}}$) $< p_{j_{[p^{н}]}}$, то $k = p^{н}$). Тогда решение на текущей итерации достигается полиномиальным подалгоритмом с трудоемкостью $O(n_k^2)$, где n_k – количество заданий на текущей итерации, которое оптимизируется.

Утверждение 22. Полиномиальная составляющая ПДС-алгоритма реализуется:

1) алгоритмическими процедурами проверки выполнения достаточных признаков оптимальности №№1–5, 7, которые в общем алгоритме реализуются до выполнения итераций оптимизации;

2) если не выполняется п. 1), то алгоритмическими процедурами проверки выполнения достаточных признаков оптимальности №№10–12, которые включают в себя выполнение подалгоритмов полиномиальной сложности, как показано выше в утверждениях 18–20 и примечании к ним;

3) если ДППР выполняется на каждой итерации, то полиномиальным подалгоритмом получено оптимальное расписание на всем множестве заданий.

Утверждение 23. 1) Если в начале текущей итерации оптимизации выполняется достаточный признак оптимальности №9 (утверждение 17), то текущая и последующие итерации оптимизации не выполняются, получено оптимальное решение;

2) если после выполнения текущей итерации оптимизации выполнился достаточный признак оптимальности №6 или 8 (утверждения 10 и 16), то получено оптимальное расписание на всем множестве заданий, и последующие итерации оптимизации не выполняются;

3) если ДППР выполняется на каждой итерации, то реализуется достаточное условие, при котором экспоненциальная составляющая превращается в алгоритм полиномиальной сложности.

Приведенные свойства позволили создать ПДС-алгоритм решения задачи [3].

Свойства 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.

Основные этапы решения задачи:

– упорядочение заданий по невозрастанию приоритетов заданий (последовательность $\sigma^{уп}$);
– выполнение свободных перестановок и определение множества конкурирующих заданий (последовательность $\sigma^{сп}$);
– выполнение итераций оптимизации для использования резервов предшествующих заданий каждым конкурирующим заданием $j_{[k]}$ (последовательность σ^{k}).

Для каждого этапа сформулированы условия (утверждения 18–22), при выполнении которых оптимальное решение задачи достигается за полиномиальное время.

Если в последовательности, упорядоченной по неубыванию директивних сроков заданий, суммарное запаздывание равно нулю (достаточный признак оптимальности №1, следствие к теореме 1), то она оптимальна на всем множестве заданий, и решение достигается за полиномиальное время с трудоемкостью $O(n \log n)$. Если в последовательности $\sigma^{уп}$ выполняются достаточные признаки оптимальности №2, 3, 4 (утверждения 1, 2, следствие к утверждению 4), то эта последовательность оптимальна на всем множестве заданий, и решение достигается за полиномиальное время с трудоемкостью $O(n \log n)$. Иначе выполняются свободные перестановки и анализируется полученная последовательность $\sigma^{сп}$. При выполнении достаточных признаков оптимальности №5, 7 (утверждение 6, следствие 2 к утверждению 10), последовательность $\sigma^{сп}$ оптимальна на всем множестве заданий, трудоемкость алгоритма определяется функцией $O(n^2)$. Если в последовательности $\sigma^{уп}$ выполняется достаточный признак оптимальности №10 (утверждение 18), то реализуется подалгоритм полиномиальной сложности с трудоемкостью $O(n \log n)$. Если в последовательности $\sigma^{сп}$ выполняется хотя бы один из достаточных признаков оптимальности №11 или 12 (утверждения 19, 20), то реализуется подалгоритм полиномиальной сложности с трудоемкостью, соответственно, $O(n)$ или $O(n \log n)$. Если на каждой итерации оптимизации выполняется ДППР, то полиномиальным подалгоритмом получено оптимальное расписание на всем множестве заданий. Если не выполняется ни один из достаточных признаков оптимальности или ДППР, то полиномиальная составляющая алгоритма не реализуется.

Общая трудоемкость выполнения полиномиальной составляющей ПДС-алгоритма 1 не превышает $O(n^3)$.

Экспериментальные исследования показали статистическую значимость полиномиальной составляющей. При решении индивидуальных задач выполнение итераций оптимизации реализуется далеко не для всего множества запаздывающих заданий, что существенно сокращает трудоемкость алгоритма.

Свойства экспоненциальной составляющей ПДС-алгоритма

Трудоемкий перебор различных вариантов использования резервов конкурирующими заданиями возникает, если для всех конкурирующих заданий выполняется:

$$\forall i < g, j_{[i]}, j_{[g]} \in K,\ l_{j_{[i]}} \le l_{j_{[g]}}\ \ d_{j_{[i]}} \ge d_{j_{[g]}}, \qquad (4)$$

причем приоритеты заданий отличаются на незначительную величину, а резервы таковы, что итерации оптимизации выполняются для каждого конкурирующего задания. В этом случае реализуется переход на п.14.9 алгоритма. При этом для задания $j^*_{[m]}$ ($j^{**}_{[m]}$) в свою очередь ищется другое задание, помеченное «*» («**»), которое может быть встроено после него, что приведет к уменьшению значения функционала. В этом случае осуществляется перебор заданий, помеченных «*» («**»), с последующей оптимизацией подпоследовательности, что может быть связано с экспоненциальной трудоемкостью.

Эффективность экспоненциальной составляющей ПДС-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 заданием 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 13, 16, 17, позволяющие часть заданий исключить из множества конкурирующих. В утверждениях 2, 4, 10, 12–15 сформулированы правила отсечения ветвей, не содержащих оптимального решения, а в утверждении 23 – условия существенного сокращения вычислений экспоненциальной составляющей алгоритма. Только для специальных случаев параметров задачи – например, при выполнении (4) – можно выйти на полный перебор вариантов для отдельных подпоследовательностей.

Утверждение 24. Текущая итерация оптимизации, основанная на направленных перестановках и реализованная посредством конечного числа операций оптимизационных процедур а)–г) (утверждение 11), выполняемых для очередного конкурирующего задания, входящего в состав текущей подпоследовательности, а также рекурсивно выполняемых для переупорядочения заданий на интервалах встраивания и интервалах перестановок, является конечной.

Из приведенной выше логики построения алгоритма для случая, когда ни один из достаточных признаков оптимальности в процессе реализации алгоритма не выполняются, и из утверждений 3–5, 8–15, 24 и леммы 1 очевидным образом вытекает следующее утверждение.

Утверждение 25. Алгоритм является конечным (утверждение 24) и оптимальным.

Следствие 1. Чтобы на каждой итерации приоритет текущего конкурирующего задания не превышал приоритета ни одного из заданий оптимальной подпоследовательности, полученной на предыдущей итерации, необходимо, чтобы конкурирующие задания были расположены в порядке невозрастания их приоритетов.

Следствие 2. В результате реализации алгоритма получаем одно оптимальное расписание. Для получения всего множества оптимальных расписаний достаточно выполнять перестановки незапаздывающих заданий между собой таким образом, чтобы они оставались незапаздывающими, не меняя позиции остальных заданий.

Очевидно, что достаточные признаки оптимальности (теорема 1 и утверждения 1, 2, 4, 6, 10, 16–20), а также ДППР (утверждение 21), перестановки, использующиеся в оптимизационных процедурах, при которых значение функционала уменьшается (утверждения 7, 8), необходимые условия для встраивания конкурирующего задания (утверждение 9), свойства полиномиальной и экспоненциальной составляющей и доказательства точности алгоритма (утверждения 22–25) справедливы также для задачи МСЗ [1].

Выводы

Исследованы свойства задачи, сформулированы и обоснованы достаточные признаки оптимальности получаемых решений, приведены условия, при выполнении которых одно либо несколько конкурирующих заданий исключаются из множества конкурирующих, а также правила отсечения бесперспективных перестановок и встраиваний. Приведенные свойства позволили создать ПДС-алгоритм решения задачи [3]. Сформулированы свойства полиномиальной и экспоненциальной составляющих ПДС-алгоритма, доказаны его конечность и оптимальность.

Список литературы

1. Згуровский М.З., Павлов А.А. Принятие решений в сетевых системах с ограниченными ресурсами: Монография. – К.: Наукова думка. – 2010. – 573 с.
2. Tanaka S., Fujikuma S., Araki M. An exact algorithm for single-machine scheduling without machine idle time // Journal of Scheduling, vol. 12, no. 6, pp. 575–593, 2009.
3. 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.
4. Танаев В.С., Шкурба В.В. Введение в теорию расписаний. – М.: Наука, 1975.– 256 с.