

ПДС-АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ СОСТАВЛЕНИЯ РАСПИСАНИЙ ПО КРИТЕРИЮ ОПЕРЕЖЕНИЯ/ЗАПАЗДЫВАНИЯ НА ОДНОМ ПРИБОРЕ

Предложены эффективные ПДС-алгоритмы решения NP-трудной задачи опережения/запаздывания для случаев, когда момент начала выполнения заданий фиксирован или находится в интервале времени $[t_1, t_k]$. Решение основано на ПДС-алгоритме решения задачи минимизации суммарного запаздывания заданий. Разработан эвристический алгоритм определения самого позднего момента начала выполнения заданий, при котором достигается минимальное значение функционала. Приведены примеры решения задач и результаты экспериментальных исследований.

The effective PDC-algorithms are proposed for the solution of NP-hard earliness/tardiness problem for the cases when the moment of the tasks execution start is fixed or is in the time interval $[t_1, t_k]$. The solution is based on the PDC-algorithm for the total tardiness minimization problem. A heuristic algorithm is developed to define the latest moment of the tasks execution start at which the minimal functional value is achieved. An examples of the problems solution and the results of the experiments are given.

Введение

Составлению расписаний относительно директивных сроков уделено значительное внимание в литературе. Одна из причин этого явления – возрастающая конкуренция на международных рынках: фирмы должны предложить большое разнообразие индивидуальных изделий, в то время как клиенты ожидают, что заказанные товары будут поставлены своевременно. Чтобы удовлетворить эти виды требований, были разработаны такие принципы и философии, как управление отклонением, одновременная разработка, производство «точно в срок» и т.д. Например, принцип «точно в срок» устанавливает, что необходимое количество товаров должна быть произведено и поставлено точно в необходимое время. Таким образом, можно ожидать, что задания будут выполнены вовремя, потому что запаздывающие поставки, так же, как и опережающие, рассматриваются как нежелательные [1, 2, 3, 4].

В то время как опережение заданий связано с затратами на складирование, запаздывание заданий связано с такими штрафами, как потеря доброжелательности клиента и потеря заказов. Поэтому задача минимизации опережения и запаздывания (О/З) очень важна в рыночной экономике.

Большинство исследований по задачам О/З связаны с моделями для одного прибора, хотя некоторые результаты для одного прибора были расширены на задачи для параллельных приборов. Некоторые из простейших результатов для задач О/З были получены для моделей,

в которых все задания имеют общий директивный срок ($d_j = d$).

В данной статье мы рассматриваем задачу составления расписаний по критерию минимизации суммарного опережения и запаздывания при выполнении заданий одним прибором (О/З), которая может быть сформулирована таким образом.

Постановка задачи О/З

Множество из n независимых заданий $J = \{j_1, j_2, \dots, j_n\}$ должно быть назначено без прерываний на один прибор, который может выполнять не более, чем одно задание одновременно. Прибор и задания предполагаются постоянно доступными с момента времени ноль, а простои прибора не допускаются. Задание j , где $j = 1, 2, \dots, n$, требует времени выполнения l_j и должна в идеале быть завершена в свой директивный срок d_j . Для любого заданного расписания опережение и запаздывание задания j могут быть определены выражениями, соответственно:

$$E_j = \max(0, d_j - C_j) = (d_j - C_j)^+ \quad (1)$$

$$T_j = \max(0, C_j - d_j) = (C_j - d_j)^+, \quad (2)$$

где C_j – момент завершения задания j .

Цель состоит в том, чтобы найти расписание, минимизирующее суммарное опережение и запаздывание всех заданий:

$$\sum_{j=1}^n (E_j + T_j). \quad (3)$$

Задача относится к классу NP-трудных. Рассматриваются следующие задачи:

ОЗ1. Момент начала выполнения заданий r фиксированный и равен нулю;

ОЗ2. Момент начала выполнения заданий находится в интервале времени $[t_1, t_k]$, $t_k > t_1 > 0$;

ОЗ3. Момент начала выполнения заданий не фиксирован.

Данные задачи включены в состав математического обеспечения четырехуровневой модели планирования, принятия решений и оперативного управления в сетевых системах с ограниченными ресурсами [5]: на третьем уровне при построении плана выполнения заданий и на четвертом уровне при оперативной коррекции построенного плана в случае его частичного невыполнения.

Задачи опережения и запаздывания рассматривались в работе Кэннета [6]. С использованием системы обозначений Лоулера и др. [7] статическая задача составления расписаний О/З может быть описана как $1||\Sigma(E_j + T_j)$. Бейкер и Скуддер [8] приводят обзор открытой литературы по задаче составления расписаний О/З, описывая разные модели О/З и разные штрафы за опережение и запаздывание заданий. Полезный обзор задачи также приведён в книге [9]. Когда директивные сроки задач определены, но различны, задачи О/З являются NP-полными [10]. Яно и Ким [11] разработали эвристику для частного случая взвешенных штрафов, в котором вес пропорционален длительностям заданий. Они используют процедуру попарной перестановки, чтобы улучшить начальные эвристические решения.

Оу и Мортон [12] исследуют функцию на основе приоритета и разновидность фильтрованного лучевого поиска для задач О/З, делая вывод, что качество решения значительно ухудшается, если затраты на опережение заданий игнорируются в пользу рассмотрения только стоимости запаздывания. Планировщик Дэвиса и Кейнта [13] может быть встроен в существующее эвристическое решение для улучшения качества расписания О/З при низких вычислительных затратах. Шварц и Мухопадьяй [14] разработали подход группировки (кластеризации) для определения оптимального момента запуска заданий на примере задачи О/З при предположении, что последовательность заданий известна априорно. Подход группировки осуществляет оптимальный выбор времени для 500 заданий за несколько секунд вычислительного времени. Сридхаран и Жоу [15] используют эвристику диспетчирования для задач О/З, позволяющую вставку допустимого резерва прибора.

Алгоритм, описанный в статье [16], основан на табу-поиске. Алгоритмы для определения оптимальной последовательности относительно слабо исследованы. Шварцем [17] был предложен алгоритм ветвей и границ, основанный на сортировке смежных заданий с разными весами штрафов. Лии и Чой [18] предложили генетический алгоритм, а Яно и Ким [11] изучили некоторые доминантные свойства последовательностей заданий со штрафами, пропорциональными времени обработки.

Наиболее значительные результаты для расписаний с запрещенными и разрешенными нарушениями директивных сроков представлены в статьях Бэнка и Вэрнена [19], Гордона [20], Валенте, Альвесса [21], Фельдмана и Бискапа [1], а также Тунг-ай Тсая [22].

Хугевен и Ван де Вельд [23] предложили алгоритм, основанный на методе ветвей и границ. По причине того, что алгоритм имеет экспоненциальный рост времени вычисления, он может использоваться только для задач с количеством заданий, не превосходящим 20. Сложность решения задачи больших размерностей обусловила необходимость на практике почти всегда использовать приближенные алгоритмы.

В настоящей статье представлены эффективные ПДС-алгоритмы решения исследуемых задач ОЗ1 и ОЗ2.

ПДС-алгоритмы [24] – это алгоритмы с полиномиальной и экспоненциальной составляющими, при этом в некоторых случаях возможна декомпозиция на подзадачи меньшей размерности, каждая из которых решается независимо. Например, в задаче минимизации суммарного взвешенного момента окончания выполнения заданий [24, гл. 2] осуществляется декомпозиция на множества максимального приоритета, в каждом из которых оптимизация осуществляется независимо.

В отличие от известных алгоритмов, ПДС-алгоритмы относят решаемую индивидуальную задачу к подклассу полиномиально разрешимых в процессе анализа ее решения. С этой целью для каждой труднорешаемой задачи строго выводятся логико-аналитические условия, проверяемые в процессе ее решения, в случае выполнения которых данная произвольная индивидуальная задача точно решается полиномиальным алгоритмом. Логико-аналитические условия и полиномиальный алгоритм их проверки составляют полиномиальную составляющую ПДС-алгоритма.

Полиномиальный алгоритм, проверяющий признак оптимальности допустимого решения,

может быть: а) частью экспоненциальной составляющей либо полиномиальной аппроксимации экспоненциальной составляющей ПДС-алгоритма [24, гл. 1, 2]; б) совпадать полностью с полиномиальной аппроксимацией экспоненциальной составляющей ПДС-алгоритма [24, гл. 3]; в) отдельным полиномиальным алгоритмом, не совпадающим с экспоненциальной составляющей либо полиномиальной аппроксимацией экспоненциальной составляющей ПДС-алгоритма (алгоритмы, приведенные в настоящей статье).

Декомпозиционная составляющая [24, гл. 2] реализуется только тогда, когда теоретически строго полиномиальным алгоритмом можно декомпозировать задачу на подзадачи меньшей размерности.

Таким образом, необходимым и достаточным условием построения ПДС-алгоритмов является нахождение теоретически обоснованных признаков оптимальности допустимого решения. Из этого определения следует, что ПДС-алгоритмы бывают двух типов. ПДС-алгоритмы первого типа включают полиномиальную составляющую и экспоненциальную составляющую, которая представлена точным экспоненциальным алгоритмом, построенным как на теоретически обоснованных направленных перестановках, так и на эффективных отсечениях. В результате решения [25, 24, гл. 1] задачи либо реализуется полиномиальная составляющая алгоритма, либо после выполнения всех перестановок и отсечений получаем решение, оптимальность которого строго обоснована.

В ПДС-алгоритмах второго типа экспоненциальная составляющая представлена ее полиномиальной аппроксимацией, которая может быть либо приближенным полиномиальным алгоритмом, если получена оценка отклонения решения от оптимального, либо эвристическим алгоритмом, если такой оценки нет. В результате решения получаем: а) оптимальное значение функционала, если в процессе выполнения алгоритма реализованы признаки оптимальности получаемых решений; б) приближенное решение с оценкой отклонения от оптимального; в) эвристическое решение, если такая оценка не может быть получена.

Алгоритмы решения исследуемых в статье задач относятся к ПДС-алгоритмам второго типа.

С целью повышения практической эффективности ПДС-алгоритмов в целом введем понятие эвристического признака оптимальности.

Определение 1. Эвристическим признаком оптимальности допустимого расписания называются логико-аналитические условия, выполнение которых в допустимом решении статистически значимо приводит к оптимальному решению.

Эти свойства допустимого решения качественно обосновываются и подтверждаются численными экспериментами. В силу того, что точное доказательство этих свойств отсутствует, проверка этого эвристического признака оптимальности выполняется экспоненциальной составляющей ПДС-алгоритма или ее полиномиальной аппроксимацией.

Введение эвристического признака оптимальности допустимого решения четко определяет пути дальнейших теоретических исследований – точное доказательство этого признака либо, скорее всего, его частного случая, позволит включить этот признак вместе с соответствующим полиномиальным алгоритмом в полиномиальную составляющую ПДС-алгоритма, что резко повышает его практическую эффективность.

Приведем краткий план решения поставленных задач.

Задача ОЗ1. Задача О/З с фиксированным моментом начала выполнения заданий.

1.1. Полиномиальная составляющая ПДС-алгоритма решения задачи.

1.1.1. Формулировка признака оптимальности допустимого решения.

1.1.2. Полиномиальный алгоритм, проверяющий признак оптимальности. Алгоритм основан на ПДС-алгоритме решения задачи минимизации суммарного запаздывания (МСЗ) [24, гл. 1], включает в себя как его полиномиальную составляющую, так и его экспоненциальную составляющую с наложением дополнительного ограничения на число выполняемых итераций.

1.2. Полиномиальная аппроксимация экспоненциальной составляющей ПДС-алгоритма решения задачи ОЗ1. Эвристический алгоритм с введением качественно обоснованных эвристических признаков оптимальности.

1.3. Пример решения задачи.

Задача ОЗ2. Момент начала выполнения заданий находится в интервале времени $[t_1, t_k]$.

2.1. Полиномиальная составляющая ПДС-алгоритма решения задачи сводится к полиномиальной составляющей ПДС-алгоритма решения Задачи ОЗ1.

2.2. Полиномиальная аппроксимация экспоненциальной составляющей ПДС-алгоритма решения задачи ОЗ2.

2.2.1. Эвристический алгоритм решения задачи ОЗ2, который базируется на алгоритме решения задачи ОЗ3.

2.2.2. Эвристический алгоритм решения задачи ОЗ3.

Алгоритмы решения задач ОЗ2 и ОЗ3 также базируются на основе положений построения экспоненциальной составляющей ПДС-алгоритма решения задачи МСЗ [24, гл. 1].

Задача ОЗ1. Фиксированный момент начала выполнения заданий ($r = 0$)

1.1 Полиномиальная составляющая ПДС-алгоритма решения задачи

1.1.1 Формулировка признака оптимальности допустимого решения

Признаком оптимальности решения Задачи ОЗ1 является отсутствие опережающих заданий в оптимальном расписании, полученном в результате решения задачи минимизации суммарного запаздывания (МСЗ).

Обоснование. Расписание, минимизирующее только суммарное запаздывание, представляет нижнюю границу минимального суммарного запаздывания, которое может быть достигнуто. Если в таком расписании суммарное опережение равно нулю, то такое расписание оптимально по критерию опережения/запаздывания (3).

1.1.2. Полиномиальный алгоритм, проверяющий признак оптимальности

Алгоритм основан на ПДС-алгоритме решения задачи МСЗ [24, гл. 1]. Полиномиальная составляющая ПДС-алгоритма решения задачи О/З реализуется полиномиальной составляющей ПДС-алгоритма решения задачи МСЗ либо экспоненциальной составляющей ПДС-алгоритма задачи МСЗ с наперед заданным ограничением по числу выполняемых итераций. Если в этом случае алгоритм реализуется на всем множестве заданий, т.е. если количество операций до останова алгоритма меньше введенного полиномиального ограничения, то полученное решение также является реализацией полиномиальной составляющей алгоритма.

Рассмотрим более подробно условия реализации полиномиальной составляющей ПДС-алгоритма решения задачи МСЗ.

Введем необходимые определения [24].

Определение 2. Последовательностью $\sigma^{уп}$ (сигма упорядоченная) называется последовательность заданий множества J , $j = \overline{1, n}$, в которой задания упорядочены по неубыванию длительностей l_j , т. е. $\forall j, i, j < i : l_j < l_i$, а при $l_j = l_i$, $d_j \leq d_i$.

Определение 3. 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 4. Последовательность заданий, полученную в результате выполнения всех свободных перестановок в последовательности $\sigma^{уп}$, назовем $\sigma^{сн}$.

В следующих утверждениях 1, 2, 6 [24, гл. 1] сформулированы признаки оптимальности, соответственно, последовательности $\sigma^{уп}$, $\sigma^{сн}$, σ^k (текущей последовательности, полученной в результате выполнения очередной итерации алгоритма). В утверждениях 3, 4 сформулированы условия сокращения перестановок в случае наличия заданий с отрицательными директивными сроками. В утверждении 5 сформулировано условие декомпозиции последовательности $\sigma^{сн}$.

Утверждение 1. 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 хотя бы одного из следующих условий:

1. $\exists j_{[i]}, p \leq i \leq g \mid r_{j_{[i]}} > 0, d_{j_{[i]}} > d_{j_{[g]}}$. На интервале встраивания задания $j_{[g]}$ есть задания $j_{[i]}$ с резервами времени, для которых $d_{j_{[i]}} > d_{j_{[g]}}$. p – позиция, на которой 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 опоздание 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:

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

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

4. $\forall i \mid p \leq i \leq g \mid r_{j_{[i]}} \leq 0$, но $\exists j_{[k]}, k < p \mid d_{j_{[k]}} > d_{j_{[g]}}$, $r_{j_{[k]}} > 0$, $d_{j_{[k]}} > C_{j_{[p-1]}}$. На интервале перестановки задания $j_{[g]}$ резервы отсутствуют, но существует задание $j_{[k]}, k < p$, директивный срок которого больше $d_{j_{[g]}}$ и резерв больше нуля.

Утверждение 2. Для последовательности $\sigma^{\text{сп}}$ не существует перестановок и встраиваний, приводящих к улучшению целевой функции, если выполняется хотя бы одно из условий:

- $d_{j_{[i]}} - C_{j_{[i]}} \geq 0, \forall i = \overline{1, n}$;
- $d_{j_{[i]}} - C_{j_{[i]}} \leq 0, \forall i = \overline{1, l}, d_{j_{[i]}} - C_{j_{[i]}} \geq 0, \forall i = \overline{l+1, n}$;
- $d_{j_{[i]}} - C_{j_{[i]}} \leq 0, \forall i = \overline{1, n}$;
- для каждого задания $j_{[k]}$, принадлежащего множеству заданий с резервами (R), и каждого задания $j_{[g]}$, принадлежащего множеству запаздывающих заданий (Y), выполняется: $d_{j_{[k]}} \leq d_{j_{[g]}}$, $l_{j_{[k]}} \leq l_{j_{[g]}} \forall k \in \overline{1, n}, g \in Y$.

Утверждение 3 [26]. В последовательности $\sigma^{\text{сп}}$ для заданий с отрицательными директивными сроками справедливо: при $i < j: l_i \leq l_j$.

Утверждение 4 [26]. Если на интервале встраивания задания $j_{[g]}$ с отрицательным значением директивного срока есть другое задание $j_{[k]}$ с отрицательным значением директивного срока, то при выполнении процедуры встраивания задание $j_{[g]}$ займет позицию $k+1$.

Утверждение 5 [26]. Если в последовательности $\sigma^{\text{сп}}$ первые позиции $i = [1, k]$ занимают задания, для которых $r_{j_{[i]}} \leq 0$, то последовательность $\sigma^{\text{сп}}$ декомпозируется на две подпоследовательности: σ^1 включает задания от $j_{[1]}$ до $j_{[k]}$, σ^2 – от $j_{[k+1]}$ до $j_{[n]}$. Подпоследовательность σ^1 оптимальна, и задания, принадлежащие этой подпоследовательности, в перестановках не участвуют. Оптимизация выполняется только для заданий подпоследовательности σ^2 .

Утверждение 6. Пусть в последовательности σ^k $j_{[g]}$ – запаздывающее задание. Уменьшение значения функционала при перемещении $j_{[g]}$ на более ранние позиции возможно только при выполнении хотя бы одного из следующих условий:

1. $\exists j_{[i]}, p \leq i \leq g \mid r_{j_{[i]}} > 0, d_{j_{[i]}} > d_{j_{[g]}}$. На интервале встраивания задания $j_{[g]}$ есть задания $j_{[i]}$, для которых $d_{j_{[i]}} > d_{j_{[g]}}$, $d_{j_{[i]}} > C_{j_{[i]}}$, p – позиция, на которой запаздывание по заданию $j_{[g]}$ минимально (или равно нулю).

2. $\exists j_{[q]}, q < g, d_{j_{[q]}} > C_{j_{[g]}}$.

3. $\exists j_{[l]}, l < g, d_{j_{[l]}} > C_{j_{[g]}} - l_{j_{[g]}}$;

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

4. $\forall i \mid p \leq i \leq g \mid r_{j_{[i]}} \leq 0$, но $\exists j_{[k]}, k < p \mid d_{j_{[k]}} > d_{j_{[g]}}$, $d_{j_{[k]}} > C_{j_{[k]}}$, $d_{j_{[k]}} > C_{j_{[p-1]}}$.

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

Характеристика полиномиальной составляющей ПДС-алгоритма решения задачи МСЗ. Справедливы следующие утверждения [24].

Утверждение 7. Пусть в последовательности $\sigma^{\text{сп}}$ $j_{[g]}$ – первое запаздывающее задание, и выполняется: $\max r(\sigma^{\text{сп}}) \leq l_{j_{[g]}}$. В этом случае оптимальное значение функционала достигается за полиномиальное время с трудоемкостью, определяемой функцией $O(n \log n)$.

Утверждение 8. Пусть $j_{[g]}$ – первое запаздывающее 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 $j_{[s]}, j_{[t]}$ выполняется $l_{j_{[s]}} \leq l_{j_{[t]}}$, $d_{j_{[s]}} \leq d_{j_{[t]}}$, 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 9. Пусть $j_{[g]}$ – первое запаздывающее задание в последовательности $\sigma^{\text{сп}}$. Если на интервале $\overline{1, g-1}$ все задания упорядочены по неубыванию значений длительности выполнения; $\max r_i \leq l_{j_{[g]}}$, $i = \overline{1, g-1}$ и $\forall j_{[s]}, s = \overline{g+1, n}, d_{j_{[s]}} \leq C_{j_{[s]}}$, то задача решается за полиномиальное время с трудоемкостью $O(n^2)$.

Утверждение 10. Пусть при выполнении каждой итерации оптимизации при встраивании очередного запаздывающего задания $j_{[g]}$ (конкурирующего или порожденного) выполняется: а) фактическая позиция встраивания p^H , опре-

деляемая в процессе выполнения алгоритма, больше позиции задания, встроенного последним на предыдущих шагах алгоритма; б) на интервале $\overline{1, p^H - 1}$ нет заданий $j_{[k]}$, для которых выполняются условия: $d_{j_{[k]}} > d_{j_{[g]}}$, $d_{j_{[k]}} > C_{j_{[k]}}$, $d_{j_{[k]}} > C_{j_{[p^H - 1]}}$; в) на интервале $\overline{1, p^H - 1}$ для заданий, помеченных «*» («**»), условие $C_{j_{[p]}} - d_{j_{[m]}} < \sum_{i=m}^{p^H} \max(0, C_{j_{[i]}} - d_{j_{[i]}})$, позволяющее сместить эти задания на более поздние позиции, не выполняется. Тогда задача решается за полиномиальное время с трудоемкостью $O(n^3)$.

Утверждение 11. Если условия утверждений 1 или 6 (для всех конкурирующих заданий) не выполняются, или выполняются условия хотя бы одного из утверждений 2 или 7–10, то задача решается посредством полиномиальной составляющей ПДС-алгоритма с трудоемкостью $O(n^3)$.

Если в расписании, полученном полиномиальной составляющей ПДС-алгоритма решения задачи О/З, нет опережающих заданий, это расписание оптимально по критерию (3).

Характеристика экспоненциальной составляющей ПДС-алгоритма решения задачи МСЗ. Алгоритм относится к ПДС-алгоритмам первого типа. Приведем общую структуру ПДС-алгоритма [24]. Решение рассматриваемой задачи осуществляется в 2 этапа: предварительного и оптимизационного.

На *предварительном этапе* реализуется:

- а) построение последовательности $\sigma^{уп}$;
- б) выполнение свободных перестановок (получена последовательность $\sigma^{сп}$);
- в) формирование множества конкурирующих заданий, т. е. запаздывающих заданий в последовательности $\sigma^{сп}$, которым в этой последовательности предшествуют задания с резервами времени и большим директивным сроком. Если таких заданий в последовательности $\sigma^{сп}$ нет, то последовательность $\sigma^{сп}$ оптимальна.

Этап оптимизации: на этом этапе последовательность $\sigma^{сп}$ декомпозируется на подпоследовательности меньшего размера. На каждой итерации строится оптимальное расписание для последовательности σ^k для заданий $j_{[i]}$, $i = \overline{1, g_k}$, где $j_{[g_k]}$ – очередное конкурирующее задание. На первой итерации рассматривается подпоследовательность $\sigma^1 \subset \sigma^{сп}$, включающая

задания, которые занимают в последовательности $\sigma^{сп}$ позиции $\overline{1, g_1}$, где g_1 – позиция первого конкурирующего запаздывающего задания. На второй итерации полученная оптимальная подпоследовательность дополняется заданиями $j_{[l]}$

, которые в последовательности $\sigma^{сп}$ занимают позиции $l = \overline{g_1 + 1, g_2}$, и строится оптимальное расписание для подпоследовательности на интервале $\overline{1, g_2}$. На каждой итерации оптимизация осуществляется за счет использования резервов времени незапаздывающих заданий (а также резервов, полученных в результате перестановок на более поздние позиции, заданий, ранее использовавших существующие резервы – заданий, помеченных «*» («**»)) очередным конкурирующим заданием и порожденными в результате его перестановок новыми запаздывающими заданиями. Оптимизация выполняется посредством ряда перестановок и встраиваний, направленных на улучшение значения функционала на рассматриваемом подмножестве заданий. Исключаются только те перестановки, которые заведомо ухудшают значение функционала. Таким образом, строится оптимальное расписание для всего множества заданий.

На каждой текущей итерации проверяются условия, сформулированные в утверждениях 1.18–1.31 [24, гл. 1], которые позволяют уменьшить число необходимых направленных перестановок для получения оптимального решения:

- уменьшить интервалы встраивания;
- сократить бесперспективные процедуры встраивания и перестановок;
- сформулировать условия декомпозиции, условия исключения заданий из множества конкурирующих и признаки оптимальности полученных решений.

Наиболее трудоемкий перебор различных вариантов использования резервов конкурирующими заданиями возникает, если для всех конкурирующих заданий выполняется

$$\forall l_{j_{[i]}} \leq l_{j_{[g]}} \quad d_{j_{[i]}} \geq d_{j_{[g]}} \quad (4)$$

причем длительности выполнения отличаются на незначительную величину, а резервы таковы, что итерации оптимизации выполняются для каждого конкурирующего задания. 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, поме-

ченних «*» («**») с последующей оптимизацией подпоследовательности, что может быть связано с экспоненциальной трудоёмкостью.

Эффективность экспоненциальной составляющей алгоритма в целом определяется следующими факторами:

1. На этапе оптимизации последовательность $\sigma^{\text{СП}}$ декомпозируется на подпоследовательности меньшего размера. Оптимизация осуществляется на подпоследовательности, ограниченной позицией встраивания очередного конкурирующего задания и позицией, занимаемой этим заданием в последовательности $\sigma^{\text{СП}}$. 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 1.24, 1.28, 1.29 [24], позволяющие часть заданий исключить из множества конкурирующих. В утверждениях 1.28, 1.29 [24] сформулированы признаки оптимальности полученных решений, что существенно сокращает число выполняемых итераций оптимизации, а следовательно, и время решения задачи.

3. Экспериментальные исследования в целом показали, что при невыполнении условия (4) при решении индивидуальных задач существует лишь незначительный перебор. И только для специальных случаев параметров задачи можно выйти на полный перебор вариантов для отдельных подпоследовательностей.

1.2. Полиномиальная аппроксимация экспоненциальной составляющей ПДС-алгоритма для Задачи ОЗ1 (алгоритм А1)

Если не реализовалась полиномиальная составляющая ПДС-алгоритма, задача ОЗ1 решается полиномиальной аппроксимацией экспоненциальной составляющей ПДС-алгоритма, основанной на положениях построения экспоненциальной составляющей ПДС-алгоритма решения задачи МСЗ.

Алгоритм А1 состоит из двух блоков:

Блок 1 – построение расписания $\sigma_{\text{опт}}$ ПДС-алгоритмом решения задачи МСЗ [24, гл. 1]; если в полученном расписании суммарное опережение равно нулю, то это расписание оптимально по критерию (3).

Блок 2 – оптимизация расписания $\sigma_{\text{опт}}$ по критерию (3). Блок 2 состоит из однотипных итераций, на каждой итерации проверяется возмож-

ность уменьшения значения функционала за счет вставки на позицию, занимаемую опережающим заданием, заданий большей длительности.

Находим в текущем расписании очередное опережающее задание $j_{[p]}$, которое занимает позицию p . На интервале $[p + 1, n]$ определяем множество заданий большей длительности:

$$N = \{i \in \overline{p+1, n} : l_{j_{[i]}} > l_{j_{[p]}}\}$$

Для заданий $j_{[k]}$ из множества N находим значение функционала на интервале переноса в результате выполнения процедуры перестановки на позицию p :

$$\left| d_{j_{[k]}} - \left(C_{j_{[k]}} - \sum_{i=p}^{k-1} l_{j_{[i]}} \right) \right| + \sum_{i=p}^{k-1} \left| (d_{j_{[i]}} - C_{j_{[i]}}) - l_{j_{[k]}} \right| + \sum_{i=k+1}^n \left| d_{j_{[i]}} - C_{j_{[i]}} \right| \quad (5)$$

Для встраивания выбираем задание $j_{[k]}$, при перестановке которого на позицию p значение функционала в соответствии с выражением (5) уменьшается на максимальную величину.

Переносим задание $j_{[k]}$ на позицию p . В полученном расписании ищем следующее опережающее задание на интервале $[p + 1, n]$, находим новое множество заданий, претендующих на перестановку, и выполняем аналогичную процедуру переноса. Такие процедуры выполняются, пока не будут рассмотрены все опережающие задания.

В полученном расписании переупорядочиваем задания на интервалах опережающих и запаздывающих заданий в соответствии с Леммами 1 и 2 [12], если это приводит к уменьшению значения функционала.

Если при реализации алгоритма не выполнялась ни одна из перестановок (см. эвристический признак оптимальности 1, приведенный ниже), то полученное расписание оптимально по критерию (3). Проверяем эвристический признак оптимальности 2, если он выполняется, то полученное расписание также оптимально по критерию (3). Конец алгоритма.

Обоснование алгоритма А1. Известны следующие свойства задачи ОЗ1.

Лемма 1 [12]. Если в последовательности, упорядоченной по неубыванию длительностей заданий, нет опережающих заданий, то такая последовательность оптимальна по критерию (3).

Лемма 2 [12]. Если в последовательности, упорядоченной по невозрастанию длительностей

заданий, нет запаздывающих заданий, то такая последовательность оптимальна по критерию (3).

Алгоритм основан на следующей эвристике.

Эвристика 1. Поиск решения при фиксированном моменте начала выполнения заданий осуществляется в окрестности, образованной множеством последовательностей, построенных в результате выполнения уменьшающих значение функционала (3) направленных перестановок в последовательности, оптимальной по критерию МСЗ, на позиции, занимаемые опережающими заданиями, заданий большей длительности. Перестановки на позиции, занимаемые заданиями без опережения, приведет к увеличению запаздывания, а перестановка заданий меньшей длительности, чем опережающие, в соответствии с Леммой 1, заведомо приведут к ухудшению значения функционала.

Эвристический признак оптимальности 1. Если в расписании, оптимальном по критерию МСЗ, не существует уменьшающих значение функционала (3) перестановок на более ранние позиции, занимаемые опережающими заданиями, заданий большей длительности, то такое расписание оптимально по критерию (3).

Обоснование. Если в результате выполнения алгоритма А1 не реализовалась ни одна из перестановок, уменьшающих значение функционала в соответствии с выражением (5), то расписание, оптимальное по критерию МСЗ, является оптимальным по критерию (3).

Эвристический признак оптимальности 2. Если в расписании, оптимальном по критерию МСЗ, позиции от 1 до k занимают запаздывающие задания, а позиции от $k + 1$ до n – опережающие задания, упорядоченные по невозрастанию длительностей, то такое расписание оптимально по критерию (3).

Обоснование. Перестановка запаздывающих заданий на более поздние позиции приведет к увеличению запаздывания по запаздывающим заданиям и опережения по опережающим заданиям, т.к. в результате перестановки они сместятся на более ранние позиции. Аналогично, перестановка опережающих заданий на более ранние позиции приведет к увеличению опережения по этим заданиям и запаздывания по запаздывающим заданиям, т.к. в результате перестановки они сместятся на более поздние позиции.

Трудоёмкость алгоритма А1 определяется трудоёмкостью алгоритма решения задачи МСЗ, равной $O(n^3)$.

1.3. Пример решения задачи алгоритмом А1

Таблица 1 – Исходные данные для примера 1

j	l_j	d_j	j	l_j	d_j
1	65	180	5	116	500
2	81	334	6	118	400
3	92	303	7	132	650
4	96	724	8	133	691

Таблица 2 – Оптимальное расписание по критерию МСЗ при $r = 0$

j	l_j	d_j	C_j	T_j
1	65	180	65	
2	81	334	146	
3	92	303	238	
6	118	400	356	
5	116	500	472	
7	132	650	604	
4	96	724	700	
8	133	691	833	142

Функционал: 142

В этом расписании для каждого опережающего задания находим множество заданий большей длительности, претендующих на перестановку, и из них выбираем задание, при перестановке которого на позицию опережающего задания значение функционала в соответствии с выражением (5) уменьшается на максимальную величину.

Таблица 3 – Оптимальное расписание по критерию О/З для примера 1

j	l_j	d_j	C_j	$ C_j - d_j $
6	118	400	118	282
1	65	180	183	3
3	92	303	275	28
2	81	334	356	22
5	116	500	472	28
7	132	650	604	46
8	133	691	737	46
4	96	724	833	109

Функционал: 564

Это расписание совпадает с оптимальным расписанием, полученным точным алгоритмом.

Задача ОЗ2. Момент начала выполнения заданий находится в интервале времени $[t_1, t_k]$

2.1. Полиномиальная составляющая ПДС-алгоритма решения задачи

Аналогично Задаче ОЗ1, признаком оптимальности решения Задачи ОЗ2 является отсутствие опережающих заданий в оптимальном расписании, полученном ПДС-алгоритмом решения задачи МСЗ [24, гл. 1] для момента начала выполнения заданий, равного t_1 .

Обоснование. Действительно, момент начала выполнения заданий должен находиться в отрезке $[t_1, t_k]$, и если для момента t_1 в расписании

нет опережающих заданий, то дальнейшее увеличение момента начала выполнения заданий приведет к ухудшению значения функционала.

Таким образом, *полиномиальная составляющая ПДС-алгоритма решения задачи* сводится к полиномиальной составляющей ПДС-алгоритма решения Задачи ОЗ1.

2.2. Полиномиальная аппроксимация экспоненциальной составляющей ПДС-алгоритма решения задачи ОЗ2

Если расписание, полученное ПДС-алгоритмом решения задачи МСЗ при $r = t_1$, не удовлетворяет признаку оптимальности, выполняется *полиномиальная аппроксимация экспоненциальной составляющей ПДС-алгоритма*, в состав которой входят алгоритмы А1, А2 и А3. *Эвристический алгоритм А3* определения фактического момента начала выполнения заданий в заданном интервале r_{ϕ} базируется на *эвристическом алгоритме А2* решения Задачи ОЗ3 (со свободным моментом начала выполнения заданий).

2.2.1. *Эвристический алгоритм решения Задачи ОЗ3 (алгоритм А2: определение самого позднего момента начала выполнения заданий, при котором достигается минимальное значение функционала О/З)*

Обозначим: $R_j = (d_j - C_j)^+$ – резерв задания j ; $R_{\min} = \min_j R_j$; N_R – число заданий с резервами; N_T – число запаздывающих заданий. N_T включает задания с нулевым резервом.

Алгоритм А2 состоит из трех блоков:

1. Построение расписания σ_0 ПДС-алгоритмом решения задачи МСЗ [24, гл. 1] для момента начала выполнения заданий $r = 0$.

2. Построение расписания, оптимального по критерию МСЗ, для $r_i > 0$.

3. Оптимизация расписания по алгоритму А1.

Блок 1

1.1. Построение расписания σ_0 , оптимального по критерию МСЗ для $r_0 = 0$. Если в расписании σ_0 нет опережающих заданий, оно оптимально по критерию О/З, $r_{\text{opt}} = 0$, конец алгоритма. Иначе переход на шаг 1.2.

1.2. Определение в расписании σ_0 N_R , N_T и R_{\min} .

1.3. Если $N_R > N_T$ (или $N_R - N_T > 1$, если число заданий нечетное), переход на 1.4, иначе на блок 3.

1.4. $r_{i+1} = r_i + R_{\min}$; $i = i + 1$. Переход на блок 2.

Блок 2 – коррекция расписания σ_0 по запаздывающим заданиям по алгоритму задачи МСЗ [24, гл. 1]. Если в полученном расписании σ_i нет опережающих заданий, оно оптимально по критерию О/З, $r_{\text{opt}} = r_i$, конец алгоритма. Иначе, если в полученном расписании σ_i $N_R > N_T$ (или $N_R -$

$N_T > 1$, если число заданий нечетное), переход на шаг 1.4, иначе на блок 3.

Блок 3 – оптимизация расписания σ_i при полученном оптимальном значении момента $r_i = r_{\text{opt}}$ алгоритмом А1. Проверка эвристических признаков оптимальности алгоритма А1. Если в полученном расписании по критерию О/З $N_R = N_T$ и $R_{\min} = k > 0$, то момент начала выполнения заданий можно увеличить на k единиц, т.е. имеем множество оптимальных моментов запуска заданий с тем же значением функционала: $M = \{r_{\text{opt}}, r_{\text{opt}} + 1, \dots, r_{\text{opt}} + k\}$. Конец алгоритма.

Обоснование алгоритма А2. Справедливы следующие утверждения.

Утверждение 12. Если в расписании σ $N_R > N_T$, то при увеличении момента начала выполнения заданий на величину R_{\min} значение функционала О/З уменьшается на $R_{\min}(N_R - N_T)$.

Доказательство. Действительно, в этом случае уменьшение суммарного опережения равно $R_{\min}N_R$, а увеличение суммарного запаздывания равно $R_{\min}N_T$. Следовательно, уменьшение значения функционала равно $R_{\min}(N_R - N_T)$.

Утверждение 13. Пусть в оптимальном расписании σ , полученном ПДС-алгоритмом решения задачи МСЗ, $N_R > N_T$. Если при увеличении момента начала выполнения заданий на величину R_{\min} и последующей оптимизации по критерию МСЗ в полученном расписании нет опережающих заданий, то такое расписание оптимально по критерию О/З для рассматриваемого момента начала выполнения заданий.

Утверждение 14. Уменьшение значения функционала при увеличении момента начала выполнения заданий происходит, пока выполняется $N_R > N_T$. При $N_T > N_R$ значение функционала увеличивается.

Доказательство основано на утверждении 12.

Утверждение 15. Пусть в расписании, оптимальном по критерию (3), $R_{\min} = k$ и $N_R = N_T$. Момент начала выполнения заданий можно увеличить на k единиц без изменения значения функционала.

Доказательство. Утверждение справедливо, т.к. в этом случае не меняются значения N_R и N_T .

Утверждение 16. Для каждого заданного момента начала выполнения r_i оптимальное расписание по критерию (3) строится на основе оптимального расписания по критерию МСЗ для рассматриваемого момента начала выполнения заданий.

Алгоритм А2 основан на следующей эвристике.

Эвристика 2. Поиск оптимального момента начала выполнения заданий и оптимального

расписания по критерию О/З осуществляется следующим образом. Для момента начала выполнения заданий $r_0 = 0$ строим оптимальное расписание по критерию МСЗ (σ_0) (п. 1.1 алгоритма). Если в расписании $\sigma_0 N_R \leq N_T$, то оптимальный момент начала выполнения заданий $r_{opt} = 0$. И если в σ_0 нет опережающих заданий, то оно оптимально по критерию О/З. Если же $N_R > N_T$, момент начала выполнения заданий увеличим на R_{min} и полученное расписание оптимизируем по критерию МСЗ (получаем расписание σ_{new}) (п. 1.3–1.4 и блок 2 алгоритма, утверждения 12, 13 и 16). Такие процедуры увеличения момента начала выполнения заданий на R_{min} с последующей оптимизацией по критерию МСЗ выполняются до тех пор, пока $N_R > N_T$ (п. 1.3–1.4 алгоритма, утверждение 14). Если в расписании $\sigma_{new} N_R \leq N_T$ для чётного числа заданий или $N_R - N_T \leq 1$ для нечётного числа заданий, то полученный момент начала выполнения заданий является оптимальным, и ему соответствует оптимальное расписание по критерию О/З ($N_R < N_T$ может быть только в том случае, если при последнем увеличении момента начала выполнения минимальное опережение, равное R_{min} , было у нескольких заданий). Расписание σ_{new} оптимизируется по критерию (3) алгоритмом А1, проверяются эвристические признаки оптимальности алгоритма А1. Если в полученном расписании по критерию О/З $N_R = N_T$ и $R_{min} = k > 0$, то момент начала выполнения заданий можно увеличить на k единиц (блок 3 алгоритма, утверждение 15).

Таким образом, *эвристическим признаком оптимальности* самого позднего момента начала выполнения заданий r_{opt} , при котором достигается минимальное значение функционала (3), является выполнение условия: в расписании, оптимальном по критерию МСЗ, полученном в результате выполнения эвристики 2, при $r = 0$ $N_R \leq N_T$, либо при $r > 0$ $N_R \leq N_T$, если число заданий чётное, или $N_R - N_T \leq 1$, если число заданий нечётное (см. пример 4 ниже).

Примечание. Если при $N_R = N_T$ в расписании σ увеличение момента начала выполнения заданий на R_{min} приводит к расписанию, в котором также $N_R = N_T$, то момент начала выполнения заданий увеличивается на R_{min} (см. пример 3).

Трудоемкость алгоритма А2 определяется трудоемкостью алгоритма задачи МСЗ, равной $O(n^3)$.

Для оценки эффективности разработанного алгоритма А2 (т.к. не были найдены результаты решения рассматриваемых задач для сравнения у других авторов), были проведены следующие исследования. Для различных моментов начала выполнения заданий с заданным шагом строи-

лись оптимальные по критерию (3) расписания точным алгоритмом. Оптимальные расписания и самые поздние моменты начала выполнения заданий, полученные алгоритмом А2, совпали с результатами, полученными точными алгоритмами.

Ниже приведены примеры решения задачи ОЗЗ разработанным алгоритмом А2, таблицы, содержащие оптимальное расписание и самые поздние моменты начала выполнения заданий, полученные точным алгоритмом, и графики, отражающие зависимости функционала О/З от моментов начала выполнения заданий.

Примеры решения задачи алгоритмом А2

Таблица 4 – Исходные данные для примера 2

j	l_j	d_j	j	l_j	d_j
1	50	500	7	100	900
2	79	534	8	102	668
3	80	700	9	121	528
4	83	604	10	130	605
5	88	987	11	147	537
6	96	951	12	150	1100

Таблица 5 – Оптимальное расписание по критерию МСЗ при $r = 0$

j	l_j	d_j	C_j	T_j	E_j
1	50	500	50		450
2	79	534	129		405
3	80	700	209		491
9	121	528	330		198
11	147	537	477		60
4	83	604	560		44
8	102	668	662		6
10	130	605	792	187	
7	100	900	892		8
5	88	987	980		7
6	96	951	1076	125	
12	150	1100	1226	126	

Функционал: **438**

В этом расписании $N_R = 9, N_T = 3, R_{min} = 6$. Принимаем момент начала выполнения заданий $r = 6$.

Таблица 6 – Оптимальное расписание по критерию МСЗ при $r = 6$

j	l_j	d_j	C_j	T_j	E_j
1	50	500	56		444
2	79	534	135		399
3	80	700	215		485
9	121	528	336		192
11	147	537	483		54
4	83	604	566		38
8	102	668	668	0	
10	130	605	798	193	
7	100	900	898		2
5	88	987	986		1
6	96	951	1082	131	
12	150	1100	1232	132	

Функционал: **456**

В этом расписании $N_R = 8, N_T = 4, R_{min} = 1$. Момент начала выполнения заданий $r = 6+1 = 7$.

Таблица 7 – Оптимальное расписание по критерию МСЗ при $r = 7$

j	l_j	d_j	C_j	T_j	E_j
1	50	500	57		443
2	79	534	136		398
3	80	700	216		484
9	121	528	337		191
11	147	537	484		53
4	83	604	567		37
8	102	668	669	1	
10	130	605	799	194	
7	100	900	899		1
5	88	987	987	0	
6	96	951	1083	132	
12	150	1100	1233	133	

Функционал: **460**

В этом расписании $N_R = 7, N_T = 5, R_{\min} = 1$. Момент начала выполнения заданий $r = 7+1 = 8$.

Таблица 8 – Оптимальное расписание по критерию МСЗ при $r = 8$

j	l_j	d_j	C_j	T_j	E_j
1	50	500	58		442
2	79	534	137		397
3	80	700	217		483
9	121	528	338		190
11	147	537	485		52
4	83	604	568		36
8	102	668	670	2	
10	130	605	800	195	
7	100	900	900	0	
5	88	987	988	1	
6	96	951	1084	133	
12	150	1100	1234	134	

Функционал: **465**

В этом расписании $N_R = N_T = 6$. Дальнейшее увеличение момента начала выполнения заданий приведет к увеличению N_T . Выбираем $r = 8$. Расписание при $r = 8$ оптимизируем по алгоритму А1 и получаем оптимальное расписание по критерию О/З (табл. 9):

Таблица 9 – Оптимальное расписание по критерию О/З для примера 2

j	l_j	d_j	C_j	$ C_j - d_j $
11	147	537	155	382
10	130	605	285	320
9	121	528	406	122
1	50	500	456	44
2	79	534	535	1
4	83	604	618	14
3	80	700	698	2
8	102	668	800	132
7	100	900	900	0
5	88	987	988	1
6	96	951	1084	133
12	150	1100	1234	134

Функционал: **1285**

Для примера 2 были получены такие расписания (табл. 10):

Таблица 10 – Оптимальные расписания по критерию О/З для примера 2

r	σ_{opt}	f_{opt}	r	σ_{opt}	f_{opt}
0		1307	10		1289
1		1305	20		1329
2		1303	30	11,10,9,1,2,4, 3,8,7,6,5,12	1369
3		1301	40		1409
4		1297	50		1449
5	11,10,9,1,2,4, 3,8,7,6,5,12	1293	60	11,9,2,1,10,4,	1492
6		1289	70	3,8,6,5,7,12	1506
7		1285	80	11,9,2,1,4,8,3,	1527
8		1285	90	10,6,5,7,12	1535
9		1287	100	11,9,2,1,4,8,3, 7,6,5,10,12	1541

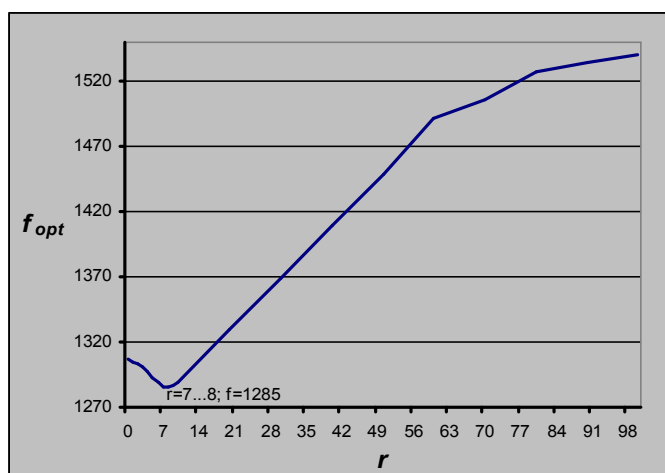


Рис. 1. Зависимость функционала О/З от r для примера 2

Таблица 11 – Исходные данные для примера 3

j	l_j	d_j	j	l_j	d_j
1	50	500	6	100	900
2	79	534	7	102	668
3	83	604	8	121	528
4	88	987	9	130	605
5	96	951	10	147	537

Таблица 12 – Оптимальное расписание по критерию МСЗ при $r = 0$

j	l_j	d_j	C_j	T_j	E_j
1	50	500	50		450
2	79	534	129		405
3	83	604	212		392
8	121	528	333		195
10	147	537	480		57
9	130	605	610	5	
7	102	668	712	44	
6	100	900	812		88
5	96	951	908		43
4	88	987	996	9	

Функционал: **58**

Имеем $N_R = 7, N_T = 3, R_{\min} = 43$. Принимаем момент начала выполнения заданий $r = 43$.

Таблица 13 – Оптимальное расписание по критерию МСЗ при $r = 43$

j	l_j	d_j	C_j	T_j	E_j
1	50	500	93		407
2	79	534	172		362
3	83	604	255		349
8	121	528	376		152
10	147	537	523		14
9	130	605	653	48	
7	102	668	755	87	
6	100	900	855		45
5	96	951	951	0	
4	88	987	1039	52	

Функционал: **187**

$N_R = 6, N_T = 4, R_{\min} = 14$. Момент начала выполнения заданий $r = 43 + 14 = 57$.

Таблица 14 – Оптимальное расписание по критерию МСЗ при $r = 57$

j	l_j	d_j	C_j	T_j	E_j
1	50	500	107		393
2	79	534	186		348
3	83	604	269		335
8	121	528	390		138
10	147	537	537	0	
9	130	605	667	62	
7	102	668	769	101	
6	100	900	869		31
5	96	951	965	14	
4	88	987	1053	66	

Функционал: **243**

В этом расписании $N_R = N_T = 5, R_{\min} = 31$. Момент начала выполнения заданий $r = 57 + 31 = 88$.

Таблица 15 – Оптимальное расписание по критерию МСЗ при $r = 88$

j	l_j	d_j	C_j	T_j	E_j
1	50	500	138		362
2	79	534	217		317
8	121	528	338		190
10	147	537	485		52
3	83	604	568		36
7	102	668	670	2	
9	130	605	800	195	
6	100	900	900	0	
4	88	987	988	1	
5	96	951	1084	133	

Функционал: **331**

В этом расписании $N_R = N_T = 5$. Дальнейшее увеличение момента начала выполнения заданий приведет к увеличению N_T . Выбираем $r = 88$. Расписание при $r = 88$ оптимизируем по алгоритму А1 и получаем оптимальное расписание по критерию О/З (табл. 16):

Таблица 16 – Оптимальное расписание по критерию О/З для примера 3

j	l_j	d_j	C_j	$ C_j - d_j $
10	147	537	235	302
8	121	528	356	172
2	79	534	435	99
1	50	500	485	15
3	83	604	568	36
7	102	668	670	2
9	130	605	800	195
6	100	900	900	0
4	88	987	988	1
5	96	951	1084	133

Функционал: **955**

В этом расписании $N_R = N_T = 5$, минимальный резерв R_{\min} – у задания 1 – равен 15. Это дает возможность увеличить момент начала выполнения заданий на 15 (до 103) без изменения значения функционала.

Таблица 17 – Оптимальные расписания по критерию О/З для примера 3

r	σ_{opt}	f_{opt}	r	σ_{opt}	f_{opt}
0	10,9,8,1,2,3,7,6,5,4	1097	103	10,8,2,1,3,7,9,6,4,5	955
70	10,8,2,1,9,3,7,6,5,4	982	104	10,8,2,1,3,7,9,6,4,5	957
80	10,8,2,1,9,3,7,6,5,4	966	105	10,8,2,1,3,7,9,6,4,5	959
85	10,8,2,1,9,3,7,6,4,5	962	110	10,8,2,1,3,7,9,6,4,5	969
87	10,8,2,1,3,7,9,6,4,5	957	120	10,8,1,2,3,7,9,6,4,5	984
88	10,8,2,1,3,7,9,6,4,5	955	130	10,8,1,2,3,7,9,6,4,5	996
89	10,8,2,1,3,7,9,6,4,5	955	140	10,8,1,2,3,7,9,5,4,6	1020
90	10,8,2,1,3,7,9,6,4,5	955	150	10,8,1,2,3,7,9,5,4,6	1054
100	10,8,2,1,3,7,9,6,4,5	955			

Зависимость функционала О/З от r для примера 3 приведена на рис. 2.

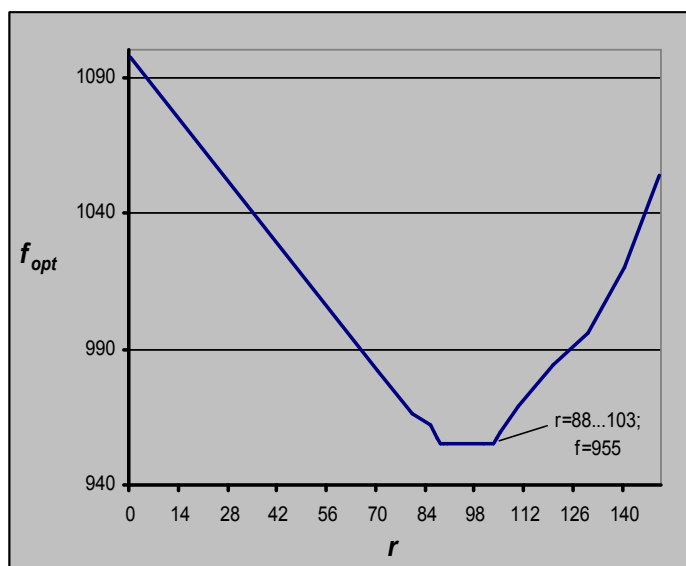


Рис. 2. Зависимость функционала О/З от r для примера 3

Таблица 18 – Исходные данные для примера 4

j	l_j	d_j	j	l_j	d_j
1	65	180	6	118	400
2	81	334	7	132	650
3	92	303	8	133	500
4	96	724	9	150	691
5	116	500			

Таблица 19 – Оптимальное расписание по критерию МСЗ при $r = 0$

j	l_j	d_j	C_j	T_j	E_j
1	65	180	65		115
2	81	334	146		188
3	92	303	238		65
6	118	400	356		44
5	116	500	472		28
8	133	500	605	105	
4	96	724	701		23
7	132	650	833	183	
9	150	691	983	292	

Функционал: 580

Имеем $N_R = 6, N_T = 3, R_{min} = 23$. Принимаем момент начала выполнения заданий $r = 23$.

Таблица 20 – Оптимальное расписание по критерию МСЗ при $r = 23$

j	l_j	d_j	C_j	T_j	E_j
1	65	180	88		92
2	81	334	169		165
3	92	303	261		42
6	118	400	379		21
5	116	500	495		5
8	133	500	628	128	
4	96	724	724	0	
7	132	650	856	206	
9	150	691	1006	315	

Функционал: 649

В этом расписании $N_R = 5, N_T = 4$.

Таблица 21 – Оптимальное расписание по критерию О/З при $r = 23$

j	l_j	d_j	C_j	$ C_j - d_j $
3	92	303	115	188
1	65	180	180	0
2	81	334	261	73
6	118	400	379	21
5	116	500	495	5
8	133	500	628	128
4	96	724	724	0
7	132	650	856	206
9	150	691	1006	315

Функционал: 936

Анализ статистических исследований. Были проведены многочисленные исследования на размерностях до 20 заданий. Оптимальные расписания и самые поздние моменты начала вы-

Таблица 22 – Оптимальные расписания по критерию О/З для примера 4

r	σ_{opt}	f_{opt}	r	σ_{opt}	f_{opt}
0	6,1,3,2,8,5,7,4,9	939	13	3,1,2,6,8,5,4,7,9	953
1	6,1,3,2,8,5,7,4,9	942	14	3,1,2,6,8,5,4,7,9	952
2	6,1,3,2,8,5,7,4,9	945	15	3,1,2,6,8,5,4,7,9	951
3	6,1,3,2,8,5,7,4,9	948	20	3,1,2,6,5,8,4,7,9	945
4	6,1,3,2,8,5,7,4,9	951	22	3,1,2,6,5,8,4,7,9	939
5	6,1,3,2,8,5,7,4,9	954	23	3,1,2,6,5,8,4,7,9	936
6	6,1,3,2,8,5,7,4,9	955	24	3,1,2,6,5,8,4,7,9	937
7	6,1,3,2,8,5,7,4,9	956	25	3,1,2,6,5,8,4,7,9	938
8	6,1,3,2,8,5,7,4,9	957	30	3,1,2,6,5,8,4,7,9	947
9	6,1,3,2,8,5,7,4,9	958	35	2,1,3,6,5,8,4,7,9	962
10	3,1,2,6,8,5,4,7,9	958	40	1,3,2,6,5,8,4,7,9	970
11	3,1,2,6,8,5,4,7,9	955	45	1,3,2,6,5,7,4,8,9	977
12	3,1,2,6,8,5,4,7,9	954			

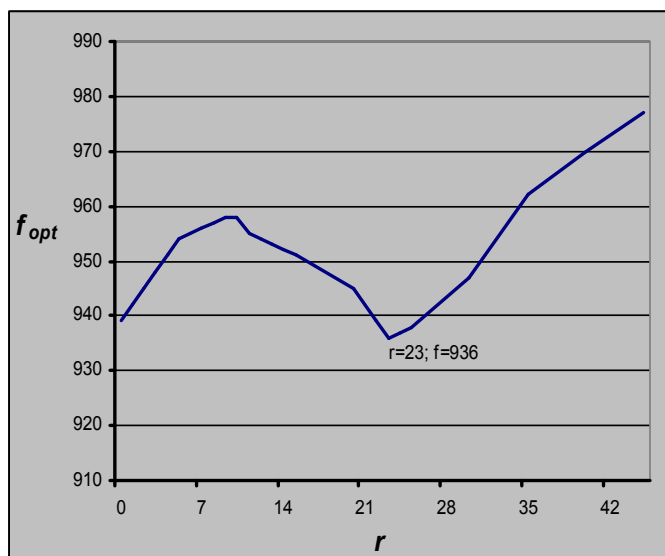


Рис. 3. Зависимость функционала О/З от r для примера 4

полнения заданий, полученные алгоритмом А2, совпали с результатами, полученными точными алгоритмами.

Кривые на графиках, отражающих зависимость функционала О/З от r , условно можно разбить на участки трех типов (см., напр., рис. 3):

1. При увеличении момента начала выполнения заданий значение функционала уменьшается за счет уменьшения опережения заданий с резервами, т.к. их число больше числа запаздывающих заданий (утверждение 14, участок [10, 23] на рис. 3).

2. При увеличении момента начала выполнения заданий значение функционала в отдельных случаях может возрастать, т.к. резервы уменьшаются, и задания большей длительности могут их не использовать (участок [0, 10] на рис.3).

3. После получения самого позднего момента начала выполнения заданий, при котором достигается оптимальное значение функционала О/З, наблюдается монотонный рост суммарного опе-

режения и запаздывания (участок [23, 45] на рис. 3).

2.2.2. Эвристический алгоритм решения Задачи ОЗ2 (алгоритм А3: определение фактического момента начала выполнения заданий в заданном интервале)

1. Построение расписания σ_0 , оптимального по критерию МСЗ для $r_0 = t_1$.

2. Находим множество оптимальных значений момента запуска заданий M , при которых достигается минимальное значение функционала О/З, алгоритмом А2. Если $M \cap [t_1, t_k] \neq \emptyset$, то множество фактических моментов запуска заданий $M_\phi = M \cap [t_1, t_k]$, переход на шаг 3. Иначе, если $\forall r \in M, r < t_1$, то фактический момент запуска заданий $r_\phi = t_1$, переход на шаг 3. Иначе если $\forall r \in M, r > t_k$, то для построения оптимального расписания по критерию О/З выполняем модифицированный алгоритм А2, конец алгоритма.

3. Для момента r_ϕ или любого из моментов $r \in M_\phi$ алгоритмом А1 строим расписание по критерию О/З. Проверяем эвристические признаки оптимальности алгоритма А1.

Модифицированный алгоритм А2:

Блок 1

1.1. Определение в расписании $\sigma_0 N_R, N_T$ и R_{\min} .

1.2. Если $N_R > N_T$ (или $N_R - N_T > 1$, если число заданий нечётное), переход на 1.3, иначе на блок 3.

1.3. $r_{i+1} = \min(r_i + R_{\min}, t_k), i = i + 1$, переход на блок 2.

Блок 2 – коррекция расписания σ_0 по запаздывающим заданиям по алгоритму задачи МСЗ [24, гл. 1]. Переход на шаг 1.2.

Блок 3 – для всех $r_i \in [t_1, t_k]$, а также моментов t_1 и t_k , алгоритмом А1 построить расписания по критерию О/З. Проверить эвристические признаки оптимальности алгоритма А1. Выбрать момент r_i^* , при котором достигается наименьшее значение функционала. Если в этом расписании $\sigma_i^* N_R = N_T$, то находим задание с наименьшим резервом R_{\min}^* . Если $R_{\min}^* = k > 0$, то с таким же значением функционала будут все расписания с моментом начала выполнения заданий $r \in [r_i^*, \min(r_i^* + k, t_k)]$ (утверждение 15). Конец алгоритма.

Обоснование алгоритма А3. Уменьшение значения функционала происходит за счет двух факторов: за счет использования резервов опережающих заданий заданиями большей длительности (алгоритм А1) и за счет увеличения момента начала выполнения заданий, при котором уменьшается суммарное опережение.

Утверждение 17. После достижения опти-

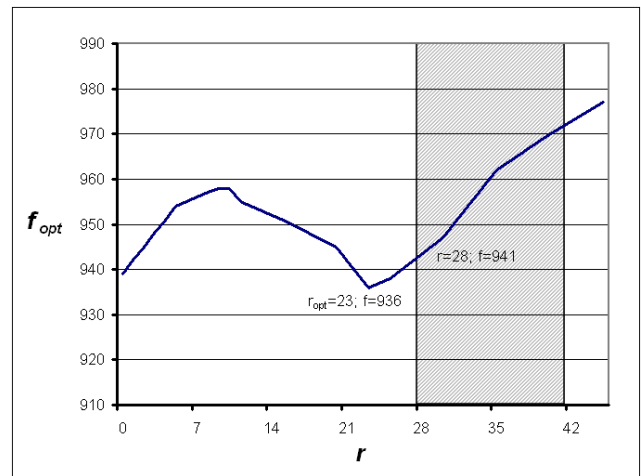
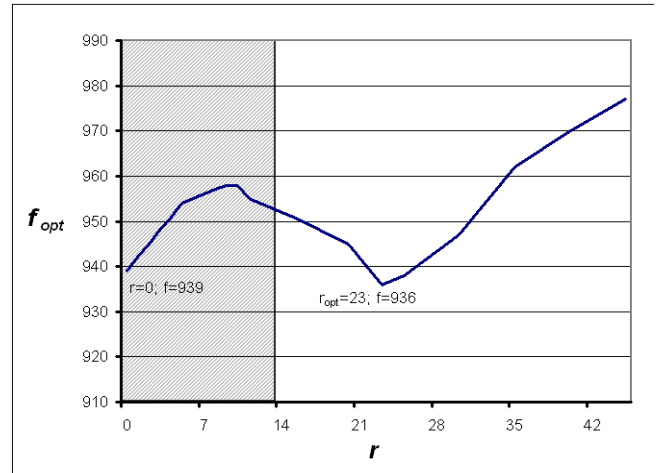


Рис. 5. Зависимость функционала О/З от r для примера 6 (задача ОЗ2)

мального момента начала выполнения заданий при его дальнейшем увеличении значение функционала монотонно возрастает.

Доказательство. В этом случае идет ухудшение по обоим факторам: менее эффективно используются резервы опережающих заданий, а число запаздывающих заданий становится больше числа опережающих (утверждение 14). ■

Если $M \cap [t_1, t_k]$ не пусто, то это пересечение и является множеством фактических моментов запуска заданий M_ϕ . Если множество M находится до t_1 (см. пример 5 ниже), то фактический момент запуска заданий равен t_1 (утверждение 17, см. п. 3 анализа статистических исследований). Если же множество M находится после t_k (см. пример 6 ниже), то выполняется модифицированный алгоритм А2, поскольку возможно как увеличение, так и уменьшение значения функционала в зависимости от момента начала выполнения заданий (п. 2 анализа статистических исследований). Это связано с тем, что в таких расписаниях существуют большие резервы (опережения), которые используют задания большой длительности, а при увеличении мо-

мента начала выполнения заданий опережения уменьшаются, и их уже не могут использовать задания, ранее их использовавшие.

Примеры решения задачи O32

Для решения задачи O32 рассмотрим пример 4 для задачи O33 (исходные данные и решение см. в таблицах 18–22 и рис. 3).

Пример 5. Задан интервал времени $[t_1, t_k] = [28, 42]$ (на рис. 4 интервал выделен штриховкой). Алгоритмом A2 получен оптимальный момент начала выполнения заданий $r_{opt} = 23 < 28$. Поэтому фактический момент запуска заданий $r_{\phi} = 28$, т.к. при нем достигается минимальное значение функционала O/3.

Пример 6. Задан интервал времени $[t_1, t_k] = [0, 14]$ (на рис. 5 интервал выделен штриховкой). Алгоритмом A2 получен оптимальный момент начала выполнения заданий $r_{opt} = 23 > 14$. Поэтому выполняем модифицированный алгоритм A2, при этом проверяются моменты начала выполнения 0 и 14, выбираем $r_{\phi} = 0$, т.к. при нем достигается минимальное значение функционала O/3.

Выводы

Разработаны эффективные ПДС-алгоритмы решения задач минимизации суммарного опережения и запаздывания на одном приборе при фиксированном моменте начала выполнения заданий или когда момент начала выполнения заданий находится в интервале времени $[t_1, t_k]$. Разработан эвристический алгоритм определения самого позднего момента начала выполнения заданий, при котором достигается минимальное значение функционала (3).

Эффективность разработанных алгоритмов подтверждается статистическими исследованиями с большим количеством примеров. Оптимальные расписания и самые поздние моменты начала выполнения заданий, полученные разработанными и точным алгоритмом, совпадают, что позволяет сделать вывод о необходимости дальнейших теоретических и статистических исследований

Список литературы

1. Feldmann M., Biskup D. Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches // *Computers & Industrial Engineering*. – 2003. – №44. – P.307–323
2. Tütüncüoğlu R.A. Sequencing with earliness and tardiness penalties. – Department of Industrial Engineering, Bilkent University, 1999. – 14 p.
3. Valente J. M. S., Alves R. A. F. S. Improved heuristics for the early/tardy scheduling problem with no idle time // *Computers & Operations Research*. – 2005. – №32. – P.557–569
4. Jin S., Mason S.J. Minimizing earliness and tardiness costs on a single machine with uncommon job due dates // Department of Industrial Engineering, Bell Engineering Center, University of Arkansas Fayetteville, 2004. – 23 p.
5. Павлов А.А., Мисюра Е.Б., Лисецкий Т.Н., Сперкач М.О., Халус Е.А. Четырехуровневая модель планирования, принятия решений и оперативного управления в сетевых системах с ограниченными ресурсами // *Вісник НТУУ «КПІ». Серія «Інформатика, управління та обчислювальна техніка»*. – К.: “БЕК+”, 2013. – №58 –с. 11–23.
6. Kanett J.J. Minimizing the average deviation of job completion times about a common due date // *Nav Res Logist // Naval Research Logistics*. – 1981. – №28. – P. 643–651.
7. Lawler E.L. A fully polynomial approximation scheme for the total tardiness problem [Текст] / E.L. Lawler // *Operations Research Letters*. – 1982. – №1. – P.207–208
8. Baker K.R. Sequencing with earliness and tardiness penalties: a review [Текст] / K.R. Baker, G.D. Scudder // *Operations Research*. – 1990. – Vol.1. – №38. – P.22–36.
9. T'kindt V., Billaut J.-C. Multicriteria scheduling: theory, models and algorithms: Springer, Berlin; Heidelberg, New York. – 2002.
10. Garey M.R. One-processor scheduling with symmetric earliness and tardiness penalties [Текст] / M.R. Garey, R.E. Tarjan, G.T. Wilfong // *Mathematics of Operations Research*. – 1988. – №13. – P.330–348.
11. Yano C.A. Algorithms for a class of single-machine weighted tardiness and earliness problems [Текст] / C.A. Yano, Y.D. Kim // *European Journal of Operations Research*. – 1991. – Vol.52. – P.167–178.
12. Ow P.S. The single machine early/tardy problem [Текст] / P.S. Ow, T.E. Morton // *Management Science*. – 1989. – Vol.2. – №35. – P.177–191.
13. Davis J.S. Single-machine scheduling with early and tardy completion costs [Текст] / J.S. Davis, J.J. Kanet // *Naval Research Logistics*. – 1993. – №40. – P. 85–101.
14. Szwarc W. Optimal timing scheduling in earliness-tardiness single machine sequencing [Текст] / W. Szwarc, S.K. Mukhopadhyay // *Naval Research Logistics*. – 1995. – №42. – P.1109–1114.

15. Sridharan V. A decision theory based scheduling procedure for single-machine weighted earliness and tardiness problem [Текст] // V. Sridharan, Z. Zhou // *European Journal of Operations Research*. – 1996. – Vol.94. – P.292–301.
16. Wan G., Yen B.P.C. Tabu search for single-machine scheduling with distinct due windows and weighted earliness/tardiness penalties // *European Journal of Operations Research*. – №142. – 2002. – P.271–281
17. Szwarc W. Adjacent ordering in single-machine scheduling with earliness and tardiness penalties // *Naval Research Logistics*. – 1993. – №40. – P.229–243
18. Lee C.Y., Choi J.Y. A generic algorithm for job sequencing problem with distinct due dates and general early-tardy penalty weights // *Computers & Operational Research*. – 1995. – №22. – P.857–869.
19. Bank J., Werner F. Heuristic algorithm for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties // *Math Comput Model*. – 2001. – №33. – P.363–383.
20. Gordon V., Proth J.P., Chu C. A survey of the state-of-art of common due date assignment and scheduling research // *European Journal of Operations Research*. – 2002. – №139. – P.1–25
21. Valente J.M.S., Alves R.A.F.S. Filtered and recovering beam search algorithms for the early/tardy scheduling problem with no idle time // *Computers & Industrial Engineering*. – 2005. – №48(2) . – P.363–375
22. Tung-I Tsai. A genetic algorithm for solving the single machine earliness/tardiness problem with distinct due dates and ready times // *International Journal of Advanced Manufacturing Technologies*. – 2007. – №32. – P.994–1000
23. Hoogeveen J.A., Van de Velde L.S. A branch-and-bound algorithm for single-machine earliness-tardiness scheduling with idle time // *INFORMS Journal of Computing*. – 1996. – №8. – P.402–412.
24. Згуровский М.З., Павлов А.А. Принятие решений в сетевых системах с ограниченными ресурсами: Монография. – К.: Наукова думка, – 2010. – 573 с.
25. Павлов О.А., Мисюра О.Б., Шевченко К.Ю. Побудова ПДС-алгоритму розв'язання задачі мінімізації сумарного зваженого запізнення виконання робіт на одному приладі // *Вісник НТУУ “КПІ”. Серія «Інформатика, управління та обчислювальна техніка»*. – К.: “БЕК+”, 2012. – №56. – С. 58–70.
26. Павлов А.А., Мисюра Е.Б., Костик Д.Ю. Минимизация суммарного запаздывания при наличии заданных с отрицательными значениями директивных сроков // *Вісник НТУУ “КПІ”. Серія «Інформатика, управління та обчислювальна техніка»*. – К.: “БЕК+”, 2011. – №53. – С.3–5.