УДК 519.854.2

ЗГУРОВСКИЙ М.З.,
ПАВЛОВ А.А.,
МИСЮРА Е.Б.,
МЕЛЬНИКОВ О.В.,
МУХА И.П.,
ЛИЩУК Е.И.

Эвристический алгоритм решения задачи суммарного взвешенного запаздывания на одном приборе

Рассматривается NP-трудная в сильном смысле задача комбинаторной оптимизации по критерию минимизации суммарного взвешенного опоздания, входящая в состав математического обеспечения четвертого уровня четырехуровневой модели планирования (включая оперативное) и принятия решений. На основе исследования свойств ранее разработанного ПДС-алгоритма решения задачи и трудоемкости отдельных его процедур создан новый эвристический алгоритм, позволяющий решать задачи реальной практической размерности. Исследована его эффективность.

We consider NP-hard in the strong sense problem of combinatorial optimization with the criterion of minimizing the total weighted tardiness. This problem is part of the mathematical support of the four-level planning model (including operational) and decision making on its fourth level. On the basis of the properties research of the previously developed PSC-algorithm for the problem solving and the complexity study of its individual procedures, a new heuristic algorithm was created that allows to solve problems of real practical dimension. Its efficiency is investigated.

Ключевые слова: Составление расписаний, эвристики, NP-трудные задачи

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 $d_j \geq 0$. 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 выполнения задания j.

Данная задача входит в состав математического обеспечения четвертого уровня четырехуровневой модели планирования (включая оперативное) и принятия решений [1], на котором осуществляется оперативная корректировка плана производства, построенного на третьем уровне модели, в случае срывов в процессе его выполнения.

В [2] приведен ПДС-алгоритм решения данной задачи, включающий полиномиальную составляющую и точный экспоненциальный подалгоритм. ПДС-алгоритм строит оптимальное расписание для задач размерности до 1000 заданий, а практические задачи могут включать десятки тысяч заданий. Таким образом, актуальным остается создание эвристических алгоритмов для решения задач практической размерности.

Описание логики построения ПДС-алгоритма

ПДС-алгоритм основан на перестановках, направленных на использование запаздывающими заданиями резервов времени ($d_j - C_j > 0$) предшествующих незапаздывающих заданий.

Алгоритм состоит из k однотипных итераций, где k – количество конкурирующих запаздывающих заданий в последовательности $\sigma^{сп}$ ($\sigma^{сп}$ – последовательность, упорядоченная по приоритетам заданий ω_j/l_j, в которой выполнены перестановки незапаздывающих заданий на более поздние позиции в случае, если на интервале перестановки имеются запаздывающие задания). В процессе решения задачи часть конкурирующих заданий по определенным правилам исключается из множества конкурирующих, в связи с чем число выполняемых итераций уменьшается.

На первой итерации строится оптимальное расписание для заданий на интервале $\overline{1, g_1}$ последовательности $\sigma^{сп}$, где $j_{[g_1]}$ – первое запаздывающее конкурирующее задание. На следующей итерации рассматривается подпоследовательность заданий на интервале $\overline{1, g_2}$, в которой позиции $\overline{1, g_1}$ занимает оптимальная подпоследовательность, полученная на первой итерации; позиции $\overline{g_1+1, g_2}$ занимают задания, стоящие на этих позициях в последовательности $\sigma^{сп}$; $j_{[g_2]}$ – следующее запаздывающее конкурирующее задание в последовательности $\sigma^{сп}$. И т.д., пока не будут выполнены итерации оптимизации для всего множества конкурирующих заданий.

Рассмотрим реализацию итерации оптимизации 1. Выполняется процедура встраивания первого конкурирующего запаздывающего задания $j_{[g_1]}$ на позицию p, на которой взвешенное запаздывание задания $j_{[g_1]}$ будет минимальным или равным нулю; задания на интервале встраивания упорядочиваются по невозрастанию их приоритетов и выполняются свободные перестановки. На интервале встраивания могут появиться порожденные запаздывающие задания. Необходимо построить оптимальное расписание на подпоследовательности $\overline{1, g_1}$ посредством перераспределения резервов незапаздывающих заданий между встроенным заданием и порожденными запаздывающими заданиями. С этой целью выполняется пошаговая оптимизация на текущей подпоследовательности для каждого очередного запаздывающего задания. При этом используются резервы как незапаздывающих заданий на интервале встраивания $\overline{p, g-1}$, так и заданий, занимающих позиции $\overline{1, p-1}$, соответствующих следующему требованию: эти задания должны иметь резервы на интервале встраивания очередного запаздывающего задания. Выполнение итерации заканчивается, когда выполнена оптимизация для всех запаздывающих заданий текущей подпоследовательности и построена оптимальная подпоследовательность для множества заданий на интервале $\overline{1, g_1}$.

На каждой следующей итерации для очередного конкурирующего запаздывающего задания $j_{[g_k]}$ реализуется процедура встраивания его на более раннюю позицию в оптимальную подпоследовательность, полученную на предыдущей итерации, и затем, аналогично итерации 1, осуществляется пошаговая оптимизация на текущей подпоследовательности для каждого очередного запаздывающего задания и перераспределение резервов на интервале $\overline{1, g_k}$.

На каждой итерации значение функционала уменьшается или остается неизменным, что позволяет на основе ПДС-алгоритма строить приближенные или эвристические алгоритмы.

В соответствии с утверждением 11 [3], ПДС-алгоритм включает следующие основные процедуры, выполняемые как для конкурирующих заданий, так и для запаздывающих заданий на интервале встраивания:

а) определение и расширение интервала встраивания (позиции p, на которой запаздывание задания будет минимальным);

б) определение наличия резервов времени на расширенном интервале встраивания;

в) в случае наличия резервов времени на расширенном интервале встраивания выполнение процедуры встраивания запаздывающего задания на позицию, определенную интервалом встраивания. Все задания, занимающие позиции после встроенного запаздывающего задания, упорядочиваются по невозрастанию их приоритетов 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;

д) уменьшение взвешенного запаздывания запаздывающих заданий за счет резервов, освобожденных заданиями, ранее их использовавшими (помеченными заданиями), с помощью перестановки их на более поздние позиции.

Выделим процедуры реализации текущей итерации точного ПДС-алгоритма, связанные с экспоненциальной трудоемкостью.

1. В результате выполнения процедуры встраивания и дальнейшей пошаговой оптимизации на интервале встраивания задания на этом интервале переупорядочиваются и расписание перестраивается. Если для очередного запаздывающего задания $j_{[k]}$ позиция встраивания меньше, чем у предыдущего, для которого выполнялась процедура встраивания, то все задания, включая задания на предыдущем интер-

вале встраивания, переупорядочиваются, и на этом расширенном интервале заново запускаются процедуры оптимизации, что может привести к экспоненциальной трудоемкости.

2. Процедура д) выполняется в случае, если в результате выполнения оптимизации (процедура в)) задание остается запаздывающим. Процедура д) включает рекурсивный вызов процедур а)–г) для подпоследовательности заданий на интервале перестановки помеченного задания. При этом для помеченного задания в свою очередь ищется другое помеченное задание, которое может быть встроено после него, что приведет к уменьшению значения функционала. В этом случае осуществляется перебор помеченных задач с последующей оптимизацией подпоследовательности, что может быть связано с экспоненциальной трудоемкостью. Процедуры а)–г) имеют полиномиальную трудоемкость.

3. Самый трудоемкий перебор вариантов использования резервов конкурирующими заданиями в точном алгоритме может возникнуть, если для подпоследовательности K конкурирующих заданий выполняется:

$$\forall i < g, j_{[i]}, j_{[g]} \in K,\ l_{j_{[i]}} - l_{j_{[g]}} \le \Delta_1,$$

$$d_{j_{[i]}} - d_{j_{[g]}} \ge \Delta_2,\ \frac{\omega_{j_{[i]}}}{l_{j_{[i]}}} - \frac{\omega_{j_{[g]}}}{l_{j_{[g]}}} \ge \Delta_3,$$

где Δ_1, Δ_2, Δ_3 – незначительные величины, причем резервы таковы, что итерации оптимизации выполняются для каждого конкурирующего задания.

Очевидно, что все остальные процедуры имеют полиномиальную трудоемкость.

В данной статье разработаны эвристически обоснованные модификации процедуры д), позволяющие проводить сокращенный направленный перебор помеченных заданий, что приводит к получению наиболее эффективного решения по показателю качества.

Эвристические модификации процедур ПДС-алгоритма

Приведем и обоснуем эвристики, позволяющие исключить перебор и построить полиномиальный эвристический алгоритм решения задачи на основе точного ПДС-алгоритма.

Эвристика 1. В последовательности $\sigma^{сп}$ все конкурирующие задания множества K упорядочиваются по неубыванию директивных сроков. В результате исключаются условия, при которых после переупорядочения заданий на расширенном интервале встраивания в соответствии с приоритетами требуется повторный запуск процедур оптимизации (см. п. 1 выше). При этом выполняются только перестановки, улучшающие функционал (свободные перестановки и перестановки в соответствии с утверждениями 7 и 8 [3]). Они имеют полиномиальную трудоемкость.

Эвристика 2. Процедура д) выполняется для рассматриваемого запаздывающего задания следующим образом. Звездочкой (двумя звездочками) помечаются только конкурирующие задания. Для перестановки на более поздние позиции выбирается помеченное задание с самым низким приоритетом, проверяется условие:

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

(см. утверждение 14 [3]), и если оно выполняется, то перестановка реализуется и на интервале перестановки осуществляются перестановки, улучшающие функционал (свободные перестановки и перестановки в соответствии с утверждениями 7 и 8 [3]). Если задание, для которого выполнялась процедура д), стало незапаздывающим, переходим к следующему запаздывающему заданию. Иначе ищем следующее помеченное задание и выполняем аналогичные процедуры. Таким образом, исключается перебор различных комбинаций перестановки помеченных заданий.

Трудоемкость выполнения модифицированной процедуры д) определяется функцией $O(n^2)$.

Эвристический алгоритм решения задачи

1. Построение последовательности $\sigma^{уп}$ (упорядочение заданий по невозрастанию приоритетов).

2. Построение последовательности $\sigma^{сп}$ (выполнение свободных перестановок).

3. Определение множества конкурирующих заданий K в последовательности $\sigma^{сп}$ и упорядочение их по неубыванию директивних сроков.

4. Выбираем очередное конкурирующее задание $j_{[g]}$ из последовательности $\sigma^{сп}$.

5. Определяем позицию $p^{н}$ встраивания задания $j_{[g]}$ (процедура а)).

6. Анализ резервов на интервале $\overline{1, p^{н}-1}$ (процедура б)). Если задания, необходимые для расширения интервала встраивания, не найдены, выполняем коррекцию позиции встраивания (процедура г)). Встраиваем задание $j_{[g]}$ на

определенную позицию $p^{н}$ и помечаем его звездочкой (процедура в)).

7. Если задание $j_{[p^i]}$ остается запаздывающим, то для каждого очередного запаздывающего задания на интервале встраивания, включая задание $j_{[p^i]}$, выполняем улучшающие перестановки (свободные перестановки и перестановки в соответствии с утверждениями 7 и 8 [3]).

8. Если рассматриваемое задание осталось запаздывающим, то выполняем модифицированную процедуру д), рассматривая для освобождения резервов только помеченные конкурирующие задания, начиная с помеченного задания с самым низким приоритетом.

9. Если просмотрены все конкурирующие задания, конец алгоритма. Иначе, переход на шаг 4.

Статистические исследования разработанного эвристического алгоритма показали его высокую эффективность. Решались задачи с размерностью до 500 заданий. Расписания, полученные эвристическим алгоритмом, сравнивались с расписаниями, полученными точным ПДС-алгоритмом. Отклонение показателя качества полученных расписаний находилось в пределах 8–10% от полученных ПДС-алгоритмом.

Выводы

На основании ранее разработанного ПДС-алгоритма для задачи суммарного взвешенного запаздывания при выполнении независимых заданий с директивными сроками одним прибором [2] представлен эвристический алгоритм с трудоемкостью O(n^2). Приведены и обоснованы эвристики, позволяющие исключить экспоненциальный перебор в ПДС-алгоритме различных вариантов выполняемых перестановок. Показано, что все остальные процедуры имеют полиномиальную трудоемкость. Экспериментальные исследования на задачах с размерностью до 500 заданий показали высокую эффективность разработанного алгоритма.

Список литературы

1. Згуровский М.З. Методология построения четырехуровневой модели планирования, принятия решений и оперативного планирования в сетевых системах с ограниченными ресурсами / М.З. Згуровский, А.А. Павлов, Е.Б. Мисюра и др. // Вісник НТУУ “КПІ”. Серія «Інформатика, управління та обчислювальна техніка». – К.: “ВЕК+”, 2014. – №61. – с.60–84
2. 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.
3. Згуровский М.З., Павлов А.А. Теоретические свойства ПДС-алгоритма для задачи минимизации суммарного взвешенного запаздывания на одном приборе // Вісник НТУУ “КПІ”. Серія «Інформатика, управління та обчислювальна техніка». – К.: “ВЕК+”, 2017. – №65 – С.4-14.