УДК 519.854.2

ПДС-АЛГОРИТМЫ И ТРУДНОРЕШАЕМЫЕ ЗАДАЧИ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ

М.З. ЗГУРОВСКИЙ, А.А. ПАВЛОВ, Е.Б. МИСЮРА

Формулируются базовые принципы построения ПДС-алгоритмов. Приводятся обобщения повышения эффективности их использования. Анализируется возможность создания новых ПДС-алгоритмов на основе существующих.

ВВЕДЕНИЕ

Большая часть дискретных математических моделей, построенных для анализа, синтеза и функционирования сложных систем, является либо NP-полными задачами [1], либо не менее сложными. В частности, к классу NP-полных задач относится подавляющее большинство задач теории расписаний.

Вычислительная сложность точных алгоритмов решения комбинаторных задач оптимизации из класса NP определяется выражением $O(2^{p(n)})$, где p — полиномиальная функция; n — размерность конкретной системы однозначной записи задачи. Иными словами, точные алгоритмы решения таких задач, представляющие модели реальных систем, создают проблему трансвычислительной сложности [2]. Поэтому практические задачи решаются приближенными либо эвристическими методами. Приближенных методов с полиномиальной оценкой сложности, гарантирующих наперед заданную точность (в предположении $P \neq NP$), не существует. Эвристические методы не гарантируют качества решений [1]. Можно также отметить, что в некоторых случаях ограничения, накладываемые на формулировку задачи комбинаторной оптимизации из класса NP, приводят к решению по полиномиальной вычислительной схеме. Например, труднорешаемая задача комбинаторной оптимизации «Минимизация суммарного взвешенного момента окончания выполнения частично упорядоченного множества заданий при отношении порядка, заданном ориентированным ацикличным графом» в случае, когда ориентированный граф является деревом либо последовательно-параллельным, точно решается полиномиальным алгоритмом [1].

В работах [3–5] развивается новый подход к построению точных эффективных алгоритмов для труднорешаемых задач комбинаторной оптимизациизации (ПДС-алгоритмы). Под ПДС-алгоритмом понимается точный алгоритм, содержащий полиномиальную и декомпозиционную составляющую.

Общая методология построения алгоритма заключается в следующем. На основе теоретических свойств исследуемой комбинаторной задачи:

а) находятся логико-аналитические условия (*p*-условия), выполнение которых в процессе решения ПДС-алгоритмом произвольной индивидуальной задачи данного класса приводит к нахождению оптимального решения вычислительной процедурой полиномиальной сложности (при этом заранее определить, будут ли для произвольной индивидуальной задачи выполняться эти условия, невозможно);

б) в процессе решения исходная задача декомпозируется на совокупность задач меньшей размерности.

В теории расписаний на основе этого подхода построены эффективные точные ПДС-алгоритмы для следующих задач [4–8]:

- Минимизация суммарного запаздывания при выполнении независимых заданий на одном приборе (МСЗ).
- Минимизация суммарного взвешенного момента окончания выполнения частично упорядоченного множества заданий при отношении порядка, заданном ориентированным ацикличным графом (МВМ).
- Минимизация суммарного запаздывания при выполнении независимых заданий с равными директивными сроками параллельными приборами (МСЗП).

ПОСТАНОВКА ЗАДАЧИ МСЗ

Дано множество независимых заданий $J = \{j_1, j_{2,} \ldots, j_n\}$, состоящих из одной операции. Для каждого из них известны длительность l_j и директивный срок d_j выполнения. Задания поступают в систему одновременно в момент времени $a_j = 0,\ j = \overline{1, n}$. Прерывания не допускаются. Необходимо построить расписание выполнения заданий для одного прибора, минимизирующее суммарное опоздание при выполнении заданий.

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

где C_j — момент завершения выполнения задания j.

Согласно Д. Ду и Д. Люнгу [9], задача является NP-трудной. Обзор известных методов ее решения приведен в работе [10].

ПОСТАНОВКА ЗАДАЧИ МВМ

Имеется частично-упорядоченное множество заданий $J = \{j_1, j_{2,} \ldots, j_n\}$, которые, начиная с нулевого момента времени, обслуживаются одним прибором, без прерываний, не более, чем одно в единицу времени. Каждому заданию j поставлены в соответствие его длительность $l_j > 0$ и вес

ω_j (произвольное действительное число, в приложении теории расписания >0). Орграф частичного упорядочивания ацикличен.

Необходимо найти последовательность обслуживания заданий, суммарный взвешенный момент окончания выполнения работ в которой минимален.

$$f=\sum_{k=1}^{n}\omega_{j[k]}C_{j[k]}\to\min\,, \tag{2}$$

где $C_{j[k]}=\sum_{s=1}^{k}l_{j[k]}$ — момент завершения выполнения задания, стоящего в допустимом расписании на k-й позиции.

Эта задача является NP-трудной в сильном смысле и останется таковой, если все длительности или все веса заданий равны 1. Она разрешима за полиномиальное время, если порядок отношений предшествования является лесом или последовательно-параллельным графом [11].

ПОСТАНОВКА ЗАДАЧИ МСЗП

Дано множество заданий J, число приборов m, для каждого $j\in J$ известна длительность выполнения l_j. Все задания имеют общий директивный срок d. Необходимо построить расписание σ выполнения заданий $j\in J$ на m приборах равной производительности такое, чтобы достигался минимум функционала

$$f(\sigma)=\sum_{j\in J}\max[0;C_j(\sigma)-d], \tag{3}$$

где $C_j(\sigma)$ — момент завершения выполнения задания j в последовательности σ.

Предполагается, что все задания множества J поступают на выполнение одновременно и обслуживаются без прерываний.

Сформулированная задача относится к классу NP-трудных задач [1] и разрешима за псевдополиномиальное время при $m=2$.

ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ПДС-АЛГОРИТМОВ

Принадлежность к классу перестановочных расписаний

Критерии (1)–(3) принадлежат к одному и тому же классу, называемому классом регулярных критериев. Регулярный критерий R является возрастающей функцией момента окончания каждого из n заданий.

Теорема 1 [12]. Для системы $n|1$ расписание, оптимальное относительно регулярного критерия, принадлежит классу, исключающему прерывания или искусственные простои.

Основным результатом этой теоремы является то, что поиск оптимального решения должен проводиться в классе перестановочных расписаний

[12], общее количество которых в случае n работ задается одной из $n!$ возможных перестановок их номеров.

Направленность перестановок

Во избежание полного перебора необходимо определить возможность проведения направленных перестановок, позволяющих отсечь бесперспективные варианты решения. Рассмотрим типы направленных перестановок для каждой из трех задач.

Задача МСЗ

Справедлива следующая теорема.

Теорема 2 [4]**.** Если в последовательности, упорядоченной по неубыванию значений длительности заданий (последовательность $\sigma^{\text{уп}}$), запаздывающим заданиям не предшествуют задания $j_{[i]}$, для которых $r_{j_{[i]}} = d_{j_{[i]}} - C_{j_{[i]}} > 0$, то не существует переносов заданий (перестановок и встраиваний), приводящих к улучшению целевой функции, где $r_{j_{[i]}} = d_{j_{[i]}} - C_{j_{[i]}}$ — резерв времени задания $j_{[i]}$, а i — позиция, занимаемая заданием $j_{[i]}$ в последовательности $\sigma^{\text{уп}}$.

Введем следующие определения.

Определение 1. Перестановкой (EFSR — extraction and forward-shifted reinsertion — извлечение и повторная вставка со сдвигом вперед) называется процедура переноса задания $j_{[g]}$ на позицию k $(k > g)$ и одновременно заданий, занимающих позиции $g+1, g+2, \ldots, k-1, k$, на позиции $g, g+1, \ldots, k-1$, соответственно.

Определение 2. Интервалом перестановки задания $j_{[g]}$ на позицию k называется множество заданий, находящихся на позициях $g+1, g+2, \ldots, k-1, k$ до выполнения перестановки.

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

Определение 4. Интервалом встраивания $I_{j_{[g]}}$ задания $j_{[g]}$ называется множество заданий, стоящих до встраивания на позициях $p, p+1, \ldots, g-1$, где p определяется из условия

$$\sum_{i=p-1}^{g-1} l_{j_{[i]}} < C_{j_{[g]}} - d_{j_{[g]}} \le \sum_{i=p}^{g-1} l_{j_{[i]}} . \tag{4}$$

Если же условие (4) не выполняется ни для одной позиции, то $p = 1$. Таким образом, опоздание по заданию j на позиции p должно быть минимально или равно нулю.

Справедлива следующая теорема.

Теорема 3 [4]. Пусть в последовательности $\sigma^{\text{уп}}$ $j_{[g]}$ — 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:

$$1.\ \exists\, j_{[i]},\ P_{\min} \le i < g \,|\, r_{j_{[i]}} > 0,\ d_{j_{[i]}} > d_{j_{[g]}}.$$

На интервале встраивания задания $j_{[g]}$ есть задания $j_{[i]}$ с резервами времени, для которых $d_{j_{[i]}} > d_{j_{[g]}}$. $P_{\min}$ — позиция, где запаздывание по заданию $j_{[g]}$ минимально или равно нулю.

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

В последовательности $\sigma^{\text{уп}}$ 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 его окончания. 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 $j_{[q]}$.

$$4.\ \forall\, i \,|\, P_{\min} \le i < g \,|\, r_{j_{[i]}} \le 0,$$

но
$$\exists\, j_{[k]},\, |\, k < p \,|\, d_{j_{[k]}} > D_{j_{[g]}},\ r_{j_{[k]}} > 0,\ d_{j_{[k]}} > C_{j_{[p-1]}}.$$

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.

Выполнение одного из условий 1–4 означает, что на интервале встраивания задания $j_{[g]}$ существуют резервы (условия 1–3) или они будут образованы в результате перестановок (условие 4). При невыполнении условия 4 перестановка заданий, занимающих позиции $\overline{1, p-1}$, на интервал встраивания задания $j_{[g]}$ приведет к запаздыванию этих заданий, и так как они имеют меньшую длительность, чем $j_{[g]}$, то встраивание задания $j_{[g]}$ приведет к увеличению значения функционала. Следовательно, если не выполняется ни одно из условий $1-4$, то не существует встраиваний задания $j_{[g]}$ на более ранние позиции, приводящих к улучшению целевой функции.

Следствие. Пусть в последовательности $\sigma^{\text{уп}}$ число запаздывающих заданий $n_з > 1$. Тогда, если в последовательности $\sigma^{\text{уп}}$ ни для одного из запаздывающих заданий $j_{[g]}$ нет предшествующих заданий $j_{[s]}$, $s < g$, для которых выполняется хотя бы одно из условий 1–4, то последовательности $\sigma^{\text{уп}}$ отвечает оптимальное значение функционала.

Разработанный ПДС-алгоритм построен на перестановках и встраиваниях, направленных на оптимальное использование запаздывающими заданиями резервов времени незапаздывающих заданий. В процессе решения используются следующие типы перестановок и встраиваний.

- Перестановки (EFSR):

а) заданий, для которых резерв времени больше нуля;

б) заданий, ранее использовавших резервы в результате перестановок и встраиваний.

Перестановки осуществляются, если при их выполнении значение функционала уменьшается.

- Встраивания (EBSR):

а) при наличии резервов на интервале встраивания $\overline{p, g-1}$ запаздывающего задания $j_{[g]}$, где p — позиция встраивания задания $j_{[g]}$;

б) если резервы на интервале $\overline{p, g-1}$ отсутствуют, при наличии резервов на интервале $\overline{1, p-1}$ у заданий $j_{[i]}$, для которых $d_{j_{[l]}} > d_{j_{[g]}}$, $d_{j_{[l]}} > C_{j_{[l]}}$, $d_{j_{[l]}} > C_{j_{[p-1]}}$.

В качестве примера приведем условия реализации полиномиальной составляющей алгоритма, полученные на основе исследования свойств последовательности $\sigma^{\text{уп}}$.

Теорема 4 [4]. Пусть в последовательности $\sigma^{\text{уп}}$ $j_{[g]}$ — первое запаздывающее задание, и выполняется $\max r(\sigma^{\text{уп}}) \le l_{j_{[g]}}$. В этом случае оптимальное значение функционала достигается за полиномиальное время с трудоемкостью, определяемой функцией $O(n \log n)$.

Задача МВМ

Направленность перестановок основывается на следующей теореме.

Теорема 5 [5]**.** Оптимальное расписание является *P*-упорядоченным.

P-упорядоченным расписанием [5] является такая допустимая последовательность заданий, которая обладает следующими свойствами:

а) первым выполняется подмножество $S_1 \subset J$ (J — множество всех заданий), имеющее максимально возможный приоритет, т.е.

$$P(S_1) \ge P(U), \ \forall U \subset J; \tag{5}$$

б) подмножество S_1 содержит максимально возможное количество заданий при условии выполнения неравенства (5);

в) вторым выполняется подмножество S_2, удовлетворяющее свойствам а) и б) на множестве $J \setminus S_1$, и т.д.

Теорема 6 [5]**.** Пусть на множестве расписаний P задан функционал $f(\Pi) = \sum_{k=1}^{n} \omega_{j_{[k]}} C_{j_{[k]}}$. Тогда для двух произвольных перестановок $\Pi' = (\Pi^{(1)}, \Pi^{(a)}, \Pi^{(b)}, \Pi^{(2)})$ и $\Pi'' = (\Pi^{(1)}, \Pi^{(b)}, \Pi^{(a)}, \Pi^{(2)})$, принадлежащих P, существует функция $f(\Pi)$, которая удовлетворяет таким свойствам: из условия $f(\Pi^{(a)}) > f(\Pi^{(b)})$ следует неравенство $f(\Pi') \le f(\Pi'')$ и из $f(\Pi^{(a)}) = f(\Pi^{(b)})$ — равенство $f(\Pi') = f(\Pi'')$.

Данная функция называется приоритетом перестановки и обозначается

$$P(\Pi) = \sum_{j \in \{\Pi\}} \omega_j \Big/ \sum_{j \in \{\Pi\}} l_j .$$

Алгоритм основан на перестановках, направленных на построение P-упорядоченного расписания.

В работе [5] введены понятия цепочек и конструкций общего вида и определены их свойства. На этой основе разработан ПДС-алгоритм, строящий оптимальное расписание для индивидуальной задачи МВМ, отношения предшествования которых имеют вид цепочек, простых или сложных конструкций произвольной степени вложения. Оптимальное расписание получается также в том случае, если направленные ребра, нарушающие p-условия алгоритма, не принимают участия в построении оптимального расписания. Например, когда алгоритм не использует отношения предшествования, нарушающие p-условия при построении множеств максимальных приоритетов, а сами множества имеют вид цепочек или конструкций произвольного вида. Алгоритм основывается на анализе приоритетов и использовании перестановочного приема для следующих непосредственно друг за другом структур взаимосвязанных заданий.

Задача МСЗП

Направленность выполнения перестановок определяется на основе таких теорем [6].

Пусть исходное множество заданий J упорядочено по неубыванию значений длительностей выполнения и разбито на (необязательно непустые) подмножества $(J_1, J_2, \ldots, J_m)$, попарно не имеющие общих элементов, где J_i — множество заданий, обслуживаемых i-м прибором, $i = \overline{1, m}$.

Поиск оптимального расписания в соответствии с известной теоремой [13] можно ограничить рассмотрением расписаний, при которых каждый прибор обслуживает задания в порядке возрастания их длительностей.

Теорема 7 [6]**.** Существует оптимальное расписание, при котором выполняются условия:

1. $P \bigcup S = \{1, 2, \ldots, |P \bigcup S\}$.

2. Если $|P \bigcup S| < n$, то $\sum\limits_{j \in P_i \bigcup S_i} l_j \ge d$ и $Q_i \setminus S_i$ содержит те и только те элементы, которые отличаются от $|P \bigcup S| + i$ на величину, кратную m, $i = \overline{1, m}$, где

$$P = \bigcup_{i=\overline{1,m}} P_i\ ,\ S = \bigcup_{i=\overline{1,m}} S_i\ ,\ Q = \bigcup_{i=\overline{1,m}} Q_i\ ,$$

$P_i(\sigma)$ — множество незапаздывающих заданий в расписании прибора i;

$S_i(\sigma)$ — множество запаздывающих заданий в расписании прибора i, для которых выполняются условия

$$S_j^{\text{н}} < d\ ,\ C_j > d\ ,\ \forall\, j \in S_i(\sigma)\ ,$$

где $S_j^{\text{н}}$ — момент начала выполнения задания j;

$Q_i(\sigma)$ — множество запаздывающих заданий в расписании прибора i, для которых выполняются условия

$$S_j^{\text{н}} > d\ ,\ \forall\, j \in Q_i(\sigma)\ ;$$

R_i — резерв времени прибора i в расписании σ

$$R_i = d - \sum_{j \in P_i(\sigma)} l_j\ ;$$

$\Delta_i(\sigma)$ — запаздывание в выполнении задания $f \in S_i(\sigma)$ относительно директивного срока

$$\Delta_i = \sum_{j \in P_i(\sigma) \bigcup S_i(\sigma)} l_j - d\ .$$

Пусть Ψ_{PS} — класс расписаний, удовлетворяющий условию теоремы 7; $\Psi_P \subset \Psi_{PS}$ — класс расписаний, удовлетворяющий дополнительно следующим условиям:

1. $P = \{1, 2, \ldots, |P\}$.
2. $\min\limits_{j \in S(\sigma)} l_j > \max\limits_{i=\overline{1,m}} R_i(\sigma)$.
3. $S_{j_k}^H \le S_{j_l}^H$, если $l_{j_k} \le l_{j_l}$, $\forall\, j_k, j_l \in S(\sigma)$.

Теорема 8 [6]**.** Если в расписании $\sigma \in \Psi_P$, $\Omega_\Sigma(\sigma) = \min\{R_\Sigma(\sigma), \Delta_\Sigma(\sigma)\} = 0$, то расписание оптимально.

Величина $\Omega_\Sigma(\sigma)$ является основной характеристикой расписания, где $R_\Sigma(\sigma)$, $\Delta_\Sigma(\sigma)$ показывают, насколько можно теоретически уменьшить значение функционала, чтобы получить оптимальное расписание.

Теорема 9 [6]**.** Расписание с одинаковым числом запаздывающих заданий на приборах (равномерное расписание) является оптимальным.

Перестановки направлены на уменьшение значения $\Delta_\Sigma(\sigma)$ на приборах с бóльшим числом запаздывающих заданий за счет резервов $R_\Sigma(\sigma)$ на приборах с меньшим числом и на построение равномерных расписаний.

Если в процессе решения произвольной индивидуальной задачи выполняются условия, сформулированные в теоремах 8 и 9, то задача решается полиномиальной составляющей ПДС-алгоритма.

Полиномиальная составляющая всех трех разработанных ПДС-алгоритмов определяется только на основе направленных перестановок.

Структура алгоритмов

Разработанные ПДС-алгоритмы решения задач МСЗ, МВМ и МСЗП условно можно представить в виде следующих основных блоков:

а) упорядочения;

б) построения начального расписания;

в) оптимизации.

В блоке упорядочения выполняется упорядочение заданий в соответствии с их приоритетами (последовательность $\sigma^{\text{уп}}$). Для заданий с весами и директивными сроками приоритет задания для всех трех задач в общем случае определяется отношением ω_j / l_j. Чем больше вес задания и меньше его длительность, тем выше приоритет этого задания.

В задаче МСЗ все веса равны, поэтому упорядочение осуществляется по неубыванию значений длительности выполнения заданий (последовательности $\sigma^{\text{уп}}$).

В задаче МВМ все задания упорядочиваются по невозрастанию отношения ω_j / l_j, и если полученная последовательность допустима по выполнению, то она оптимальна.

В задаче МСЗП, аналогично задаче МСЗ, все задания упорядочиваются по неубыванию значений их длительностей.

В блоке б) по определенным правилам на основе последовательности $\sigma^{\text{уп}}$ строятся начальные расписания, для которых затем выполняются процедуры оптимизации.

В задаче МСЗ в последовательности $\sigma^{\text{уп}}$ выполняются свободные перестановки, т.е. перестановки незапаздывающих заданий с резервами на более поздние позиции таким образом, чтобы эти задания оставались незапаздывающими. Это выполняется только в том случае, если на интервале перестановки есть запаздывающие задания (последовательность $\sigma^{\text{сп}}$).

В задаче МВМ строится допустимая по выполнению последовательность заданий.

В задаче МСЗП предварительно распределяются задания по приборам, следуя правилам, определенным в теореме 7.

Для выполнения блока оптимизации в задаче МСЗП разработана окрестность поиска оптимального решения, полученная с помощью различных типов перестановок. Поиск оптимального решения по этой окрестности осуществляется посредством направленных перестановок в соответствии с условиями теорем 8, 9. Величина $\min\{R_\Sigma(\sigma), \Delta_\Sigma(\sigma)\}$ является оценкой полученных решений и показывает, насколько можно теоретически уменьшить значение функционала, чтобы получить оптимальное решение.

Для выполнения блоков оптимизации исследуемых задач МСЗ и МВМ разработаны адаптивные алгоритмы поиска оптимального решения, суть которых состоит в том, что на каждой k-й итерации анализируется решение $k-1$-й итерации, на основании чего выбирается дальнейшая стратегия поиска.

В задаче МСЗ число итераций алгоритма определяется количеством конкурирующих заданий.

Определение 5. 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

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

На каждой итерации проверяется возможность использования резервов времени предшествующих незапаздывающих заданий очередным конкурирующим заданием и строится оптимальное расписание рассматриваемой последовательности. На первой итерации строится оптимальное расписание для заданий на интервале $\overline{1, g_1}$, где $j_{[g_1]}$ — первое запаздывающее конкурирующее задание в последовательности $\sigma^{\text{сп}}$. На следующей итерации рассматривается последовательность заданий на интервале $\overline{1, g_2}$, где $j_{[g_2]}$ — следующее запаздывающее конкурирующее задание.

Пусть уже выполнена $k-1$ итерация и построено оптимальное расписание для последовательности заданий на интервале $\overline{1, k-1}$. Переходим к очередному запаздывающему конкурирующему заданию $j_{[k]}$ и строим оптимальное расписание для подпоследовательности σ^k, включающей задания на интервале $\overline{1, k}$. Аналогичные процедуры выполняются до тех пор, пока не будет рассмотрена вся последовательность $\sigma^{\text{сп}}$ и построено оптимальное расписание на всем множестве заданий.

Алгоритм решения задачи МВМ также является итерационным. Он основывается на анализе приоритетов и использовании перестановочного приема для следующих непосредственно друг за другом подпоследовательностей заданий. Данный алгоритм на каждой k-й итерации строит P-упорядоченное расписание следующим образом: к оптимальному расписанию, полученному на $k-1$-й итерации, добавляется очередное запаздывающее задание из начальной допустимой последовательности и проводится последующая оптимизация. В конце последней итерации получаем оптимальное решение для всего множества заданий.

Алгоритмы решения задач МСЗ и МВМ построены таким образом, что на каждой итерации улучшается (или остается равным предыдущему) значение оптимизируемого функционала. Последнее обстоятельство позволяет остановить процесс решения при достижении граничного времени и оценить отклонение найденного решения от искомого оптимального, так как для решенного класса задач получены оценки, величины которых зависят от текущего решения.

Отметим еще одно свойство разработанных алгоритмов для задач МСЗ и МВМ. Структура алгоритмов позволяет «загрублять» предлагаемые решения, т.е. выигрывать во времени за счет потери точности получаемых результатов, исключая из алгоритмов те процедуры, которые статистически лишь незначительно влияют на значение функционала. Это свойство позволяет строить эффективные приближенные решения.

Декомпозиция

В задаче МСЗ [4] на этапе оптимизации последовательность $\sigma^{\text{сп}}$ декомпозируется на подпоследовательности меньшего размера. Оптимизация осуществляется на подпоследовательности, ограниченной позицией встраивания очередного конкурирующего задания и позиции, 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.

В задаче МВМ построение P-упорядоченного расписания заключается в том, что на каждой итерации k на основе рассматриваемого задания $j_{[k]}$ по определенным правилам строится структура взаимосвязанных заданий (цепочка, простая конструкция, вложенная конструкция), которая встраивается в соответствии с приоритетом в допустимой последовательности на более ранние позиции, не нарушая допустимости всей последовательности. Аналогично строится P-упорядоченное расписание для всего множества заданий, которое в соответствии с определением представлено упорядоченной подпоследовательностью подмножеств максимальных приоритетов.

Разбиение исходного множества заданий на подмножества максимальных приоритетов является декомпозицией исходной задачи на подзадачи меньшей размерности и выполняется по полиномиальной вычислительной схеме [5].

В задаче МСЗП декомпозиция осуществляется на всем множестве заданий при построении начального расписания на подмножестве незапаздывающих и запаздывающих заданий с учетом условий, сформулированных в теореме 7. В соответствии с теоремой 8 выполняются перестановки для уменьшения $\Delta_{\Sigma}(\sigma)$ на приборах с бóльшим числом запаздывающих заданий за счет резервов $R_{\Sigma}(\sigma)$ на приборах с меньшим числом запаздывающих заданий и для построения равномерных расписаний [6]. Причем в перестановках участвуют только задания, принадлежащие множеству незапаздывающих заданий.

КЛАССИФИКАЦИЯ ПДС-АЛГОРИТМОВ

Разработанные ПДС-алгоритмы реализуются по-разному. Существуют два типа ПДС-алгоритмов.

ПДС-алгоритмы первого типа (задачи МВМ и МСЗП) содержат полиномиальную составляющую и приближенный алгоритм и строятся только на направленных перестановках. В результате решения задачи получаем либо строго оптимальное решение полиномиальной составляющей алгоритма, либо приближенное с верхней оценкой отклонения от оптимального (задача МВМ). В задаче МСЗП задается полиномиальное ограничение на число вычислений от размерности задачи, и характеристика $\min\{R_{\Sigma}(\sigma), \Delta_{\Sigma}(\sigma)\}$ является оценкой отклонения от оптимального значения функционала. При этом всегда известно, является ли полученное решение точным или приближенным.

ПДС-алгоритм второго типа (задача МСЗ) имеет полиномиальную и экспоненциальную составляющие. Если в процессе решения произвольной индивидуальной задачи выполняются логико-аналитические условия, полученные на основании исследования ее свойств, то данная задача решается полиномиальным подалгоритмом. Иначе, выполняется экспоненциальное число вычислений. В процессе решения реализуются следующие отсечения, позволяющие исключить бесперспективные варианты перестановок и существенно сократить вычисления экспоненциальной составляющей алгоритма для получения оптимального решения.

Теорема 10. Необходимые условия для встраивания конкурирующего задания $j_{[g]}$ в последовательности 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 $d_{j_{[i]}} > d_{j_{[g]}}$, $j_{[i]} \in I_{j_{[g]}}$;

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

$$d_{j_{[i]}} > d_{j_{[g]}},\ d_{j_{[i]}} > C_{j_{[i]}},\ d_{j_{[i]}} > C_{j_{[p-1]}}. \tag{6}$$

Теорема 11. Если на интервале встраивания задания $j_{[g]}$ находится задание $j_{[k]}$, $d_{j_{[k]}} \le d_{j_{[g]}}$, $l_{j_{[k]}} \le l_{j_{[g]}}$, то интервал $j_{[g]}$ определяется позициями $\overline{k+1, g-1}$.

Теорема 12. Если на позиции *p* встраивания задания $j_{[g]}$ находится запаздывающее задание или задание с нулевым резервом и на позициях $\overline{1, p-1}$ нет заданий, отвечающих условиям (6), то задание $j_{[g]}$ встраивается на позицию $p+1$. При выполнении аналогичных условий на позиции $p+1$ — на позицию $p+2$ и т.д. до позиции g.

Теорема 13. Если при выполнении итерации оптимизации для конкурирующего задания $j_{[g]}$ для всех предшествующих незапаздывающих заданий $j_{[i]}$ выполняется $d_{j_{[i]}} \le d_{j_{[g]}}$ или на интервале $\overline{1, g-1}$ резервы отсутствуют, то задание $j_{[g]}$ остается на занимаемой позиции и подпоследовательность на интервале $\overline{1, g}$ оптимальна.

Теорема 14. Если в последовательности σ^k при встраивании запаздывающего задания $j_{[g]}$ на интервале $\overline{1, g-1}$ нет предшествующих заданий, для которых $d_{j_{[i]}} > d_{j_{[g]}}$ и $d_{j_{[i]}} > C_{j_{[i]}}$, но есть задания $j^{*}_{[m]}$ ($j^{**}_{[m]}$), для которых $l_{j^{*}_{[m]}}$ ($l_{j^{**}_{[m]}}$) $> l_{j[g]}$, то оптимизация осуществляется за счет резервов, освобожденных заданиями $j^*_{[m]}$ ($j^{**}_{[m]}$) при встраивании их после задания $j_{[g]}$

$$C_{j_{[g]}} - d_{j^*_{[m]}} < \sum_{i=m}^{g} \max\left(0, C_{j_{[i]}} - d_{j_{[i]}}\right). \tag{7}$$

Теорема 15. Если конкурирующее задание $j_{[g]}$ в результате выполнения для него итерации оптимизации, определяющей возможность использования резервов этим заданием, заняло позицию $k > p$, то следующие за ним конкурирующие задания $j_{[l]}$, для которых $d_{j_{[l]}} \ge d_{j_{[g]}}$, в оптимальном расписании не смогут занять позицию меньшую, чем $k+1$.

Теорема 16. Если конкурирующее задание $j_{[g]}$ в результате выполнения для него итерации оптимизации осталось на исходной позиции и все задания на позициях $\overline{g+1, n}$ запаздывающие, то задания, для которых $d_{j_{[l]}} \ge d_{j_{[g]}}$, $i \in \overline{g+1, n}$, исключаются из числа конкурирующих.

Теорема 17 [4]**.** Запаздывающее задание $j_{[g]}$ на k-й итерации может занять более раннюю позицию, что приведет к улучшению целевой функции, только в том случае, если хотя бы у одного задания $j_{[k]}$ на интервале $\overline{1, g-1}$ есть резерв и выполняется $d_{j_{[k]}} > d_{j_{[g]}}$, либо (в случае отсутствия резервов) на интервале $\overline{1, g-1}$ есть задания с метками.

Теорема 18 [4]. Пусть в последовательности σ^k для запаздывающих заданий $j_{[k]}, j_{[r]} \in J$ выполняется $l_{j_{[k]}} \le l_{j_{[r]}}$, $d_{j_{[k]}} \le d_{j_{[r]}}$. Тогда в оптимальном расписании задание $j_{[k]}$ будет предшествовать заданию $j_{[r]}$.

Теорема 19 [4]**.** Неконкурирующие задания в последовательности σ^k не могут занять более ранние позиции, чем позиции, занимаемые ими в $\sigma^{\text{сп}}$.

Теорема 20. Пусть уже построена оптимальная подпоследовательность на интервале $\overline{1, g-1}$. Запаздывающее конкурирующее задание $j_{[g]}$ на k-й итерации, выполняемой для определения возможности использования резервов этим заданием, не может быть перемещено на более раннюю позицию, если на интервале $\overline{1, g-1}$ ни у одного из заданий нет резервов и для всех помеченных заданий выполняется

$$l_{j^*_{[i]}} \le l_{j_{[g]}}, \quad i = \overline{1, g-1}.$$

Теорема 21 [4]**.** Если в последовательности σ^k конкурирующее задание $j_{[g]}$ в результате выполнения для него итерации оптимизации не использовало существующие резервы, минимальное значение функционала соответствует позиции g, занимаемой этим заданием, и если $\forall\, j_{[r]}$, $r = \overline{g+1, n}$, $d_{j_{[r]}} \ge d_{j_{[g]}}$ и $C_{j_{[r]}} \ge d_{j_{[r]}}$, то задания $j_{[r]}$ исключаются из множества конкурирующих, а последовательность σ^k оптимальна.

Теорема 22. Пусть на итерации k, выполняющейся для очередного конкурирующего запаздывающего задания $j_{[g]}$, для всех $j_{[l]}$, $l=\overline{1, g-1}$, справедливо $d_{j_{[l]}} \le C_{j_{[l]}}$, для всех $j^*_{[l]}$ выполняется $l_{j^*_{[l]}} \le l_{j_{[g]}}$ и на интервале $\overline{g, n}$ резервы отсутствуют. Тогда запаздывающие задания, занимающие позиции $\overline{g, n}$, исключаются из множества конкурирующих, а последовательности σ^k отвечает оптимальное значение функционала.

Теорема 23. Если на интервале встраивания конкурирующего задания $j_{[g]}$ есть задание $j_{[r]}$, для которого $d_{j_{[r]}} < d_{j_{[g]}}$, $l_{j_{[r]}} < l_{j_{[g]}}$, и $j_{[r]}$ опаздывает, то осуществляется декомпозиция рассматриваемой подпоследовательности, и при выполнении итерации оптимизации для $j_{[g]}$ задания на интервале $\overline{1, r}$ не рассматриваются, а оптимизация осуществляется на интервале $\overline{r+1, g-1}$.

Теорема 24. Пусть в последовательности σ^k задания $j_{[g]}$ и $j_{[k]}$ конкурирующие ($g<k$) и выполняется $d_{j_{[k]}} < d_{j_{[g]}} < C_{j^{\text{н}}_{[p]}} - l_{j^{\text{н}}_{[p]}}$ ($p^{\text{н}}$ — позиция встраивания задания $j_{[g]}$), а на интервале $\overline{1, p^{\text{н}}}$ резервы отсутствуют. Если в результате выполнения итерации оптимизации задание $j_{[g]}$ заняло позицию $s > p^{\text{н}}$, то задание $j_{[k]}$ в оптимальном расписании не сможет занять позицию меньшую, чем $s+1$.

При невыполнении условия теоремы 10 существующие резервы в последовательности σ^k недостаточны для перемещения задания $j_{[g]}$ на более ранние позиции. Процедура встраивания с последующей оптимизацией для этого задания не выполняется, что существенно сокращает число выполняемых перестановок. В теоремах 11–13, 15, 18, 23, 24 сформулированы условия, позволяющие уменьшать интервал встраивания очередного запаздывающего задания, и так как процедура оптимизации осуществляется на этом интервале, соответственно сокращается число перестановок.

Условия теоремы 14 отсекают выполнение процедур встраивания запаздывающего задания на более ранние позиции, а оптимизация осуществляется за счет резервов, освобожденных заданиями, ранее их использовавшими, и в выражении (7) определяется возможность выполнения таких перестановок. При выполнении условий теорем 16, 19, 21, 22 часть конкурирующих заданий исключается, так как для этих заданий не выполняются итерации оптимизации, что сокращает объем вычислений. В теоремах 17 и 20 сформулированы условия, определяющие возможность уменьшения значения функционала при перемещении запаздывающего задания на более ранние позиции в текущей последовательности.

ПЕРСПЕКТИВЫ РАЗВИТИЯ ПДС-АЛГОРИТМОВ

Свойства ПДС-алгоритма решения задачи МСЗ могут быть использованы при построении ПДС-алгоритмов для решения задач «Минимизация суммарного взвешенного запаздывания при выполнении независимых заданий на одном приборе» (МСВЗ) и «Минимизация суммарного опережения и запаздывания относительно директивных сроков при выполнении независимых задач одним прибором» (О/З).

Для задачи МСВЗ, как и для задачи МСЗ, приоритет определяется отношением ω_j / l_j. Поэтому в ПДС-алгоритме будут использоваться те же типы и правила перестановок с учетом различных весов заданий, что и для ПДС-алгоритма задачи МСЗ. Это позволит учитывать дополнительные отсечения, что существенно сократит трудоемкость алгоритма. ПДС-алгоритм задачи МСЗ войдет в состав ПДС-алгоритма решения задачи МСВЗ как его наиболее трудоемкий частный случай.

Алгоритм решения задачи О/З базируется на следующих теоремах.

Обозначим $r_j = d_j - C_j$ резерв задания j. Очевидно, $r_j = E_j$, где E_j — опережение задания j относительно его директивного срока. Следовательно, уменьшение опоздания за счет резервов незапаздывающих заданий приводит к уменьшению функционала О/З.

Теорема 25 [15]. Последовательность, отсортированная по невозрастанию значений длительности выполнения, не имеющая опережающих работ, является оптимальной по критерию О/З.

Теорема 26 [15]. Последовательность, отсортированная по неубыванию значений длительности выполнения, не имеющая запаздывающих работ, является оптимальной по критерию О/З.

Обозначим $r_{\min} = \min\{r_j\}$; N_r — число заданий с резервами ($r_j > 0$); $N_з$ — число запаздывающих заданий ($r_j \le 0$).

Теорема 27 [16] (использование резервов опережающих заданий). Если в последовательности σ $N_r > N_з$, то при увеличении начала выполнения заданий на величину $r_{\min}$ значение функционала О/З уменьшается также на $r_{\min}$.

На каждой итерации при увеличении моментов начала выполнения заданий в текущей последовательности на $r_{\min}$ при условии $N_r > N_з$ значение функционала также уменьшается на $r_{\min}$. При $N_r \le N_з$ получена последовательность σ_R.

Теорема 28 [16] (признак оптимальности последовательности σ_R). Если в последовательности σ_R резервы отсутствуют, то она оптимальна по критерию О/З.

Приведенные результаты будут использованы при создании ПДС-алгоритма решения задачи О/З для определения направленности перестановок, а также при определении условия выполнения полиномиальной составляющей алгоритма.

СТАТИСТИЧЕСКИЕ ИССЛЕДОВАНИЯ АЛГОРИТМОВ И ПУТИ ПОВЫШЕНИЯ ИХ ЭФФЕКТИВНОСТИ

Предварительные исследования МСЗ на задачах с размерностью до 1000 заданий показали, что данный алгоритм не имеет аналогов среди существующих алгоритмов и может применяться для задач размерности значительно большей, чем 500 заданий. Алгоритм не только конструирует оптимальные расписания для больших задач, но и делает это за приемлемое время.

Для МВМ проведено 7700 испытаний с размерностью до 300 заданий (решались также индивидуальные задачи с размерностью до 1000 заданий). В 70% случаев результирующее расписание теоретически оптимально. Анализ результатов статистического моделирования показал, что полиномиальная составляющая ПДС–алгоритма задачи МВМ является статистически значимой для моделируемых произвольно индивидуальных задач.

Для МСЗП проведены испытания на задачах с размерностью до 3000 заданий с числом приборов до 30. В 87% случаев результирующее расписание теоретически оптимально.

Для МВМ и МСЗП дальнейшие исследования проводятся с целью обоснования новых логико-аналитических условий для нахождения точного решения задачи полиномиальной составляющей алгоритма.

Так, для МВМ в работе [5] предложен алгоритм построения оптимального расписания для индивидуальных задач МВМ, отношения предшествования которых имеют вид цепочек, простых или сложных конструкций с произвольной степенью вложения. Рассматриваемая индивидуальная задача МВМ может быть решена за полиномиальное время, когда реализуются условия выполнения полиномиальной составляющей ПДС-алгоритма, т.е. в результате решения исходной индивидуальной задачи алгоритм строит только структуры, имеющие вид цепочек либо конструкций произвольного вида.

В работе [7, 14] приведены новые условия полиномиальной составляющей ПДС-алгоритма задачи МВМ. Доказано, что оптимальное расписание может быть получено за полиномиальное время для множеств максимальных приоритетов, не являющихся в общем случае совокупностью конструкций.

В работе [8] рассмотрены новые условия полиномиальной составляющей, реализация которых дает точное решение за полиномиальное время. Выведены дополнительные условия, определенные для задачи, все множество заданий которой является одним множеством максимального приоритета. Предложена модификация полиномиальной составляющей ПДС-алгоритма, реализующего эти условия.

Для задачи МСЗ дальнейшие исследования будут проводиться по двум направлениям, дополняющим друг друга.

1. Выделение случаев, когда с учетом декомпозиции реализуется трудоемкий перебор различных вариантов использования резервов предшествующих заданий конкурирующими заданиями и доказательство эффективности экспоненциальной составляющей алгоритма на основе экспериментальных исследований.

2. Выделение случаев, когда с учетом декомпозиции реализуется трудоемкий перебор вариантов решения задачи и теоретическое обоснование расширения области определения полиномиальной составляющей алгоритма в тех случаях, когда экспоненциальная составляющая приводит к незначительному перебору вариантов.

Распараллеливание вычислений

Общим универсальным принципом повышения эффективности ПДС-алгоритмов является принцип распараллеливания вычислений, позволяющий:

- реализовать выполнение независимых перестановок различных типов на параллельных процессорах (МСЗП);
- останавливать вычислительный процесс при выполнении одного из условий полиномиальной составляющей алгоритма (МСЗ, МСЗП);
- управлять дальнейшим процессом вычислений на основе текущего анализа результатов параллельных вычислений (МСЗ, МВМ, МСЗП);
- гарантированно уменьшать общее время вычислений (МСЗ, МВМ, МСЗП).

ЗАКЛЮЧЕНИЕ

Анализируются свойства ПДС-алгоритмов, построенных для отдельных классов одноэтапных задач теории расписаний, а также общие свойства и принципиальные отличия алгоритмов. Обосновываются пути их модификации с целью повышения эффективности. Рассматривается возможность построения новых ПДС-алгоритмов с использованием теоретических свойств известных.

ЛИТЕРАТУРА

1. *Гери М.Р., Джонсон Д.С.* Вычислительные машины и труднорешаемые задачи.— М.: Мир, 1982. — 416 с.
2. *Згуровський М.З., Панкратова Н.Д.* Основи системного аналізу. — Київ: Видавнича група BHV, 2007. — 544 с.
3. *Павлов О.А., Павлова Л.О.* ПДС-алгоритми для важкорозв'язуємих комбінаторних задач. Теорія і методологія розробки. — Ужгород: Поличка «Карпатського краю», 1998. — 320 с. (англ. мовою).
4. *Павлов А.А., Мисюра Е.Б.* Эффективный точный ПДС-алгоритм решения задачи о суммарном запаздывании для одного прибора // Системні дослідження та інформаційні технології. — 2004. — № 4. — С.30–59.
5. *Конструктивные* полиномиальные алгоритмы решения индивидуальных задач из класса NP / А.А. Павлов и др. — Киев: Техніка. — 1993. — 128 с.
6. *Павлов А.А., Теленик С.Ф.* Информационные технологии и алгоритмизация в управлении. — Киев: Техніка, 2002. — 344 с.
7. *Павлов О.А., Аксьонова Л.О.* «Мінімізація сумарного зваженого моменту закінчення робіт» як перший рівень моделі дрібносерійного виробництва та способи її розв'язання // Системні дослідження та інформаційні технології. — 2002. — № 1. — С.119–130.

8. *Аксенова Л.А.* Новые полиномиальные подклассы труднорешаемой задачи «Минимизация суммарного взвешенного момента» для множества одного приоритета // Управляющие системы и машины. — 2002. — № 6. — С. 21–28.
9. *Du J. and Leung J.Y.-T.* Minimizing total tardiness on one processor is NP-hard // Math. Oper. Res. — 1990. — № 15. — P. 483–495.
10. *Shwarz W. and Mukhopadhyay S.K.* Decomposition of the single machine total tardiness problem // Operations Research Letters. — 1996. — № 19. — P. 243–250.
11. *Lawler E.L.* Sequencing jobs to minimize total weighted completion time subject to precedence constraints // Ann. Discrete Math. — 1978. — № 2. — P. 75–90.
12. *Конвей Р.В., Максвелл У.Л.* Теория расписаний. — М.: Наука, 1975. — 359 с.
13. *Танаев В.С., Шкурба В.В.* Введение в теорию расписаний. — М.: Наука, 1975. — 256 с.
14. *Павлов А.А., Аксенова Л.А.* Новые условия полиномиальной составляющей ПДС-алгоритма задачи «Минимизация суммарного взвешенного момента» // Проблемы программирования. — 2001. — № 1–2. — С. 69–75.
15. *Ow P.S., Morton T.E.* The Single Machine Early: Tardy Problem // Management Science. — 1989. — **35**, № 2. — P. 177–191.
16. *Павлов О.А., Місюра О.Б., Мельников О.В.* Дослідження властивостей та розв'язання задачі «Мінімізація сумарного штрафу як за випередження, так і за запізнення відносно директивних строків при виконанні незалежних завдань одним приладом» // Вісн. НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. — 2008. — № 48. — 7 с.

Поступила 29.04.2009