

УДК 681.3.681.5

Модифицированный алгоритм Романовского быстрого нахождения приближённого решения однородной распределительной задачи

Р. А. Нейдорф, А. А. Жикулин

(Донской государственный технический университет)

Разработана модификация алгоритма Романовского приближённого решения однородной распределительной задачи, обладающая низкими требованиями к ресурсам по сравнению с исходным алгоритмом. Описаны внесённые в оригинальный алгоритм изменения, которые позволяют обеспечить более эффективную процедуру формирования загрузки исполнителей заданиями. Приведены результаты вычислительных экспериментов.

Ключевые слова: теория расписаний, однородная задача, минимаксный критерий, метод ветвей и границ, дерево вариантов.

Введение. Исследования в области классической теории расписаний (КТР) сохраняют актуальность, несмотря на большое количество научно-исследовательских работ, посвящённых этой теме [1, 2]. Для КТР характерны два типа задач, существенно отличающиеся постановкой задачи и методами решения: однородные (ОРЗ) и неоднородные (НРЗ) распределительные задачи (РЗ). Несмотря на более упрощённую форму, ОРЗ принадлежат к классу NP-полных задач с показательным ростом сложности при увеличении размерности. Одной из наиболее популярных и перспективных тем исследований является разработка эффективных приближённых алгоритмов, обладающих высокой скоростью решения РЗ.

Постановка задачи. В связи со сформулированной во введении проблемой в данной работе ставится задача создания приближённого алгоритма решения ОРЗ, который бы обладал достаточно хорошими точностными свойствами при низких требованиях к ресурсам. Общая математическая модель постановки и решения ОРЗ, предложенная в работе [3], выглядит следующим образом. Рассматривается исполнительная система (ИС), состоящая из m идентичных, параллельно работающих исполнителей $E = \{e_1, \dots, e_m\}$. На вход ИС поступает множество n независимых заданий (работ) $W = \{w_1, \dots, w_n\}$, которые необходимо распределить между исполнителями. Известен ресурс выполнения каждого i -го задания r_i , и он одинаков для любого j -го исполнителя e_j . Таким образом, множеству W сопоставлено n -множество ресурсов $R = \{r_1, \dots, r_n\}$. Решением ОРЗ является множество $D^w = \{W_1, \dots, W_m\}$, в котором подмножества заданий $W_j = \{w_k \mid w_k \in W\}$ отвечают обязательному свойству:

$$\forall j, k \in [1, m] \rightarrow \bigcup_{j=1}^m W_j = W; \quad W_j \cap W_k = \emptyset.$$

Запланированная вариантом D^w загрузка заданиями каждого исполнителя e_j оценивается ресурсом R_j , где $R_j = \sum r_k; r_k : w_k \in W_j$. В результате решению ОРЗ в виде конкретного варианта D^w сопоставляется оценочное множество $D^r = \{R_1, \dots, R_m\}$. Для оценки решения РЗ формируется функция $Q^m[D^r]$, отражающая требования к свойствам этого решения. Например, максимальная по ресурсу загрузка одного из исполнителей

$$Q^m[D^r] = \max_{\bar{R}} \{R_j \mid j = \overline{1, m}\} \quad (1)$$

представляет собой оценку ресурсоёмкости решения и считается наиболее эффективной среди функционалов, а если в качестве ресурса выступает время, то выражение (1) является оценкой производительности ИС.

При любом критерии наилучшим способом конкретизации и обеспечения эффективности решения является оптимизационный подход. Например, от оценки (1) целесообразно задаться условиями минимальности значения в окончательном её решении. Это порождает минимаксный критерий (ММК) оптимизации:

$$Q^{mm}[D^r] = \min_{E, W} \max_R \{R_j \mid j = \overline{1, m}\}.$$

Современные методы решения ОРЗ. Одним из подходов к решению поставленной задачи является разработка нового алгоритма решения РЗ на основе уже существующего метода путём его модификации с целью повышения его эффективности. Для решения ОРЗ зарубежными и российскими учёными разработано множество алгоритмов — как точных, так и приближённых. Наиболее известным и широко распространённым точным методом решения является алгоритм Романовского (АР), описанный и исследованный во многих источниках [4], [5]. Методологически АР принадлежит к классу методов «ветвей и границ», которые значительно снижают ресурс решения РЗ по сравнению с алгоритмом прямого перебора. Однако в худшем случае алгоритм осуществляет проверку всех возможных вариантов решений, что приводит к большим объёмам вычислений для нахождения оптимума. Таким образом, даже для задач невысокой размерности получение оптимального решения за приемлемое время становится невозможным. Несмотря на этот недостаток, АР обладает высокой скоростью сходимости к области решений, близких к абсолютному оптимуму, по сравнению с другими точными алгоритмами. Это обусловлено грамотным применением в АР механизма отсека неперспективных ветвей поиска. Поэтому, по мнению авторов, в качестве базового метода для создания эффективного приближённого алгоритма решения ОРЗ наиболее целесообразно использовать АР.

Алгоритм Романовского точного решения ОРЗ. АР заключается в последовательных попытках получения распределения заданий по исполнителям так, что загрузка каждого исполнителя не превышала максимально допустимой загрузки исполнителя z . Для выделения последующих z -задач используется итеративная формула

$$z_k = F_z(fa_{k-1}, fb_{k-1}),$$

где fa — нижняя граница (оценка снизу) поиска, fb — верхняя граница (оценка сверху) поиска, F_z — функция изменения z , причём $z_k \in (fa_{k-1}, fb_{k-1})$, k — шаг итерации. Если на k -м шаге итерации z -задача имеет решение, то значение верхней границы изменяется, т. е. $fb_k = Q^m$, иначе изменяется значение нижней границы, т. е. $fa_k = z_k$.

В результате последовательного рассмотрения z -задач происходит уменьшение интервала поиска оптимального распределения, которое завершается при выполнении условия

$$fb_k = fa_k + 1.$$

Для решения z -задач применяется метод ветвей и границ с односторонним обходом дерева вариантов. Если представить, что исполнители могут обеспечить ресурсами в количестве, равном z , то у ИС должно оставаться свободное количество ресурсов

$$\text{slack} = zm - \sum_{i=1}^m R_i,$$

где z — количество ресурса у исполнителя, m — количество исполнителей, R_i — частичная загрузка исполнителей на определённом шаге алгоритма.

Задания назначаются исполнителям последовательно из списка заданий, в котором они упорядочены по уменьшению ресурса выполнения, таким образом, чтобы каждый раз назначалось задание с самым большим ресурсом выполнения. Назначение заданий продолжается при соблюдении условия

$$\text{slack} > 0. \quad (2)$$

Иначе необходимо произвести отмену назначения текущего задания и попытаться назначить другое. В худшем случае алгоритму требуется перебрать все варианты для получения решения.

Модифицированный АР (МАР) быстрого нахождения приближённого решения ОРЗ.

Основным недостатком АР является высокая требовательность к ресурсам при определённых условиях ОРЗ, поскольку в худшем случае он осуществляет перебор всех возможных вариантов решений, что ведёт к большим объёмам вычислений. Это присуще всем алгоритмам, построенным по схеме методов «ветвей и границ». Таким образом, для увеличения быстродействия АР наиболее целесообразно изменить алгоритм решения z-задачи, а итеративный принцип выделения z-задачи оставить без изменения. Более того, с целью получения наибольшей производительности разрабатываемого алгоритма решение z-задачи необходимо осуществлять по оригинальной стратегии АР до тех пор, пока не будет осуществлена отмена назначения текущего задания. Это обусловлено тем, что АР довольно быстро сходится к области решений, близких к абсолютному оптимуму.

Поскольку для решения z-задачи АР осуществляет односторонний обход дерева всех возможных решений, то при неблагоприятных условиях ОРЗ будет осуществлён перебор большого количества вариантов. Поэтому необходимо изменить способ перебора распределений заданий по исполнителям с целью минимизации вариантов, рассматриваемых алгоритмом при поиске решения. Таким образом, авторами предлагается внести следующие изменения в оригинальный алгоритм решения z-задачи.

Во-первых, необходимо назначать задания каждому j -му исполнителю так, чтобы его загрузка R_j была максимальна для значения z . Таким образом, необходимо формировать набор заданий $W_j = \{w_k \mid w_k \in \overline{W}\}$, где $\overline{W}$ — множество ещё не назначенных исполнителям заданий с максимальным значением загрузки среди всех возможных комбинаций заданий из множества $\overline{W}$. Тогда как в АР исполнителю назначается первая рассмотренная комбинация заданий, удовлетворяющих условию (2). В отличие от оригинального алгоритма, в разрабатываемом алгоритме назначается такой набор заданий, который минимизирует дефицит загрузки исполнителя. По мнению авторов, этот эвристический подход позволит значительно быстрее достигать решения, удовлетворяющего z .

Во-вторых, если для исполнителя e_j не существует набора заданий $W_j = \{w_k \mid w_k \in \overline{W}\}$, который удовлетворяет условию (2), то алгоритм осуществляет процедуру отката не к предыдущему исполнителю e_{j-1} , как в АР, а к первому исполнителю e_1 . Эта стратегия позволит значительно сократить перебор вариантов решений за счёт максимального отсечения нижних ветвей дерева поиска. В то же время данный подход рассматривает широкий диапазон разнообразных решений, поскольку осуществляется проход по всем верхним ветвям дерева вариантов. Однако это не позволит перебрать все возможные решения, что в худшем случае может привести к пропуску искомого оптимума. Поэтому эта модификация АР является приближённым методом решения ОРЗ в отличие от исходного алгоритма.

Добавим вышеописанные изменения в оригинальный АР — и алгоритм решения z-задачи примет следующий вид.

Ш.1 Решаем z-задачу по оригинальной стратегии АР. Если на этапе выполнения алгоритма необходимо произвести отмену назначения текущего задания, т. е. не выполняется усло-

вие (2), то осуществляем быстрый поиск решения z-задачи по модифицированной стратегии, т. е. переходим на Ш.2. Иначе решение z-задачи найдено, переходим на Ш.11.

Ш.2 $\overline{W} = W$, $j = 1$.

Ш.3 Пока $j \leq m$, иначе — на Ш.11.

Ш.4 $W_{best} = \emptyset$, $R_{best} = 0$, где W_{best} — набор заданий, имеющих max значение загрузки $\forall W_j = \{w_k \mid w_k \in \overline{W}\}$.

Ш.5 Осуществляем перебор всех возможных $W_j = \{w_k \mid w_k \in \overline{W}\}$.

Ш.6 Если $R_j = z$, где R_j — загрузка W_j , то $W_{best} = W_j$, $R_{best} = R_j$ — и переходим на Ш.8.

Ш.7 Иначе если $slack > 0$ и $R_j > R_{best}$, то $W_{best} = W_j$, $R_{best} = R_j$ — и переходим на Ш.5.

Ш.8 Если W_{best} не пустое множество, то W_{best} назначается исполнителю e_j , $\overline{W} = \overline{W} / W_j$, $j = j + 1$ — и переходим на Ш.3.

Ш.9 Иначе для исполнителя e_j не найден набор заданий, удовлетворяющих условию (2). Если $j = 1$, то просмотрены все рассматриваемые варианты решений, т. е. решение не найдено. Поэтому — выход из алгоритма.

Ш.10 В противном случае осуществляем откат к первому исполнителю e_1 , т. е. $j = 1$, $\overline{W} = W$, $W_{best} = \emptyset$, $R_{best} = 0$, $W_j = W_{next}$, где W_{next} — набор заданий, который ещё не рассматривался алгоритмом, — и переходим на Ш.6.

Ш.11 Решение найдено, формируем результат.

Сравнительный анализ эффективности алгоритмов решения ОРЗ. Для сравнительного анализа эффективности алгоритмов решения ОРЗ помимо модифицированного и исходного АР также использовались метод критического пути (МКП) и эволюционно-генетический алгоритм (ЭГА). Эти алгоритмы являются приближёнными методами решения и характеризуются высокими ресурсно-точностными показателями среди алгоритмов этого класса. С целью исследования эффективности вышеуказанных алгоритмов были проведены вычислительные эксперименты при разных значениях параметров задачи. Такими параметрами являются: n — количество заданий, m — количество исполнителей, $[z_1, z_2]$ — диапазон генерации ресурсов заданий. В качестве эволюционно-генетической модели используется модель, описанная в работе [5]. В ходе экспериментов были случайным образом сгенерированы по 100 векторов ресурсов заданий в диапазоне $[z_1, z_2]$. Для анализа точностных характеристик приближённых алгоритмов на малых размерностях ОРЗ используется процент нахождения оптимальных решений P_{opt} а ресурсных — среднее время выполнения алгоритма t_{cp} (в секундах) по 100 опытам (табл. 1).

По результатам, приведённым в таблице 1, видно, что точностные свойства МАР значительно лучше других приближённых алгоритмов. Поскольку при неблагоприятных условиях МАР находил оптимальные решения в 92 % случаев, тогда как МКП — 0 %, а ЭГА — 70 %. По ресурсным характеристикам МАР находится на уровне МКП и на порядок быстрее, чем АР и ЭГА.

Согласно исследованиям [5], для АР одним из самых неблагоприятных условий ОРЗ является случай, когда $m = 3$, $n = 17$ и диапазон ресурсов заданий $[25..30]$. Поэтому проведены дополнительные вычислительные эксперименты, результаты которых подтвердили высокие точностные характеристики МАР: МАР в 100 % случаях находил оптимум, МКП — 0 %, ЭГА — 97 %.

Таблица 1

Результаты экспериментов на малых размерностях ОРЗ

m	n	$[z_1, z_2]$	АР		МАР		МКП		ЭГА	
			$P_{\text{опт}}$	$t_{\text{ср}}$	$P_{\text{опт}}$	$t_{\text{ср}}$	$P_{\text{опт}}$	$t_{\text{ср}}$	$P_{\text{опт}}$	$t_{\text{ср}}$
2	12	[50, 55]	100	<0.001	100	<0.001	94	<0.001	100	0.171
2	12	[30, 65]	100	<0.001	100	<0.001	36	<0.001	100	0.207
2	14	[50, 55]	100	0.001	100	<0.001	92	<0.001	100	0.239
2	14	[30, 65]	100	<0.001	100	<0.001	51	<0.001	100	0.196
3	13	[40, 60]	100	0.896	99	<0.001	0	<0.001	99	0.245
4	12	[50, 55]	100	0.032	100	<0.001	85	<0.001	100	0.221
4	12	[30, 65]	100	0.249	92	<0.001	36	<0.001	89	0.264
4	14	[50, 55]	100	25.329	100	<0.001	0	<0.001	96	0.280
4	14	[30, 35]	100	3.618	99	<0.001	0	<0.001	70	0.333

Также проведены эксперименты на более высоких размерностях ОРЗ: $m \geq 3$, $n \geq 31$. Поскольку АР в поставленных экспериментах довольно часто не позволял получить решение за приемлемое время (менее 30 минут), в качестве показателя, оценивающего точностные характеристики алгоритмов, используется процент решений $P_{\text{мин}}$, имеющих минимальную оценку по опыту (табл. 2).

Таблица 2

Результаты экспериментов на больших размерностях ОРЗ

m	n	$[z_1, z_2]$	МАР		МКП		ЭГА	
			$P_{\text{мин}}$	$t_{\text{ср}}$	$P_{\text{мин}}$	$t_{\text{ср}}$	$P_{\text{мин}}$	$t_{\text{ср}}$
3	31	[50, 55]	100	0.023	0	<0.001	57	0.795
3	31	[30, 65]	100	0.001	0	<0.001	93	0.703
3	51	[50, 55]	100	0.018	89	<0.001	100	0.609
3	51	[30, 65]	100	0.002	73	<0.001	81	0.823
4	41	[40, 60]	100	0.092	0	<0.001	31	0.802
5	31	[50, 55]	100	0.010	0	<0.001	26	0.848
5	31	[30, 65]	100	0.001	0	<0.001	90	0.548
5	51	[50, 55]	100	0.087	0	<0.001	1	1.535
5	51	[30, 35]	100	0.002	0	<0.001	0	1.139

По результатам, приведённым в таблице 2, МАР не только подтвердил, но и значительно улучшил свои точностные характеристики по сравнению с другими алгоритмами. Во всех проведённых экспериментах МАР находил решение с min оценкой по опыту, тогда как в других алгоритмах этот показатель сильно изменялся: МКП — от 0 до 89 %, ЭГА — от 0 до 100 %. Однако среднее время расчёта значительно увеличилось по сравнению с ранее проведёнными экспериментами на малых размерностях ОРЗ, а именно: с 1 мс до 92 мс. Согласно этому факту, можно сделать предположение, что МАР имеет экспоненциальный рост времени выполнения относительно размерности задачи. Таким образом, необходимы дальнейшие исследования алгоритма для проверки этой гипотезы. Хочется отметить, что ни в одном опыте количество вариантов, рассматриваемых МАР, при поиске решений не достигло полного перебора, что указывает на эффективность внесённых изменений в АР.

Выводы. МАР обладает высокими ресурсно-точностными характеристиками при решении ОРЗ невысоких размерностей ($m \leq 5$, $n \leq 51$) по сравнению с другими приближёнными алгоритмами. Разработанный алгоритм позволяет в большинстве случаев получать решения с более низкой оценкой по ММК, чем МКП (в среднем в 67,9 % случаев) и ЭГА (25,3 %). К тому же время выполнения МАР во всех экспериментах не превысило 100 мс, что указывает на эффективность внесённых в АР изменений. Однако быстрый рост времени работы алгоритма относительно размерности задачи требует дальнейшего исследования и улучшения МАР.

Библиографический список

1. Коффман, Э. Г. Теория расписания и вычислительные машины / Э. Г. Коффман. — Москва: Наука, 1984. — 334 с.
2. Конвей, Р. В. Теория расписаний / Р. В. Конвей, В. Л. Максвелл, Л. В. Миллер. — Москва: Наука, 1975. — 360 с.
3. Нейдорф, Р. А. Методологические проблемы теории расписаний / Р. А. Нейдорф, В. Г. Кобак // Системный анализ, управление и обработка информации: 1-й межвуз. сб. науч. ст. / ДГТУ; ТТИ ЮФУ. — Ростов-на-Дону, 2007. — С. 101—108.
4. Романовский, И. В. Алгоритмы решения экстремальных задач / И. В. Романовский. — Москва: Наука, 1977. — 352 с.
5. Будиловский, Д. М. Оптимизация решения задач теории расписаний на основе эволюционно-генетической модели распределения заданий: дисс. ... канд. техн. наук. — Ростов-на-Дону, 2007. — 212 с.

Материал поступил в редакцию 14.06.2012.

References

1. Koffman, E. G. Teoriya raspisaniya i vy`chislitel`ny`e mashiny` / E. G. Koffman. — Moskva: Nauka, 1984. — 334 s. — In Russian.
2. Konvej, R. V. Teoriya raspisaniy / R. V. Konvej, V. L. Maksvell, L. V. Miller. — Moskva: Nauka, 1975. — 360 s. — In Russian.
3. Nejdorf, R. A. Metodologicheskie problemy` teorii raspisaniy / R. A. Nejdorf, V. G. Kobak // Sistemny`j analiz, upravlenie i obrabotka informacii: 1-j mezhvuz. sb. nauch. st. / DGTU; TTI YuFU. — Rostov-na-Donu, 2007. — S. 101—108. — In Russian.
4. Romanovskij, I. V. Algoritmy` resheniya e`kstremal`ny`x zadach / I. V. Romanovskij. — Moskva: Nauka, 1977. — 352 s. — In Russian.
5. Budilovskij, D. M. Optimizaciya resheniya zadach teorii raspisaniy na osnove e`volyucionno-geneticheskoy modeli raspredeleniya zadaniy: diss. ... kand. texn. nauk. — Rostov-na-Donu, 2007. — 212 s. — In Russian.

MODIFIED ROMANOVSKY ALGORITHM FOR QUICK FINDING OF APPROXIMATE SOLUTION TO HOMOGENEOUS ALLOCATION PROBLEM

R. A. Neydorf, A. A. Zhikulin

(Don State Technical University)

The modification of Romanovsky algorithm for the approximate solution to the homogeneous allocation problem is developed. The modification involves lower resource requirements as against the original algorithm. The changes in the original algorithm which permit to provide a more effective procedure of loading executors with tasks are described. The computational experiments are resulted.

Keywords: scheduling theory, homogeneous problem, minimax criterion, branch-and-bound method, search tree.