

Алгоритм повышения быстродействия минимаксной оптимизации решений распределительных задач в однородных системах***Р. А. Нейдорф, А. А. Жикулин**

Описывается и обосновывается алгоритм оптимизации решения однородных распределительных задач теории расписаний. Он представляет собой модификацию известного в этой предметной области алгоритма Романовского — классического варианта метода ветвей и границ с односторонним обходом дерева решений. Проведено системное исследование этого алгоритма, которое позволило выявить причины увеличения времени его работы при обходе некоторых ветвей дерева решений. Это даёт возможность предложить свободную от выявленного недостатка модификацию, названную комбинационно-модифицированным алгоритмом Романовского. Сущность данной модификации заключается в следующем. В процедуре решения распределительной задачи избирательно пропускаются те правила, этапы и шаги, которые приводят к перебору на исполнителях наборов заданий, заведомо повторяющих проверенный ранее результат. Сущность нового алгоритма поясняется примером. Приводятся результаты статистически представительных исследований. Они позволили продемонстрировать возможности алгоритма на распределительных задачах высокой размерности. (Решение таких задач классическим алгоритмом невозможно из-за ограниченных временных ресурсов.) Результаты обработки этих решений показали, что новая модификация не позволяет решить проблему NP-полноты распределительных задач, но обеспечивает ресурсно-временной выигрыш, связанный с существенным снижением показателя степени экспоненциальной модели роста среднего времени решения.

Ключевые слова: теория расписаний, распределительная задача, алгоритм Романовского, дерево решений, перебор комбинаций, размер задания, сочетания, перестановки.

Введение. Масштабы современного производства, задачи координации действий больших технических, экономических и социальных систем усложнили функции организационного управления. Оно требует жёсткого планирования работ, процессов, действий во всех сферах человеческой деятельности. Эффективность планирования определяет технико-экономические показатели многочисленных бизнес-процессов. Поэтому всё больше внимания уделяется теории расписаний, суть которой можно сформулировать как решение проблемы упорядочивания задач, решаемых последовательно параллельно во времени.

Квинтэссенцией проблемы упорядочивания является задача построения оптимального расписания процесса выполнения конечного множества заданий выделенной для этого совокупностью исполнителей. Возникает задача построения такого алгоритма распределения заданий между исполнителями, который обеспечивает оптимум критерия оценки результирующего расписания. Указанная задача относится к классу NP-полных, является сложной как для теоретического, так и для экспериментального исследования [1–7].

С этим свойством распределительной задачи (РЗ) связана необходимость использования значительного временного ресурса, что затрудняет нахождение оптимального решения даже в простейших случаях работы с однородными РЗ (ОРЗ). Наиболее известный и эффективный в настоящее время подход к решению ОРЗ демонстрирует алгоритм, предложенный И. В. Романовским [8]. Однако и он требует доработки. Если речь идёт о задачах с размерностью по количеству исполнителей более 2 и по количеству заданий более 15 с некоторыми структурными особенностями совокупности распределяемых заданий, то время решения РЗ алгоритмом Романовского (АР) может значительно превысить разумные или технологические пределы. Таким образом, актуальна задача повышения ресурсно-временной эффективности данного алгоритма за счёт модификации,

* Работа выполнена по тематическому плану Министерства образования РФ № 2.3.13 «Метод сбалансированной дискретизации для задач имитационного моделирования динамических процессов в распределённых объектах».

которая позволит повысить быстродействие, но не нарушит основного свойства нахождения абсолютного оптимума. Иными словами, модификация метода Романовского должна сохранить свойство метода ветвей и границ.

Состояние проблемы. Исследованию проблем, связанных с решением распределительных задач теории расписаний, посвящены труды многих отечественных учёных — таких, как В. С. Танаев, В. В. Шкурба, И. В. Романовский, О. Г. Алексеев, Б. А. Головкин, С. Д. Пашкеев, А. М. Ларионов, С. А. Майоров, Г. И. Новиков, В. М. Португал, А. А. Лазарев, С. В. Севастьянов, А. Н. Безгинов, Р. А. Нейдорф, В. Г. Кобак и др. Не меньшее внимание данной проблеме уделяли и зарубежные исследователи: Р. Беллман, С.-М. Джонсон, Дж.-Р. Джексон, Э.-Г. Коффман, Р.-В. Конвей, В.-Л. Максвелл, Л.-В. Миллер, М. Гэри, Д. Джонсон, П. Брукер, Дж. Келли, М. Уолкер, М. Крон, Х.-А. Таха, М.-П. Пинедо, Дж. Блазевиц и многие другие.

В последнее время интерес к чисто теоретическим РЗ несколько ослаб. Этому способствовали разные обстоятельства. Абстрактный характер РЗ, создаваемый множеством допущений и ограничений (особенно у ОРЗ), сформировал в научной среде мнение об их незначительной практической ценности. Кроме того, долгие попытки построить точные и быстрые алгоритмы решения даже ОРЗ чаще всего давали несущественные результаты — не более нескольких процентов. Поэтому основные усилия были направлены на построение приближённых алгоритмов. Однако, по мнению авторов, эти задачи являются основой любых реальных прикладных задач и достойны исследования. Результаты данной статьи показывают, что возможности усовершенствования известных методов решения РЗ не исчерпаны.

Цель и постановка задачи. Целью настоящей работы является повышение ресурсной эффективности метода точного решения ОРЗ теории расписаний при сохранении его свойства точности решения.

Для этого исследуются:

- возможности существующих алгоритмических средств точного решения ОРЗ с помощью АР;
- алгоритмические особенности АР, обуславливающие возможность сокращения количества рассматриваемых вариантов решения.

Таким образом возможно получить решение задачи разработки точного алгоритма решения ОРЗ с существенно более высокими по сравнению с АР ресурсно-временными характеристиками. Общие сведения о факте решения поставленной задачи и некоторые свойства нового алгоритма опубликованы в виде краткого описания [9].

Анализ ресурсно-временных возможностей АР при решении ОРЗ. Результаты оценки ресурсно-временных характеристик АР приведены в табл. 1.

В 1-м столбце указаны номера опытов в проведённом двухуровневом трёхфакторном эксперименте. В каждом опыте решалось по 100 ОРЗ. Во 2-м, 3-м и 4-м столбцах приведены значения варьируемых в эксперименте факторов — основных параметров ОРЗ: количество исполнителей m ; количество распределяемых заданий n ; ширина диапазона $\Delta_{разм}$ в пределах которого случайным образом равномерно распределены размеры заданий.

В блоке таблицы S_{mid} помещены результаты обработки основной совокупности оценок времени решения ОРЗ: $P_{выб}^{mid}$ — процент задач, попавших в данную совокупность; a_t^{mid} — средние выборочные a_t значения времени решений РЗ (в мс) и Δ_t^{mid} — их среднеквадратичные отклонения Δ_t . В следующем блоке S_{max} помещены такие же характеристики для совокупности максимальных временных оценок.

В предпоследнем столбце приведены значения отношения средних по совокупностям максимальных и основных оценок:

$$\lambda_{\text{mid}}^{\text{max}} = \frac{a_t^{\text{max}}}{a_t^{\text{mid}}} \quad (1)$$

В последнем столбце приведены средние выборочные a_t^{gen} значения времени решений ОРЗ по всей совокупности оценок.

Таблица 1

Анализ ресурсно-временных характеристик классического АР

№ п/п	m	n	$\Delta_{\text{разм}}$	S_{mid}			S_{max}			$\lambda_{\text{mid}}^{\text{max}}$	a_t^{gen}
				$p_{\text{выб}}^{\text{mid}}$	a_t^{mid}	Δ_t^{mid}	$p_{\text{выб}}^{\text{max}}$	a_t^{max}	Δ_t^{max}		
0	3	13	20	90	32,2	43,42	10	302,78	44,01	9,40	59,26
1	2	12	10	97	0,062	0,028	3	0,326	0,112	5,26	0,069
2	2	12	30	98	0,069	0,047	2	2,96	0,257	42,96	0,127
3	2	14	10	94	0,176	0,143	6	1,19	0,272	6,74	0,237
4	2	14	30	93	0,094	0,094	7	0,834	0,186	8,87	0,146
5	4	12	10	96	0,033	0,005	4	0,09	0,019	2,73	0,035
6	4	12	30	94	0,074	0,045	6	0,316	0,048	4,27	0,089
7	4	14	10	97	10,74	5,14	3	44,35	3,71	4,13	11,75
8	4	14	30	95	1,29	2,66	5	26,38	11,85	20,45	2,54

Анализ с помощью критерия Стьюдента множеств оценок каждого решения задачи данного опыта показал, что каждая структура РЗ характеризуется ощутимой неоднородностью (табл. 1) [10]. Причём наблюдается до 10 % выбросов в сторону больших значений времени и ни одного — в сторону меньших. Естественно предположить, что первоочередной является задача устранения выбросов, порождаемых взаимодействием свойств АР и неблагоприятной структуры ОРЗ. Это подтверждается и результатами эксперимента, который построен по плану двухуровневого многофакторного [11]. Анализ табл. 1 показывает, что применительно к классическому АР все рассматриваемые факторы оказывают приблизительно равнозначное воздействие на среднее время его работы (как по общей выборке результатов, так и по выборке кардинально больших ресурсов). Увеличение количества исполнителей и распределяемых заданий увеличивает средний ресурс, необходимый для решения РЗ исследуемым алгоритмом. Рост ширины диапазона размеров заданий увеличивает его только при $n = 12$, а при $n = 14$, наоборот, уменьшает.

Данные выводы о влиянии параметров РЗ на ресурсно-временные характеристики АР хорошо иллюстрируются представленной на рис. 1 диаграммой. Здесь отображены оценки среднего времени работы алгоритма по основной совокупности результатов a_t^{mid} и совокупности максимальных ресурсов a_t^{max} . Также отчётливо видно, что среднее время работы АР как по основной совокупности оценок, так и по совокупности максимальных оценок очень нелинейно зависит от параметров РЗ.

На долю кардинально больших ресурсов $p_{\text{выб}}^{\text{max}}$ в исследуемом диапазоне РЗ влияют главным образом количество распределяемых заданий и разброс значений их размеров (табл. 1). Количество исполнителей практически не оказывает воздействия на данный параметр. Отношение $\lambda_{\text{mid}}^{\text{max}}$ представляет собой дополнительную к среднеквадратичному отклонению оценку разброса времени решения РЗ. Достаточно хорошо видна зависимость значений $\lambda_{\text{mid}}^{\text{max}}$ от количества испол-

нителей и диапазона размеров распределяемых заданий. При этом увеличение количества исполнителей уменьшает значения данного параметра, а рост ширины диапазона размеров заданий, наоборот, увеличивает его. Влияние количества распределяемых заданий на отношение λ_{mid}^{max} оценить сложно, т. к. в некоторых случаях увеличение значений этого фактора приводит к росту его величины, а в других, наоборот, к уменьшению.

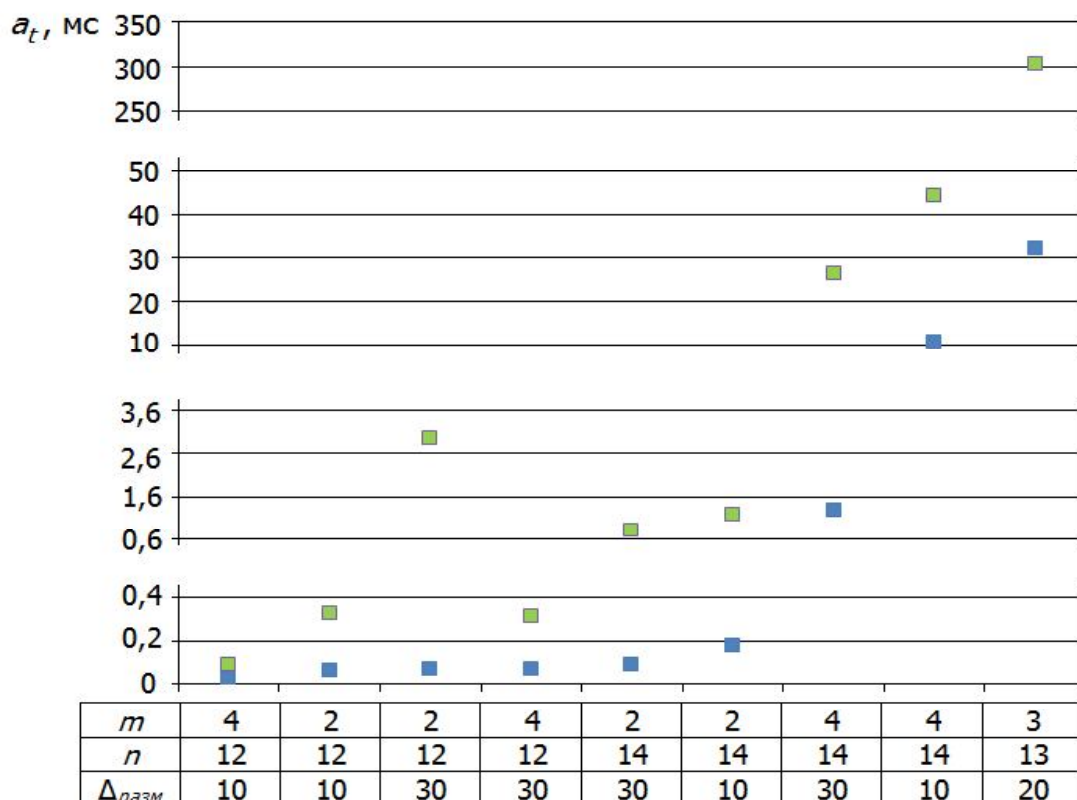


Рис. 1. Влияние параметров ОРЗ (m , n , $\Delta_{разм}$) на ресурсно-временные характеристики АР: ■ — a_t^{mid} ; ■ — a_t^{max}

В целом, по сравнению с полным перебором классический алгоритм Романовского позволяет значительно сократить количество рассматриваемых во время поиска решений. В среднем он требует на 2–6 порядков меньше вычислительных ресурсов, чем алгоритм полного перебора (АПП), для оценки временного ресурса которого возможно применение и аналитических зависимостей. В пределах эксперимента даже в худших случаях среднее время работы рассматриваемого алгоритма оказалось примерно в 15 раз меньше в сравнении с АПП. Однако при решении некоторых ОРЗ АР существенно более ресурсоемок по сравнению с АПП. Например, при решении ОРЗ с параметрами $m = 3$, $n = 19$ и $\Delta_{разм} = 10$ АПП требуется в среднем 11 мин, тогда как АР только 2 из 100 задач решил в ограниченное по эксперименту время $T = 30$ мин, а в остальных 98 случаях не завершил работу в срок. Этот феномен требует разъяснения, т. к. теоретически лишь в самых неблагоприятных ситуациях АР может приближаться по временному ресурсу к АПП, но никак не должен быть хуже. В связи с этим и была поставлена задача исследования АР с целью нахождения возможного способа повышения его ресурсной эффективности.

Исследование возможных способов повышения быстродействия АР при решении ОРЗ. Системное исследование алгоритмического механизма решения z-задачи АР позволило выявить в нём следующие избыточности, приводящие к увеличению времени решения.

Первая избыточность: АР при некоторых условиях осуществляет перебор на каждом исполнителе всех возможных перестановок заданий, а не их различных сочетаний (комбинаций). Однако для ОРЗ порядок выполнения заданий на исполнителе не важен — и в этих случаях алгоритм затрачивает лишнее время на рассмотрение одинаковых по составу наборов заданий, отличающихся лишь порядком следования в них элементов. Поскольку количество перестановок заданий существенно больше количества их сочетаний, то данный дефект может значительно увеличивать время решения ОРЗ алгоритмом Романовского. Причём в этих случаях, как показывает практика, время работы АР может быть даже существенно больше времени работы АПП, поскольку в последнем осуществляется перебор на каждом исполнителе только различных сочетаний (комбинаций) заданий, а не их перестановок.

Вторая избыточность связана с тем, что для РЗ с малым разбросом размеров заданий характерно наличие существенного количества одинаковых по размеру групп заданий. Поэтому значительно и количество различных множеств заданий с одинаковыми сочетаниями их размеров, которые рассматриваются АР во время поиска решения. Однако АР не учитывает это свойство ОРЗ. Вследствие этого на рассмотрение большого количества одинаковых по сочетанию размеров заданий ветвей дерева решений ОРЗ затрачивается много времени.

Выявленные особенности АР иллюстрируются примером. Пусть требуется построить оптимальное с точки зрения минимаксного критерия [1–3] расписание выполнения 17 заданий на 4-исполнительной системе. Размеры заданий сгенерированы случайным образом из узкого диапазона $r_i \in [47, 53]$ (табл. 2). С помощью табл. 2 можно рассмотреть подробно ход работы АР при решении ОРЗ. Причём рассматривается один из наиболее ресурсоёмких при решении данной ОРЗ этапов работы, а именно решение z-задачи с максимально допустимой загрузкой исполнителей $z = 235$.

Таблица 2

Распределяемое множество заданий и их размеры

Задание	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9
Размер задания r_i	53	47	50	48	48	50	51	49	51
Задание	W_{10}	W_{11}	W_{12}	W_{13}	W_{14}	W_{15}	W_{16}	W_{17}	
Размер задания r_i	47	51	53	52	49	48	49	48	

Из элементов приведённого в табл. 2 множества распределяемых заданий строится очередь L по убыванию их размеров (табл. 3). Вычисляется *slack* — общий ещё не использованный исполнителями запас «недогруженности»: для $z = 235$ *slack* = 96.

Таблица 3

Очередь заданий по убыванию их размеров

№ элемента очереди	1	2	3	4	5	6	7	8	9
Задание	W_1	W_{12}	W_{13}	W_7	W_9	W_{11}	W_3	W_6	W_8
Размер задания r_i	53	53	52	51	51	51	50	50	49
№ элемента очереди	10	11	12	13	14	15	16	17	
Задание	W_{14}	W_{16}	W_4	W_5	W_{15}	W_{17}	W_2	W_{10}	
Размер задания r_i	49	49	48	48	48	48	47	47	

Далее из очереди L назначаются задания на выполнение 1-му исполнителю e_1 так, чтобы его суммарная по размерам заданий загрузка R_1^z удовлетворяла условиям

$$\left. \begin{array}{l} a) R_1^z \leq z, \\ b) z - R_1^z \leq \text{slack}. \end{array} \right\} \quad (2)$$

Первым рассматриваемым алгоритмом набором заданий, удовлетворяющим (2), является $W_1 = \{w_1, w_{12}, w_{13}, w_7\}$. Назначенные 1-му исполнителю задания исключаются из очереди L , и она принимает следующий вид:

$$L = (w_9, w_{11}, w_3, w_6, w_8, w_{14}, w_{16}, w_4, w_5, w_{15}, w_{17}, w_2, w_{10}). \quad (3)$$

Таким образом, загрузка 1-го исполнителя составляет $R_1^\Sigma = 209$, а общий ещё не использованный исполнителями запас «недогруженности» — $slack := slack - (z - R_1^\Sigma) = 70$.

Далее осуществляется переход к следующему по порядку исполнителю e_2 и в соответствии с (2) и (3) ему назначается набор заданий $W_2 = \{w_9, w_{11}, w_3, w_6\}$. Соответственно, загрузка 2-го исполнителя составляет $R_2^\Sigma = 202$, $slack = 37$, а очередь L имеет вид

$$L = (w_8, w_{14}, w_{16}, w_4, w_5, w_{15}, w_{17}, w_2, w_{10}). \quad (4)$$

Затем осуществляется назначение заданий 3-му по порядку исполнителю e_3 . Однако, перебрав все возможные варианты его загрузки, АР не находит набор заданий, удовлетворяющих (2), и производит откат ко 2-му исполнителю для проверки других допустимых вариантов его загрузки. Для этого алгоритм отменяет назначение последнего задания 2-го исполнителя w_6 , возвращает его в очередь (4) и назначает следующий за ним элемент — w_8 . Таким образом, на 2-го исполнителя назначается $W_2 = \{w_9, w_{11}, w_3, w_8\}$ с $R_2^\Sigma = 201$. Соответственно, $slack = 36$, а очередь L имеет вид

$$L = (w_6, w_{14}, w_{16}, w_4, w_5, w_{15}, w_{17}, w_2, w_{10}). \quad (5)$$

Далее для 3-го исполнителя алгоритм снова не находит допустимую загрузку, производит откат ко 2-му исполнителю и аналогичным образом в качестве последнего задания последовательно рассматривает остальные задания из очереди (5), т. е. на 2-м исполнителе рассматриваются множества заданий $\{w_9, w_{11}, w_3, w_{14}\}$, $\{w_9, w_{11}, w_3, w_{16}\}$ и т. д.

Перебрав все возможные варианты назначений из очереди (5) на место последнего задания 2-го исполнителя, АР не находит решения z -задачи. Поэтому он осуществляет откат к предыдущему (предпоследнему) назначенному заданию w_3 , отменяет его назначение и назначает вместо него задание w_6 , т. е. на 2-го исполнителя назначается $W_2 = \{w_9, w_{11}, w_6\}$ с $R_2^\Sigma = 152$, а очередь L принимает вид

$$L = (w_3, w_8, w_{14}, w_{16}, w_4, w_5, w_{15}, w_{17}, w_2, w_{10}). \quad (6)$$

Будем обозначать «недогруженность» j -го исполнителя sl_j . Поскольку «недогруженность» 2-го исполнителя $sl_2 = 83$ больше максимального размера распределяемых заданий $r_{\max} = 53$, то в соответствии с описанным в [8] правилом выбора очередного задания из очереди в W_2 включается первое задание очереди (6) — w_3 , т. е. исполнителю назначается $W_2 = \{w_9, w_{11}, w_6, w_3\}$. Однако данное множество по составу совпадает с ранее рассмотренным (первым по порядку) на 2-м исполнителе множеством $\{w_9, w_{11}, w_3, w_6\}$. Поэтому перебор различных загрузок 3-го исполнителя также не даёт положительного результата.

Далее аналогичным образом на 2-м исполнителе алгоритм рассматривает ещё четыре таких же по составу множества заданий, отличающихся лишь порядком следования в них элементов: $\{w_9, w_3, w_{11}, w_6\}$, $\{w_9, w_3, w_6, w_{11}\}$, $\{w_9, w_6, w_{11}, w_3\}$, $\{w_9, w_6, w_3, w_{11}\}$ и т. д.

Для иллюстрации второго дефекта АР в табл. 4 приведены сочетания размеров заданий, рассмотренных алгоритмом на 2-м исполнителе во время поиска решения данной z -задачи. Из таблицы видно, что даже в тех случаях, когда АР рассматривает на 2-м исполнителе различные

по составу наборы заданий, сочетания размеров могут совпадать. При этом количество таких случаев довольно велико: из 8 первых рассмотренных различных множеств заданий только 3 (меньше половины) имеют различные сочетания размеров.

Таблица 4

Рассмотренные АР на 2-м исполнителе сочетания размеров заданий

№ п/п	Набор заданий	Размеры заданий	Суммарный размер
1	$\{w_9, w_{11}, w_3, w_6\}$	$\{51, 51, 50, 50\}$	202
2	$\{w_9, w_{11}, w_3, w_8\}$	$\{51, 51, 50, 49\}$	201
3	$\{w_9, w_{11}, w_3, w_{14}\}$	$\{51, 51, 50, 49\}$	
4	$\{w_9, w_{11}, w_3, w_{16}\}$	$\{51, 51, 50, 49\}$	
5	$\{w_9, w_{11}, w_3, w_4\}$	$\{51, 51, 50, 48\}$	200
6	$\{w_9, w_{11}, w_3, w_5\}$	$\{51, 51, 50, 48\}$	
7	$\{w_9, w_{11}, w_3, w_{15}\}$	$\{51, 51, 50, 48\}$	
8	$\{w_9, w_{11}, w_3, w_{17}\}$	$\{51, 51, 50, 48\}$	
...	...	...	...

В итоге, перебрав все возможные варианты загрузки для каждого исполнителя, АР не смог получить удовлетворяющее ограничению $z = 235$ распределение заданий по исполнителям. При этом количество рассмотренных АР ветвей дерева вариантов составило около 1,5 миллиарда, тогда как только около 37 тысяч (более чем на 4 порядка меньше!) из них имеют различные сочетания размеров заданий. В результате АР затратил на решение этой z -задачи 316,05 с.

Далее АР переходит к решению z -задач с ограничением $z > 235$ и находит оптимальное решение ОРЗ при $z = 239$. В результате общее время решения РЗ составило 911,85 с.

Таким образом, можно существенно сократить количество рассматриваемых классическим АР во время поиска вариантов распределения заданий по исполнителям. Для этого в указанный алгоритм следует добавить механизм обеспечения комбинаторной уникальности загрузки исполнителя при обходе ветвей дерева вариантов решений. Это позволит значительно повысить ресурсно-временную эффективность АР при решении РЗ, сохранив при этом точность его работы.

Комбинаторная модификация АР точного решения ОРЗ. Из предшествующего анализа видно, что для обеспечения комбинаторной уникальности загрузки исполнителя при обходе ветвей дерева вариантов решений в классический АР необходимо внести ряд изменений. Это нужно сделать так, чтобы не изменить сущность реализации метода ветвей и границ.

1. Для обеспечения перебора на исполнителях только различных сочетаний заданий (т. е. для устранения первого выявленного выше дефекта АР) необходимо изменить правило выбора алгоритмом очередного задания из очереди ещё не распределённых заданий. Для этого следует исключить из данного правила условие, при котором алгоритм назначает j -му исполнителю задания из головы очереди ещё не распределённых заданий, если «недогруженность» s_j этого исполнителя больше максимального размера распределяемых заданий $r_{\max} = \max_R \{r_i \mid r_i \in R\}$ (R — множество размеров распределяемых заданий). При этом нужно осуществлять выбор алгоритмом очередного задания w_{i_k} из очереди $L = (w_{i_1}, w_{i_2}, \dots, w_{i_S})$ для включения его в рассматриваемое W_j только из той части очереди, которая ещё не была проанализирована алгоритмом, т. е. соблюдая условие

$$\left. \begin{aligned} a) r_{i_k} &\leq sl_j, \\ b) r_{i_k} &= \max_{i_p < i \leq i_s} r_i, \end{aligned} \right\} \quad (7)$$

где i_p — индекс извлечённого на предыдущей итерации алгоритма задания из очереди L .

Такая модификация AP обусловлена тем, что с назначением на следующих итерациях задания из головы очереди возможно повторное назначение уже рассмотренного задания. Это происходит в случае, если на предыдущих итерациях алгоритм отменил назначение с j -го исполнителя ряд уже проанализированных им заданий и вернул их обратно в очередь. При выборе заданий из той части очереди, которая ещё не была проанализирована алгоритмом, будет обеспечен перебор на исполнителе только различных сочетаний заданий.

Второе. Требуется обеспечить перебор на исполнителях только различных по сочетанию размеров наборов заданий (для устранения второго выявленного выше дефекта AP). Для этого в случае необходимости отмены назначения задания на исполнителе, вместо него назначается не следующее задание из очереди (как в AP), а то задание, которое имеет отличный от отменённого размер. В качестве пояснения: пусть $L^q = (w_{i_1}, w_{i_2}, \dots, w_{i_s})$ — очередь расположенных по убыванию размеров $(r_{i_1} \geq r_{i_2} \geq \dots \geq r_{i_s})$ нераспределённых на q -м шаге алгоритма заданий, а $w_{i_k} \in L^q$ — задание, назначение которого отменено алгоритмом с e_j исполнителя. Тогда размер r_{i_p} назначаемого на e_j исполнителя задания $w_{i_p} \in L^q$ ($i_p > i_k$) (вместо задания $w_{i_k} \in L^q$) должен удовлетворять условию

$$r_{i_p} \neq r_{i_k}. \quad (8)$$

Предложенное изменение позволяет обеспечить перебор на исполнителе только различных по сочетанию размеров наборов заданий, и вот почему. В AP осуществляется односторонний обход дерева вариантов решений, а назначение заданий исполнителям производится из упорядоченной по убыванию размеров очереди. В силу этого перебор различных по сочетанию размеров комбинаций заданий для исполнителя алгоритм осуществляет упорядоченно, т. е. сначала рассматриваются комбинации заданий с большими значениями их размеров (табл. 4). Такой перебор комбинаций заданий обеспечивает последовательное (друг за другом) рассмотрение AP различных наборов заданий с одинаковым сочетанием размеров. Следовательно, алгоритму при переборе наборов заданий достаточно проверять отличие сочетания размеров от рассмотренного на предыдущем шаге в случае отката — таким образом обеспечивается комбинационная уникальность загрузки исполнителя при обходе ветвей дерева вариантов решений.

Соблюдение (8) позволит алгоритму пропустить рассмотрение тех заданий очереди, которые имеют одинаковый с отменяемым заданием размер, исключив из перебора одинаковые по сочетанию размеров заданий ветви дерева вариантов решений задачи. В изменении остальных этапов работы классического AP нет необходимости.

Пример применения комбинационно модифицированного AP (КМАР). Работу КМАР хорошо иллюстрирует уже рассмотренный пример решения z -задачи. Решение её КМАР осуществляется по AP до того момента, когда оказывается необходимым сформировать на 2-м исполнителе новый набор заданий. С этого места необходимо рассмотреть работу алгоритма более подробно.

Для формирования ещё не рассматривавшегося на 2-м исполнителе набора заданий КМАР отменяет выбор последнего назначенного на данного исполнителя задания w_6 , возвращает его в очередь (4) и назначает вместо него из данной очереди задание w_8 , удовлетворяющее (8). В данном случае из очереди выбирается то же задание, что и классическим AP. Таким образом, на 2-го

исполнителя назначается множество заданий $W_2 = \{w_9, w_{11}, w_3, w_8\}$ с размерами $R_2 = \{51, 51, 50, 49\}$. Соответственно, загрузка 2-го исполнителя $R_2^\Sigma = 201$, $slack = 36$, а очередь L имеет вид

$$L = (w_6, w_{14}, w_{16}, w_4, w_5, w_{15}, w_{17}, w_2, w_{10}). \quad (9)$$

Для 3-го исполнителя алгоритм не находит допустимую загрузку и откатывается ко 2-му исполнителю. Аналогично предыдущему, КМАР вместо задания w_8 назначает не следующее за ним в очереди (9) задание w_{14} (как базовый АР), а удовлетворяющее (8) задание w_4 . В результате, на 2-го исполнителя назначается множество заданий $W_2 = \{w_9, w_{11}, w_3, w_4\}$, отличное от предыдущего как по составу, так и по сочетанию размеров $R_2 = \{51, 51, 50, 48\}$. В результате КМАР исключил из рассмотрения наборы заданий $\{w_9, w_{11}, w_3, w_{14}\}$ и $\{w_9, w_{11}, w_3, w_{16}\}$, характеризующиеся одинаковым с полученным на предыдущем шаге сочетанием размеров, и сократил временные затраты на проверку соответствующих ветвей дерева вариантов.

Аналогичным образом на 2-м исполнителе рассматривается набор заданий $W_2 = \{w_9, w_{11}, w_3, w_2\}$ с $R_2 = \{51, 51, 50, 47\}$, причём с пропуском наборов $\{w_9, w_{11}, w_3, w_5\}$, $\{w_9, w_{11}, w_3, w_{15}\}$ и $\{w_9, w_{11}, w_3, w_{17}\}$ с уже рассмотренным сочетанием размеров $\{51, 51, 50, 48\}$ и т. д.

Перебрав все возможные варианты назначений из очереди (9) на место последнего задания 2-го исполнителя, КМАР не находит решения z-задачи. Поэтому он осуществляет откат к предыдущему (предпоследнему) назначенному заданию w_3 , отменяет его назначение и назначает вместо него удовлетворяющее (9) задание w_8 , т. е. на 2-го исполнителя назначается $W_2 = \{w_9, w_{11}, w_8\}$ с $R_2 = \{51, 51, 49\}$. Соответственно, суммарная загрузка составляет $R_2 = \{51, 51, 49\}$ $R_2^\Sigma = 151$, а очередь L принимает вид

$$L = (w_3, w_6, w_{14}, w_{16}, w_4, w_5, w_{15}, w_{17}, w_2, w_{10}). \quad (10)$$

Затем алгоритм в соответствии с модифицированным правилом выбора очередного задания из очереди ещё не распределённых заданий (см. выше) назначает не первое задание очереди — w_3 (как базовый АР), а удовлетворяющее (7) задание w_{14} , т. е. на 2-го исполнителя назначается $W_2 = \{w_9, w_{11}, w_8, w_{14}\}$ с $R_2 = \{51, 51, 49, 49\}$. Таким образом, КМАР исключил из рассмотрения множество заданий $\{w_9, w_{11}, w_8, w_3\}$, которое имеет одинаковый с ранее рассмотренным множеством $\{w_9, w_{11}, w_3, w_8\}$ набор заданий, но отличается лишь порядком следования в нём элементов.

Далее КМАР аналогичным образом исключает из рассмотрения на 2-м исполнителе ещё четыре одинаковых по составу с ранее рассмотренным $\{w_9, w_{11}, w_3, w_8\}$ множества заданий, отличающихся лишь порядком следования в них элементов, и т. д.

В итоге, перебрав все возможные варианты загрузки для каждого исполнителя, КМАР (так же, как и классический АР) не смог получить удовлетворяющее ограничению $z = 235$ распределение заданий по исполнителям. Однако вследствие того, что КМАР во время решения данной z-задачи рассматривал только различные по сочетанию размеров наборы заданий на исполнителях, объём рассмотренных им вариантов распределений существенно меньше (более чем на 4 порядка) по сравнению с базовым АР. В результате ресурсно-временные затраты на решение этой z-задачи снизились с 316,05 с до 0,0159 с.

Далее КМАР переходит к решению z -задач с ограничением $z > 235$ и находит совпадающее с полученным базовым АР оптимальное решение РЗ при $z = 239$. В результате применение КМАР позволяет уменьшить общее время решения РЗ с 911,85 с до 0,0517 с.

Экспериментальное исследование ресурсно-временных характеристик КМАР. Проведённое с помощью специально разработанного программного средства экспериментальное исследование КМАР показало его высокие ресурсно-временные характеристики, позволяющие сократить время точного решения практически любой ОРЗ в диапазоне от 2 до 8 исполнителей и от 10 до 50 заданий. В ряде случаев применение КМАР (исследования проводились более чем на 16000 задач) позволило получить 200000-кратный временной выигрыш. Средний же выигрыш по всей совокупности решённых задач — 5600-кратный.

На диаграмме рис. 2 представлены результаты одного из проведённых вычислительных экспериментов, иллюстрирующие высокую ресурсно-временную результативность КМАР. На этом рисунке представлены значения отношения среднего времени работы АР к среднему времени КМАР $\beta_{алг} = a_t^{AP} / a_t^{КМАР}$ по основной совокупности оценок и по совокупности максимальных оценок при решении наиболее ресурсоёмких для классического АР ОРЗ с узким диапазоном размеров распределяемых заданий $\Delta_{разм} = 6$. В каждом опыте эксперимента алгоритмами решалось по 100 РЗ.

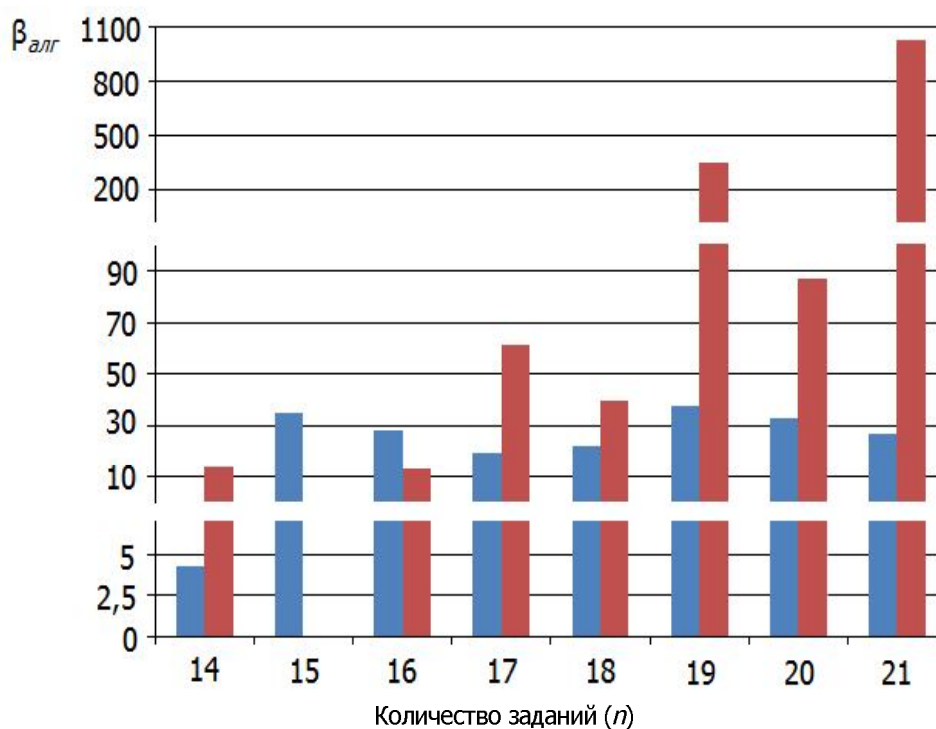


Рис. 2. Временной относительный выигрыш КМАР для 7-исполнительной системы: ■ — S_{mid} ; ■ — S_{max}

Экспериментальные исследования КМАР также показали, что, во-первых, данная модификация позволяет не только сократить время решения ОРЗ по сравнению с базовым алгоритмом, но и значительно (в несколько раз по количеству распределяемых заданий) расширить диапазон доступных для решения за приемлемое время (до 70 мин) задач. Во-вторых, зависимость среднего времени решения КМАР ОРЗ с узким диапазоном размеров распределяемых задач ($\Delta_{разм} \leq 10$) от количества заданий имеет периодичный «пилообразный» характер с экспоненциальным ростом значений. В качестве примера на рис. 3 такая зависимость показана для $m = 7$. Фактические исследования были проведены для $m = 2...8$, и при всех значениях m характер зависимости подтвердился.

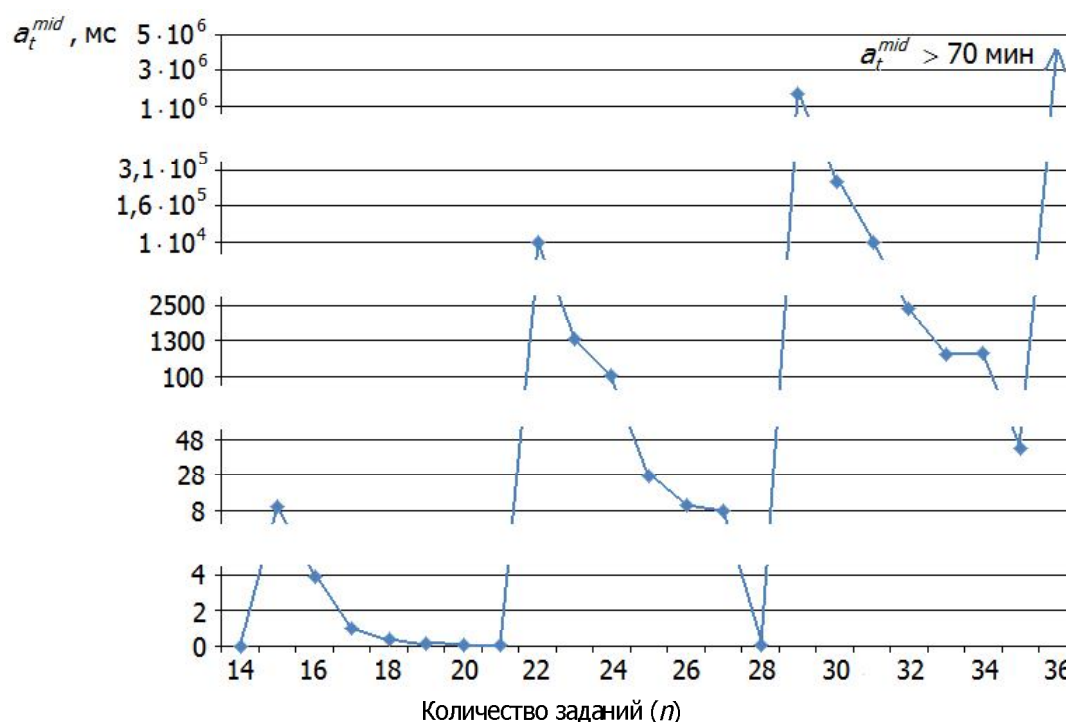


Рис. 3. Анализ временного ресурса КМАР для 7-исполнительной системы

Выводы.

1. Свойства описанной в статье комбинационной модификации АР исследованы в достаточно широком диапазоне структурных параметров ОРЗ (m — до 8 исполнителей и n — до ~50 заданий) и для предельно трудного варианта распределения близких по величине заданий ($\Delta_{\text{разм}} = 6$). Несмотря на это, статистически весьма представительные исследования (более 16000 вычислительных экспериментов) показали, что эта модификация позволяет существенно (в основном примерно на 1–3 порядка, а в частных случаях более чем на 4 порядка) снизить время и расширить диапазон оптимизационного решения ОРЗ по сравнению с классическим АР.

2. Однако и теоретический анализ, и экспериментальное изучение свойств КМАР показали, что он остаётся NP-полным алгоритмом, и его средние ресурсно-временные показатели растут экспоненциально с ростом размерности задачи. Таким образом, разработанная комбинационная модификация позволила лишь уменьшить среднее значение показателя степени общей экспоненциальной зависимости временного ресурса (важнейшей характеристики сложности алгоритма), но не изменила качественно сложности модифицированного метода ветвей и границ, что естественно.

3. Полученные весьма обнадеживающие результаты модификации классического алгоритма решения распределительной задачи показывают актуальность и перспективность исследований в области совершенствования фундаментальных методов решения задач теории расписаний.

Библиографический список

1. Jackson, J.-R. Scheduling a production line to minimize maximum tardiness / J.-R. Jackson // Research Report. Los Angeles University of California Management Sciences Research Project. — 1955. — № 43. — 72 p.
2. Bellman, R. Mathematical aspects of scheduling theory / R. Bellman // Journal of the Society for Industrial and Applied Mathematics. — 1956. — Vol. 4. — Pp. 168–205.

3. Коффман, Э. Г. Теория расписания и вычислительные машины / Э. Г. Коффман. — Москва : Наука, 1984. — 334 с.
4. Танаев, В. С. Теория расписаний. Одностадийные системы / В. С. Танаев, В. С. Гордон, Я. М. Шафранский. — Москва : Наука, 1984. — 384 с.
5. Танаев, В. С. Теория расписаний. Многостадийные системы / В. С. Танаев, Ю. Н. Сотсков, В. А. Струевич. — Москва : Наука, 1989. — 328 с.
6. Scheduling Computer and Manufacturing Processes / J. Blazewicz [et al.]. — New York : Springer, 2001. — 485 p.
7. Лазарев, А. А. Теория расписаний. Задачи и алгоритмы / А. А. Лазарев, Е. Р. Гафаров. — Москва : Моск. гос. ун-т им. М. В. Ломоносова, 2011. — 222 с.
8. Романовский, И. В. Алгоритмы решения экстремальных задач / И. В. Романовский. — Москва : Наука, 1977. — 352 с.
9. Нейдорф, Р. А. Повышение ресурсной эффективности алгоритма точного решения однородной распределительной задачи / Р. А. Нейдорф, А. А. Жикулин // Математические методы в технике и технологиях — ММТТ-27 : сб. тр. XXVII Междунар. науч. конф. — Тамбов : Тамб. гос. техн. ун-т, 2014. — Т. 3. — С. 42–46.
10. Бочаров, П. П. Теория вероятностей. Математическая статистика / П. П. Бочаров, А. В. Печинкин. — Москва : ФИЗМАТЛИТ, 2005. — 296 с.
11. Адлер, Ю. П. Планирование эксперимента при поиске оптимальных условий / Ю. П. Адлер, Е. В. Маркова, Ю. В. Гранковский. — Москва : Наука, 1976. — 278 с.

Материал поступил в редакцию 04.06.2014.

References

1. Jackson, J.-R. Scheduling a production line to minimize maximum tardiness. Research Report. Los Angeles University of California Management Sciences Research Project, 1955, no. 43, 72 p.
2. Bellman, R. Mathematical aspects of scheduling theory. Journal of the Society for Industrial and Applied Mathematics, 1956, vol. 4, pp. 168–205.
3. Koffman, E. G. Teoriya raspisaniya i vychislitelnyye mashiny. [Scheduling theory and computers.] Moscow : Nauka, 1984, 334 p. (in Russian).
4. Tanayev, V. S., Gordon, V. S., Shafranskiy, Y. M. Teoriya raspisaniy. Odnostadiynnye sistemy. [Scheduling theory. Single-stage systems.] Moscow : Nauka, 1984, 384 p. (in Russian).
5. Tanayev, V. S., Sotskov, Y. N., Strusevich, V. A. Teoriya raspisaniy. Mnogostadiynnye sistemy. [Scheduling theory. Multistage systems.] Moscow : Nauka, 1989, 328 p. (in Russian).
6. Blazewicz, J., et al. Scheduling Computer and Manufacturing Processes. New York : Springer, 2001, 485 p.
7. Lazarev, A. A., Gafarov, E. R. Teoriya raspisaniy. Zadachi i algoritmy. [Scheduling theory. Problems and algorithms.] Moscow: Lomonosov Moscow State University, 2011, 222 p. (in Russian).
8. Romanovskiy, I. V. Algoritmy resheniya ekstremalnykh zadach. [Algorithms for solving extremum problems.] Moscow : Nauka, 1977, 352 p. (in Russian).
9. Neydorf, R. A., Zhikulin, A. A. Povysheniye resursnoy effektivnosti algoritma tochnogo resheniya odnorodnoy raspredelitelnoy zadachi. Matematicheskiye metody v tekhnike i tekhnologiyakh — ММТТ-27: сб. тр. XXVII Mezhdunar. nauch. konf. [Resource efficiency increase of the algorithm for homogeneous allocation problem exact solution. Mathematical methods in Engineering and Technology — ММТТ-27: Proc. XXVII Int. Sci. Conf.] Tambov : TGTU, 2014, vol. 3, pp. 42–46 (in Russian).
10. Bocharov, P. P., Pechinkin, A. V. Teoriya veroyatnostey. Matematicheskaya statistika. [Probability calculus. Mathematical statistics.] Moscow : FIZMATLIT, 2005, 296 p. (in Russian).

11. Adler, Y. P., Markova, E. V., Grankovskiy, Y. V. Planirovaniye eksperimenta pri poiske optimalnykh usloviy. Planning an experiment under search for optimal conditions.] Moscow : Nauka, 1976, 278 p. (in Russian).

SPEEDING ALGORITHM FOR MINIMAX OPTIMIZATION OF ALLOCATION PROBLEM SOLUTIONS IN HOMOGENEOUS SYSTEMS*

R. A. Neydorf, A. A. Zhikulin

The optimization algorithm for solving homogeneous allocation problems of the scheduling theory is described and proved. It is a modification of Romanovsky's algorithm known in this problem domain. Romanovsky's algorithm is a classical version of the branch-and-bound method with one-way traversing a decision tree. A systematic study of this algorithm that allows revealing reasons for its operate time increment when traversing some decision tree branches is carried out. It allows proposing modification free of the revealed shortcoming. It is called a combinatorial-modified Romanovsky's algorithm. The essence of this modification is as follows. In the process of solving the allocation problem, those rules, stages, and steps that lead to the sorting on executors the sets of tasks deliberately duplicating the previous effects are selectively skipped. The essence of the new algorithm is illustrated by an example. The statistically presented studies are resulted. They demonstrate the algorithm capabilities on the high dimensional allocation problems. (Such problems cannot be solved by the classical algorithm due to the limited timing budgets.) The results of processing these solutions have shown that the new modification does not solve the problem of NP-complete allocation tasks, but it provides a resource-time gain associated with the significant reduction in the exponential model index of the average solution time.

Keywords: scheduling theory, allocation problem, Romanovsky's algorithm, decision tree, combination search, job size, combinations, permutations.

* The research is done on theme no. 2.3.13 "Balanced sampling method for the problems of dynamic process simulation in distributed objects" on the thematic research plan of the RF Ministry of Education and Science.