ТЕХНИЧЕСКИЕ НАУКИ

УДК 519.87

Сравнение точностных характеристик простейших алгоритмов решения однородной минимаксной задачи

В.Г. Кобак, Г.А. Гаврилов
Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

Аннотация
Исследованы методы эффективного распределения задач между вычислительными ресурсами в современных информационных технологиях. Рассмотрены алгоритм критического пути, алгоритм Пашкеева и точный алгоритм Романовского. Каждый из них направлен на равномерное распределение задач с целью оптимизации производительности. В вычислительном эксперименте задействовали различные объемы исходных данных. Использовалась программа, написанная на языке «Пайтон» (Python). В таблицах представлены результаты сравнения работы алгоритмов при различных условиях. Полученные данные позволяют сделать выводы об их эффективности.

Ключевые слова: алгоритм критического пути, алгоритм Пашкеева, точный алгоритм Романовского, теория расписаний, списочные алгоритмы, равномерное распределение задач между вычислительными устройствами.

Для цитирования. Кобак В.Г., Гаврилов Г.А. Сравнение точностных характеристик простейших алгоритмов решения однородной минимаксной задачи. *Молодой исследователь Дона.* 2024;9(4):4–9.

Comparison of Accuracy Characteristics of the Simplest Algorithms for Solving a Homogeneous Minimax Problem

Valerii G. Kobak, Gleb A. Gavrilov
Don State Technical University, Rostov-on-Don, Russian Federation

Abstract
Methods for efficiently distributing tasks between computing resources in modern information technologies have been studied. The critical path algorithm, Pashkeev's algorithm and Romanovsky's exact algorithm are considered. Each of them aims to distribute tasks evenly to optimize performance. The computational experiment involved various amounts of initial data. A program written in Python was used. The tables present the results of comparing the performance of the algorithms under various conditions. The data obtained allow us to draw conclusions about their effectiveness.

Keywords: critical path algorithm, Pashkeev's algorithm, Romanovsky's exact algorithm, scheduling theory, list algorithms, uniform distribution of tasks between computing devices

For citation. Kobak VG, Gavrilov GA. Comparison of Accuracy Characteristics of the Simplest Algorithms for Solving a Homogeneous Minimax Problem. *Young Researcher of Don.* 2024;9(2):4–9.

Введение. Эффективное управление вычислительными устройствами особенно важно при выполнении задач, требующих высокой производительности. Отметим, что быстродействие аппаратной части, включая микропроцессоры, не единственное условие оперативности [1]. Решающую роль играет адекватное распределение нагрузки между вычислительными устройствами. Например, при использовании комплекса из четырех устройств рационально распределять нагрузку равномерно, чтобы повысить эффективность вычислительного процесса и сократить время выполнения операций.

Для создания плана равномерного распределения используют точные и приближенные алгоритмы. Точные дают оптимальные решения, однако требуют значительных временных затрат. Иногда это неприемлемо, особенно если превышается время выполнения вычислительной задачи. Приближенные алгоритмы отличаются более высокой скоростью, но могут уступать в точности [2]. Для нахождения приближенного решения предлагается рассмотреть списочные алгоритмы — критического пути и Пашкеева.

Цель данной научной работы заключается в исследовании и сравнении списочных и точного алгоритмов для оптимизации производительности в однородных системах и повышения эффективности информационных технологий. Основное внимание уделяется анализу временных и точностных характеристик алгоритмов, определяются их преимущества и недостатки.

Основная часть

Алгоритм критического пути. Алгоритм критического пути представляет собой списочный метод, ориентированный на эффективное распределение задач между обработчиками. Ниже перечислены его шаги.

1. Задания матрицы загрузки сортируются в порядке убывания значений элементов. Таким образом, задания с наибольшей загрузкой будут первыми в списке.

2. Распределение текущего задания. На каждом шаге выбирается текущее задание из отсортированного списка. Затем оно распределяется на прибор с наименьшей загрузкой. Если у нескольких приборов загрузка одинакова, то задание распределяется на прибор, который стоит слева.

3. Алгоритм завершает работу, когда все задания успешно распределены по обработчикам.

Проиллюстрируем это на примере.

Пусть количество заданий $m = 7$, а количество приборов $n = 3$. Все задания равномерно распределены в интервале от 10 до 20. Сгенерированная матрица M изображена на рис. 1.

$$M := \begin{pmatrix} 16 & 16 & 16 \\ 14 & 14 & 14 \\ 15 & 15 & 15 \\ 16 & 16 & 16 \\ 18 & 18 & 18 \\ 11 & 11 & 11 \\ 18 & 18 & 18 \end{pmatrix}$$

Рис. 1. Сгенерированная матрица M

На рис. 2 — переупорядоченная матрица $M1$.

$$M1 := \begin{pmatrix} 18 & 18 & 18 \\ 18 & 18 & 18 \\ 16 & 16 & 16 \\ 16 & 16 & 16 \\ 15 & 15 & 15 \\ 14 & 14 & 14 \\ 11 & 11 & 11 \end{pmatrix}$$

Рис. 2. Матрица после переупорядочивания

На рис. 3 — решение после применения алгоритма критического пути.

$$\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 18 & 18 & 16 \\ 15 & 14 & 16 \\ 33 & 32 & 32 \\ 0 & 11 & 0 \\ 33 & 43 & 32 \end{pmatrix} \end{array}$$

$$results := 43$$

Рис. 3. Решение матрицы алгоритмом критического пути

Алгоритм Пашкеева. Алгоритм Пашкеева — это тоже списочный метод для эффективного распределения задач между обработчиками. Ниже описан принцип его действия.

1. Задания матрицы загрузки сортируются в порядке убывания значений элементов. Таким образом, задания с наибольшей загрузкой будут первыми в списке.
2. Производится оценка текущей загрузки первого и последнего обработчиков.
3. Задания распределяются последовательно по N приборам, начиная с крайнего, с наименьшим значением нагрузки.
4. Работа алгоритма завершается, когда все задания успешно распределены по обработчикам.

Проиллюстрируем данный алгоритм.

Использование матрицы показано на рис. 2. Применение алгоритма Пашкеева дает решение, представленное на рис. 4.

$$
\begin{array}{c}
\begin{array}{ccc} 1 & 2 & 3 \end{array} \\
\begin{pmatrix} 18 & 18 & 16 \\ 14 & 15 & 16 \\ 32 & 33 & 32 \\ 0 & 0 & 11 \\ 32 & 33 & 43 \end{pmatrix}
\end{array}
$$

$$results := 43$$

Рис. 4. Решение матрицы алгоритмом Пашкеева

Таким образом, алгоритм Пашкеева также направлен на равномерное распределение задач. Однако он учитывает загрузку крайних обработчиков, начиная с того, у которого наименьшая текущая загрузка. Это позволяет более эффективно использовать ресурсы и уменьшить общее время выполнения вычислительных задач.

Все эти алгоритмы дают приближенные решения. Рассмотрим точный алгоритм Романовского.

Алгоритм Романовского. Он представляет собой точный метод решения задачи распределения заданий по приборам. Данный подход основан на последовательном распределении заданий с учетом доступных ресурсов каждого прибора. Цель алгоритма — максимизация загрузки каждого прибора, учитывая ограничение по количеству ресурсов z. Каждое задание распределяется таким образом, чтобы не превышать доступные ресурсы прибора. При этом учитывается свободное место на приборе, которое вычисляется из остатка свободного места (*slack*). Важно, чтобы остаток свободного места оставался неотрицательным на протяжении всего процесса распределения заданий. Если значение *slack* становится отрицательным, последнее действие отменяется в соответствии с правилами одностороннего обхода [3].

Рассмотрим данный алгоритм на примере.

Для решения используется матрица (см. рис. 2). Найдем верхнюю оценку, которая является суммой заданий, то есть $rec = 108$, а также нижнюю границу, которая находится по формуле $est = rec \div 36$. В данном примере $est = 36$. Попробуем решить эту задачу с $z = 40$. Свободное место будет составлять $40 \times 3 - 108 = 12$. Решение с использованием данного алгоритма показано на рис. 5.

1-й прибор. 40 22 4, остаток свбодного места 12 - 4 = 8
18 18
2-й прибор. 40 24 8, остаток свбодного места 8 - 8 = 8
16 16
3-й прибор. 40 25 11 0, остаток свбодного места 0 - 0 = 0
15 14 11

Рис. 5. Решение матрицы алгоритмом Романовского

Вычислительный эксперимент позволил определить временные и точностные характеристики алгоритмов.

Вычислительный эксперимент. Для оценки эффективности двух списочных алгоритмов (критического пути и Пашкеева) провели вычислительный эксперимент на различных объемах исходных данных с использованием программы, разработанной на языке «Пайтон» (Python). В качестве вычислительной платформы выбрали стационарный компьютер с процессором AMD Ryzen 7 3700X и оперативной памятью 16 гигабайт. Исходные данные представлены в виде 100 случайно сгенерированных матриц размерами 3×50, 8×50, 13×50, 18×50, 23×50 с диапазонами значений 10–20, 10–50, 10–100, а также матриц с размерами 3×500, 8×500, 13×500, 18×500, 23×500. Результаты вычислений представлены в табл. 1 и 2.

Таблица 1

Сравнение точности приближенных (списочных) алгоритмов* для матриц 3×50, 8×50, 13×50, 18×50, 23×50

N×*M*	Интервал значений	Среднее значение алгоритма		Максимальное значение алгоритма	
		Пашкеева	критического пути	Пашкеева	критического пути
3×50	10–20	253,02	252,59	268	268
	10–50	504,60	503,90	560	560
	10–100	924,41	923,01	1111	1109
8×50	10–20	100,98	100,62	108	108
	10–50	195,21	194,61	221	221
	10–100	351,27	349,34	415	412
13×50	10–20	60,11	59,73	64	64
	10–50	120,63	119,87	142	141
	10–100	221,16	219,84	253	254
18×50	10–20	45,55	44,97	50	50
	10–50	91,29	90,03	112	112
	10–100	165,42	162,45	196	196
23×50	10–20	40,74	40,25	44	43
	10–50	74,10	72,29	83	82
	10–100	129,88	127,41	152	151
*100 матриц.					

Таблица 2

Сравнение точности приближенных (списочных) алгоритмов* для матриц 3×500, 8×500, 13×500, 18×500, 23×500

N×*M*	Интервал значений	Среднее значение алгоритма		Максимальное значение алгоритма	
		Пашкеева	критического пути	Пашкеева	критического пути
3×500	10–20	2501,61	2501,2	2553	2553
	10–50	5014,77	5013,93	5249	5249
	10–100	9174,94	9173,81	9655	9655
8×500	10–20	943,33	942,57	971	969
	10–50	1882,26	1880,62	1961	1961
	10–100	3443,65	3441,75	3660	3661
13×500	10–20	583,03	582,13	597	596
	10–50	1160,05	1158,12	1214	1211
	10–100	2119,51	2117,51	2216	2216
18×500	10–20	420,29	419,26	429	428
	10–50	838,10	836,06	869	865
	10–100	1533,08	1530,43	1620	1618
23×500	10–20	330,30	329,18	339	337
	10–50	657,82	655,78	677	675
	10–100	1203,79	1201,66	1286	1283
*100 матриц.					

В табл. 3 и 4 показано сравнение приближенных алгоритмов с точным. Это дает возможность оценить степень точности и скорости работы каждого метода на конкретных данных. Анализ результатов позволит выбрать оптимальный алгоритм для определенной задачи в зависимости от требований к точности и скорости вычислений.

Таблица 3

Сравнение алгоритмов* по точности

N×M	Интервал значений	Значения списочных алгоритмов				Значение точного алгоритма Романовского	
		среднее		максимальное			
		Пашкеева	критического пути	Пашкеева	критического пути	среднее	максимальное
4×20	10–20	75,567	75,133	82	81	74,933	81
	15–20	87,633	87,533	91	91	87,533	91
4×21	10–20	85,467	85,567	94	94	79,033	87
	15–20	103,533	103,333	108	108	93,933	104
4×22	10–20	88,233	88,100	100	99	83,467	93
	15–20	103,167	102,700	107	107	96,733	101
4×23	10–20	89,000	88,667	98	98	86,333	95
	15–20	103,567	103,367	106	106	100,200	103
4×24	10–20	90,167	89,933	97	96	89,667	96
	15–20	106,200	106,000	111	111	105,967	110
*30 матриц.							

Таблица 4

Среднее время нахождения решения алгоритмами*, с

N×M	Интервал значений	Списочные алгоритмы		Точный алгоритм Романовского
		критического пути	Пашкеева	
4×20	10–20	0,000467	0,000612	0,0232
	15–20	0,000737	0,000676	55,780
4×21	10–20	0,000567	0,000567	0,0308
	15–20	0,000768	0,000768	6425,344
4×22	10–20	0,000901	0,000802	0,0324
	15–20	0,000813	0,000801	4765,721
4×23	10–20	0,000801	0,000768	0,0274
	15–20	0,000867	0,000834	4866,886
4×24	10–20	0,000967	0,000935	0,0292
	15–20	0,000938	0,000734	5987,363
*30 матриц.				

Заключение. Рассмотрены алгоритмы критического пути, Пашкеева и Романовского. Установлено, что алгоритм Романовского требует наиболее значительных временных затрат. При увеличении количества приборов и интервала значений алгоритм Пашкеева сохраняет приемлемую точность, однако уступает по этому показателю алгоритму критического пути.

Список литературы

1. Кобак В. Г., Титов Д. В. Исследование турнирного отбора в генетическом алгоритме для решения однородной минимаксной задачи. В: *Тр. Междунар. науч. конф. «Математические методы в технике и технологиях»*. Саратов: СГТУ; 2008. №2. С. 12.

2. Кобак В.Г., Шевченко В.В. Решение однородной минимаксной задачи экспериментальным алгоритмом без возвратов. *Известия вузов. Северо-Кавказский регион. Технические науки*. 2021;4:5–10.

3. Коффман Э.Г. *Теория расписания и вычислительные машины*. Москва: Наука; 1987. 334 с.

Об авторах:

Валерий Григорьевич Кобак, доктор технических наук, профессор кафедры программного обеспечения вычислительной техники и автоматизированных систем Донского государственного технического университета (344003, РФ, Ростов-на-Дону, пл. Гагарина, 1), valera33305@mail.ru

Глеб Андреевич Гаврилов, студент Донского государственного технического университета (344003, РФ, Ростов-на-Дону, пл. Гагарина, 1), gleb.gavrilov.2014@mail.ru

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the Authors:

Valerii G. Kobak, Dr. Sci. (Eng.), Professor of the Computer Software and Automated Systems Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), valera33305@mail.ru

Gleb A. Gavrilov, Student of the Computer Software and Automated Systems Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), gleb.gavrilov.2014@mail.ru

Conflict of interest statement: the authors do not have any conflict of interest.

All authors have read and approved the final manuscript.