



УДК 681.3-181.48

© 2004 г. **В.Г. Кобак**, канд. тех. наук,

С.Е. Федоров

(Ростовская государственная академия сельскохозяйственного
машиностроения)

МОДИФИКАЦИЯ АЛГОРИТМА АЛЕКСЕЕВА ПРИ ТОЧНОМ РЕШЕНИИ МИНИМАКСНОЙ ЗАДАЧИ ТЕОРИИ РАСПИСАНИЯ

Рассматривается алгоритм Алексева точного решения минимаксной задачи теории расписания, а также способы его упрощения для случая идентичных приборов. Приведен пример использования алгоритма и результаты вычислительных экспериментов.

Введение

В настоящее время актуальной проблемой остается разработка эффективных алгоритмов построения оптимальных расписаний параллельных вычислительных систем. В основном они находят применение для весьма простых математических моделей параллельных программ и многопроцессорных систем, среди которых выделяются детерминированные модели составления расписаний, обеспечивающие минимальное время выполнения параллельных программ [1].

Формулировка минимаксной задачи

Постановка задачи планирования выполнения параллельных программ может выглядеть следующим образом. Имеется система обслуживания, состоящая из n параллельных приборов (процессоров). На обслуживание поступает множество требований мощностью m . Возможности системы определяются матрицей $[t_{ij}]$ ($i = 1, \dots, m, j = 1, \dots, n$), где t_{ij} – длительность обслуживания i -го требования j -м прибором ($t_{ij} > 0$). Приборы в общем случае не идентичны, но каждое требование может быть обслужено любым прибором. В каждый момент времени отдельный прибор обслуживает не более одного требования. Обслуживание требования, находящегося на некотором из приборов, не прерывается для передачи на другой прибор.

Необходимо распределить требования таким образом, чтобы обеспечить минимальное время их обслуживания.

При отсутствии ограничений на возможные варианты расписания и без учета момента поступления требований задача сводится к разбиению множества требований $N = \{1, \dots, m\}$ на n непересекающихся подмножеств N_j ($j = 1, \dots, n$), т.е. справедливы следующие условия: $N_k \cap N_l = \emptyset$, $k, l = 1, \dots, n$, $k \neq l$; $\bigcup_{j=1}^n N_j = N$. Критерием разбиения, обеспечивающим оптимальность планирования требований по быстрдействию, является минимаксный критерий

$$\max_{1 \leq j \leq n} T_j \rightarrow \min, \quad (1)$$

где $T_j = \sum_{i \in N_j} t_{ij}$ – общее время загрузки j -го прибора.

Алгоритмы решения распределительной задачи

На практике при решении таких задач широкое распространение получили простые и достаточно эффективные списочные расписания, основанные на эвристических алгоритмах. В общем случае подобные расписания являются приближенными.

Для построения точных расписаний в основном используются алгоритмы, основанные на схеме метода ветвей и границ (МВГ). Среди этих алгоритмов можно выделить алгоритм Романовского [2], применяемый только для вычислительных систем с однородными (идентичными) приборами, т.е. в том случае, когда время обслуживания каждого требования одинаково, независимо от того, на каком из приборов будет происходить его обслуживание. Этот алгоритм весьма эффективен но при определенных патологических условиях объем выполняемых вычислений становится близок к полному перебору, что присуще и любому другому алгоритму, построенному по схеме МВГ. Другой алгоритм, опубликованный в работе Алексеева [3], применяется как для систем с неоднородными приборами, так и с однородными приборами. Причем следует отметить, что использование этого алгоритма для систем с однородными приборами не столь эффективно в сравнении с алгоритмом Романовского. В связи с этим следует рассмотреть возможность модификации алгоритма Алексеева для случая систем с однородными приборами.

Построение расписания при помощи алгоритма Алексеева

Процесс поиска оптимального решения состоит в направленном движении по вершинам дерева вариантов распределения требований. Дерево представляет собой n -нарное иерархическое дерево, в котором каждая вершина, за исключением корневой, имеет одного предшественника, а

также каждая вершина, за исключением конечных, имеет n потомков. Пример полного дерева вариантов распределения для 2 приборов и 3 требований приведен на рис. 1.

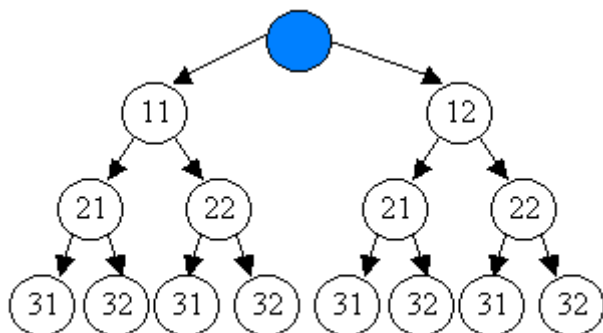


Рис. 1. Дерево решения.

Корневая вершина соответствует начальному этапу решения, когда ни одно требование еще не имеет назначения на конкретный прибор. Корень имеет n вершин-потомков, каждое из которых определяет, на какой из приборов будет распределено первое требование (поэтому вершины удобно пронумеровать

следующим образом: $z1, z2, \dots, zn$, где z – номер распределяемого требования). Данные вершины, в свою очередь, также имеют по n потомков, которые показывают распределение следующего требования, и так далее. Таким образом, дерево имеет $(m+1)$ уровней. Под z -м уровнем понимается множество вершин, для которых путь от этой вершины до корня содержит ровно $(z+1)$ вершин.

Стратегия ветвления основана на следующем. Пусть известно, как распределены первые $(z-1) < m$ требований, т.е. определено частичное решение в виде частично сформированных подмножеств требований, поступающих на обслуживание на j -й прибор N_j^z ($j=1, \dots, n$), и для частичного решения построена соответствующая часть дерева. На z -м уровне дерева формируется n вариантов распределения z -го требования, для чего последовательно принимается $z \in N_j^z$ ($j=1, \dots, n$).

Каждый из полученных вариантов распределения оценивается с помощью выражения

$$T_{zj} = \max \left\{ T_{id}^z; \max_{z+1 \leq i \leq m} \tau_i \right\}. \quad (2)$$

Здесь T_{id}^z – общее время обслуживания в системе с идеализированной дисциплиной обслуживания, для которой:

первые $z < m$ требований уже распределены по приборам, т.е. получены подмножества N_j^z ($j=1, \dots, n$) и определены частичные загрузки

$$T_j^z = \sum_{i \in N_j^z} t_{ij} \quad (j=1, \dots, n);$$

каждое из оставшихся $(m-z)$ требований будет обслужено с максимальной для него производительностью, т.е. за время

$$\tau_i = \min_{1 \leq j \leq n} t_{ij}, \quad i=1, 2, \dots, m;$$

также оставшиеся требования будут распределены между наименее загру-

женными приборами так, чтобы их загрузка была равномерной. Очевидно, номера этих приборов будут определяться первыми

$$r = \begin{cases} n, & \text{если } z \leq m - n, \\ m - z, & \text{если } z > m - n, \end{cases}$$

членами последовательности $\pi = (j_1, j_2, \dots, j_s, j_{s+1}, \dots, j_n)$, отвечающей неравенству $T_{j_1}^z \leq T_{j_2}^z \leq \dots \leq T_{j_s}^z \leq T_{j_{s+1}}^z \leq \dots \leq T_{j_n}^z$.

Выражение T_{id}^z определяется по формуле

$$T_{id}^z = \max \{ T^z; T_{j_n}^z \}, \quad (3)$$

где $T^z = \frac{1}{r} \left(\sum_{s=1}^r T_{j_s}^z + \sum_{i=z+1}^m \tau_i \right)$.

Значение T_{zj} дает оценку нижней границе решения в зависимости от соответствующего распределения первых z требований, улучшить которую при назначении оставшихся $(m-z)$ требований не представляется возможным, т.е.

$$\max_{N_j^z \subset N_j} \{ T_j \} \geq T_{zj}.$$

Вершина, соответствующая варианту с наименьшей оценкой

$$T_{zp} = \min_{1 \leq j \leq n} T_{zj}, \quad (4)$$

выбирается в качестве активной для дальнейшего ветвления. Остальные вершины данного уровня – концевые. При наличии нескольких вершин, отвечающих условию (4), выбирается любая из них, если не установлено правило предпочтения (например, по величине t_{ij}).

Процесс продолжается до тех пор, пока дальнейшее ветвление становится невозможным. Решение оптимально, если дерево вариантов не имеет концевых вершин с оценками

$$T_{ij} < T^* \text{ при } i \in N_j, j = 1, \dots, n, \quad (5)$$

где $T^* = \max_{1 \leq j \leq n} T_j$ – значение целевой функции полученного решения.

В противном случае производится проверка и уточнение решения путем ветвления из вершин, отвечающих неравенству (5). Проверку целесообразно начинать с нижних уровней, так как при этом может быть достаточно быстро достигнуто улучшение решения, что, в свою очередь, сократит число вариантов верхних уровней, подлежащих проверке. Ветвление из проверяемой вершины прекращается, если на каком-либо уровне оценка нижней границы достигнет или превысит величину T^* . В случае получения нового решения для проверки используется соответствующее значение целевой функции.

Процедура проверки может быть ускорена, если оценки проверяе-

мых вершин уточнять с учетом дополнительного ограничения

$$\tau_i \neq t_{ij}, \text{ при } t_{ij} \geq T^* - T_j^z, i=z+1, z+2, \dots, m, j=1, \dots, n. \quad (6)$$

При этом в качестве τ_i потребуется выбирать не минимальный, а один из следующих по величине элементов t_{ij} , что может привести к увеличению T^z и оценки. Если в какой-либо строке матрицы $[t_{ij}]$ все $t_{ij} \geq T^* - T_j^z$, то можно, не производя ветвления, утверждать, что при данном распределении z требований нижняя граница решения обязательно достигнет или превысит T^* . Таким образом, введение условия (6) позволяет сократить число проверяемых вариантов. Заметим, что в зависимости от структуры исходной матрицы в ряде случаев достигается равенство $T^* = T_0$, где T_0 – начальная нижняя граница решения, рассчитываемая по формуле

$$T_0 = \max \left\{ \frac{1}{n} \sum_{i=1}^m \tau_i; \max_{1 \leq i \leq m} \tau_i \right\}. \text{ Поэтому при организации вычислительной}$$

процедуры целесообразно проверять это условие прежде, чем просматривать концевые вершины.

Модификация алгоритма Алексева для случая идентичных приборов

Для увеличения эффективности данного алгоритма в случае идентичности приборов необходимо внести следующие изменения. Исходная матрица $[t_{ij}]$ длительностей обслуживания требований может быть заменена вектором $[\tau_i]$, где $\tau_i = t_{i1}$, $i = 1, 2, \dots, m$, так как для однородных приборов выполняется равенство $t_{i1} = t_{i2} = \dots = t_{in}$, $i = 1, 2, \dots, m$. Для упрощения оценки (2) нижней границы варианта распределения следует перед началом выполнения процедуры ветвления упорядочить элементы τ_i таким образом, чтобы соблюдалось неравенство

$$\tau_{i_1} \geq \tau_{i_2} \geq \dots \geq \tau_{i_k} \geq \tau_{i_{k+1}} \geq \dots \geq \tau_{i_m}, \quad (7)$$

и, следовательно, распределение требований начиналось с имеющих наибольшее время обслуживания. Упорядочивание позволяет свести оценку T_{zj} к виду

$$T_{zj} = \max \left\{ T^z; T_{j_n}^z \right\}. \quad (8)$$

Здесь

$$T^z = \begin{cases} \frac{1}{n} \sum_{k=1}^m \tau_{i_k}, & \text{если } z \leq m - n, \\ \frac{1}{m - z} \left(\sum_{s=1}^{m-z} T_{j_s}^z + \sum_{k=z+1}^m \tau_{i_k} \right), & \text{если } z > m - n, \end{cases}$$

где $T_{j_1}^z \leq T_{j_2}^z \leq \dots \leq T_{j_s}^z \leq T_{j_{s+1}}^z \leq \dots \leq T_{j_n}^z$ – последовательность частичных

загрузок приборов; $T_j^z = \sum_{i_k \in N_j^z} \tau_{i_k}, j=1, \dots, n.$

В оценке (8) лишней оказалось значение $\max_{z+1 \leq k \leq m} \tau_{i_k}$, так как для любого $z \in 1, 2, \dots, m$ выполняется неравенство

$$T_{j_n}^z \geq \max_{z+1 \leq k \leq m} \tau_{i_k},$$

потому что, максимальная по величине загрузка одного из приборов $T_{j_n}^z$ будет включать в себя длительность выполнения τ_{i_r} хотя бы одного из требований, такого, что $r \leq z$ и согласно неравенству (7)

$$\tau_{i_r} \geq \max_{z+1 \leq k \leq m} \tau_{i_k}.$$

Также в оценке (8) значительно упрощается вычисление величины T^z при $z \leq m-n$. В этом случае

$$T^z = \frac{1}{n} \sum_{k=1}^m \tau_{i_k},$$

так как для любого $z \in 1, \dots, m$ с учетом однородности приборов справедливо соотношение

$$\left(\sum_{s=1}^n T_{j_s}^z + \sum_{k=z+1}^m \tau_{i_k} \right) = \left(\sum_{k=1}^z \tau_{i_k} + \sum_{k=z+1}^m \tau_{i_k} \right) = \sum_{k=1}^m \tau_{i_k}.$$

Таким образом, нахождение оценки (8) потребует существенно меньших вычислительных затрат по сравнению с первоначальной оценкой (2), что ведет к снижению трудоемкости упрощенного варианта алгоритма Алексева.

Кроме изменений в оценке нижней границы варианта распределения необходимо ввести уточнение в применение дополнительного ограничения (6). В силу однородности приборов, а также ввиду неравенства (7) это ограничение следует использовать только для того, чтобы проверить справедливость неравенства

$$\tau_{i_k} \geq T^* - T_j^z, \text{ для } k=z+1 \text{ и } \forall j \in 1, \dots, n, \quad (9)$$

что позволяет утверждать, что при данном распределении z требований нижняя граница решения обязательно достигнет или превысит T^* . В отличие от ограничения (6), которое используется как для уточнения величины τ_i , так и для оценки перспективности дальнейшего ветвления по данной вершине, ограничение (9) применяется только для проверки перспективности ветвления (уточнять значение τ_i не имеет смысла). Следует отметить, что условие (9) проверяется только для $k=z+1$, так как если это условие не выполняется при $k=z+1$, то для $k \in z+2, z+3, \dots, m$ оно также будет неверным вследствие того, что значения τ_{i_k} упорядочены в соответствии с неравен-

ством (7).

Также однородность приборов позволяет существенно сократить количество оцениваемых вершин, если учесть, что всего возможных вариантов распределения требований по приборам n^m , а при этом имеет смысл рассматривать только n^{m-1} вариантов. Сокращение вариантов обусловлено тем, что требование, распределяемое первым, можно назначить, например, на первый прибор и далее найти решение (распределить оставшиеся требования по приборам), которое в дальнейшем нельзя улучшить, если назначить это требование на другой из приборов. Поэтому если после обхода вершин дерева вариантов распределения и возврата назад к конечным вершинам, удовлетворяющим условию (5), текущим оказывается первый уровень ветвления ($z=1$), то конечные вершины на этом уровне далее не просматриваются и найденное до этого распределение является окончательным решением.

В качестве начальной нижней границы решения T_0 можно использовать оценку

$$T_0 = \max \left\{ \frac{1}{n} \sum_{k=1}^m \tau_{i_k}; \tau_{i_1} \right\}. \quad (10)$$

Для сокращения количества проверяемых вариантов следует перед началом ветвления определить T^* в виде $T^* = T^p$, где T^p – длина расписания, найденного при помощи некоторого приближенного алгоритма (можно использовать для этого $T^* = \infty$).

Рассмотрим основные шаги вычислительного алгоритма.

1. Формируем вектор $[\tau_i]$ и упорядочиваем его в соответствии с (7).
2. По формуле (10) определяем начальную нижнюю границу T_0 ; $z=0$.
3. Увеличиваем значение z на единицу. Проверяем условие $z \leq m$. Если условие выполняется, переходим к п.4, в противном случае – к п.6.
4. Формируем вершины z -го уровня и по формуле (8) с учетом (9) оцениваем границы вариантов решения (первоначально полагаем $T^* = \infty$ или $T^* = T^p$). Выявляем вершину с наименьшей оценкой (4), проверяем для нее условие (5). Если оно выполнено, переходим к п.5, если нет – к п.9.
5. Включаем выбранную вершину в решение и возвращаемся к п.3.
6. Запоминаем решение $\{N_j\}$ ($j=1, \dots, n$) и соответствующее значение T^* целевой функции.
7. Проверяем условие $T^* = T_0$. Если оно выполняется, переходим к п.11, в противном случае – к п.8.
8. Исключаем из решения вершину m -го уровня ветвления,
9. Уменьшаем значение z на единицу. Проверяем условие $z = 1$. Если условие выполнено, переходим к п.11, в противном случае – к п.10.
10. Исключаем из решения вершину z -го уровня. Полагаем соответствующий ей элемент равным ∞ . Восстанавливаем элементы $(z+1)$ -й стро-

ки матрицы $[t_{ij}]$ и возвращаемся к п.4.

11. Выводим результаты решения и заканчиваем процесс.

Пример работы модифицированного алгоритма Алексева

Проиллюстрируем работу алгоритма. Пусть имеются $n=3$ идентичных прибора и $m=8$ независимых требования. Возможности системы определяются матрицей производительности $[t_{ij}]$ ($i = 1, \dots, m; j = 1, \dots, n$), которая преобразуется в вектор $[\tau_i]$ ($i = 1, \dots, m$). Данный вектор (табл. 1) для простоты примера уже упорядочен в соответствии с (7). Также перед началом поиска решения следует получить вектор накопленных сумм

$$\sum_{k=i_k}^m \tau_{i_k},$$

использование которого значительно упрощает вычисление оценки (8).

Таблица 1

i_k	1	2	3	4	5	6	7	8
τ_{i_k}	16	9	9	7	6	5	4	4
$\sum_{k=i_k}^m \tau_{i_k}$	60	44	35	26	19	13	8	4

Определим начальную нижнюю границу по формуле (10)

$$T_0 = \max \{60/3; 16\} = 20.$$

Сформируем вершины первого уровня ветвления ($z = 1; z \leq m-n = 5$).

Получим три варианта загрузки системы (табл. 2). Оценим их по формуле (8).

$$T_{11} = \max \{60/3; 16\} = 20, T_{12} = \max \{60/3; 16\} = 20,$$

$$T_{13} = \max \{60/3; 16\} = 20.$$

Полагаем активной первую вершину 11 и переходим к формированию второго уровня при $z = 2$ ($z \leq 5$).

Оценки нижних границ для полученных вариантов следующие:

$$T_{21} = \max \{60/3; 25\} = 25, T_{22} = \max \{60/3; 16\} = 20,$$

$$T_{23} = \max \{60/3; 16\} = 20.$$

В качестве активной вершины выбирается вершина 22.

Определим оценки нижних границ третьего уровня ветвления ($z = 3, z \leq 5$):

$$T_{31} = \max \{60/3; 25\} = 25, T_{32} = \max \{60/3; 18\} = 20,$$

$$T_{33} = \max \{60/3; 16\} = 20.$$

В качестве активной вершины выбирается вершина 32.

Далее определим оценки границ 4-го уровня ветвления ($z = 4; z \leq 5$):

$$T_{41} = \max \{60/3; 23\} = 23, T_{42} = \max \{60/3; 25\} = 25,$$

$$T_{43} = \max \{60/3; 18\} = 20.$$

Активная вершина – вершина 43.

Оценки нижних границ 5-го уровня ветвления ($z = 5; z \leq 5$):

$$T_{51} = \max \{60/3; 22\} = 22, T_{52} = \max \{60/3; 24\} = 24,$$

$$T_{53} = \max \{60/3; 18\} = 20.$$

В качестве активной вершины выбирается вершина 53.

Таким же образом, последовательно определяем оценки 6-го, 7-го и 8-го уровней ветвления (табл. 2).

Таблица 2

Уровень ветвления z	z_j	N_1^z	N_2^z	N_3^z	T_1^z	T_2^z	T_3^z	T_{zj}
0	0	\emptyset	\emptyset	\emptyset	0	0	0	20
1	11	1	\emptyset	\emptyset	16	0	0	20
	12	\emptyset	1	\emptyset	0	16	0	20
	13	\emptyset	\emptyset	1	0	0	16	20
2	21	1; 2	\emptyset	\emptyset	25	0	0	25
	22	1	2	\emptyset	16	9	0	20
	23	1	\emptyset	2	16	0	9	20
3	31	1; 3	2	\emptyset	25	9	0	25
	32	1	2; 3	\emptyset	16	18	0	20
	33	1	2	3	16	9	9	20
4	41	1; 4	2; 3	\emptyset	23	18	0	23
	42	1	2; 3; 4	\emptyset	16	25	0	25
	43	1	2; 3	4	16	18	7	20
5	51	1; 5	2; 3	4	22	18	7	22
	52	1	2; 3; 5	4	16	24	7	24
	53	1	2; 3	4; 5	16	18	13	20
6	61	1; 6	2; 3	4; 5	21	18	13	21
	62	1	2; 3; 6	4; 5	16	23	13	23
	63	1	2; 3	4; 5; 6	16	18	18	21
7	71	1; 6; 7	2; 3	4; 5	25	18	13	25
	72	1; 6;	2; 3; 7	4; 5	21	22	13	22
	73	1; 6;	2; 3	4; 5; 7	21	18	17	21
8	81	1; 6; 8	2; 3	4; 5; 7	25	18	17	25
	82	1; 6	2; 3; 8	4; 5; 7	21	22	17	22
	83	1; 6	2; 3	4; 5; 7; 8	21	18	21	21

Дальнейшее ветвление приводит к решению $N_1 = \{1; 6\}$, $N_2 = \{2; 3\}$, $N_3 = \{4; 5; 7; 8\}$ со значением целевой функции $T^* = 21$.

Следует попытаться улучшить полученное решение сначала ветвлением из конечной вершины 33 с оценкой $T_{33} = 20$ ($T_{33} < T^*$).

Для этой вершины ограничение (9) не выполняется, поэтому следует продолжить ветвление (табл. 3), предварительно сделав ее активной.

Таблица 3

Уровень ветвления z	z_j	N_1^z	N_2^z	N_3^z	T_1^z	T_2^z	T_3^z	T_{zj}
4	41	1; 4	2	3	23	9	9	23
	42	1	2; 4	3	16	16	9	20
	43	1	2	3; 4	16	9	16	20
5	51	1; 5	2; 4	3	22	16	9	22
	52	1	2; 4; 5	3	16	22	9	22
	53	1	2; 4	3; 5	16	16	15	20
6	61	1; 6	2; 4	3; 5	21	18	13	21
	62	1	2; 4; 6	3; 5	16	23	13	21
	63	1	2; 4	3; 5; 6	16	18	18	20
7	71	1; 7	2; 4	3; 5; 6	20	16	20	20
	72	1	2; 4; 7	3; 5; 6	16	20	20	20
	73	1	2; 4	3; 5; 6; 7	16	16	24	24
8	81	1; 7; 8	2; 4	3; 5; 6	24	16	20	24
	82	1; 7	2; 4; 8	3; 5; 6	20	20	20	20
	83	1; 7	2; 4	3; 5; 6; 8	20	16	24	24

Определим оценки нижних границ 4-го уровня ветвления ($z = 4$; $z \leq 5$) с учетом ограничения (9):

$$T_{41} = \max \{60/3; 23\} = 23, T_{42} = \max \{60/3; 16\} = 20,$$

$$T_{43} = \max \{60/3; 16\} = 20.$$

В качестве активной вершины выбирается вершина 42.

Оценки нижних границ 5-го уровня ветвления ($z = 5$; $z \leq 5$):

$$T_{51} = \max \{60/3; 22\} = 22, T_{52} = \max \{60/3; 22\} = 22,$$

$$T_{53} = \max \{60/3; 16\} = 20.$$

Активная вершина – вершина 53.

Дальнейшее ветвление (табл. 3) приводит к решению $N_1 = \{1; 7\}$, $N_2 = \{2; 4; 8\}$, $N_3 = \{3; 5; 6\}$ со значением $T^* = 20$. Таким образом, найденное решение совпало с нижней границей $T^* = T_0$. Лучше, чем это, решения уже не получить.

Процесс поиска оптимального распределения в виде дерева показан на рис. 2. Приведенный пример содержит 6561 вариант распределения ($n^m = 3^8 = 6561$). С помощью предложенного алгоритма оптимальное решение найдено после просмотра 39 вариантов. Решение задач показало, что этот эффект может быть более ощутимым для матриц больших размеров.

Сравнительный анализ эффективности алгоритмов решения

С целью проведения сравнительного анализа эффективности модифицированного и исходного алгоритма Алексева, а также алгоритма Романовского было реализовано несколько серий машинных экспериментов при различных начальных условиях.

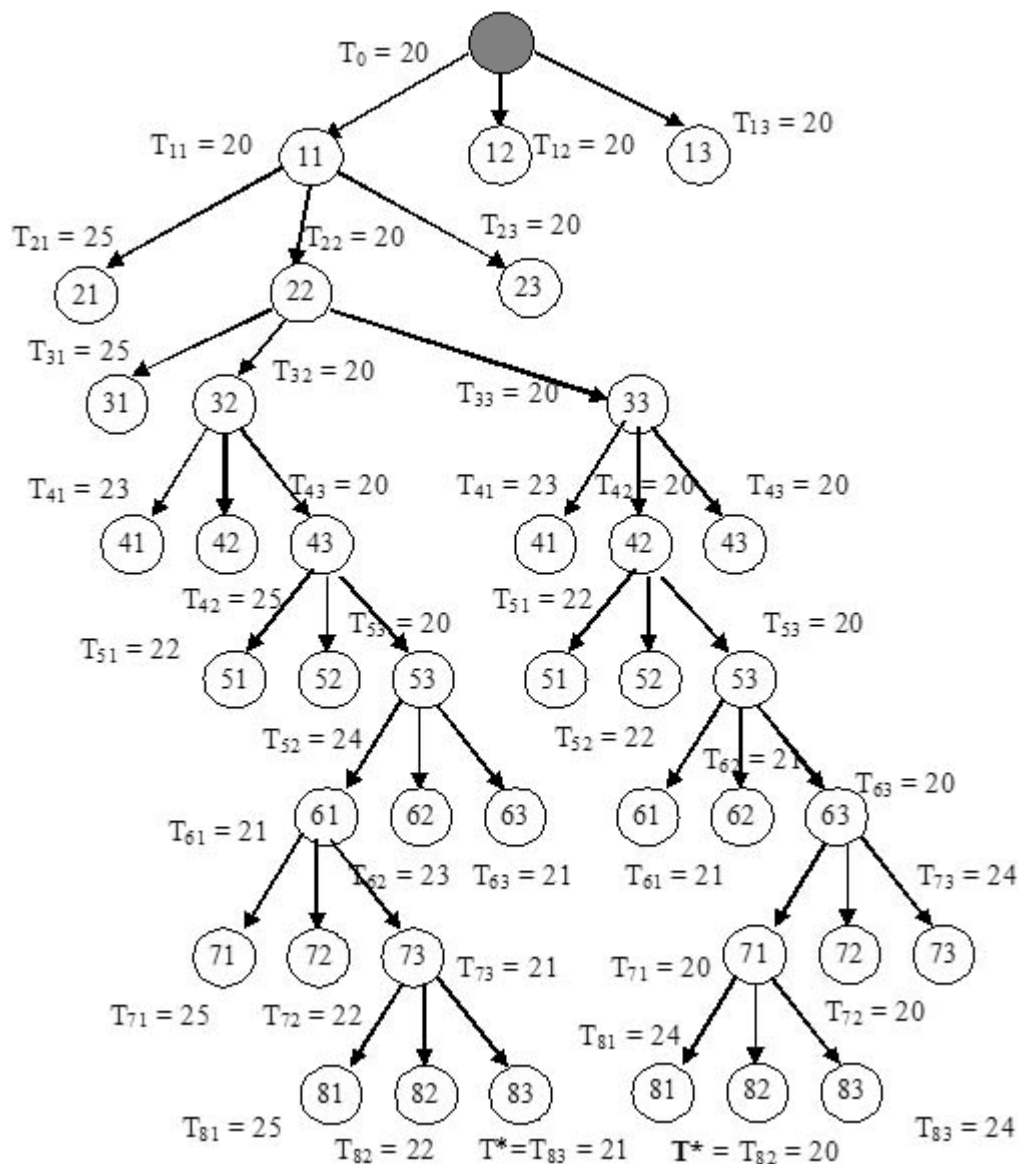


Рис. 2. Дерево вариантов решения задачи.

При этом исследовались в основном двух- ($n = 2$) и трехприборные ($n=3$) системы. Количество требований в системе варьировалось в диапазоне $m=10\div 25$. В первой серии экспериментов, в которой задача решалась как исходным, так и модифицированным алгоритмом Алексева, значения в матрице $[t_{ij}]$ являются случайными числами, равномерно распределенными на интервале от 20 до 30. Во второй серии экспериментов, в которой задача решалась модифицированным алгоритмом Алексева, а также алгоритмом Романовского, значения в матрице $[t_{ij}]$ являются случайными числами, распределенными по нормальному закону со средним 20 и дисперсией 8. Машинный эксперимент заключался в генерации 100 матриц $[t_{ij}]$ размерности $n \times m$ с последующим решением минимаксной задачи для заданных условий соответствующими алгоритмами. При этом определялись следующие временные характеристики: t_{cp} – среднее время решения (в мс) t_{max} – максимальное время решения (в мс); $S(t)$ – выборочное среднее квадратичное отклонение.

ратическое отклонение времени решения (в мс). Результаты экспериментов приведены в табл. 4 и 5.

Таблица 4

$n \times m$	Исходный алгоритм Алексева			Модиф. алгоритм Алексева		
	t_{cp}	$S(t)$	t_{max}	t_{cp}	$S(t)$	t_{max}
2x10	0,95	0,44	2,92	0,83	0,19	1,62
2x11	1,37	0,88	8,12	0,97	0,34	3,85
2x12	0,96	0,54	3,46	0,81	0,22	2,47
2x13	1,45	0,71	5,48	1,01	0,27	2,37
2x14	1,00	0,58	3,95	0,84	0,26	2,82
2x15	1,62	0,73	4,30	1,08	0,34	2,13
2x16	0,98	0,68	6,26	0,89	0,27	2,61
2x17	1,58	0,60	4,53	1,01	0,17	1,66
2x18	0,98	0,54	3,08	0,90	0,35	2,50
2x19	1,88	0,93	5,32	1,14	0,28	2,10
2x20	1,04	0,83	6,73	0,86	0,22	2,34
2x21	1,96	0,91	6,85	1,24	0,51	3,23
2x22	1,09	0,97	7,15	0,93	0,50	3,92
2x23	2,20	1,10	8,80	1,14	0,29	2,12
2x24	1,06	0,72	4,38	0,88	0,20	1,61
2x25	2,25	0,99	5,86	1,32	0,64	4,89
3x10	8,18	8,47	36,57	2,48	2,70	19,53
3x11	14,06	17,75	71,74	3,55	4,53	20,49
3x12	12,17	27,00	146,87	3,61	6,04	35,90
3x13	19,46	34,97	243,95	6,28	9,51	79,26
3x14	29,02	88,23	808,15	7,88	15,06	134,50
3x15	14,65	24,84	123,95	4,32	8,92	67,33
3x16	27,49	43,37	234,78	14,09	25,89	130,79
3x17	39,97	56,45	350,14	23,28	30,59	211,80
3x18	24,68	47,13	270,42	14,34	34,00	273,43
3x19	85,60	145,32	798,59	43,16	94,01	586,50
3x20	124,00	221,80	1842,45	61,76	136,52	743,13
3x21	66,71	207,95	1535,11	52,38	165,42	1229,42
3x22	280,86	499,86	3824,42	212,65	364,91	2796,84
3x23	409,48	830,50	6136,34	268,15	434,28	2870,34
3x24	357,95	1480,76	11668,22	118,47	528,20	4044,16
3x25	1143,30	2118,00	14968,47	1132,45	2255,91	14307,47

По результатам экспериментов можно сделать следующие выводы.

1. Рассмотренный алгоритм Алексева (исходный и модифицированный), как и алгоритм Романовского, имеет экспоненциальную сложность (как в худшем, так и в среднем случае).

2. Предложенный вариант упрощения алгоритма Алексева имеет преимущество при решении минимаксной задачи по сравнению с исходным вариантом алгоритма Алексева для всех рассмотренных в эксперименте случаев. Причем оно более существенно для более трудоемких задач (например, для задач при $n=3$).

3. Алгоритм Романовского оказывается в среднем более эффективным, чем алгоритм Алексева (как исходный, так и модифицированный), но при неблагоприятных условиях предложенная модификация алгоритма Алексева дает существенный выигрыш во времени решения (в десятки раз).

Таблица 5

$n \times m$	Алгоритм Романовского			Модиф. алгоритм Алексева		
	t_{cp}	$S(t)$	t_{max}	t_{cp}	$S(t)$	t_{max}
2x10	0,56	0,23	1,34	0,86	0,22	1,62
2x11	1,12	0,49	1,96	1,69	1,34	4,11
2x12	0,58	0,22	0,95	0,99	0,49	2,67
2x13	2,39	5,55	22,99	1,55	2,52	10,83
2x14	0,56	0,24	0,97	1,17	0,89	4,02
2x15	0,96	0,15	1,74	1,05	0,29	2,13
2x16	0,80	1,38	13,77	2,09	3,82	27,27
2x17	131,41	1303,91	13040,07	2,82	14,77	148,96
2x18	0,73	0,79	6,73	1,85	3,84	21,53
2x19	1,10	0,15	1,86	1,96	1,21	8,74
2x20	0,68	0,69	5,44	1,76	4,41	28,72
2x21	1,08	0,17	1,61	2,96	2,25	12,16
2x22	0,55	0,21	0,84	0,82	0,12	1,37
2x23	1,17	0,32	2,85	6,47	6,43	41,25
2x24	1,91	13,22	132,78	2,67	16,54	165,71
2x25	2,44	6,76	48,95	13,59	24,15	162,06
3x10	1,15	0,23	2,03	13,90	4,93	17,64
3x11	0,94	0,12	1,61	14,96	5,38	19,89
3x12	0,71	0,26	2,76	6,06	9,89	62,34
3x13	23,57	34,25	118,88	148,34	152,70	433,36
3x14	7,04	11,96	69,90	174,32	171,00	371,30
3x15	0,66	0,21	1,78	21,33	46,07	349,23
3x16	835,60	3240,72	23850,12	753,47	1830,55	7164,20
3x17	376,12	1205,25	6713,32	907,66	2461,20	8173,14
3x18	0,72	0,17	1,18	322,75	700,04	3258,81
3x19	72719,91	514947,22	4485783,73	3119,08	17026,20	132868,47
3x20	130150,96	725061,61	5869087,73	6866,07	33515,71	179876,37
3x21	0,39	0,18	1,32	610,26	2547,05	16251,08
3x22	78752,41	491005,92	4216359,06	5593,50	15763,95	111775,32
3x23	1,21	0,63	3,75	576,22	1539,62	12831,57
3x24	499,14	3947,56	37670,63	4056,02	14667,83	77771,79
3x25	17759,96	115950,05	1094554,95	20796,76	32347,22	120235,97

ЛИТЕРАТУРА

1. Кофман Э. Г. Теория расписания и вычислительные машины. М.: Наука, 1987.
2. Романовский И. В. Алгоритмы решения экстремальных задач. М.: Наука, 1977.
3. Алексева О. Г. Комплексное применение методов дискретной оптимизации. М.: Наука, 1987.

Статья представлена к публикации членом редколлегии Е.А. Ереминым.