Б.А. Есипов, В.В. Муравьев

РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ РЕШЕНИЯ ОБОБЩЕННОЙ ЗАДАЧИ О МИНИМАЛЬНОМ ПОКРЫТИИ

(Самарский государственный аэрокосмический университет им. академика С.П. Королёва (национальный исследовательский университет))

При постановке задач наиболее эффективного распределения ресурсов по отдельным объектам, а также в задачах управления организационно-техническими системами возникает так называемая задача об оптимальном покрытии. В [1] рассмотрена математическая модель обобщенной задачи о минимальном покрытии, решение которой приводит к сложным переборным алгоритмам [2]. В данной работе рассмотрен подход к разработке алгоритмов решения этой задачи, запрограммировано несколько алгоритмов ее решения и сделан анализ их быстродействия для задач большой размерности.

Постановка обобщенной задачи о покрытии отличается от классической наличием нескольких независимых покрываемых множеств и заключается в следующем.

Пусть даны множество $K=(1,...,k)$ и два набора его подмножеств $A_1,...,A_n$ и $B_1,...,B_m$, таких, что $\bigcup_{i=1}^{m} B_i \subseteq \bigcup_{j=1}^{n} A_j$. Совокупность подмножеств $A_j, j \in J_i \subseteq \{1,...,n\}$, называется покрытием множества B_i, если $\bigcup_{j \in J_i} A_j = B_i$; J_i в таком случае называется множеством покрывающих индексов. Каждой паре подмножеств (B_i, A_j) приписан вес $c_{ij} \geq 0$. Весом покрытия A_j множества B_i является $\sum_{j \in J_i} c_{ij}$. Требуется найти покрытия для всех множеств из набора $B_1,..,B_n$ минимального суммарного веса такое, что множества покрывающих индексов данных покрытий J_i попарно не пересекаются. В терминах булева программирования эта задача формулируется следующим образом: найти

$$f^*(c,A,B) = \min\{\sum_{j=1}^{n}\sum_{i=1}^{m} c_{ji}x_{ji} \mid Ax \geq B, x_{ji} \in \{0,1\}\}, \quad (1)$$

при условии $\sum_{i=1}^{m} x_{ji} \leq 1, \quad j=\overline{1,n}$.

Здесь $A=(a_{kj})-$матрица размера $k \times n$ с элементами $a_{kj}=1$, если $k \in A_j$, и $a_{kj}=0$ в противном случае; $B=(b_{ki})-$матрица размера $k \times m$ с элементами $b_{ki}=1$, если $k \in B_i$, и $b_{ki}=0$ в противном случае; $c=(c_{ji})$ - матрица весов; $x=(x_{ji})$ - матрица переменных с компонентами $x_{ji}=1$, если A_j входит в покрытие B_i, и $x_{ji}=0$ в противном случае.

В данной работе предлагается переход от булевых к целочисленным переменным:

$$f^*(c,A,B)=\min\{\sum_{j=1}^{n} c(x_j,j)\,|\,Ah\geq B, x\in\{0,...,m\}^n\} \quad (2)$$

Здесь $A=(a_{kj})$ – матрица размера $k\times n$ с элементами $a_{kj}=1$, если $k\in A_j$, и $a_{kj}=0$ в противном случае; $B=(b_{ki})$ – матрица размера $k\times m$ с элементами $b_{ki}=1$, если $k\in B_i$, и $b_{ki}=0$ в противном случае; $c(j,i)=\begin{cases} c_{ji}, & i\neq 0 \\ 0, & i=0 \end{cases}$ - весовая функция, (c_{ji}) - матрица весов; $x=(x_j)$ - вектор целочисленных переменных с компонентами $x_j=i$, если A_j входит в покрытие B_i, и $x_j=0$, если A_j не входит ни в одно покрытие; $h=(h_{ji})$ - дополнительная матрица, $h_{ji}=1$, если $x_j=i$, и $h_{ji}=0$ в противном случае.

Задача в такой формулировке не имеет недопустимых состояний, так как ограничение (1) выполняется в данном случае автоматически. Такая модификация позволяет не делать лишних проверок и переходов в процессе решения, что положительно сказывается на времени поиска.

При проверке ограничения $Ah\geq B$ в матрице произведения $D=Ah$ имеет смысл вычислять элементы d_{ki} только с такими индексами (k,i), что b_{ki}=1.

Рассмотренная задача относится к классу NP-полных задач, в общем случае для нее существует лишь экспоненциальное переборное решение.

Для разработки алгоритма и программы ее решения можно применить аддитивный метод, генетический алгоритм и алгоритм муравьиной колонии, методы лагранжевой релаксации и другие [2]. В работе рассмотрены аддитивный и генетический алгоритмы на основе вышеописанной модификации.

Для реализации работы и исследования аддитивного и генетического алгоритмов была написана программа на языке Java 7. Входными данными для программы являются величины: m, n, k, матрицы a_{kj}, b_{ki} и c_{ji}. Выходными данными программы является итоговая стоимость покрытия $\sum_{j=1}^{n} c(x_j,j)$ и множества покрывающих индексов J_i, $i=\overline{1,m}$

Ниже приведены результаты тестирования, выполненные на компьютере Asus, Core i5-3210M 2.5 GHz, 4 GB RAM.

Для всех тестов величина k была постоянной и равнялось 10; матрицы весов заполнялись случайными числами в промежутке [0;1000], матрицы a_{kj}, b_{ki} заполнялись также случайно.

Для данного тестирования было выбрано несколько промежутков количества возможных состояний решения. Каждый порядок был в 10 раз больше предыдущего. Для каждого промежутка было сгенерировано множество входных данных, на которых запускались оба алгоритма, после чего сравнивалось время их работы. Для генетического алгоритма дополнительно считалось относительное отклонение ответа от точного, данного аддитивным, а также процент неудачных попыток найти решение.

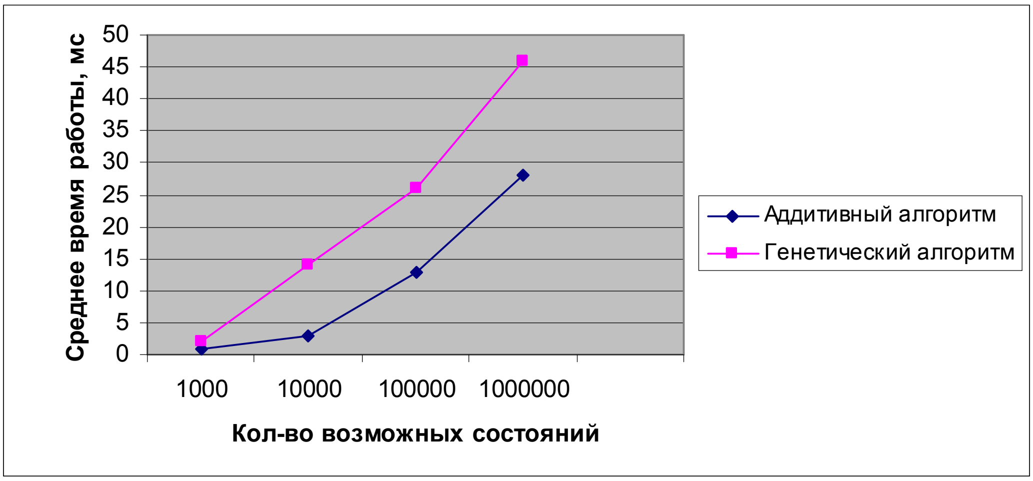


Рис. 1. Зависимость среднего времени работы алгоритмов от кол-ва возможных состояний решения

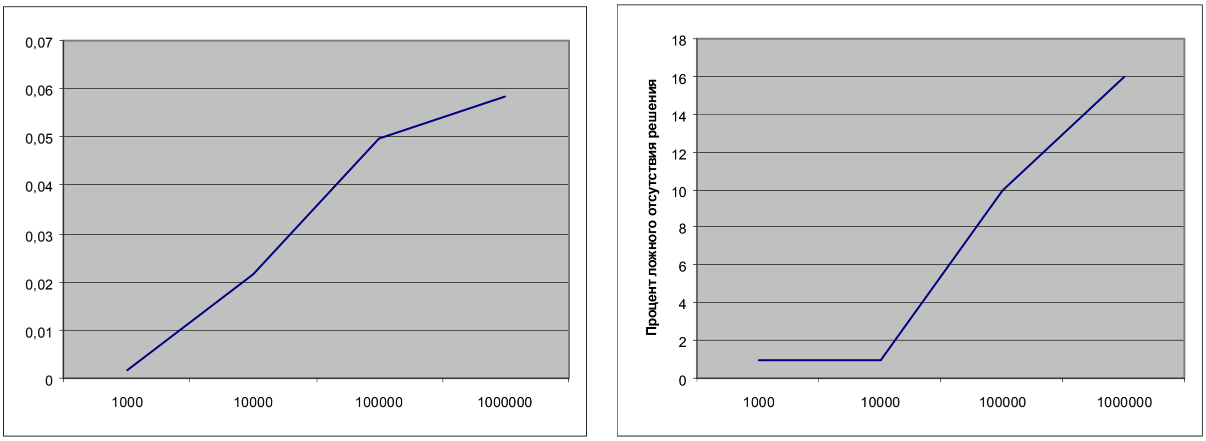


Рис. 2-3. Зависимость среднего относительного отклонения ответа и кол-ва ложного «отсутствия решения» генетического алгоритма от кол-ва возможных состояний решения

Как и следовало ожидать, аддитивный алгоритм находит ответ быстрее, чем генетический на тех же данных. С другой стороны, аддитивный алгоритм более затратен в плане используемой памяти, что не позволяет использовать его при решении задач с большим количеством состояний. Недостатком генетического алгоритма является его рандомизированность. Из рис.2 и 3 видно, что с ростом количества состояний решения увеличиваются относительное отклонение результата от минимального и количество ложных «отсутствий решения» задачи.

Для сравнения времени работы генетического алгоритма на различных входных данных было выбрано несколько промежутков количества возможных состояний решения, каждый порядок был в 10 раз больше предыдущего и для каждого промежутка было сгенерировано множество входных данных, на которых запускался алгоритм. Здесь количество возможных состояний было на порядки больше, чем при тестировании аддитивного алгоритма.

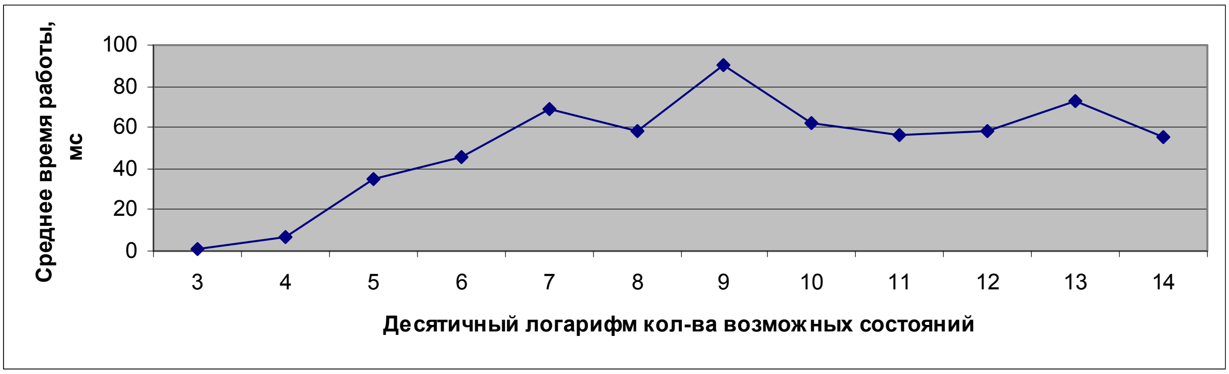
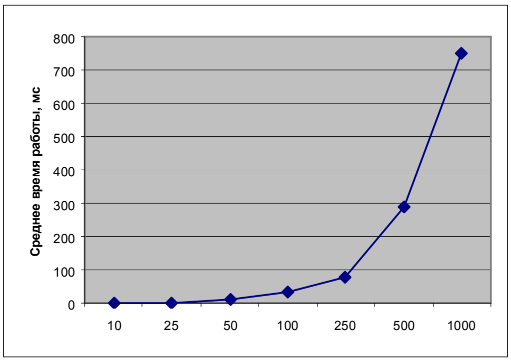


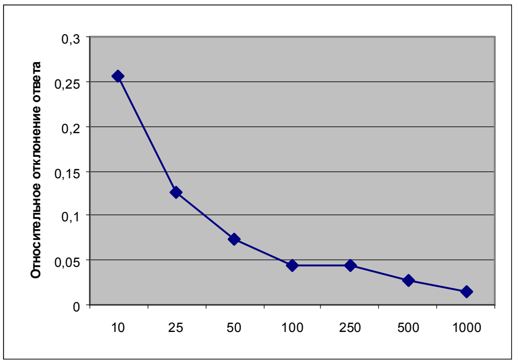
Рис. 4. Среднее время работы генетического алгоритма

Из рис. 4 видно, что, в среднем, при росте порядка количества возможных состояний решения время работы генетического алгоритма достигает определенного диапазона, в пределах которого изменяется. Это можно объяснить тем, что при большом порядке количества состояний алгоритм или быстрее начинает сходиться к правильному ответу при $n << m$, или быстрее определяет отсутствие решения при $n \approx m$.

Для анализа возможностей генетического алгоритма были выбраны несколько различных размеров популяции, начиная с 10 и заканчивая 1000. Во всех тестах количество возможных состояний находилось в промежутке $[10^5;10^6]$, для каждого теста искалось контрольное решение с помощью аддитивного алгоритма и решения для каждого размера популяции.

На рис. 5 – 7 показаны графики зависимости различных характеристик работы генетического алгоритма от размера популяции. Видно, что размер популяции порядка 100-250 является оптимальным – меньшие размеры дают слишком неточные результаты, большие – требуют намного больше времени для нахождения результата.





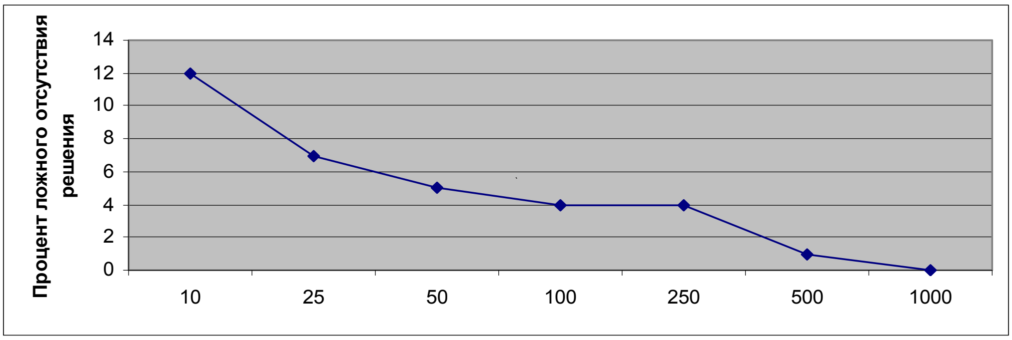


Рис.5-7. Характеристики генетического алгоритма

Таблица 1. Сравнение показателей работы аддитивного и генетического алгоритмов

Кол-во возможных состояний	Среднее время работы, мс		Отн.откл. результатов генетического алгоритма от контрольного решения	% неудачных попыток поиска решения генетическим алгоритмом
	Аддитивный алгоритм	Генетический алгоритм		
$[10^2;10^3)$	1	2	$1{,}69\cdot10^{-3}$	1
$[10^3;10^4)$	3	14	$2{,}14\cdot10^{-2}$	1
$[10^4;10^5)$	13	26	$4{,}98\cdot10^{-2}$	10
$[10^5;10^6)$	28	46	$5{,}86\cdot10^{-2}$	16

Таблица 2. Среднее время работы генетического алгоритма при различном размере входных данных.

Кол-во возможных состояний	Среднее время работы, мс	Кол-во возможных состояний	Среднее время работы, мс
$[10^2;10^3)$	1	$[10^8;10^9)$	$[10^8;10^9)$
$[10^3;10^4)$	7	$[10^9;10^{10})$	$[10^9;10^{10})$
$[10^4;10^5)$	35	$[10^{10};10^{11})$	$[10^{10};10^{11})$
$[10^5;10^6)$	46	$[10^{11};10^{12})$	$[10^{11};10^{12})$
$[10^6;10^7)$	69	$[10^{12};10^{13})$	$[10^{12};10^{13})$
$[10^7;10^8)$	58	$[10^{13};10^{14})$	$[10^{13};10^{14})$

Таблица 3. Анализ работы генетического алгоритма при различном размере популяции и одинаковом порядке количества возможных состояний перебора

Размер популяции	Среднее время работы, мс	Относительное отклонение результатов генетического алгоритма от оптимального решения	% неудачных попыток поиска решения генетическим алгоритмом
10	0	$2{,}55\cdot10^{-1}$	12
25	2	$1{,}25\cdot10^{-1}$	7
50	10	$7{,}31\cdot10^{-2}$	5
100	33	$4{,}50\cdot10^{-2}$	4
250	80	$4{,}50\cdot10^{-2}$	4
500	289	$2{,}81\cdot10^{-2}$	1
1000	752	$1{,}50\cdot10^{-2}$	0

Литература

1. Есипов, Б.А. Математическая модель и решение обобщенной задачи о покрытии [Текст]/ Есипов Б.А: - Избранные труды Международной конферен-

ции с элементами научной школы для молодежи «Перспективные информационные технологии для авиации и космоса» (ПИТ-2010), Самара, 2010, с.70-71.

2. Еремеев, А.В. Задача о покрытии множества: сложность, алгоритмы, экспериментальные исследования [Текст]/ Еремеев А.В., Заозерская Л.А., Колоколов А.А: – Дискретный анализ и исследование операций, июль—декабрь 2000. Серия 2. Том 7, № 2, с.22-46.

3. Ху, Т. Целочисленное программирование и потоки в сетях [Текст]/ Ху Т. - М.: Мир, 1974, - 520с.

4. Нгуен, Минь Ханг. Применение генетического алгоритма для задачи нахождения покрытия множества [Текст]/ Нгуен Минь Ханг:- Труды института системного анализа РАН, Изд.: Институт системного анализа РАН, Москва, 2008, том 33, с. 206-219.

А.И. Заико

ХАРАКТЕРИСТИКИ ЭРГОДИЧЕСКИХ СЛУЧАЙНЫХ ПРОЦЕССОВ

(Уфимский государственный авиационный технический университет)

Эргодическое свойство стационарных случайных процессов позволяет находить их вероятностные характеристики по одной реализации $x(t)$ осреднением по времени t, что существенно упрощает и удешевляет эксперимент [1, 2]. Однако практически это свойство используется только для определения математического ожидания m_x, дисперсии D_x и автокорреляционных $R_x(\tau)$ или взаимных корреляционных $R_{xy}(\tau)$ функций, где $\tau-$сдвиг во времени между двумя сечениями $x(t)$ и $x(t+\tau)$ процесса, а также реализациями $x(t)$ и $y(t+\tau)$ совместно эргодических процессов. Распределения вероятностей, плотностей вероятностей и их характеристические функции так до настоящего времени не находили. В докладе приводятся известные и введенные автором определения этих характеристик.

Одномерное распределение вероятности $W_1[X]$ выражается через одномерную плотность распределения вероятности $w_1[X]$, и определено выражением [2–4]

$$W_1[X] = \int_{-\infty}^{X} w_1[Z]dZ = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} 1[X - x(t)]dt, \qquad (1)$$

где $1[X - x(t)] = \begin{cases} 0, & X < x(t); \\ 1, & X > x(t) \end{cases}$ $-$единичная функция [5]; $2T-$длительность реализации $x(t)$.