

Берёза Андрей Николаевич

Южно-Российский государственный университет экономики и сервиса.

E-mail: anbirch@mail.ru.

346500, г. Шахты, ул. Шевченко д. 147.

Тел.: 89281574449.

Кафедра информатики ВИС ЮРГУЭС; заведующий кафедрой; доцент.

Bereza Andrew Nikolayevich

South Russian State University of Economics and Service.

E-mail: anbirch@mail.ru.

147, Shevchenko street, Shakhty, 346500, Russia.

Phone: 89281574449.

Head the Department; associate professor.

Ляшов Максим Васильевич

Южно-Российский государственный университет экономики и сервиса.

E-mail: max185@mail.ru.

346500, г. Шахты, ул. Шевченко, д. 147.

Тел.: 89604591974.

Кафедра информационных систем и радиотехники; аспирант.

Liyshov Maxim Vasilyevich

South Russian State University of Economics and Service.

E-mail: max185@mail.ru.

147, Shevchenko street, Shakhty, 346500, Russia.

Phone: 89604591974.

Department of Information Systems & Radio Engineering; postgraduate student.

УДК 681.3.001.63

О.Б. Лебедев

ПОКРЫТИЕ НА ОСНОВЕ МЕТОДА МУРАВЬИНОЙ КОЛОНИИ*

Предлагаются новые технологии, принципы и механизмы решения задачи покрытия, использующие математические методы, в которых заложены принципы природных механизмов принятия решений. Для компактного представления решения задачи покрытия используется матрица граничных требований. Это позволило организовать пространство решений, в рамках которого организован поисковый процесс, базирующийся на моделировании адаптивного поведения муравьиной колонии. По сравнению с существующими алгоритмами достигнуто улучшение результатов.

Покрытие; муравьиная колония; оптимизация.

O.B. Lebedev

COVERING ON THE BASIS OF THE METHOD OF THE ANT COLONY

New technologies, principles and mechanisms of the decision of a problem the coverings using mathematical methods in which principles of natural mechanisms of decision-making are put in pawn are offered. For compact representation of the decision of a problem of a covering the matrix of boundary requirements is used. It has allowed to organize space of decisions in which

* Работа выполнена при поддержке: РФФИ (грант № 09-01-00509), г/б № 2.1.2.1652.

frameworks the search process which is based on modeling of adaptive behaviour of ant colony is organized. In comparison with existing algorithms improvement of results is reached.

Covering; ant colony; optimization.

Введение. Одной из важнейших задач при построении интеллектуальных систем автоматизированного проектирования (САПР) сверхбольших интегральных схем (СБИС), является покрытие функциональной схемы элементами из библиотечного набора фрагментов топологии. Покрытие во многом определяет качество топологии СБИС. Задачи о покрытии классифицируются следующим образом: о покрытии путями, о вершинном покрытии, о покрытии множествами.

Исходными данными задачи о покрытии множествами [1] выступают конечное множество X , а также семейство его подмножеств $F = \{X_i \mid i=1,2,\dots,n\}$, таких, что $X_i \subseteq X$ и $\bigcup_i X_i = X$. Задача о минимальном покрытии множествами заключается в отыскании набора $P \subseteq F$ с минимальным числом подмножеств $X_i \in P$ и $X_i \in F$, такого, чтобы $\bigcup_i X_i = X$ и $X_i \in P$.

Непосредственное обобщение этой задачи состоит в приписывании подмножествам X_i определенных весов и требования многократного покрытия элементов множества X . В таком случае задача сводится к нахождению покрывающего набора минимальной стоимости [1]. Задача покрытия является NP -полной. В течение последних лет были предложены различные подходы к решению проблемы покрытия. Главным образом, это алгоритмы, основанные на эвристиках, обеспечивающих получение приемлемого результата за полиномиальное время [2-4]. Тем не менее, возросшие сложность решаемых задач и требования к качеству решения делают актуальной разработку новых более эффективных методов.

В последние годы интенсивно разрабатывается научное направление, объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений [5-7]. Идея муравьиного алгоритма [8] — моделирование поведения муравьёв, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи. Основу поведения муравьиной колонии составляет самоорганизация, обеспечивающая достижения общих целей колонии на основе низкоуровневого взаимодействия. В работе используется представление задачи покрытия в виде адаптивной муравьиной системы на основе сочетания принципов самообучения, самоорганизации.

Постановка задачи. Одна из первых задач, решаемых на этапе конструкторского проектирования СБИС, — преобразование функциональной схемы в электрическую. Задача покрытия функциональной схемы ячейками из заданного набора эквивалентна задаче о покрытии множествами. Пусть задана функциональная схема $S = \{s_i \mid i=1,2,\dots,w\}$, состоящая из элементов s_i и множество $E = \{e_i \mid i=1,2,\dots,n\}$ типов элементов, образующих покрываемую функциональную схему. Количественный состав схемы по типам элементов опишем вектором $B = \{b_i \mid i=1,2,\dots,n\}$, где b_i — число элементов типа e_i , входящих в состав схемы. Кроме того, задан набор покрывающих ячеек $H = \{h_j \mid j=1,2,\dots,m\}$. Каждая ячейка имеет свой набор элементов из E . Элементы внутри ячейки между собой не соединены. Количественный состав ячеек выразим с помощью матрицы $A = \|a_{ij}\|_{n \times m}$, где a_{ij} — число элементов типа e_i в ячейке типа h_j . Вектор $C = \{c_j \mid j=1,2,\dots,m\}$ конкретизирует для каждой ячейки h_j её стоимость c_j . Схема считается покрытой ячейками из набора H , если каждый элемент функциональной схемы реализуется элементами из состава выбранных ячеек.

Построим математическую модель. Введём целочисленную переменную x_j , определяющую число ячеек типа h_j , входящих в покрывающий набор. Тогда задача покрытия формулируется следующим образом:

минимизировать

$$F = \sum_{j=1}^m x_j \cdot c_j \quad (1)$$

при ограничениях:

$$\sum_{j=1}^m a_{ij} \cdot x_j \geq b_i, i = 1, 2, \dots, n;$$

$$x_j \geq 0, j = 1, 2, \dots, m; \quad a_{ij} \geq 0.$$

Решение задачи представляет набор величин $x_j, j=1, 2, \dots, m$, при которых функция F (суммарная стоимость ячеек покрывающего набора) имеет минимальное значение. Если в качестве показателя c_j принять общее число элементов, присутствующих в составе ячейки t_j , т.е.

$$c_j = \sum_{i=1}^n a_{ij},$$

то F определяет общее число элементов покрывающего набора ячеек.

Формирование пространства решений. Для удобства изложения будем осуществлять процесс формирования пространства решений на примере. Пусть покрываемая схема составлена из элементов трех типов: $E = \{e_1, e_2, e_3\}$. Количественный состав схемы задается вектором $B = \{10, 30, 20\}$. Набор покрывающих ячеек $H = \{h_1, h_2, h_3, h_4, h_5\}$. Матрица A , описывающая количественный состав ячеек, представлена на рис. 1.

Введём матрицу P , которая отражает граничные требования по количественному составу элементов, покрываемых ячейками каждого типа, $P = \|p_{ij}\|_{n \times m}$, где p_{ij} – минимальное число элементов типа e_i , которое обязательно должно быть покрыто ячейками типа $h_j, p_{ij} \geq 0, p_{ij}$ – целое число. При этом для реализации полного покрытия всех элементов в соответствии с требованиями матрицы P необходимо выполнение ограничений:

$$\sum_{j=1}^m P_{ij} = b_i, i = 1, 2, \dots, n. \quad (2)$$

Для вышеприведённого примера один из возможных вариантов матрицы P имеет вид, представленный на рис. 1.

	h_1	h_2	h_3	h_4	h_5
e_1	2	3	1	2	2
e_2	2	2	3	2	1
e_3	3	1	1	2	2

$A =$

	h_1	h_2	h_3	h_4	h_5	$b_i = \sum p_{ij}$
e_1	2	3	3	1	1	$b_1 = 10$
e_2	4	4	8	5	9	$b_2 = 30$
e_3	2	6	5	1	6	$b_3 = 20$

$P =$

Рис. 1. Матрица A , описывающая количественный состав ячеек; матрица P , отражающая граничные требования

Матрица P однозначно соответствует покрывающему набору ячеек, который формируется следующим образом. Сначала строится матрица $D = ||d_{ij}||_{n \times m}$. Элемент матрицы d_{ij} – целое число, которое определяется как большее целое от p_{ij}/a_{ij} и фактически равно *минимальному* числу ячеек типа h_j , необходимых для покрытия p_{ij} элементов типа e_i . Затем в пределах каждого j -го столбца матрицы D находится максимальное число d_{jmax} ($\forall i$) [$d_{jmax} \geq d_{ij}$]. Оно является минимальным числом $x_j=d_{jmax}$ ячеек типа h_j , гарантированно обеспечивающих покрытие p_{1j} элементов типа e_1 , p_{2j} элементов типа e_2 , ... , p_{nj} элементов типа e_n , в соответствии с требованиями матрицы P . Причём, кроме элементов типа e_i , для которых $d_{ij} = d_{jmax}$, остальные типы элементов могут быть покрыты с избытком.

Для рассматриваемого примера матрица D и покрывающий набор X имеют вид, представленный на рис. 2.

Общее число ячеек в покрывающем наборе $N_{\alpha}=2+6+5+3+9=25$. Общее число N_i элементов типа e_i , входящее в состав покрывающего набора ячеек для нашего примера: $N_1=2 \cdot 2+6 \cdot 3+5 \cdot 1+3 \cdot 2+9 \cdot 2=51$; $N_2=2 \cdot 2+6 \cdot 2+5 \cdot 3+3 \cdot 2+9 \cdot 1=46$; $N_3=2 \cdot 3+6 \cdot 1+5 \cdot 1+3 \cdot 2+9 \cdot 2=41$. Общее число элементов, образующих покрывающий набор, $N_3 = N_1 + N_2 + N_3 = 138$. Таким образом, матрица P является символьным представлением решения задачи покрытия множествами. В работе пространство решений представляется множеством матриц P . Поиск решения сводится к поиску такой матрицы P , т.е. к поиску совокупности таких значений элементов p_{ij} матрицы P , которые оптимизируют показатель качества (критерий).

$$D = \begin{array}{c|ccccc} & h_1 & h_2 & h_3 & h_4 & h_5 \\ \hline e_1 & 1 & 1 & 3 & 1 & 1 \\ e_2 & 2 & 2 & 3 & 3 & 9 \\ e_3 & 1 & 6 & 5 & 1 & 3 \end{array}$$

$$X = \begin{array}{c|ccccc} \sum x_j & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline 25 & 2 & 6 & 5 & 3 & 9 \end{array}$$

Рис. 2. Матрица D и покрывающий набор X

Организация поисковых процедур на основе моделировании адаптивного поведения муравьиной колонии. В общем случае поиск решения задачи покрытия осуществляется коллективом кластеров муравьев $Z = \{Z_k | k=1, 2, \dots, l\}$. На каждой итерации муравьи каждого кластера Z_k строят свое конкретное решение задачи покрытия в виде матрицы граничных требований $P^k = ||p^k_{ij}||_{n \times m}$. Другими словами число решений формируемых муравьями на каждой итерации равно числу кластеров муравьев. Число муравьев в каждом кластере равно числу типов элементов n в покрываемой функциональной схеме. Каждый i -й муравей z_{ik} k -го кластера соответствует типу e_i элементов, и решает задачу формирования i -й строки P^k_i матрицы граничных требований P^k путем выбора конкретных значений p^k_{ij} с соблюдением ограничений (2). Таким образом, каждый i -й муравей z_{ik} k -го кластера формирует частичное решение P^k_i . Полное решение – матрица граничных требований $P^k = \bigcup_i P^k_i$

формируется кластером муравьев $Z_k = \{z_{ik} | i=1, 2, \dots, n\}$.

Поиск решений осуществляется на n графах поиска решений $G_1- G_n$, имеющих одну и ту же структуру. В работе исследовались два способа откладывания феромона муравьями в процессе поиска решений. При первом способе муравьи откладывают феромон на ребрах графа поиска решений (ГПР), а при втором способе – на вершинах ГПР. Кроме того, муравьи из разных кластеров обмениваются

информацией в соответствии с которой осуществляется коррекция количества феромона на ребрах (или вершинах) ГПР.

Каждый граф $G_i(V_i, U_i)$ соответствует типу e_i элемента покрываемой схемы и на нем муравьем z_{ik} формируется i -я строка P_i^k матрицы граничных требований P^k . Базовая структура ГПР формируется следующим образом. Множество вершин $V_i \setminus O_i$ графа G_i разбито на $(m-1)$ стадий V_{ij} , где m – количество типов покрывающих ячеек. $\bigcup_j V_{ij} = V_i \setminus O_i, j=1, 2, \dots, (m-1)$. Стадия V_{ij} соответствует типу h_j покрывающей

ячейки. Если a_{ij} не равно нулю, то число вершин в каждой стадии равно (b_i+1) , где b_i – число элементов типа e_i , входящих в состав покрываемой схемы, $V_{ij} = \{v_{ijs} | s=0, 1, 2, \dots, b_i\}$. Вершины в стадии V_{ij} имеют вес от 0 до b_i и расположены в порядке возрастания веса сверху вниз. Вес вершины $\varphi(v_{ijs})$ равен ее порядковому номеру s в стадии V_{ij} , то есть $\varphi(v_{ijs})=s$. Если a_{ij} равно нулю, то стадия V_{ij} включает только одну вершину с нулевым весом. Вершина O_i – начальная. $\varphi(O_i)=0$. Ребра графа $G_i(V_i, U_i)$ ориентированные и связывают вершины соседних стадий V_{ij} и $V_{i,j+1}$. Задача каждого муравья z_{ik} найти в графе G_i маршрут M_{ik} из вершины O_i до одной из вершин стадии $V_{i,m-1}$. Ограничением является суммарный вес $\varphi(M_{ik})$ вершин, входящих в маршрут M_{ik} , он должен быть не больше b_i , то есть

$$\varphi(M_{ik}) \leq b_i. \quad (3)$$

Каждая вершина x_{ijs} в маршруте M_{ik} определяет значение элемента $p_{ij}^k \in P_i^k$. При этом $p_{ij}^k = \varphi(v_{ijs}), v_{ijs} \in M_{ik}$. Значение последнего элемента p_{im}^k в строке P_i^k определяется как $p_{im}^k = b_i - \varphi(M_{ik})$.

С учетом ограничения (3) ребра, связывающие вершины соседних стадий V_{ij} и $V_{i,j+1}$ формируются по следующему правилу. Вершина $v_{ijc} \in V_{ij}$ связана с такими вершинами $v_{i,j+1,s} \in V_{i,j+1}$ для которых выполняются условие

$$\varphi(v_{ijc}) + \varphi(v_{i,j+1,s}) \leq b_i. \quad (4)$$

Формирование графа G_i с соблюдением условия (4) позволяет уменьшить общее число ребер U_i и соответственно сузить пространство поиска. При таком подходе число ребер между двумя стадиями уменьшается почти в два раза и равно $(b_i+2) \cdot (b_i+1) / 2$. Поскольку стадий V_{ij} в графе G_i равно $(m-1)$, а ребра связывают только две соседние стадии, число ребер между стадиями равно $((b_i+2) \cdot (b_i+1) / 2) \cdot (m-2)$. Базовая структура ГПР для $b_i = 3$ и $m = 4$ представлена на рис. 3.

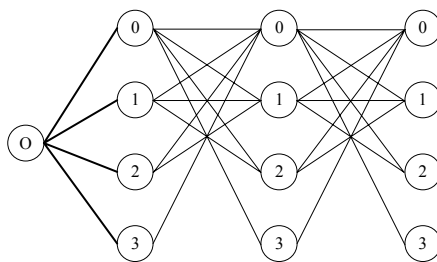


Рис. 3. Базовая структура ГПР для $b_i = 3$ и $m = 4$

В общем случае ГПР представляется совокупностью из m стадий (по числу типов покрывающих ячеек) и начальной вершины O_i . В этом случае по схеме, рассмотренной выше, начальная вершина O_i связывается с каждой стадией, в свою очередь каждая стадия связывается с остальными. Для равномерного распределе-

ния муравьев и создания равных стартовых условий можно использовать m начальных вершин, при этом каждая начальная вершина связана со своей стадией. Частным случаем является циклическая стадия: первая стадия связана со второй, вторая с третьей, ..., $(m-1)$ -я стадия с m -ой, m -я стадия с первой. Задача муравья z_{ik} найти маршрут M_{ik} , включающий по одной вершине из $(m-1)$ -й стадии. Значение последнего элемента p_{im}^k в строке P_i^k определяется как $p_{im}^k = b_i - \varphi(M_{ik})$.

Процесс поиска решений итерационный. Каждая итерация l включает четыре этапа. На первом этапе муравьями каждого кластера находится решение задачи покрытия, на втором этапе на ребрах (или вершинах) графов $G_1 - G_n$ откладывается феромон, на третьем этапе осуществляется коррекция количества феромона на ребрах (или вершинах) ГПР, на четвертом этапе осуществляется испарение феромона. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромоны откладываются муравьями на ребрах графов $G_1 - G_n$ после полного формирования решения. Не теряя общности, рассмотрим процесс поиска решения на базовой структуре ГПР.

На первом этапе каждой итерации каждый муравей z_{ik} формирует свое собственное частичное решение – маршрут M_{ik} в графе G_i . Моделирование поведения муравьев в задаче покрытия связано с распределением феромона на множестве ребер (или вершин) графов $G_1 - G_n$. Предварительно на всех ребрах (или вершинах) каждого графа G_i откладывается одинаковое (небольшое) количество феромона ε_i . Значение ε_i задается априорно. Процесс построения маршрута M_{ik} пошаговый, начиная от вершины O_i , поэтому, вначале, все i -е муравьи z_{ik} из всех кластеров помещаются в вершину O_i . Число таких муравьев равно числу кластеров. На каждом шаге агент применяет вероятностное правило выбора очередной вершины в графе G_i для включения ее формируемый маршрут M_{ik} , причем на шаге t в маршрут включается вершина стадии V_{it} . Процесс выбора осуществляется следующим образом. Обозначим как $M_{ik}(t)$ маршрут, построенный муравьем z_{ik} за t шагов, и пусть вершина $v_{itc} \in V_{it}$ является последней в маршруте $M_{ik}(t)$. На шаге $t+1$ определяется суммарный вес вершин – $\varphi(M_{ik}(t))$, входящих в состав $M_{ik}(t)$. Пусть $\Gamma_{it}(c)$ множество вершин стадии $V_{i,t+1}$, смежных вершине v_{itc} , $\Gamma_{it}(c) \subset V_{i,t+1}$. Среди множества вершин $\Gamma_{it}(c)$ выделяется подмножество $\Pi_{it}(c) \subset \Gamma_{it}(c)$ такое, что для каждой вершины $v_{i,t+1,s} \in \Pi_{it}(c)$ выполняется условие

$$\varphi(M_{ik}(t)) + \varphi(v_{i,t+1,s}) \leq b_i, \quad (5)$$

обеспечивающее выполнение ограничения (3). Вершины подмножества $\Pi_{it}(c)$ являются кандидатами для включения в маршрут на шаге $t+1$.

Агент просматривает все вершины $\Pi_{it}(c)$. При первом способе для каждой вершины $v_{i,t+1,s} \in \Pi_{it}(c)$ рассчитывается f_{cs} – суммарное количество феромона, отложенного на ребре графа G_i , связывающего $v_{itc} \in V_{it}$ с $v_{i,t+1,s} \in \Pi_{it}(c)$. При втором способе рассчитывается f_{cs} – суммарное количество феромона, отложенного на каждой вершине $v_{i,t+1,s} \in \Pi_{it}(c)$. Вероятность P_v включения на шаге $t+1$ вершины $v_{i,t+1,v} \in \Pi_{it}(c)$ в формируемый отдельным муравьем маршрут пропорциональна суммарному количеству феромона f_{cv} . P_v определяется следующим соотношением:

$$P_v = f_{cv} / \sum_s f_{cs}, \quad (s \mid v_{i,t+1,s} \in \Pi(v_{itc})). \quad (6)$$

С вероятностью P_v агент выбирает одну из вершин, которая включается в маршрут $M_{ik}(t+1)$.

После построения муравьями маршрутов (каждый муравей z_{ik} – свой маршрут M_{ik} за $m-1$ шагов), формируются частичные решения P_i^k и полное решение P^k , по-

строенное кластером муравьев Z_k . В соответствии с методом, изложенным выше, для каждого решения P^k , находится значение целевой функции F^k .

На втором этапе итерации, каждый муравей откладывает феромон на ребрах (или вершинах) построенного маршрута.

Количество феромона $\Delta\tau_{ik}(l)$, откладываемое муравьем z_{ik} на каждом ребре (или вершине) построенного маршрута M_{ik} , определяется следующим образом:

$$\Delta\tau_{ik}(l) = Q_i / F^k(l), \quad (7)$$

где l – номер итерации, Q_i – общее количество феромона, откладываемое муравьем на ребрах (или вершинах) маршрута в графе G_i , $F^k(l)$ – целевая функция для решения, полученного кластером муравьев Z_k на l -ой итерации. Чем меньше $F^k(l)$, тем больше феромона откладывается на ребрах (или вершинах) построенного маршрута и, следовательно, тем больше вероятность выбора этих ребер (или вершин) при построении маршрутов на следующей итерации.

Далее, на третьем этапе, осуществляется корректировка количества феромона, отложенного на ребрах (или вершинах) графов G_1 - G_n , путем обмена информацией о количествах элементов, выбранных агентами, которые должны быть покрыты ячейками типа h_j .

Просмотрим матрицу D^k по столбцам. В столбце j находим максимальный элемент $(d^k_{oj})_{max}$, который и определяет число x_j ячеек типа h_j в покрывающем наборе, найденное агентами k -го кластера. Величина $(d^k_{oj})_{max}$ соответствует весу вершины j -ой стадии в маршруте M_{ok} , построенном агентом z_{ok} , то есть величине $(p^k_{oj})_{max}$. Для нашего примера рассмотрим 2-й столбец матрицы D^k . Максимальное значение имеет $d^k_{32}=6$. Значение d^k_{32} определяется весом, равным 6, вершины стадии V_{32} графа G_3 в маршруте агента z_{3k} . $d^k_{32} = p^k_{32} / a_{32} = 6/1=6$. Отметим, что число ячеек $x_j = (d^k_{oj})_{max}$ будет избыточным для покрытия тех количеств p^k_{ij} элементов типа $e_i \neq e_o$, которые определены в столбце j матрицы P^k . Поскольку целью оптимизации является минимизация числа покрывающих ячеек, естественным является стремление изменить, а именно уменьшить $(d^k_{oj})_{max}$, при этом уменьшится как число покрывающих ячеек, так и число избыточно покрытых элементов. В связи с этим при построении маршрута агентом z_{ok} на следующей итерации целесообразно вершины стадии V_{oj} с весом равным и большим величины $(p^k_{oj})_{max}$ сделать менее предпочтительными. При первом способе это достигается путем уменьшения количества феромона на величину α на всех ребрах графа G_o входящих в вершины стадии V_{oj} с весами от $(p^k_{oj})_{max}$ до b_i . В нашем примере уменьшается количество феромона на всех ребрах входящих в вершины стадии V_{32} графа G_3 с весами от 6 до 20. При втором способе это достигается путем уменьшения количества феромона на величину α на всех вершинах стадии V_{oj} с весами от $(p^k_{oj})_{max}$ до b_i .

После того, как каждый агент сформировал решение, отложил феромон, и произведена коррекция количества феромона на ребрах (или вершинах) графов G_1 - G_n на четвертом этапе происходит общее испарение феромона на ребрах (или вершинах) графов G_1 - G_n в соответствии с формулой (8):

$$f_{ik} = f_{ik} \cdot (1 - \rho), \quad (8)$$

где ρ – коэффициент обновления.

После выполнения всех действий на итерации находится агент с лучшим решением, которое запоминается. Далее осуществляется переход на следующую итерацию.

При использовании первого способа откладывания феромона муравьями пространственная сложность пропорциональна числу ребер, при втором способе чис-

лу вершин. Поскольку число ребер значительно больше числа вершин, то второй способ более предпочтителен, как в отношении пространственной, так и в отношении временной сложности алгоритма.

Временная сложность этого алгоритма зависит от времени жизни колонии l (число итераций), количества вершин графа n и числа муравьев m , и определяется как $O(l \cdot n^2 \cdot m)$.

Отметим, что при построении матрицы D^k по сформированной агентами матрице P^k , минимальное число d_{ij}^k ячеек типа h_j , необходимых для покрытия p_{ij}^k элементов типа e_i , определяется как большее целое от p_{ij}^k/a_{ij} . Другими словами – d_{ij}^k кратно a_{ij} . В связи с этим целесообразно использовать значения весов вершин в каждом графе G_i кратными величине a_{ij} , то есть $\varphi(v_{ijs}) = s \cdot a_{ij}$.

В этом случае число и веса вершин в стадии определяются следующим образом. Пусть C_{ij} – меньшее целое от b_i/a_{ij} . Множество вершин стадии $V_{ij} = \{v_{ijs} | s=0, 1, 2, \dots, N_{ij}\}$. Если $(b_i - a_{ij} \cdot C_{ij}) = 0$, то есть b_i делится на a_{ij} без остатка, то $N_{ij} = C_{ij}$. Если $(b_i - a_{ij} \cdot C_{ij}) > 0$, то $N_{ij} = C_{ij} + 1$. Вершины стадии имеют следующий вес: $\varphi(v_{ij0}) = 0 \cdot a_{ij} = 0$; для s в интервале от 1 до C_{ij} вес вершин $\varphi(v_{ijs}) = s \cdot a_{ij}$. Если $(b_i - a_{ij} \cdot C_{ij}) > 0$, то для $s = C_{ij} + 1$ вес вершины $\varphi(v_{ijs}) = b_i$. Таким образом, сформированное рассмотренным способом множество вершин в стадии меньше в среднем в a_{ij} раз. При таком подходе число и веса вершин в соседних стадиях могут быть различными, но в любом случае формирование множества ребер между соседними стадиями производится с учетом условия (4), а формирование маршрута M_{ik} с учетом условия (3).

Экспериментальные исследования проводились на IBM PC. В работе для преодоления локального барьера исследовалась комбинация подходов к построению структуры ГПР. В частности, наивысшая эффективность была достигнута при использовании: циклической структуры ГПР; с числом начальных вершин равным числу стадий; значения весов вершин в каждой стадии V_{ij} графа G_i кратными величине a_{ij} ; подхода, при котором феромон откладывался на вершинах ГПР. Сравнение с известными алгоритмами показало, что при меньшем времени работы у полученных с помощью муравьиного алгоритма решений значения целевой функции лучше (меньше) в среднем на 6%.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Андерсон Д. Дискретная математика и комбинаторика. – М.: Вильямс, 2003.
2. R. Cordone, F. Ferrandi, D. Sciuto, R.W. Calvo. An Efficient Heuristic Approach to Solve the Unate Covering Problem. *IEEE Transactions on computer-aided design of integrated circuits and systems*, vol.120, No.12, December 2001. – P. 1377-1387.
3. Coudert O. "On solving covering problems", in *proceedings of 30th ACM/IEEE Design automation conference*, 1996. – P. 197-202.
4. Дендобренко Б.П., Малика А.С. Автоматизация проектирования радиоэлектронной аппаратуры. – М.: Высшая школа, 2002.
5. Лебедев Б.К., Лебедев В.Б. Планирование на основе роевого интеллекта и генетической эволюции // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 25-33.
6. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Решение задачи покрытия на основе эволюционного моделирования // Теория и системы управления. – 2009. – №1. – С. 101-116.
7. G. Di Caro, F. Ducatelle, L.M. Gambardella. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, 16(5): 443-455, 2005.
8. M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.

Лебедев Олег Борисович

Технологический институт федерального государственного образовательного учреждения высшего профессионального образования “Южный федеральный университет” в г. Таганроге.

E-mail: lbk@tsure.ru.

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: 88634371743.

Кафедра систем автоматизированного проектирования; доцент.

Lebedev Oleg Borisovich

Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: lbk@tsure.ru.

44, Nekrasovsky, Taganrog, 347928, Russia.

Phone: 88634371743.

Department of Computer Aided Design; associate professor.

УДК 681.3

В.В. Курейчик, Е.Е. Полупанова

**ЭВОЛЮЦИОННАЯ ОПТИМИЗАЦИЯ НА ОСНОВЕ АЛГОРИТМА
КОЛОНИИ ПЧЁЛ***

В статье рассматриваются основные принципы работы простого пчелиного алгоритма, улучшенного пчелиного алгоритма, а также алгоритма колонии пчел. Все эти алгоритмы моделируют поведение пчел в живой природе в поисках нектара. Проводится сравнение временной сложности предложенных алгоритмов, а также все алгоритмы сравниваются в разрезе используемых параметров.

Пчелиный алгоритм; улучшенный пчелиный алгоритм; алгоритм колонии пчёл.

V.V. Kureichik, E.E. Polupanova

**ARTIFICIAL BEE COLONY ALGORITHM OF EVOLUTIONARY
OPTIMIZATION**

In this paper, performances of basic Bees algorithm, Improved Bees algorithm and also Artificial Bee Colony Algorithm were presented. All this algorithms mimics the food foraging behavior of swarm of honey bees. Effect of problem dimensionality on the performance of the algorithms was investigated and the algorithms were compared to each other in terms of the number of control parameters to be tuned.

Bees algorithm; Improved Bees algorithm; Artificial Bee Colony algorithm.

Введение. Основной целью эволюционной оптимизации является поиск такого сочетания параметров (независимых переменных), которое бы способствовало максимизации или минимизации качественных, количественных и вероятностных характеристик решаемой задачи [1]. В последнее время широкое распространение получили интегрированные методы оптимизации, заимствующие основные принципы своей работы из живой природы [2]. Исследователи экспериментируют с различными типами представлений, так, эволюционные и генетические алгоритмы

* Работа выполнена при поддержке: РФФИ (гранты № 10-07-90010, № 09-07-00318), г/б № 2.1.2.1652.