УДК 681.325

Покрытие на основе методов роевого интеллекта

Б.К. Лебедев, О.Б. Лебедев

Южный федеральный университет

Lebedev.b.k@gmail.com

***Аннотация* — Предлагаются новые технологии, принципы и механизмы решения задачи покрытия множествами, использующие математические методы, в которых заложены принципы природных механизмов принятия решений. В работе используется представление решения задачи на основе моделирования адаптивного поведения муравьиной колонии. Разработана многостадийная структура графа поиска решений, на базе которого организован поисковый процесс. Проведены экспериментальные исследования. По сравнению с существующими алгоритмами достигнуто улучшение результатов.**

***Ключевые слова* — покрытие множествами; адаптивное поведение; муравьиная колония; оптимизация; САПР СБИС.**

I. Введение

Задача о покрытии множествами [1, 2] заключается в нахождении покрывающего набора минимальной стоимости и является *NP*-полной. К задаче о покрытии сводятся известные задачи дискретной оптимизации [1, 2]: задачи стандартизации, упаковки и разбиения множества, задача о наибольшей клике, задача минимизации полинома от булевых переменных и др. Известна также и обратная сводимость задачи о покрытии к этим задачам. На практике задачи о покрытии возникают при размещении пунктов в системах информационного поиска, при назначении экипажей на транспорте, проектировании интегральных схем и конвейерных линий и т.д. 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.

Для решения общей задачи о покрытии предложено большое число [3-6] точных алгоритмов, основанных на методах ветвей и границ, отсечения, перебора X-классов и др. Верхние оценки временной сложности точных методов сопоставимы со сложностью переборных алгоритмов.

В связи с этим при решении практических задач большую актуальность приобретают эвристические способы сокращения перебора. Для поиска приближенных решений широкое распространение получили методы лагранжевой релаксации [3, 4], генетические алгоритмы [7-9], роевые алгоритмы [10, 11], нейронные сети и другие эвристические алгоритмы [12-15].

Достаточно эффективный вариант ГА для задачи покрытия предложен Дж. Бисли и П. Ху в работе [7]. Проблема при использовании данного представления связана с возможностью получения недопустимых решений после мутации и кроссинговера.

В работе [12] были выбраны аддитивный и генетический методы дискретной оптимизации. Первый дает точное решение и использует зондирование решений с оценкой подмножества частичных решений. Аддитивный алгоритм более затратный в плане используемой памяти, что не позволяет использовать его при решении задач большой размерности. Недостатком генетического алгоритма является его рандомизированность. С ростом количества состояний решения увеличиваются относительное отклонение результата от минимального и количество ложных “отсутствий решения” задачи.

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. 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 [16-24]. В работе используется представление задачи покрытия в виде адаптивной муравьиной системы на основе сочетания принципов самообучения, самоорганизации.

II. 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. Исходными данными для решения задачи покрытия являются: функциональная схема проектируемой СБИС и схемы типовых ячеек применяемого библиотечного набора. Необходимо найти такое распределение логических функций покрываемой схемы по отдельным ячейкам, при котором достигается экстремум целевой функции. Известные в литературе алгоритмы покрытия функциональной схемы ячейками из заданного набора оптимизируют следующие показатели: суммарную стоимость ячеек, покрывающих схему; общее число ячеек, необходимое для реализации схемы; число типов используемых ячеек; число межъячеечных связей, общее число элементов, входящих в покрывающий набор ячеек [7-9].

Пусть задана функциональная схема $S = \{s_i \mid i=1,2,\ldots,w\}$, состоящая из элементов s_i, и множество $E = \{e_i \mid i=1,2,\ldots,n\}$ типов элементов, образующих покрываемую функциональную схему. Количественный состав схемы по типам элементов опишем вектором $B = \{b_i \mid i=1,2,\ldots,n\}$, где b_i – число элементов типа e_i, входящих в состав схемы. Кроме того, задан набор покрывающих ячеек $H = \{h_j \mid j=1,2,\ldots,m\}$. 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_i. Вектор $C = \{c_j \mid j=1,2,\ldots,m\}$ 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

$$F = \sum_{j=1}^{m} x_j \cdot c_j \quad , \tag{1}$$

при ограничениях F

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

$$a_{ij} \ge 0. \; x_j \ge 0, j = 1,2,\ldots,m;$$

Решение задачи представляет набор величин x_j, $j = 1, 2, \ldots, m$, 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.

$$c_j = \sum_{i=1}^{n} a_{ij} \quad ,(3)$$

то F определяет общее число элементов покрывающего набора ячеек.

III. 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 $B = \{10,30,20\}$. Следовательно, имеем 10 элементов первого типа, 30 – второго и 20 – третьего. Набор покрывающих ячеек состоит из пяти типов: $H = \{h_1,h_2,h_3,h_4,h_5\}$. Матрица A, описывающая количественный состав ячеек, представлена на рис. 1.

$A =$

	h_1	h_2	h_3	h_4	h_5
e_1	2	3	3	1	1
e_2	4	4	8	5	9
e_3	2	6	5	1	6

Рис. 1. Матрица *A*, описывающая количественный состав ячеек

Пусть имеется некоторое решение – покрывающий набор ячеек $X = \{x_j \mid j=1,2,\ldots,m\}$. Введём матрицу P, которая отражает количественный состав элементов, входящих в покрывающий набор ячеек X, $P = ||p_{ij}||_{n\times m}$, где p_{ij} – число элементов типа e_i , содержащееся в x_j ячейках типа h_j, $p_{ij} \ge 0$, p_{ij} – целое число. При этом для реализации полного покрытия всех элементов в соответствии с требованиями матрицы P необходимо выполнение ограничений (2). Пусть $X = \{2, 6, 5, 3, 4\}$. Для заданного покрывающего набора ячеек X матрица P имеет вид, представленный на рис. 2.

$P =$

	h_1	h_2	h_3	h_4	h_5	$b_i=\sum p_{ij}$
e_1	4	18	15	3	4	b_1= 44
e_2	8	24	40	15	36	b_2= 123
e_3	4	36	25	3	24	b_3=92
X	2	6	5	3	4	

Рис. 2. Матрица количественного состава элементов *P*

Общее число ячеек $N_я$ в покрывающем наборе и общее число N_i элементов типа e_i, входящее в состав покрывающего набора ячеек, определяются по формулам

$$N_я = \sum_{j=1}^{m} x_j \quad , \quad N_i = \sum_{j=1}^{m} a_{ij} \cdot x_j \quad .$$

Для нашего примера: $N_я = 20$; $N_1 = 44$; $N_2 = 123$; $N_3 = 92$. Общее число элементов, образующих покрывающий набор, $N_э = N_1 + N_2 + N_3 = 259$.

Данный набор X обеспечивает покрытие исходной схемы, но со значительным избытком покрывающих элементов. Целью разрабатываемого алгоритма является нахождение такого покрывающего набора X, который с соблюдением ограничений (2) минимизирует целевую функцию. Назовем покрывающий набор X, для которого соблюдаются ограничения (2), легитимным. В работе пространство решений представляется множеством легитимных, покрывающих набор X. Поиск решения сводится к поиску такого легитимного покрывающего набора X, который оптимизируют показатель качества (критерий).

IV. Организация поисковых процедур на основе моделирования адаптивного поведения муравьиной колонии

В общем случае поиск решения задачи покрытия осуществляется коллективом муравьев $Z = \{z_k \mid k = 1,2,\dots,N_k\}$. На каждой итерации каждый муравей z_k строит свое конкретное решение задачи покрытия в виде легитимного покрывающего набора X_k, Другими словами, число решений, формируемых муравьями на каждой итерации, равно числу муравьев l. Каждый муравей z_k решает задачу формирования легитимного покрывающего набора X_k с соблюдением ограничений (2).

Поиск решений осуществляется на графе поиска решений (ГПР) $G(V \cup O, U)$, $V \cup O$ – множество вершин, U – множество ребер. Решение представляется в виде маршрута на графе G. В общем случае ГПР представляется совокупностью из m подграфов G_j (по числу типов покрывающих ячеек) (рис.3).

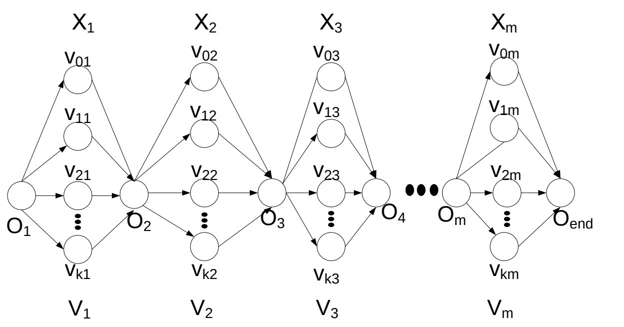

Рис. 3. Граф поиска решений

В состав подграфа G_j входит множество вершин V_j и связывающая вершина o_j. $O = \{o_j \mid j=1,2,\dots,m\}$. Базовая структура ГПР формируется следующим образом. Множество вершин V графа G разбито на m подмножеств $V_j = \{v_{ij} \mid i=0,1,2,\dots,n_i\}$, где m – количество типов покрывающих ячеек.

$$\bigcup_j V_j = V, \ j=1,2,\dots,m.$$

Вершины множества V_j в подграфе G_j имеют вес от 0 до b_i/a_{ij} и расположены в порядке возрастания веса сверху вниз. При этом

$$v_{0j} = 0, v_{nij} = b_i/a_{ij}, v_{i+1,j}=v_{ij}+1.$$

Множество вершин V_j графа G соответствует множеству возможных значений x_j – числа покрывающих ячеек типа h_j. Если $x_j = v_{0j} = 0$, то это значит, что элементы типа e_i не покрываются ячейками типа h_j. Если $x_j = v_{nij} = b_i / a_{ij}$, то это значит, что все элементы типа e_i покрываются x_j ячейками типа h_j. Во все вершины V_j входят ребра, выходящие из вершины O_j. Из всех вершин V_j выходят ребра, входящие в вершину O_{1+1},

В этом случае муравей z_k прокладывает маршрут из начальной вершины $o_{1,}$, включающий по одной вершине v_{ij} из каждого подмножества V_j и множества связывающих вершин O, чередующихся в маршруте с вершинами V, $M_k = (o_{1,}\ v_{i1},\ o_2,\ v_{i2},\ o_3,\ v_{i3},\dots\ o_{ij},\dots\ o_{im})$ Построенному маршруту будет соответствовать покрывающий набор X_k. Каждая вершина v_{ij} маршрута M_k соответствует числу x_j покрывающих ячеек типа h_j.

Для того чтобы построенный муравьем покрывающий набор X был легитимным, необходимо чтобы для него выполнялись ограничения (2).

Процесс поиска решений итерационный. Каждая итерация l включает четыре этапа. На первом этапе каждым муравьем колонии независимо друг от друга на ГПР находится решение задачи покрытия. С каждой вершиной $v_{ij} \in V$ связаны два параметра: f_{ij} – количество отложенного на v_{ij} феромона и r_{ij} – число вхождений вершины v_{ij} в нелегитимные решения. На втором этапе на вершинах подграфов V_1–V_m откладывается феромон. На третьем этапе осуществляется коррекция количества феромона на вершинах ГПР. На четвертом этапе осуществляется испарение феромона. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромоны откладываются муравьями на вершинах подграфов V_1–V_m после полного формирования решения. Не теряя общности, рассмотрим процесс поиска решения на базовой структуре ГПР.

Обозначим как $\varphi(x_j)_i$ суммарный вес (число) элементов типа e_i, покрываемых x_j ячейками типа h_j. Обозначим как $\varphi(X_k)_i$ суммарный вес (число) элементов типа e_i, покрываемых ячейками покрывающего набора X_k типа h_j. Для всех элементов типа e_i ограничением является суммарный вес $\varphi(M_{ik}) = \varphi(X_k)_i$ вершин, входящих в маршрут M_k, он должен быть не меньше b_i, то есть

$$\varphi(M_{ik}) \geq b_i. \qquad (3)$$

На первом этапе каждой итерации каждый муравей z_k формирует свое собственное частичное решение – маршрут M_k в графе G. Моделирование поведения муравьёв в задаче покрытия связано с распределением феромона на множестве вершин V графа G. Предварительно на всех вершинах множества V графа G откладывается одинаковое (небольшое) количество феромона ε, т.е. все $f_{ij} = \varepsilon$. Значение ε задается априорно. Кроме того, всем параметрам r_{ij} присваивается нулевое значение. Процесс построения маршрута M_k пошаговый, начиная от вершины o_1, поэтому вначале все муравьи помещаются в вершину o_1. Число таких муравьев задается априорно. На каждом шаге t агент, находящийся в некоторой связывающей вершине o_t, применяет вероятностное правило выбора очередной вершины в множестве V_t для включения ее в формируемый маршрут M_k. Процесс выбора осуществляется следующим образом. Для того чтобы построенный муравьем покрывающий набор X был легитимным, необходимо чтобы для него выполнялись ограничения (2).

На шаге t определяется множество вершин V_t, смежных вершине o_t.

Агент просматривает все вершины V_t. Рассчитывается f_{it} – суммарное количество феромона, отложенного на каждой вершине $v_{it} \in V_t$ за $(t-1)$ шагов, и фиксируются значения параметров r_{it}. На основе f_{it} и r_{it} рассчитываются значения показателей s_{it}, характеризующих степень притягательности вершины v_{it}:

$$s_{it} = \alpha f_{it} + \beta(t - r_{it}), \tag{4}$$

$r_{it} \le t$, чем больше r_{it}, тем меньше s_{it}; α, β – коэффициенты.

Вероятность P_{it} включения на шаге t вершины v_{it} в формируемый отдельным муравьем маршрут пропорциональна значению показателя s_{it}. P_{it} определяется следующим соотношением:

$$P_{it} = s_{it} / \sum_i s_{it}\ , (i \mid v_{it} \in V_t). \tag{5}$$

С вероятностью P_{it} агент выбирает одну из вершин, которая включается в маршрут $M_k(t)$.

После построения муравьем маршрута M_k фиксируется факт его легитимности. Если построенный маршрут легитимен, то на основе маршрута M_k формируется покрывающий набор X_k, строится матрица P_k – находится значение целевой функции F^k.

На втором этапе итерации каждый муравей откладывает феромон на вершинах построенного маршрута.

Количество феромон $\tau_{ik}(l)$, откладываемое муравьем z_k на каждой вершине построенного маршрута M_i, определяется следующим образом:

$$\tau_k(l) = Q_i / F^k(l), \tag{6}$$

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

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).

$$f_{ik} = f_{ik} \cdot (1 - \rho), \tag{7}$$

где ρ – коэффициент обновления.

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

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

Алгоритм покрытия на основе метода муравьиной колонии формулируется следующим образом.

1. В соответствии с исходной функциональной схемой $S = \{s_i \mid i=1,2,\dots,w\}$ определяется множество $E = \{e_i \mid i=1,2,\dots,n\}$ типов элементов, количественный состав схемы по типам элементов – вектор $B = \{b_i \mid i=1,2,\dots,n\}$, набор покрывающих ячеек $H = \{h_j \mid j=1,2,\dots,m\}$, количественный состав ячеек – матрица $A = ||a_{ij}||_{n\times m}$, вектор $C = \{c_j \mid j=1,2,\dots,m\}$ стоимости ячеек.

2. Формируется граф поиска решений $G(V \cup O, U)$ в виде совокупности подграфов G_1–G_m. (рис. 3). Для каждого подграфа $G_j = (0_j\ \ V_j)$ формируется подмножество вершин $V_j = \{v_{ij} \mid i=0,1,2,\dots,n_i\}$, $\ V_j = V$, $j = 1,2,\dots,m$, где m – количество типов покрывающих ячеек. В соответствии с матрицей A вершинам множества V_j в подграфе G_j задается вес.

3. Задаются значения управляющих параметров муравьиного алгоритма: α, β, ε, Q. Число муравьев – N_z.

4. На всех вершина подграфов G_1–G_n откладывается начальное количество феромона. Всем параметрам r_{it} присваивается нулевое значение.

5. Задается число итераций – N_l.

6. $k = 1$.

7. $t = 1$.

8. Муравей z_k с вероятностью P_{it}, вычисленной по формулам (4) и (5), включает вершину v_{it} в маршрут M_k.

9. Если $t < m$, то $t = t + 1$ и переход к пункту 8, иначе переход к пункту 9.

10. Если маршрут M_k легитимный, то рассчитывается значение целевой функции $F^k(l)$.

11. Если $k < N_z$, то $k = k + 1$ и переход к пункту 7, иначе переход к пункту 12.

12. Для каждого маршрута M_k: если он легитимный, то на вершинах ГПР, соответствующих маршруту M_k, откладывается феромон в количестве, рассчитываемом по формуле 6; если он нелегитимный, то на вершинах, соответствующих маршруту M_k, значение параметров r_{it} увеличивается на 1.

13. На каждой вершине ГПР количество феромона уменьшается в соответствии с формулой (7).

14. Если $l < N_l$, то $l = l + 1$ и переход к пункту 6, иначе переход к пункту 15.

15. Конец работы алгоритма.

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 распределения муравьев и создания равных стартовых условий можно использовать m начальных вершин, при этом каждая начальная вершина связана со своей стадией. Частным случаем является циклическая структура связей стадий: первая стадия связана со второй, вторая с третьей, …, (m-1)-я стадия с m-ой, m-я стадия с первой. Задача муравья z_k найти маршрут M_k, включающий по одной связывающей вершине из подграфа, связывающего две соседние связывающие вершины.

V. Экспериментальные исследования

Экспериментальные исследования проводились на ЭВМ типа IBM PC. Для анализа точности получаемых решений был синтезирован ряд примеров с априори известным оптимальным значением целевой функции. Исследованию подвергались примеры, содержащие до 1000 элементов. В среднем 120 итераций достаточно для нахождения лучшего решения. Вероятность получения оптимального решения составила 0.95, а оценки локально оптимальных решений отличались от глобального оптимума в среднем на 1%. Сравнение с известными алгоритмами [22-24] показало, что при меньшем времени работы у полученных с помощью муравьиного алгоритма решений значения целевой функции лучше (меньше) в среднем на 6%. Для преодоления локального барьера исследовались комбинации подходов к построению структуры ГПР. В частности, повышение эффективности было достигнуто при использовании: циклической структуры ГПР с использованием муравьями в качестве начальных вершин связывающих вершин множества O. В результате экспериментов установлено, что при объеме популяции $M = 100$ алгоритм сходится в среднем на 120-й итерации. При этом отклонения в сторону увеличения этой оценки составляли до 10%, а в сторону уменьшения – до 35%.

В работе для численных экспериментов использовались тестовые задачи из OR-библиотеки [49] (http://mscmga.ms.ic.ac.uk/mfo.html) и DIMACS Challange II [60] (ftp://dimacs.rutgers.edn/pub/ challenge/graph/), которые получены с помощью генератора псевдослучайных чисел.

Для сравнения алгоритмов решения задачи о покрытии используется много примеров, построенных случайным образом. Наиболее известны серии 4, 5, 6 с размерностями задач n = 1000–2000 и m = 200, предложенные в работе Э. Балаша и А. Ху [17], а также серии A, B, ..., H задач большей размерности, сгенерированные Дж. Бисли [32] аналогично сериям 4, 5, 6. Эти задачи синтезированы для матриц с заданной разреженностью (2% единиц в сериях 4, 5, A, C, G; 5% единиц в сериях 6, B, D, H; 10% и 20% единиц в сериях E и F соответственно. Веса выбраны случайным образом в диапазоне от 1 до 100. Широкий разброс весов и небольшая плотность матриц этих задач позволяют в точных алгоритмах существенно уменьшить число анализируемых вариантов по сравнению с переборными алгоритмами [1-5]. Исходные данные описанных задач доступны в библиотеке OR-Library.

Рассмотрим результаты решения наиболее трудного набора 6[]. Набор 6 состоит из пяти задач: 6.1, 6.2, 6.3, 6.4 и 6.5. В них m = 200, n = 1000, заполненность матриц ненулевыми элементами составляет 5%. Результаты решения (значения целевых функций) сведены в таблице1.

В пакете ЦЛП задачи линейного целочисленного программирования решаются симплекс-методом.

Сравнение с известными алгоритмами показало, что при меньшем времени работы у полученных с помощью муравьиного алгоритма решений значения целевой функции лучше (меньше). Разработанный алгоритм позволяет решать ЗПМ довольно больших размерностей с результатом, отличающимся от оптимального решения примерно на 1%, за приемлемое время счета.

Таблица 1

Результаты решения тестового набора 6

Имя	ЦЛП	A[22]	A[23]	A[24]	ACO
6.1	138	145	141	142	140
6.2	146	153	146	156	146
6.3	145	148	145	145	145
6.4	131	132	131	132	131
6.5	161	168	163	170	162

Повышения эффективности представленного алгоритма можно добиться путем адаптивного управления многостадийной структурой связей.

Разработанный алгоритм покрытия имеет универсальный характер и может быть использован для решения широкого круга задач линейного целочисленного программирования. Экспериментальные исследования проводились на ЭВМ типа IBM PC. В работе для преодоления локального барьера исследовались комбинации 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 кратны величине a_{ij}; подхода, при котором феромон откладывался на вершинах ГПР

VI. Заключение

Отличительной особенностью представленного алгоритма является то, что поиск решений осуществляется на графе поиска решений, представленного совокупностью m подграфов G_1–G_m, имеющих многостадийную структуру связей. С другой стороны, построенный муравьем маршрут M_k соответствует различным разбиениям элементов по ячейкам. Основу поведения муравьиной колонии составляет самоорганизация, обеспечивающая достижение общих целей колонии на основе низкоуровневого взаимодействия между муравьями.

Перспективными путями улучшения муравьиных алгоритмов являются различные адаптации параметров Q_i, α, β, ε.

Повышения эффективности представленного алгоритма можно добиться путем адаптивного управления многостадийной структурой связей.

Разработанный алгоритм покрытия имеет универсальный характер и может быть использован для решения широкого круга задач линейного целочисленного программирования.

Поддержка

Работа выполнена при финансовой поддержке грантов РФФИ №15-01-05297, №13-01-00596.

Литература

[1] Андерсон Д. Дискретная математика и комбинаторика. М.: Вильямс, 2003.

[2] O.Coudert, “On solving covering problems”, *in proceedings of 30th ACM/IEEE Design automation conference*, 1996, pp. 197-202.

[3] 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 , pp.1377-1387.

[4] Еремеев А.В., Заозерская Л.А., Колоколов А.А. Задача о покрытии множества:сложность, алгоритмы, экспериментальные исследования // Дискрет. анализ и исслед. опе-раций. Сер. 2. 2000. Т. 7, № 2. С. 22–46.

[5] Забиняко Г.И.Реализация алгоритмов решения задачи о покрытии множеств и анализ их эффективности.// Вычислительные технологии Том 12, № 6, 2007. – С. 50-58.

[6] Еремеев А. В. Генетический алгоритм для задачи о покрытии // Дискрет, анализ и исслед. операций. Сер. 2. 2000. Т. 7, № 1. С. 47-60.

[7] Beasley J. E., Chu P. C. A genetic algorithm for the set covering problem // European J. Oper. Res. 1996. V. 94, N 2. P. 394-404

[8] Нгуен Минь Ханг. Применение генетического алгоритма для задачи нахождения покрытия множества // Труды института системного анализа РАН, 2008. Т. 33. – С. 206-21

[9] 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.

[10] Лебедев Б.К., Лебедев В.Б. Покрытие на основе метода роя частиц // Сборник научных трудов XIII Всероссийской научно-технической конференции "НЕЙРОИНФОРМАТИКА-2011”.Ч.2 – М.: Изд-во Физматлит, 2011. - С. 93-103.

[11] V.M. Kurejchik, B.K. Lebedev, O.B. Lebedev. A Simulated Evolution-Based Solution of the Cover Problem. //Journal of Computer and Systems Sciences International, 2009, Vol. 48, №16 pp. 95-109

[12] Есипов Б.А,.Муравьев. В.В Исследование алгоритмо решения обобщенно задачи о минимальном покрытии. // Известия Самарского научного центра Российской академии наук, 2014. Т. 16. №4(2).

[13] Mazumder P., Rudnick E. Genetic Algorithm For VLSI Design, Layout & Test Automation. India, Pearson Education, 2003.

[14] Caprara A., Toth P., Fischetti M. Algorithms for the set covering problem // Anals of Oper. Res. 2000. Vol. 98. P. 353–371.

[15] 17.Balas E. , Ho A. Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study // Math. Program. Study. 1980. V. 12. P. 37-60.

[16] A. P. Engelbrecht. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.

[17] МакКоннелл Дж. Основы современных алгоритмов. Москва, Техносфера, 2004.

[18] Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поисковая адаптация: теория и практика. М.: Физматлит, 2006.

[19] M. Dorigo and T. Stützle. Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.

[20] Лебедев О.Б. Модели адаптивного поведения муравьиной колонии в задачах проектирования. - Таганрог: Изд-во ЮФУ, 2013.

[21] Лебедев О.Б. Покрытие методом муравьиной колонии /Двенадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2010. Труды конференции. Т. 2. М.: Физматлит, 2010. С. 423-431.

[22] Beasley J.E. An algorithm for set covering problems // European J. Oper. Res. 1987. Vol. 31.P. 85–93.

[23] Balas E., Carrera M.C. A dynamic subgradient-based branch-and-bound procedure for set covering // Oper. Res. 1996. Vol. 44, N 6. P. 875–890.

[24] Caprara A., Fishetti M., Toth P. A heuristic method for the set covering problem //Oper. Res. 1999. Vol. 47, № 5. P. 730–743.

Bioinspired VLSI chip planning methods

B.K. Lebedev, O.B . Lebedev

Southern Federal University, lebedev.b.k@gmail.com

Abstract

***Keywords* — computer-aided design, planning, VLSI, ant colony, ant tree, adaptive behavior, optimization.**

The planning problem is formulated as follows. There is a set of modules $M = \{m_i \mid i=1,2,\ldots,n\}$. Floorplan for the set of modules M is a rectangle R divided by vertical and horizontal lines into the set of rectangular blocks r; each such block contains module m_i. In our work, we use slicing floorplan with a slicing structure. Slicing structure can be represented by a binary tree. The leaves of this tree are vertices corresponding to regions, and the internal vertices correspond to cuts. The purpose of optimization is minimization of the total area of plan *R*, at observance of restrictions. In the work, for the decision of VLSI planning problem a new swarm intelligence paradigm, offered by authors, is used – *an ant tree* (*trees ant colony optimization* (T-ACO)), based on idea of an indirect exchange – Stigmergy, allowing to carry out tree synthesis. Presentation of the optimization problem as a paradigm of T-ACO is based on two key points: formation of the graph to find solutions *G*, and cultivation of trees on graph *G*. Graph $G = (S \cup C \cup M,\ U)$, $C = \{c_i \mid i = 1,2,\ ...,\ n_c\}$, corresponding to the cuts, $M = \{m_i \mid i=1,2,...,n\}$ – the set of modules, *U* – the set of edges, are constructed in the following manner. In the beginning, a complete graph is formed on the vertices of set *C*. The edges are undirected. The starting vertex *S* is introduced, which is linked with each vertex of *C*. Further, each vertex c_i is linked with all vertices n_m of the set *M*. In general, the search for a solution of the problem is carried out by team of ants $A = \{a_k \mid k = 1,2,\ ...,\ n_k\}$. Construction of tree by ants begins with starting vertex *S*. Process of finding solutions is iterative. At each iteration of the ant tree algorithm each ant a_k grows a binary tree in graph *G*, which is the solution of the problem. Each iteration *l* consists of three phases. In the first phase of each iteration, each ant a_k grows its own binary tree D_k. Binary tree is grown on the basis of graph *G* from the root to the leaves. At each step of growing the tree, vertex of graph *G* is selected, which is linked with one of the previously selected vertices and included in the tree edges. In the second stage of iteration, each ant lay pheromone on the edges of the tree constructed by him. In the third stage, the total evaporation of pheromone on the edges of the graph *G is calculated*. After all of the steps in the iteration the agent with the best solution is found, which is stored. The time complexity of the algorithm is $O(n^2)$, where *n* is the number of modules.

REFERENCES

[1] Anderson D. Diskretnaja matematika i kombinatorika. M.: Vil'jams, 2003.

[2] O.Coudert, “On solving covering problems”, in proceedings of 30th ACM/IEEE Design automation conference, 1996, pp. 197-202.

[3] 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 , pp.1377-1387.

[4] Eremeev A.V., Zaozerskaja L.A., Kolokolov A.A. Zadacha o pokrytii mnozhestva:slozhnost', algoritmy, jeksperimental'nye issledovanija // Diskret. analiz i issled. ope-racij. Ser. 2. 2000. T. 7, № 2. S. 22–46.

[5] Zabinjako G.I.Realizacija algoritmov reshenija zadachi o pokrytii mnozhestv i analiz ih jeffektivnosti.// Vychislitel'nye tehnologii Tom 12, № 6, 2007. – C. 50-58.

[6] Eremeev A. V. Geneticheskij algoritm dlja zadachi o pokrytii // Diskret, analiz i issled. operacij. Ser. 2. 2000. T. 7, № 1. S. 47-60.

[7] Beasley J. E., Chu P. C. A genetic algorithm for the set covering problem // European J. Oper. Res. 1996. V. 94, N 2. P. 394-404

[8] Nguen Min' Hang. Primenenie geneticheskogo algoritma dlja zadachi nahozhdenija pokrytija mnozhestva // Trudy instituta sistemnogo analiza RAN, 2008. T. 33. – S. 206-21

[9] Kurejchik V.M., Lebedev B.K., Lebedev O.B. Reshenie zadachi pokrytija na osnove jevoljucionnogo modelirovanija //Teorija i sistemy upravlenija. 2009, №1, S. 101-116.

[10] Lebedev B.K., Lebedev V.B. Pokrytie na osnove metoda roja chastic // Sbornik nauchnyh trudov XIII Vserossijskoj nauchno-tehnicheskoj konferencii "NEJROINFORMATIKA-2011”.Ch.2 – M.: Izd-vo Fizmatlit, 2011. - C. 93-103.

[11] V.M. Kurejchik, B.K. Lebedev, O.B. Lebedev. A Simulated Evolution-Based Solution of the Cover Problem. //Journal of Computer and Systems Sciences International, 2009, Vol. 48, №1b pp. 95-109

[12] Esipov B.A,.Murav'ev. V.V Issledovanie algoritmo reshenija obobshhenno zadachi o minimal'nom pokrytii. // Izvestija Samarskogo nauchnogo centra Rossijskoj akademii nauk, 2014. T. 16. №4(2).

[13] Mazumder P., Rudnick E. Genetic Algorithm For VLSI Design, Layout & Test Automation. India, Pearson Education, 2003.

[14] Caprara A., Toth P., Fischetti M. Algorithms for the set covering problem // Anals of Oper. Res. 2000. Vol. 98. P. 353–371.

[15] 17.Balas E. , Ho A. Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study // Math. Program. Study. 1980. V. 12. P. 37-60.

[16] A. P. Engelbrecht. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.

[17] MakKonnell Dzh. Osnovy sovremennyh algoritmov. Moskva, Tehnosfera, 2004.

[18] Kurejchik V.M., Lebedev B.K., Lebedev O.B. Poiskovaja adaptacija: teorija i praktika. M.: Fizmatlit, 2006.

[19] M. Dorigo and T. Stützle. Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.

[20] Lebedev O.B. Modeli adaptivnogo povedenija murav'inoj kolonii v zadachah proektirovanija. - Taganrog: Izd-vo JuFU, 2013.

[21] Lebedev O.B. Pokrytie metodom murav'inoj kolonii /Dvenadcataja nacional'naja konferencija po iskusstvennomu intellektu s mezhdunarodnym uchastiem KII-2010. Trudy konferencii. T. 2. M.: Fizmatlit, 2010. S. 423-431.

[22] Beasley J.E. An algorithm for set covering problems // European J. Oper. Res. 1987. Vol. 31.P. 85–93.

[23] Balas E., Carrera M.C. A dynamic subgradient-based branch-and-bound procedure for set covering // Oper. Res. 1996. Vol. 44, N 6. P. 875–890.

[24] Caprara A., Fishetti M., Toth P. A heuristic method for the set covering problem //Oper. Res. 1999. Vol. 47, № 5. P. 730–743.