

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *John Koza*. Genetically Breeding Populations of Computer Programs to Solve Problems of Artificial Intelligence. Proceedings of the Second International Conference on Tools for AI. Herndon, Virginia, November 6-9, 1990. Los Alamitos, CA: IEEE Computer Society Press. – P. 819-827.
2. *John Koza*. Integrating Symbolic Processing into Genetic Algorithms. Workshop on Integrating Symbolic and Neural Processes at AAAI-90 in Boston. July 29, 1990.
3. *John Koza*. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: The MIT Press, 1992. – 840 p.
4. *Zelinka Ivan*. Symbolic regression – an overview. <http://www.mafy.lut.fi/EcmiNL/older/ecmi35/node70.html>.
5. *Ricardo Poli, William B. Langdon, Nicolas F. McPhee*. A field guide to Genetic programming. <http://www.gp-field-guide.org.uk>. 2008. – 250 p.
6. *Курейчик В.М., Родзин С.И.* Эволюционные алгоритмы: генетическое программирование // Известия академии наук. Теория и системы управления. – 2002. – № 1.

Лисяк Мария Владимировна

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.
E-mail: maria-lisyak@yandex.ru.
347928, г. Таганрог, пер. Некрасовский, 44.
Тел.: 88634360524.

Lisyak Maria Vladimirovna

Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.
E-mail: maria-lisyak@yandex.ru.
44, Nekrasovskiy, Taganrog, 347928, Russia.
Phone: +78634360524.

УДК 681.3.001.63

О.Б. Лебедев

ПЛАНИРОВАНИЕ СБИС НА ОСНОВЕ МЕТОДА МУРАВЬИНОЙ КОЛОНИИ*

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

Планирование СБИС; муравьиная колония; оптимизация.

O.B. Lebedev

PLANNING VLSI ON THE BASIS OF THE ANT COLONY METHOD

New technologies, principles and mechanisms of the planning VLSI decision using mathematical methods in which principles of natural mechanisms of decision-making are put in pawn are offered. For compact representation of the slicing floorplan a modification Polish expression

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

is used. It has allowed to create the space of decisions in which 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.

Planning VLSI; ant colony; optimization.

Введение. Планирование СБИС заключается в размещении на поле кристалла блоков, имеющих форму прямоугольников. В результате планирования строится план кристалла, представляющий собой охватывающий прямоугольник, разделенный горизонтальными и вертикальными сегментами на непересекающиеся прямоугольники, в которые следует поместить соответствующие блоки. Основной целью оптимизации является минимизация общей площади кристалла.

Задача планирования относится к классу *NP*. В течение последних лет были предложены различные подходы к решению проблемы планирования. Эти подходы могут быть классифицированы следующим образом: линейное и квадратичное программирование [1]; имитация отжига; основанные на ограничениях [2]; сила направленная парадигма [3]; основанные на геометрической дуализации списков связей [3]; иерархические методы сверху-вниз и снизу-вверх [4]; метод кластеризации [5]; генетические алгоритмы (ГА); на основе поисковой адаптации [6,7] и др. Анализ существующих подходов к решению поставленной задачи показал, что удачными являются подходы, основанные на методах эволюционного моделирования [8,9]. Тем не менее, в последнее время для решения различных «сложных» задач, к которым относятся и задачи планирования всё чаще используются способы, основанные на применении методов искусственного интеллекта [10].

В последние годы интенсивно разрабатывается научное направление, объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений [11-13]. Наблюдение живых организмов подтверждает, что адаптация, выступает движущей силой эволюционного развития. Механизмы генетического поиска, реализующие случайные изменения, часто не находят ту их цепочку, которая приводит к оптимальному решению. Для этого нужны «осмысленные» изменения, направленные в сторону глобального оптимума. Такие свойства как раз присущи адаптивным поисковым процедурам на основе самообучения и самоорганизации. Адаптация на основе самообучения и самоорганизации доминирует в процессе существования и развития живых организмов. Идея муравьиного алгоритма [14] - 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 благодаря которому, в целом, колония представляет собой разумную многоагентную систему [14]. Особенности являются наличие непрямого обмена, который и используется в муравьиных алгоритмах. Непрямой обмен представляет собой разнесённое во времени взаимодействие, при котором одна особь изменяет некоторую область окружающей среды, а другие используют эту информацию позже, когда в неё попадают. В работе используется представление задачи планирования в виде адаптивной муравьиной системы, на основе сочетания принципов самообучения, самоорганизации.

Основные положения. Проблема планирования формулируется следующим образом [1]. Имеется множество модулей $M = \{m_i | i=1,2,\dots,n\}$. Каждый модуль характеризуется тройкой $\langle S_i, l_i, t_i \rangle$, где S_i – площадь модуля, а параметры l_i и t_i задают нижнюю и верхнюю границу значения h_i / w_i , т.е.

$$l_i \leq h_i / w_i \leq t_i \quad (1)$$

где h_i – высота модуля, w_i – ширина модуля. В качестве плана кристалла будем использовать план, полученный путем рекурсивного использования “гильтинного разреза”, т.е. последовательного разрезания прямоугольников на две части. На рис. 1,а представлен план, а на рис.1,б – соответствующее ему дерево $D=\{d_j \mid j = 1, 2, \dots, 2n-1\}$ “гильтинного разреза”, листьями которого являются вершины, соответствующие блокам, а внутренние вершины соответствуют разрезам: V – вертикальный, H – горизонтальный.

План для множества модулей M представляет собой прямоугольник R , разрезанный вертикальными и горизонтальными линиями на множество областей r_i , в каждую из которых помещается соответственно модуль m_i . На дереве цифрами помечены вершины, соответствующие разрезам, причем V – вертикальный разрез, а H – горизонтальный разрез. Буквами помечены вершины, соответствующие областям. Каждая область r_i , предназначенная для размещения модуля m_i , имеет размеры x_i и y_i . При соблюдении ограничений (1) размеры области должны также соответствовать ограничениям

$$S_i \leq x_i \cdot y_i, h_i \leq y_i, w_i \leq x_i. \quad (2)$$

Цель оптимизации – минимизация общей площади плана R , при соблюдении ограничений (1), (2).

Формирование плана методом свертки. Задать план это: задать структуру дерева разрезов, т.е. последовательность бинарных разрезов; для внутренних вершин дерева, соответствующих разрезам, указать тип разреза H или V ; помечить листья дерева номерами модулей; для модулей с фиксированными размерами указать их ориентацию.

На основе этой информации построение плана осуществляется путем последовательной бинарной свертки областей по дереву разрезов, начиная от листьев дерева. Каждой внутренней вершине дерева разрезов будет соответствовать область, полученная в результате бинарной свертки поддерева, имеющего корнем эту внутреннюю вершину. Будем считать, что разрез с номером i , разрезает вершину d_i (область u_i). Вначале свертки каждой вершине d_i , являющейся листом дерева разрезов, ставится в соответствие область u_i с размерами $x_i = h_i$, $y_i = w_i$, равными размерам модуля m_i . Пусть вершины d_i и d_j являются дочерними вершинами вершины d_k и пусть для областей u_i и u_j , соответствующих d_i и d_j , определены нижние границы их размеров (x_i, y_i) , (x_j, y_j) .

Процесс бинарной свертки представляет собой слияние областей u_i и u_j , формирование области u_k , определение размеров для u_k и новых размеров для u_i и u_j . Обозначим через $\max(x_1, x_2)$ максимальное значение из x_1 и x_2 . При слиянии по горизонтали $y_k = \max(y_i, y_j)$; $x_k = x_i + x_j$; y_i и y_j будут иметь размер, равный $\max(y_i, y_j)$. При слиянии по вертикали $y_k = y_i + y_j$; $x_k = \max(x_i, x_j)$; x_i и x_j будут иметь размер, равный $\max(x_i, x_j)$. Пусть для примера (рис. 1) при фиксированных размерах модулей ($m_l - m_g$) после метризации план с размещенными в областях модулями имеет вид, представленный на рис. 1,в. Будем различать размеры (h_i, w_i) модуля m_i от размеров (x_i, y_i) области u_i , в которую помещен модуль (блок) m_i .

Размеры областей и описывающего прямоугольника плана определяются путем последовательной свертки областей по дереву разрезов, исходя из размеров модулей, помещаемых в эти области.

Общая структура представления решений. Введём алфавит $A = \{M, R\}$. Структуру дерева разрезов можно задать используя на базе алфавита A польское выражение для бинарного дерева, где множество $M = \{m_i \mid i=1, 2, \dots, n_x\}$ соответствует листьям дерева разрезов (областям), а множество $R = \{H, V\}$ – соответствует

разрезам. Польское выражение для дерева, представленного на рис. 1,б имеет вид: $D = m_1 m_2 V m_3 m_4 V H m_5 m_6 H m_7 m_8 m_9 H V H V$.

Процесс восстановления дерева по польскому выражению достаточно прост. Последовательно слева направо просматривается польское выражение, и отыскиваются буквы **H** или **V**, соответствующие разрезам. Каждый такой разрез объединяет два ближайших образованных на предыдущих шагах подграфа, расположенных в польской записи слева от буквы **H** или **V**. Проиллюстрируем процесс свертки с помощью скобок: $D = ((m_1 m_2 V) (m_3 m_4 V) H) (m_5 m_6 H) (m_7 (m_8 m_9 H) V) H) V$.

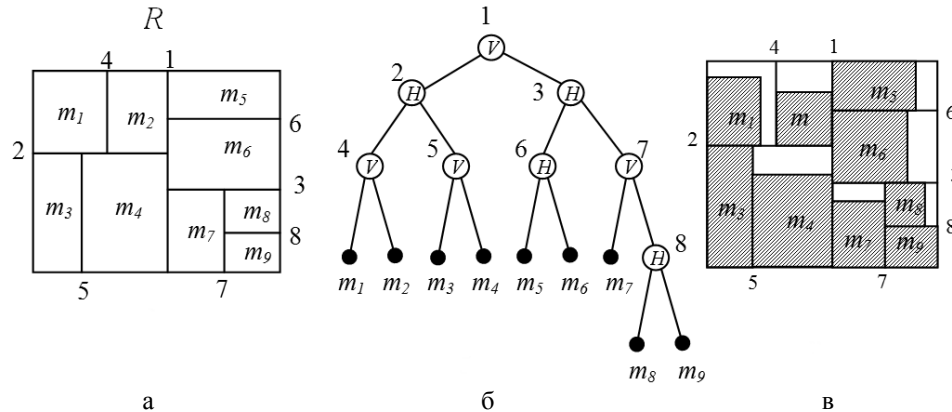


Рис. 1. а – план гильотинного разреза; б – дерево гильотинного разреза; в – план с размещенными модулями

Отметим основные свойства польского выражения, выполнение которых необходимо, чтобы записи соответствовало дерево разрезов. Обозначим через n_M – число элементов польского выражения принадлежащих множеству M , а через n_R – число элементов, соответствующих разрезам.

1. Для каждого модуля m_i возможны два способа (две ориентации) размещения в области u_i . Пусть первой ориентации модуля m_i соответствует обозначение – m_i^1 , а при второй ориентации m_i^2 .

2. В состав выражения входят по одному разу все элементы множества $M = \{m_i \mid i=1, 2, \dots, n_M\}$ с одной из меток – m_i^1 либо m_i^2 .

3. Для дерева разрезов всегда выполняется равенство $n_M = n_R + 1$.

4. Если в польском выражении провести справа от буквы **H** или **V** сечение, то слева от сечения число элементов, принадлежащих множеству M , больше числа элементов, соответствующих разрезам, по крайней мере, на единицу.

5. При просмотре выражения слева направо, первый элемент, соответствующий разрезам в польском выражении может появиться только после двух элементов, принадлежащих множеству M .

Назовем польское выражение D , построенное на базе алфавита $A = \{M, R\}$ легитимным, если оно удовлетворяет вышеперечисленным условиям. Таким образом, легитимное выражение D является символьным представлением решения задачи планирования. Различные решения получаются путём комбинирования взаимным расположением элементов алфавита $A = \{M, R\}$, удовлетворяющими условиям (1-5). В работе пространство решений представляется множеством легитимных выражений D . Поиск решения сводится к поиску такого легитимного выражения D , которое оптимизирует показатель качества (критерий).

Организация поисковых процедур на основе моделировании адаптивного поведения муравьиной колонии. Поиск решений осуществляется на полном

графе поиска решений $G=(X,U)$, где $X= X_1 \cup X_2 \cup X_3 \cup X_4$. Вершины множества $X_1 = \{x_{1i} | i=1,2,\dots, n_M\}$ соответствуют модулям $m^1_{i_s}$, размещаемым в первой ориентации. Вершины множества $X_2 = \{x_{2i} | i=1,2,\dots, n_M\}$ соответствуют модулям $m^2_{i_s}$, размещаемым во второй ориентации. Вершины множества $X_3 = \{x_{3i} | i=1,2,\dots, n_R\}$ соответствуют горизонтальным разрезам H . Вершины множества $X_4 = \{x_{4i} | i=1,2,\dots, n_R\}$ соответствуют вертикальным разрезам V .

В общем случае поиск решения задачи планирования осуществляется коллективом муравьев $Z=\{z_k | k=1,2,\dots,l\}$. На каждой итерации муравьиного алгоритма каждый муравей z_k строит свое конкретное решение задачи планирования. Решением является маршрут в графе $G=(X,U)$, включающий n_M вершин, принадлежащих множествам X_1 или X_2 , n_R вершин, принадлежащих множествам X_3 или X_4 , построенный в соответствии с условиями 1-5. В этом случае построенный маршрут представляется в виде легитимного выражения D .

Для равномерного распределения муравьев и создания равных стартовых условий в качестве начальных вершин у формируемых муравьями маршрутов используются вершины множеств X_1 и X_2 общим числом $2 \cdot n_M$. Другими словами число решений формируемых муравьями на каждой итерации равно $2 \cdot n_M$.

Моделирование поведения муравьев в задаче планирования связано с распределением феромона на ребрах графа G . На начальном этапе на всех ребрах графа G откладывается одинаковое (небольшое) количество феромона Q/v , где $v=|U|$. Параметр Q задается априори. Процесс поиска решений итерационный. Каждая итерация l включает три этапа. На первом этапе муравей находит решение, на втором этапе откладывает феромон, на третьем этапе осуществляется испарения феромона. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромон откладывается агентом на ребрах после полного формирования решения. На первом этапе каждой итерации каждый k -й муравей формирует свой собственный маршрут D_k . Процесс построения маршрута D_k пошаговый. На каждом шаге t агент применяет вероятностное правило выбора следующей вершины для включения ее формируемый маршрут $D_k(t)$. Для этого формируется множество вершин $X_k(t) \in X$, таких, что каждая из вершин $x_i \in X_k(t)$ может быть добавлена в формируемый маршрут $D_k(t)$ с соблюдением условий 1-5. Пусть $e_k(t)$ – последняя вершина маршрута $D_k(t)$. Агент просматривает все вершины $x_i \in X_k(t)$. Для каждой вершины $x_i \in X_k(t)$ рассчитывается параметра f_{ik} – суммарный уровень феромона на ребре графа G , связывающего x_i с вершиной $e_k(t)$.

Вероятность P_{ik} включения вершины $x_i \in X_k(t)$ в формируемый маршрут $D_k(t)$ определяется следующим соотношением

$$P_{ik} = f_{ik} / \sum_i f_{ik} . \quad (3)$$

Агент с вероятностью P_{ik} выбирает одну из вершин, которая включается маршрут $D_k(t)$.

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

Количество феромона $\Delta\tau_k(l)$, откладываемое муравьем z_k на каждом ребре построенного маршрута D_k , определяется следующим образом:

$$\Delta\tau_k(l) = Q / F_k(l), \quad (4)$$

где l – номер итерации, Q – общее количество феромона, откладываемое муравьем на ребрах маршрута D_k , $F_k(l)$ – целевая функция для решения, полученного муравьем z_k на l -ой итерации. Чем меньше $F_k(l)$, тем больше феромона откладывается

на ребрах построенного маршрута и, следовательно, тем больше вероятность выбора этих ребер при построении маршрутов на следующей итерации.

После того, как каждый агент сформировал решение и отложил феромон, на третьем этапе происходит общее испарение феромона на ребрах полного графа G в соответствии с формулой (5).

$$f_{ik} = f_{ik}(1 - \rho), \quad (5)$$

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 и 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 исходными данными формируется граф поиска решений G .
2. Определяются число агентов и вершины, в которые они помещаются.
3. Задается значение параметра Q_i и число итераций – N_i .
4. На всех ребрах графа G откладывается начальное количество феромона.
5. На первом этапе каждой итерации на ГПР G каждым агентом z_k строится легитимный маршрут D_k .
6. Для каждого решения D_k , с помощью алгоритма свертки находится значение целевой функции $F_k(l)$.
7. На ребрах каждого найденного маршрута в графе G откладывается феромон. Количество феромона, откладываемого каждым агентом, пропорционально $F_k(l)$.
8. Испарение феромона на ребрах (или вершинах) графов.
9. Выбор лучшего решения, полученного на протяжении всех выполненных итераций.
10. Если все итерации выполнены, то конец работы алгоритма, в противном случае переход к пункту 5 для выполнения очередной итерации.

Заключение. Для компактного представления решения задачи планирования используется модифицированная польская запись. Это позволило создать пространство решений, в рамках которого организован поисковый процесс, базирующийся на моделировании адаптивного поведения муравьиной колонии.

Экспериментальные исследования проводились на IBM PC. 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 %. В среднем запуск программы обеспечивает нахождения решения, отличающегося от оптимального менее чем на 1 %.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Kahng A.B. "Classical Floorplanning Harmful", *ISPD 2000*. – P. 207-213.
2. J. Cong et al. "Microarchitecture Evolution With Physical Planning", *DAC, 2003*. – P. 32-35.
3. N. Adya et al. "Unification of Partitioning, Placement and Floorplanning". *DAC, 2004*. – P. 550-557.
4. Liu H. Chen T. Chou A. Aziz D.F. Wong. "Integrated Power Supply Planning and Floorplanning", *IEEE Trans. on CAD, 2004*.
5. Cheng L., D.F. Wong. "Floorplan Design For Multi-million Gate FPGAs", *DAC, 2004*. – P. 292-313.

6. *Лебедев В. Б.* Планирование СБИС методом адаптивного поиска // Известия ТРТУ. – 2000. – № 2 (16). – С. 168-177.
7. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Поисковая адаптация: теория и практика. – М.: Физматлит, 2006.
8. *Лебедев Б.К.* Методы поисковой адаптации в задачах автоматизированного проектирования СБИС: Монография. – Таганрог: Изд-во ТРТУ, 2000.
9. *Курейчик В.М.* Модифицированные генетические операторы // Известия ЮФУ. Технические науки. – 2009. – № 12 (101). – С. 7-15.
10. *МакКоннелл Дж.* Основы современных алгоритмов. – М.: Техносфера, 2004.
11. *Di Caro G., Ducatelle F., Gambardella L.M.* AntHocNet: An adaptive nature inspired algorithm for routing in mobile ad hoc networks // European Transactions on Telecommunications. – 2005. – № 16(5). – P. 443-455.
12. *Engelbrecht P.* Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.
13. *Курейчик В.В., Полуфанова Е.Е.* Эволюционная оптимизация на основе алгоритма колонии пчел // Известия ЮФУ. Технические науки. – 2009. – № 12. – С. 41-46.
14. *Dorigo M. and Stützle T.* Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.

Лебедев Олег Борисович

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования “Южный федеральный университет” в г. Таганроге.
E-mail: lbk@tsure.ru.
347928, г. Таганрог, пер. Некрасовский, 44.
Тел.: 88634371743.

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: +8634371743.