

МОДИФИЦИРОВАННЫЙ МУРАВЬИНЫЙ АЛГОРИТМ ПЛАНИРОВАНИЯ СБИС НА БАЗЕ КОМПОЗИТНОЙ МОДЕЛИ ПРОСТРАНСТВА РЕШЕНИЙ

Б.К. Лебедев

lebedev.b.k@gmail.com

О.Б. Лебедев

lebedev.ob@mail.ru

А.А. Жиглатый

artemiy.zhiglaty@gmail.com

Инженерно-технологическая академия ЮФУ,
Таганрог, Ростовская обл., Российская Федерация

Аннотация

Сформирован план кристалла путем рекурсивного использования «гильотинного разреза». Задать план — это задать структуру дерева разрезов, т. е. последовательность бинарных разрезов; для внутренних вершин дерева указать тип разреза H или V ; пронумеровать листья дерева и указать ориентацию модулей. Структуру бинарного дерева разрезов можно задать, используя на базе алфавита $A = \{M, TR\}$ польское выражение, где множество букв $M = \{m_i | i = 1, 2, \dots, n_M\}$ соответствует листьям дерева разрезов (областям), а множество $R = \{H, V\}$ соответствует разрезам. Предложен способ и методы решения задач планирования СБИС на основе модифицированной муравьиной колонии. Задача синтеза дерева разрезов плана с выбором типов разрезов, идентификацией и ориентацией модулей сведена к задаче формирования модифицированного польского выражения с идентификацией элементов на композитной модели пространства решений, включающей в себя множество альтернативных вершин. Для отражения коллективной эволюционной памяти в течение жизни популяции муравьев и для формирования решения задачи использован полный граф $G = (X, U)$ с альтернативными состояниями вершин. Каждая вершина может находиться в одном из двух альтернативных состояний (α или β), соответствующих ориентации модуля или типу разреза. Задача синтеза польского выражения сформулирована как задача поиска минимального по стоимости маршрута на графе

Ключевые слова

Планирование, СБИС, дерево разрезов, модифицированная польская запись, композитная структура, муравьиная колония, гибридизация

поиска решений $G = (X, U)$. Отличительная особенность заключается в том, что при построении маршрута одновременно с выбором вершины $x_i \in X$ осуществляется выбор состояния этой вершины. Временная сложность алгоритма составляет $O(n^2)$. Эксперименты показали, что при больших размерностях временные показатели разработанного алгоритма превосходят показатели сравниваемых алгоритмов при лучших значениях целевой функции

Поступила 29.05.2019

© Автор(ы), 2019

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант РФФИ № 17-07-00997а)

Введение. Проблема планирования формулируется следующим образом [1–3]. 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 границы значения h_i/w_i , т. е. $l_i \leq h_i/w_i \leq t_i$, h_i — высота модуля, w_i — ширина модуля.

Основные проблемы задачи планирования кристалла СБИС — поиск подхода к представлению решения (плана) и построение оптимизационной процедуры поиска решения [1–3].

Все множество представлений принято разбивать на два класса: 1) гильотинный; 2) негильотинный. Гильотинная структура может быть получена путем рекурсивного деления прямоугольника на две части горизонтальными и/или вертикальными разрезами. Гильотинное представление имеет такие преимущества, как небольшие затраты на кодирование и меньшее пространство поиска, приводящие к более быстрому построению плана. Кроме того, оно является гибким при работе с фиксированными, предварительно размещенными, гибкими и прямолинейными модулями, но в пространстве решений гильотинной структуры могут отсутствовать оптимальные решения.

В течение последних лет предложены различные подходы к решению проблемы планирования. Эти подходы могут быть классифицированы следующим образом: линейное и квадратичное программирование [2]; имитация отжига [4]; основанные на ограничениях [5]; силовонаправленная парадигма [6]; основанные на геометрической дуализации списков связей [7]; иерархические методы сверху-вниз и снизу-вверх [8]; метод кластеризации [9]; генетические алгоритмы (ГА) [10]; эволюционные алгоритмы [11]; на основе поисковой адаптации [12]; биоинспирированные [13, 14]; меметические [15] и др. Анализ существующих подходов к реше-

нию поставленной задачи показал, что удачными являются подходы, основанные на методах эволюционного моделирования.

В работе предложены гибридный подход, способы и методы представления задачи планирования СБИС, базирующиеся на моделировании адаптивного поведения муравьиной колонии.

Представление плана кристалла СБИС. План для множества модулей M представляет собой прямоугольник R , разрезанный вертикальными и горизонтальными линиями на множество областей r_i , в каждую из которых помещается модуль m_i . План кристалла формируется путем рекурсивного использования «гильотинного разреза», т. е. последовательного разрезания прямоугольников на две части. Модель плана представляется в виде дерева «гильотинного разреза» DR , листьями которого являются вершины, соответствующие блокам, а внутренние вершины соответствуют разрезам: V — вертикальный, H — горизонтальный. Каждая область r_i , предназначенная для размещения модуля m_i , имеет размеры x_i и y_i . При соблюдении ограничений размеры области также должны соответствовать ограничениям:

$$S_i < x_i y_i, \quad h_i < y_i, \quad w_i < x_i.$$

Цель оптимизации — минимизация общей площади плана R при соблюдении ограничений.

Зададим план (рис. 1), для чего зададим структуру дерева разрезов, т. е. последовательность бинарных разрезов; для внутренних вершин дерева, соответствующих разрезам, укажем тип разреза H или V ; присвоим листьям дерева номера модулей; для модулей с фиксированными размерами укажем их ориентацию. Структуру дерева разрезов можно задать с использованием на базе алфавита $A = \{M, TR\}$ польского выражения для бинарного дерева, где $M = \{m_i | i = 1, 2, \dots, n_M\}$ — множество букв, соответствующее листьям дерева разрезов (областям), а $R = \{H, V\}$ — разрезам. Польское выражение для дерева, представленного на рис. 1, имеет вид $D = m_1 m_2 V m_3 m_4 V H m_5 m_6 H m_7 m_8 m_9 H V H V$. Здесь n_M — число листьев дерева; n_R — число разрезов. Выявлены основные свойства польского выражения, выполнение которых необходимо для соответствия записи дереву разрезов:

- 1) для дерева разрезов всегда выполняется равенство $n_M = n_R + 1$;
- 2) слева от любой буквы $r_j = \{H, V\}$ число букв $m_i \in M$ больше числа букв r_j .

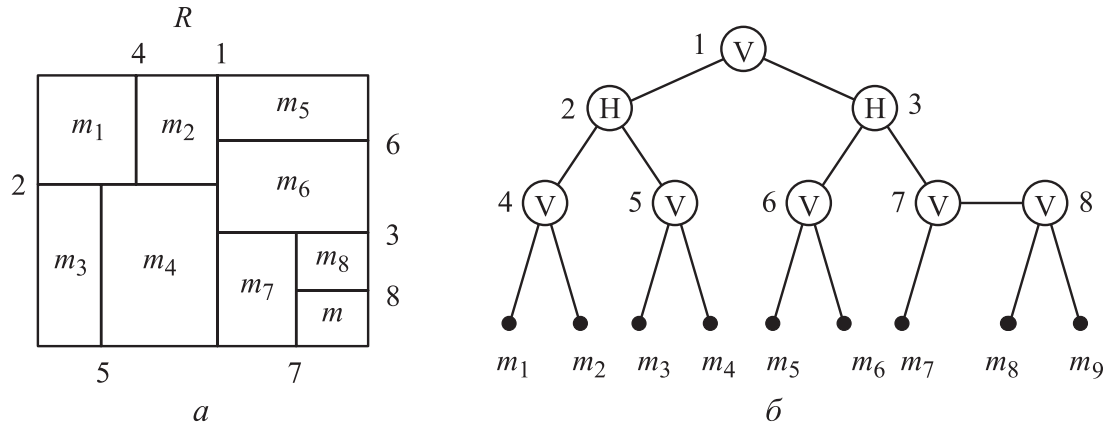


Рис. 1. План (а) и дерево (б) гильотинного разреза

Каноническая структура польского выражения не позволяет отображать ориентацию разрезов и модулей.

Задача синтеза дерева разрезов плана с выбором типов разрезов и ориентаций модулей сводится к задаче формирования модифицированного польского выражения с идентификацией элементов.

Предварительно формируется заготовка в виде вектора $D = \langle \bullet \bullet 1 \bullet 2 \bullet 3 \bullet 4 \dots \bullet m \rangle$ с пронумерованными позициями между знаками \bullet , которые соответствуют листьям дерева, знаки \bullet обозначают внутренние вершины дерева.

Муравьиный алгоритм планирования. Для того чтобы отразить коллективную эволюционную память в течение жизни популяции муравьев и сформировать решение задачи, использован полный граф $G = (X, U)$ с альтернативными состояниями вершин. Задача синтеза польского выражения формулируется как задача поиска минимального по стоимости маршрута на графе поиска решений $G = (X, U)$.

Поиск решений осуществляется на полном графе поиска решений $G = (X, U)$, где $X = X_1 \cup X_2$. Вершины множества $X_1 = \{x_{1i} | i = 1, 2, \dots, n_M\}$ соответствуют множеству модулей $M = \{m_i | i = 1, 2, \dots, n_M\}$. Каждая вершина $x_{1i} \in X_1$ может находиться в одном из двух альтернативных состояний (α или β), соответствующих типу ориентации модуля. Если модуль m_i размещается в первой ориентации, то x_{1i} находится в первом состоянии α . Если модуль m_i размещается во второй ориентации, то x_{1i} находится во втором состоянии β . Вершины множества $X_2 = \{x_{2i} | i = 1, 2, \dots, n_R\}$ соответствуют разрезам. Каждая вершина $x_{2i} \in X_2$ может находиться в одном из двух альтернативных состояний, соответствующих типу разреза. Если разрез горизонтальный H , то x_{2i} находится в первом состоянии α , если верти-

кальный V , то — во втором состоянии β . Если ориентация модулей зафиксирована, то вершины множества X_1 находятся в одном состоянии α .

Задача формулируется как задача поиска минимального по стоимости маршрута на графе поиска решений $G = (X, U)$. Отличительная особенность поисковой процедуры — при построении маршрута одновременно с выбором вершины $x_i \in X$ осуществляется выбор состояния этой вершины. Обозначим состояние вершины $x_i \theta_i = \{\alpha, \beta\}$. Другими словами, для каждой пары взаимоисключающих (альтернативных) состояний (значений) вершины в процессе прокладки маршрута выбирается один альтернативный вариант из двух возможных. Обозначим вершину x_i в первом состоянии как $x_{\alpha i}$, а во втором — как $x_{\beta i}$.

В общем случае поиск решения задачи планирования осуществляется коллективом муравьев $Z = \{z_k | k = 1, 2, \dots, l\}$. Для равномерного распределения муравьев и создания равных стартовых условий в качестве начальных вершин у формируемых муравьями маршрутов используются вершины множества X_1 общим числом n_M , т. е. число формируемых муравьями на каждой итерации решений равно n_M .

Моделирование поведения муравьев в задаче планирования связано с распределением феромона на ребрах и вершинах графа G . Для учета оценок состояний вершины x_i вводятся два счетчика α_i и β_i . На начальном этапе на всех ребрах графа G и на всех счетчиках вершин откладывается одинаковое (начальное) количество феромона ε (параметр задается априори). Процесс поиска решений итерационный. Каждая итерация l включает в себя три этапа. На первом этапе муравей находит решение, на втором откладывает феромон, на третьем осуществляется испарения феромона. В работе использован циклический (*ant-cycle*) метод муравьиных систем. В этом случае феромон откладывается агентом на ребрах после полного формирования решений популяцией. На первом этапе каждой итерации каждый муравей z_k формирует свой собственный маршрут M_k . Процесс построения маршрута M_k пошаговый. На каждом шаге t агент применяет вероятностное правило выбора следующей вершины для включения ее в формируемый маршрут $M_k(t)$. Для этого формируется множество вершин $X_k(t) \in X$, таких, что каждая вершина $x_i \in X_k(t)$ может быть добавлена в формируемый маршрут $M_k(t)$ с соблюдением указанных условий. Пусть $e_k(t)$ — последняя вершина маршрута $M_k(t)$. Агент просматривает все вершины $x_i \in X_k(t)$.

Для каждой вершины $x_i \in X_k(t)$ рассчитывается суммарный уровень феромона f_{ik} на ребре графа G , связывающего x_i с вершиной $e_k(t)$, а также суммарные уровни α_{ik} и β_{ik} феромона в счетчиках первого и второго состояний вершины x_i . Вероятность включения вершины $x_i \in X_k(t)$ с заданным состоянием θ_i в формируемый маршрут $D_k(t)$ определяется соотношениями:

$$\begin{aligned} P_{ik}^{\alpha} &= (f_{ik} + \alpha_{ik}) / \sum_i [(f_{ik} + \alpha_{ik}) + (f_{ik} + \beta_{ik})], \text{ если } x_i \text{ в состоянии } \alpha (\theta_i = \alpha); \\ P_{ik}^{\beta} &= (f_{ik} + \beta_{ik}) / \sum_i [(f_{ik} + \alpha_{ik}) + (f_{ik} + \beta_{ik})], \text{ если } x_i \text{ в состоянии } \beta (\theta_i = \beta). \end{aligned} \quad (1)$$

В соответствии с рассчитанным с помощью соотношений (1) распределением вероятности агент случайным образом выбирает одну из вершин $x_i \in X_k(t)$ в некотором состоянии θ_i , которая включается в маршрут $D_k(t)$. Построенный агентом z_k на итерации l маршрут D_k трансформируется в польское выражение, трансформирующееся в дерево разрезов с идентификацией вершин, а затем в план, для которого рассчитывается оценка $F_k(l)$. Эта оценка рассматривается в качестве оценки маршрута D_k .

На втором этапе итерации каждый муравей откладывает феромон на ребрах и счетчиках вершин построенного маршрута. Если $x_i \in D_k(t)$ в первом состоянии ($\theta_i = \alpha$), то феромон откладывается на счетчике α_i , если во втором, то — на счетчике β_i .

Количество феромона $\tau_k(l)$, откладываемое муравьем z_k на каждом ребре и счетчике вершины, включенных в построенный маршрут D_k , определяется по формуле:

$$\tau_k(l) = \delta Q / F_k(l), \quad (2)$$

где l — номер итерации; δ — коэффициент отложения; Q_i — базовое количество феромона, откладываемое муравьем на ребрах маршрута D_k ; $F_k(l)$ — целевая функция для решения, полученного муравьем z_k на l -й итерации. Чем меньше $F_k(l)$, тем больше феромона откладывается на ребрах и счетчиках вершин построенного маршрута и, следовательно, тем выше вероятность выбора этих ребер и состояний вершин 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 и счетчиках вершин полного графа G в соответствии с формулой

$$f_{ik} = f_{ik}(1 - \rho), \quad (3)$$

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), числа вершин графа n и числа муравьев m и определяется как $O(ln^2m)$.

Алгоритм планирования на основе метода муравьиной колонии формулируется следующим образом.

1. В соответствии с исходными данными формируется полный граф поиска решений G с альтернативными состояниями вершин.

2. Для каждой вершины графа поиска решений G формируются счетчики состояний α_i, β_i .

3. Задается число N_a агентов и вершин графа G , в которые они помещаются.

4. Задаются значения управляющих параметров: ε — начальная доза феромона; Q_i — базовая доза феромона; ρ — коэффициент обновления; N_l — число итераций.

5. На всех ребрах и счетчиках состояний вершин графа G откладывается начальное количество феромона ε .

6. $l = 1$ (l — номер итерации).

7. $k = 1$ (k — номер агента).

8. Агентом z_k строится на графе G легитимный маршрут $M_k(l)$.

9. Осуществляется трансформация маршрута $M_k(l)$ в представление плана в виде польской записи.

10. Представление плана в виде польской записи трансформируется в дерево разрезв $T_k(l)$.

11. Дерево разрезв $T_k(l)$ с помощью алгоритма свертки трансформируется в план $R_k(l)$, и находится значение целевой функции $F_k(l)$ плана, которая рассматривается как оценка качества маршрута $M_k(l)$.

12. Если $k < N_k$, то $k = k + 1$ и осуществляется переход к п. 8, иначе переход к п. 13.

13. На ребрах и счетчиках вершин каждого найденного маршрута в графе G откладывается феромон. Если $x_i \in M_k(t)$ в первом состоянии, то феромон откладывается на счетчике α_i , если $x_i \in D_k(t)$ во втором состоя-

нии, то — на счетчике β_i . Количество феромона $\tau_k(l)$, откладываемое муравьем z_k , пропорционально $F_k(l)$.

14. Феромон испаряется с ребер и счетчиков вершин графа G .

15. Осуществляется выбор лучшего решения, полученного на протяжении всех выполненных итераций.

16. Если все итерации выполняются, то конец работы алгоритма, в противном случае переход к п. 5 для очередной итерации.

Экспериментальные исследования. Эксперименты показали, что начальное количество феромона Ω должно быть в 10–12 раз больше среднего количества феромона $\tau_k(l)$, откладываемого муравьями на каждой итерации.

Для проведения экспериментов программы планирования методом муравьиной колонии (ПМК) использована процедура синтеза контрольных примеров с известным оптимумом $F_{\text{опт}}$ по аналогии с известным методом AFEKO — *Floorplanning Examples with Known Optimal Area* [16, 17]. Исследованию подвергались примеры, содержащие до 1000 модулей.

При проведении испытаний для каждого эксперимента фиксировалось число итераций, при которых алгоритм приходил к какому-либо решению. Оценкой качества служит «степень качества» — величина $F_{\text{опт}}/F$, где F — оценка качества полученного решения. На основе обработки экспериментальных исследований построена средняя зависимость степени качества от числа итераций (рис. 2, а) и от размера популяции (рис. 2, б). Исследования показали, что число итераций, при которых алгоритм находил лучшее решение, лежит в пределах 100–115. В среднем на 110-й итерации решение близко к оптимальному. При этом отклонение числа итераций в сторону увеличения составляло до 10 %, а в сторону уменьшения — до 35 %. Сравнение значений критерия, полученных муравьиным алгоритмом на тесто-

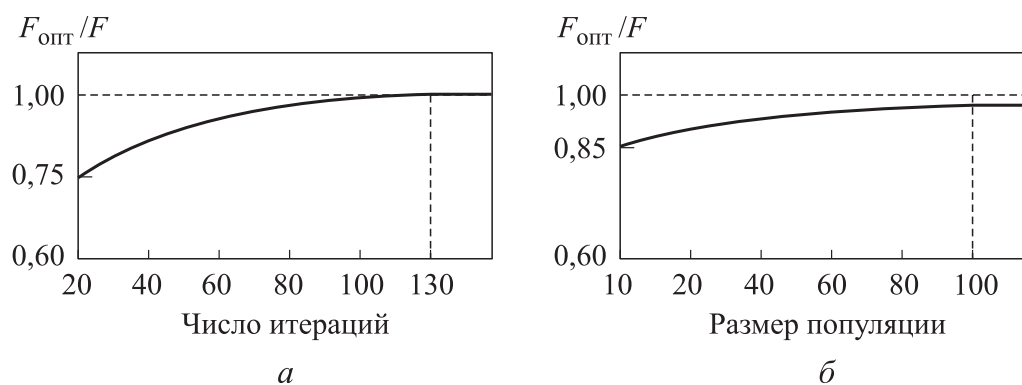


Рис. 2. Зависимости качества алгоритма от числа итераций (а) и размера популяции (б)

вых примерах, с известным оптимумом показало, что у 60 % примеров полученное решение оптимально, у 15 % примеров решения на 5 % хуже, а у 25 % примеров решения хуже не более чем на 2 %.

Проведены экспериментальные исследования, характеризующие зависимость времени выполнения алгоритма от размера задачи, и серия экспериментов по измерению продолжительности работы алгоритма от числа блоков. Общая оценка временной сложности лежит в пределах $O(n^2) - O(n^3)$.

Эксперименты проводились на MCNC (*Microelectronics Center of North Carolina*) образцах элементов. Использовано пять типов корпусов, в которых число элементов варьировалось от 9 до 49. Самый большой корпус — для платы *ami49* на 49 элементов с 42 интерфейсами ввода-вывода, 408 связями и 931 ножкой.

Результаты пяти прогонов программы планирования ПМК на основе парадигмы АСО для MCNC-бенчмарков приведены в табл. 1. Сравнительный анализ разработанных алгоритмов на основе АСО с другими алгоритмами планирования проведен на стандартных тестовых примерах и схемах (бенчмарках) [18, 19]. Данные примеры представляют собой стандартные промышленные цепи (блоки, схемы, ИС, БИС, СБИС). Сравнение выполнено по значению площади.

Таблица 1

Результаты работы программы планирования ПМК на основе АСО

Бенч-марк	Номер прогона					Среднее значение площади	Среднее время, мс
	1	2	3	4	5		
<i>apte</i>	49 396 354	49 396 354	49 396 354	49 468 310	49 468 310	49 425 136,4	105
<i>xerox</i>	24 895 479	24 873 723	24 957 366	24 946 938	25 099 333	24 954 567,8	106
<i>hp</i>	12 194 845	12 176 343	12 393 236	11 980 127	12 417 874	12 232 485	116
<i>ami33</i>	3 538 838	3 525 858	3 490 074	3 465 157	3 530 940	3 510 173,4	326
<i>ami49</i>	137 343 334	134 285 950	138 591 325	132 488 414	137 959 049	136 133 614,4	569

Для сравнения экспериментальных данных алгоритмов планирования были выбраны наиболее известные эволюционные алгоритмы: ГЕН — генетический [20]; АД — адаптивный [1]; ГА — гибридный [12]; МА — меметический [15]; ПМК — муравьиный [14]. Отметим, что результаты у сравниваемых алгоритмов получены на базе платформы *Ultra1-Spark*, а эксперименты разработанных алгоритмов проводились на ЭВМ типа *IBM PC* с процессором *Pentium*.

Представленные алгоритмы на основе парадигмы АСО находят решения, превосходящие результаты существующих алгоритмов по площади, кроме того, они оптимизируют длину соединений (табл. 2).

Таблица 2

Сравнение результатов работы алгоритмов

Бенч-марк	Алгоритм				
	ГЕН	АД	ГА	МА	ПМК
<i>apte</i>	49 425 136,4	49 425 136,4	49 396 354	49 396 354	49 396 354
<i>xerox</i>	24 954 567,8	24 954 567,8	24 895 479	24 873 723	24 957 366
<i>hp</i>	12 232 485	12 232 485	12 194 845	12 176 343	12 393 236
<i>ami33</i>	3 530 173,4	3 510 173,5	3 538 838	3 520 854	3 490 074
<i>ami49</i>	136 133 614,4	136 133 614,4	137 343 334	134 285 950	138 591 325

Экспериментальная временная сложность представленных алгоритмов на одной итерации при фиксированных значениях управляющих параметров составляет $O(n)$, а временная сложность существующих алгоритмов — $O(n^2)$. Эксперименты показали, что при больших размерностях временные показатели разработанных алгоритмов превосходят показатели сравниваемых алгоритмов при лучших значениях целевой функции. При меньшем времени работы у полученных с помощью разработанного алгоритма решений отклонение целевой функции от оптимального значения в среднем менее 6 %.

Заключение. Рассмотрены способы и методы представления задач планирования и размещения СБИС в виде эволюционных процессов, основанных на интеграции моделей адаптивного поведения биологических систем, композитных архитектур поиска решений, позволяющих работать с задачами большой размерности и получать качественные результаты за приемлемое время. Каноническая структура польского выражения не позволяет отображать ориентацию разрезов и модулей в дереве «гильотинного разреза». Задача синтеза дерева разрезов плана с выбором типов разрезов и ориентаций модулей в работе сведена к задаче формирования модифицированного польского выражения с идентификацией элементов. В свою очередь, задача синтеза модифицированного польского выражения сформулирована как задача поиска минимального по стоимости маршрута на графе поиска решений. В отличие от канонической парадигмы муравьиного алгоритма использована композитная модель пространства решений, граф поиска решений включает в себя множество альтернативных вершин. Это расширяет область применения и круг решаемых задач.

ЛИТЕРАТУРА

- [1] Курейчик В.М., Лебедев Б.К., Лебедев В.Б. Планирование сверхбольших интегральных схем на основе интеграции моделей адаптивного поиска. *Известия РАН. Теория и системы управления*, 2013, № 1, с. 84–101.
- [2] Лебедев Б.К., Лебедев О.Б., Лебедев В.Б. Методы, модели и алгоритмы размещения. Ростов н/Д, Изд-во ЮФУ, 2015.
- [3] Лебедев О.Б. Модели адаптивного поведения муравьиной колонии в задачах проектирования. Ростов н/Д, Изд-во ЮФУ, 2013.
- [4] Qi L., Xia Y., Wang L. Simulated annealing based thermal-aware floorplanning. *ICECC*, 2011, pp. 463–466. DOI: 10.1109/ICECC.2011.6067654
- [5] Chen J., Zhu W. A hybrid genetic algorithm for VLSI floorplanning. *ICIS*, 2010, pp. 128–132. DOI: 10.1109/ICICISYS.2010.5658785
- [6] Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. М., Изд-во МГТУ им. Н.Э. Баумана, 2014.
- [7] Chen G., Guo W., Chen Y. A PSO-based intelligent decision algorithm for VLSI floorplanning. *Soft Comput.*, 2010, vol. 14, iss. 12, pp. 1329–1337. DOI: <https://doi.org/10.1007/s00500-009-0501-6>
- [8] Banerjee P., Sangtani M., Sur-Kolay S. Floorplanning for Partially Reconfigurable FPGAs. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2011, vol. 30, iss. 1, pp. 8–17. DOI: 10.1109/TCAD.2010.2079390
- [9] Ерошенко И.Н. Разработка генетического алгоритма кластерного планирования СБИС. *Известия ЮФУ. Технические науки*, 2010, № 7, с. 54–60.
- [10] Лебедев Б.К., Лебедев В.Б. Планирование на основе роевого интеллекта и генетической эволюции. *Известия ЮФУ. Технические науки*, 2009, № 4, с. 25–33.
- [11] Курейчик В.В., Курейчик В.В. Архитектура гибридного поиска при проектировании. *Известия ЮФУ. Технические науки*, 2012, № 7, с. 22–27.
- [12] Kureichik V.M., Lebedev B.K., Lebedev O.B. Hybrid evolutionary algorithm of planning VLSI. *Proc. GECCO'10*. Portland, OR, 2010, pp. 821–822.
- [13] Лебедев Б.К., Лебедев О.Б. Биоинспирированные методы планирования кристалла СБИС. *Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)*, 2014, № 1, с. 171–176.
- [14] Лебедев О.Б. Планирование СБИС на основе метода муравьиной колонии. *Известия ЮФУ. Технические науки*, 2010, № 7, с. 67–73.
- [15] Potti S., Pothiraj S. GPGPU implementation of parallel memetic algorithm for VLSI floorplanning problem. In: Nagamalai D., Renault E., Dhanuskodi M. (eds). *Trends in Computer Science, Engineering and Information Technology. CCSEIT 2011. Communications in Computer and Information Science*, vol. 204. Berlin, Heidelberg, Springer, pp. 432–441. DOI: https://doi.org/10.1007/978-3-642-24043-0_44
- [16] Shanavas I.H., Gnanamurthy R.K. Evolutionary algorithmical approach for VLSI floorplanning problem. *IJCTE*, 2009, vol. 1, no. 4, pp. 461–464. DOI: 10.7763/IJCTE.2009.V1.75

- [17] Silvaso Library Platform™. URL: <http://www.nangate.com> (дата обращения: 15.02.2019).
- [18] MCNC: веб-сайт компании. URL: www.mcnc.org (дата обращения: 15.02.2019).
- [19] hMetis. URL: <http://glaros.dtc.umn.edu/gkhome/metis/hmetis/download> (дата обращения: 15.02.2019).
- [20] HB Suite. cadlab.cs.ucla.edu: веб-сайт.
URL: <http://cadlab.cs.ucla.edu/cpmo/HBsuite.html> (дата обращения: 15.02.2019).
- [21] Ma Q., Young E.F.Y. Multivoltage floorplan design. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2010, vol. 29, iss. 4, pp. 607–617.
DOI: 10.1109/TCAD.2010.2042895
- [22] Лебедев Б.К., Лебедев О.Б., Лебедева Е.М. Гибридный алгоритм ситуационного планирования траектории на плоскости в условиях частичной неопределенности. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2018, № 1, с. 76–93.
DOI: 10.18698/0236-3933-2018-1-76-93

Лебедев Борис Константинович — д-р техн. наук, профессор кафедры систем автоматизированного проектирования Инженерно-технологической академии ЮФУ (Российская Федерация, 347900, Ростовская обл., Таганрог, Некрасовский переулок, д. 44).

Лебедев Олег Борисович — канд. техн. наук, доцент кафедры систем автоматизированного проектирования Инженерно-технологической академии ЮФУ (Российская Федерация, 347900, Ростовская обл., Таганрог, Некрасовский переулок, д. 44).

Жиглатый Артемий Александрович — ассистент кафедры математического обеспечения и применения ЭВМ Инженерно-технологической академии ЮФУ (Российская Федерация, 347900, Ростовская обл., Таганрог, Некрасовский переулок, д. 44).

Просьба ссылаться на эту статью следующим образом:

Лебедев Б.К., Лебедев О.Б., Жиглатый А.А. Модифицированный муравьиный алгоритм планирования СБИС на базе композитной модели пространства решений. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2019, № 5, с. 49–63. DOI: 10.18698/0236-3933-2019-5-49-63

MODIFIED ANT ALGORITHM OF VLSI PLANNING ON THE BASIS OF THE COMPOSITE MODEL OF THE SOLUTION SPACE

В.К. Lebedev
О.В. Lebedev
А.А. Zhiglaty

lebedev.b.k@gmail.com
lebedev.ob@mail.ru
artemiy.zhiglaty@gmail.com

**Academy for Engineering and Technologies, Southern Federal University,
Taganrog, Rostov Region, Russian Federation**

Abstract

In this paper, the crystal plan is formed by the recursive use of a “guillotine cut”. To set the plan means to set the structure of the binary tree of the cuts, i.e., sequence of binary cuts; for internal tree vertices, to indicate the type of the cut H or V ; to number the leaves of the tree and indicate the orientation of the modules. The structure of the binary tree of the cuts can be set using the Polish expression on the base of the alphabet $A = \{M, TR\}$, where the set of letters $M = \{m_i | i = 1, 2, \dots, n_M\}$ corresponds to the leaves of the section tree (regions), and the set $R = \{H, V\}$ corresponds to the cuts. We propose a way and methods for solving the problems of planning VLSI based on a modified ant colony. The task of synthesizing the section tree of the plan with the choice of types of sections, identification and orientation of the modules in the work is reduced to the task of forming a modified Polish expression with the identification of elements on the composite model of the solution space, including many alternative vertices. To keep the collective evolutionary memory during the life of the ant population and to form the solution of the problem, we use the complete graph $G = (X, U)$ with alternative vertex states. Each vertex may be in one of two alternative states, i.e., α or β , corresponding to the orientation of the module or the type of the cut. The task of synthesizing the Polish expression is formulated as the task of finding the least-cost route on the solution search graph $G = (X, U)$. A distinctive feature is that when building a route, simultaneously with the choice of the vertex $x_i \in X$, the state of this vertex is selected. The time complexity of the algorithm is $O(n^2)$. Experiments have shown that for large dimensions, the time indicators of the developed algorithm exceed those of the compared algorithms with the best values of the objective function

Keywords

VLSI planning, section tree, modified Polish notation, composite structure, ant colony, hybridization

Received 29.05.2019

© Author(s), 2019

This work was supported by the Russian Foundation for Basic Research (RFBR grant no. 17-07-00997a)

REFERENCES

- [1] Kureichik V.M., Lebedev B.K., Lebedev V.B. VLSI floorplanning based on the integration of adaptive search models. *J. Comput. Syst. Sci. Int.*, 2013, vol. 52, iss. 1, pp. 80–96. DOI: <https://doi.org/10.1134/S1064230712060056>

- [2] Lebedev B.K., Lebedev O.B., Lebedev V.B. *Metody, modeli i algoritmy razmeshcheniya* [Placement methods, models and algorithms]. Rostov-on-Don, SFedU Publ., 2015.
- [3] Lebedev O.B. *Modeli adaptivnogo povedeniya muravinoy kolonii v zadachakh proektirovaniya* [Adaptive behavior models of ant colony in design problems]. Rostov-on-Don, SFedU Publ., 2013.
- [4] Qi L., Xia Y., Wang L. Simulated annealing based thermal-aware floorplanning. *ICECC*, 2011, pp. 463–466. DOI: 10.1109/ICECC.2011.6067654
- [5] Chen J., Zhu W. A hybrid genetic algorithm for VLSI floorplanning. *ICIS*, 2010, pp. 128–132. DOI: 10.1109/ICICISYS.2010.5658785
- [6] Karpenko A.P. *Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vдохновlennyye prirodoy* [Modern search optimization algorithms. Algorithms inspired by nature]. Moscow, BMSTU Publ., 2014.
- [7] Chen G., Guo W., Chen Y. A PSO-based intelligent decision algorithm for VLSI floorplanning. *Soft Comput.*, 2010, vol. 14, iss. 12, pp. 1329–1337. DOI: <https://doi.org/10.1007/s00500-009-0501-6>
- [8] Banerjee P., Sangtani M., Sur-Kolay S. Floorplanning for partially reconfigurable FPGAs. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2011, vol. 30, iss. 1, pp. 8–17. DOI: 10.1109/TCAD.2010.2079390
- [9] Eroshenko I.N. The development of genetic algorithm for clustered floorplanning. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, no. 7, pp. 54–60 (in Russ.).
- [10] Lebedev B.K., Lebedev V.B. Planning on a basis swarm intelligence and genetic evolution. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2009, no. 4, pp. 25–33 (in Russ.).
- [11] Kureychik V.V., Kureychik V.V. The architecture of hybrid search for design. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, no. 7, pp. 22–27 (in Russ.).
- [12] Kureichik V.M., Lebedev B.K., Lebedev O.B. Hybrid evolutionary algorithm of planning VLSI. *Proc. GECCO'10*. Portland, OR, 2010, pp. 821–822.
- [13] Lebedev B.K., Lebedev O.B. [Bioinspired methods for planning the VLSI crystal]. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem (MES)* [Problems of development of promising micro- and nanoelectronic systems], 2014, no. 1, pp. 171–176 (in Russ.).
- [14] Lebedev O.B. Planning VLSI on the basis of the ant colony method. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, no. 7, pp. 67–73 (in Russ.).
- [15] Potti S., Pothiraj S. GPGPU implementation of parallel memetic algorithm for VLSI floorplanning problem. In: Nagamalai D., Renault E., Dhanuskodi M. (eds). *Trends in Computer Science, Engineering and Information Technology. CCSEIT 2011. Communications in Computer and Information Science*, vol. 204. Berlin, Heidelberg, Springer, pp. 432–441. DOI: https://doi.org/10.1007/978-3-642-24043-0_44

- [16] Shanavas I.H., Gnanamurthy R.K. Evolutionary algorithmical approach for VLSI floorplanning problem. *IJCTE*, 2009, vol. 1, no. 4, pp. 461–464.
DOI: 10.7763/IJCTE.2009.V1.75
- [17] Silvaco Library Platform™. Available at: <http://www.nangate.com> (accessed: 15.02.2019).
- [18] MCNC: company website. Available at: www.mcnc.org (accessed: 15.02.2019).
- [19] hMetis. Available at: <http://glaros.dtc.umn.edu/gkhome/metis/hmetis/download> (accessed: 15.02.2019).
- [20] HB Suite. cadlab.cs.ucla.edu: website.
Available at: <http://cadlab.cs.ucla.edu/cpmo/HBsuite.html> (accessed: 15.02.2019).
- [21] Ma Q., Young E.F.Y. Multivoltage floorplan design. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2010, vol. 29, iss. 4, pp. 607–617.
DOI: 10.1109/TCAD.2010.2042895
- [22] Lebedev B.K., Lebedev O.B., Lebedeva E.M. Hybrid algorithm of situational trajectory planning under partial uncertainty. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2018, no. 1, pp. 76–93 (in Russ.).
DOI: 10.18698/0236-3933-2018-1-76-93

Lebedev B.K. — Dr. Sc. (Eng.), Professor, Department of Computer Aided Design Systems, Academy for Engineering and Technologies, Southern Federal University (Nekrasovsky pereulok 44, Taganrog, Rostov Region, 347900 Russian Federation).

Lebedev O.B. — Cand. Sc. (Eng.), Assoc. Professor, Department of Computer Aided Design Systems, Academy for Engineering and Technologies, Southern Federal University (Nekrasovsky pereulok 44, Taganrog, Rostov Region, 347900 Russian Federation).

Zhiglaty A.A. — Assistant, Department of Mathematical Software and Computer Applications, Academy for Engineering and Technologies, Southern Federal University (Nekrasovsky pereulok 44, Taganrog, Rostov Region, 347900 Russian Federation).

Please cite this article in English as:

Lebedev B.K., Lebedev O.B., Zhiglaty A.A. Modified ant algorithm of VLSI planning on the basis of the composite model of the solution space. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2019, no. 5, pp. 49–63 (in Russ.). DOI: 10.18698/0236-3933-2019-5-49-63