

15. Kureychik V.V., Kureychik V.M., Gladkov L.A., Sorokoletov P.V. Bionspirirovannyye metody v optimizatsii [Inspirovane methods in optimization]. Moscow: Fizmalit, 2009, 384 p.
16. Kureychik V.M., Lebedev B.K., Lebedev O.B. Razbienie na osnove modelirovaniya adaptivnogo povedeniya biologicheskikh sistem [Partitioning based on simulation of adaptive behavior of biological systems], *Neyrokomp'yutery: razrabotka, primeneniye* [Neurocomputers: development, application], 2010, No. 2, pp. 28-34.
17. Lebedev B.K., Lebedev V.B. Razmeshcheniye na osnove metoda pchelinoy kolonii [Plasement on the basis of the bee colony method], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 12 (113), pp. 12-19.
18. Gladkov L.A. Gibridnyy geneticheskiy algoritm resheniya zadachi razmeshcheniya elementov SBIS s uchedom trassiruемости soedineniy [A hybrid genetic algorithm for solving the placement of elements VLSI traceability connections], *Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya* [Vestnik Rostovskogo Gosudarstvennogo Universiteta Putey Soobshcheniya], 2011, No. 3, pp. 58-66.
19. Kureychik V.V., Kureychik V.I. Bioinspirirovannyy algoritm razbieniya skhem pri proektirovani SBIS [Bioinspired algorithms for partitioning VLSI circuits in design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 23-29.
20. Kuliev E.V., Lezhebokov A.A. Issledovaniye kharakteristik gibridnogo algoritma razmeshcheniya [Research parameters of hybrid algorithm for placement], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 3 (140), pp. 255-261.

Статью рекомендовала к опубликованию д.т.н., профессор Л.С. Лисицына.

**Щеглов Сергей Николаевич** – Южный федеральный университет; e-mail: srg\_sch@mail.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371625; кафедра систем автоматизированного проектирования; к.т.н.; доцент.

**Shcheglov Sergey Nikolaevich** – Southern Federal University; e-mail: srg\_sch@mail.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371625; the department of computer aided design; cand. of eng. sc.; associate professor.

УДК 681.325

**О.Б. Лебедев, О.А. Пурчина**

### **РОЕВОЙ АЛГОРИТМ ТРАССИРОВКИ В ПРИКАНАЛЬНОЙ НАДЪЯЧЕЕЧНОЙ ОБЛАСТИ\***

*Представлен алгоритм синтеза эскиза однослойной трассировки в надъячеечной области (НЯО) топологии СБИС. Разработаны новые механизмы решения задачи трассировки, использующие математические методы, в которых заложены принципы природных механизмов принятия решений. Для синтеза эскиза однослойной трассировки используется новая парадигма комбинаторной оптимизации муравьиное дерево (trees ant colony optimization (T-ACO)), основанная на идеях адаптивного поведения муравьиной колонии и в первую очередь на идее непрямого обмена – стигмержи (stigmergy), позволяющая осуществлять синтез дерева. В отличие от канонической парадигмы муравьиного алгоритма муравьем на специальном графе поиска решений  $G=(X,U)$  строится решающее дерево. Рассматривается структура и методика формирования специального графа поиска решений  $G=(X,U)$ . Описываются принципы интерпретации полученного муравьями решения в виде эскиза однослойной трассировки. Задача решается в различных постановках, отличающихся типом границ области трассировки. Программа трассировки в НЯО была реализована на языке  $C^{++}$  для IBM PC. Во всех перечисленных постановках был получен результат, близкий к оптимальному. Трудоемкость программы имеет вид  $O(n^2)$ , где  $n$  – число*

\* Работа выполнена при поддержке Министерства образования и науки РФ. Проект № 8.823.2014.

фрагментов. При использовании “ломаных” границ, процедуры “прилипания” и последовательной трассировки НЯО удавалось дополнительно снизить общую плотность каналов до 4 0%. Предложенный трассировщик полностью применим для “бессеточной” трассировки соединений разной ширины. Экспериментальные исследования проводились на IBM PC. Временная сложность алгоритма (BCA), полученная экспериментальным путем, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет  $(BCA \approx O(n^2))$ . Для проведения объективных экспериментов были использованы известные тестовые задачи, представленные в литературе и в Интернет. Для составления достоверных выводов был проведен не один, а серия опытов-экспериментов. По сравнению с существующими алгоритмами достигнуто улучшение результатов на 2–3 %.

Роевой интеллект; муравьиная колония; адаптивное поведение; муравьиное дерево; трассировка; надъячеечная область; оптимизация.

**O.B. Lebedev, O.A. Purchina**

### SWARMS OVER-THE-CELL TRACING ALGORITHM

*This paper presents a synthesis algorithm of single layer routing sketch in over-the-cell (OTC) field VLSI layout. Develop new mechanisms for solving the problem tracing, using mathematical methods, which lays down the principles of the natural mechanisms of decision-making. For the synthesis of single layer routing sketch uses a new paradigm of combinatorial optimization tree ant colony optimization (T-ACO), based on the ideas of adaptive behavior of an ant colony and first of all on the idea of indirect exchange – stigmergy, which allows for the synthesis of a tree. Unlike the canonical paradigm of ant algorithm the ant at a special graph to find solutions  $G = (X, U)$  is constructed decision trees. The structure and method of forming a special graph to find solutions  $G = (X, U)$  are described. Are offered the principles of interpretation of the ants solutions in the form of a single layer trace sketch. The problem is solved in various productions with different type of trace areas boundaries. Program OTC trace was implemented in C++ for the IBM PC. In all of these productions was received result close to optimum. When using “broken” border procedures “sticking” and consistent trace OTC able to further reduce the overall density of channels up to 40 %. The proposed router is fully applicable to “gridless” trace of different widths connections. Experimental researches were carried out on IBM PC. The time complexity of the algorithm, obtained by experiment, almost the same as the theoretical research and test problems is considered about  $O(n^2)$ , where  $n$  – number of connections. For the objective of the experiments were used known test problems represented in the literature and the Internet. To provide reliable conclusions was held not one but a series of experiments, experiments. In comparison with existing algorithms improve the results achieved on 2–3 %.*

*Swarm intelligence; ant colony; adaptive behavior; trees ant colony optimization; trace; over the cell areas.*

**1. Проблемная формулировка, термины и обозначения.** Канальная трассировка – одна из важнейших задач, решаемых при проектировании топологии СБИС.

Разработано большое количество канальных трассировщиков и некоторые из них позволяют получать решения, близкие к оптимальным [1–5]. Тем не менее, по-прежнему актуальной является проблема минимизации области канальной трассировки. С этой целью предлагается использовать области, примыкающие к каналу сверху и снизу с единственным трассировочным слоем, расположенным над ячейками (over-the-cell) (рис. 1). В этом случае некоторые из соединений или их фрагментов могут быть направлены в надъячеечную область (НЯО), что приводит к уменьшению числа магистралей, используемых в канале [1, 5].

Задача «надъячеечного» (over-the-cell) трассировщика – построение максимального планарного эскиза топологии в надъячеечной области с целью минимизации плотности канала [5].

Среди разработанных алгоритмов трассировки в НЯО [1] наибольшее распространение получили алгоритмы на основе методов динамического программирования (МДП) авторов N. Holmes, N. Sherwani, M. Sarrafzadeh и генетический алгоритм (ГА) авторов В.М. Goni, Т. Arslan, В. Torton. Задача относится к классу NP. В связи с резким ростом сложности и размеров решаемых задач классические методы МДП становятся неприемлемыми в связи с их трудоемкостью. По этой же причине в ГА резко возрастает трудоемкость декодирования и проблема получения легальных решений. Кроме того, возникла проблема надъячеечной трассировки в области с ломанными границами. В связи с этим разработка алгоритма трассировки в НЯО, отвечающего современным требованиям, является актуальной.

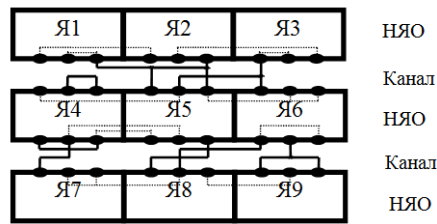


Рис. 1. Структура коммутационного поля

В работе рассматриваются два варианта верхней границы НЯО. В первом варианте граница НЯО представляет собой прямую (тип Т1) линию, а нижняя граница может быть либо прямой, либо ломаной зигзагообразной (тип Т2) линией.

Во втором случае верхняя граница может быть зигзагообразной, а нижняя граница либо прямой (тип Т3), либо зигзагообразной линией (тип Т4) (рис. 2).

Предусмотрен случай трассировки соединения различной ширины. В этом случае используется “бессеточный” (“gridless”) способ трассировки, при котором магистрали, как таковые, в НЯО отсутствуют.

Аналогично большинству работ в этой области будем считать, что имеется два слоя для канальной трассировки и один слой для надъячеечной (НЯ) трассировки [1, 5].

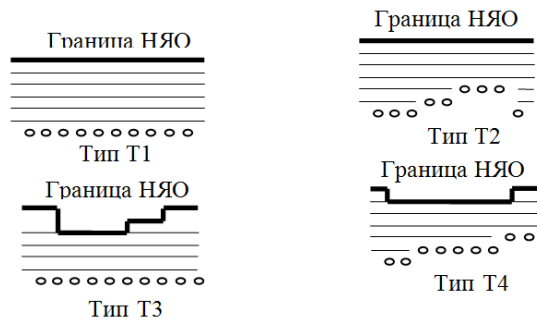


Рис. 2. Виды границ НЯО

Предварительно формируется набор соединений – кандидатов для перевода из канала в надъячеечную область. Каждая многовыводная цепь (рис. 3,а) на каждой стороне канала разбивается на двухвыводные цепи. Двухвыводная цепь связывает соседние в линейке выводы цепи. На рис. 3,б это  $f_1, f_2, f_3, f_4, f_5$ . Фрагмент  $f_6$  связывает верхний и нижний участки цепи. Каждая такая двухвыводная цепь может располагаться либо в канале, либо в НЯО.

Вначале осуществляется однослойная трассировка в надъячеечных областях. Затем, при наличии альтернативных вариантов, решается вопрос о выборе соединения для связи верхнего и нижнего участков цепи ( $f_6$  на рис. 3,б). Выбор таких связей осуществляется с учетом плотности канала и вертикальных ограничений и с учетом назначения двух выводных цепей в НЯО. После этого осуществляется трассировка в канале не попавших в НЯО соединений.

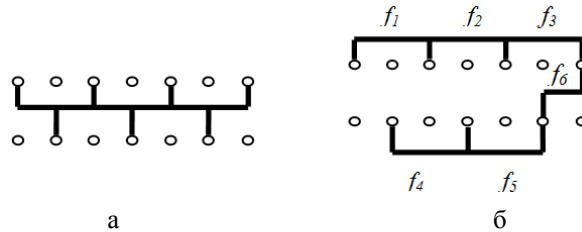


Рис. 3. Разбиение многовыводной цепи

Не теряя общности, рассмотрим одну (верхнюю) сторону канала. На рис. 2 НЯО расположена выше канала, и магистрали НЯО пронумерованы снизу вверх.

Обозначим через  $l_i$  левый край фрагмента  $f_i$ , а через  $r_i$  – правый. Пусть  $x(l_i)$  и  $x(r_i)$  координаты краёв фрагмента  $f_i$  по оси  $x$ , распространяющейся вдоль линейки выводов.

Назовём  $f_i$  вложенным в  $f_j$ , если:

–  $x(l_j) < x(l_i)$ ,  $x(r_j) > x(r_i)$ ;

–  $f_i$  и  $f_j$  размещены в НЯО без пересечений, то есть  $f_j$  выше  $f_i$  (рис. 4).

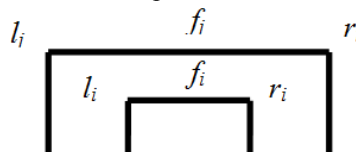


Рис. 4. Вложение цепей

Назовём два фрагмента  $f_i$  и  $f_j$  непересекающимися, если  $x(r_i) < x(l_j)$ . В этом случае фрагменты  $f_i$  и  $f_j$  могут быть размещены в НЯО без взаимного пересечения.

Построим граф решений  $G=(V,U)$ . Вершина  $v_i \in V$  соответствует фрагменту  $f_i$ , то есть  $f_i = \Gamma(v_i)$ . Между  $v_i$  и  $v_j$  существует ребро, направленное от  $v_i$  к  $v_j$  тогда и только тогда, когда  $f_j$  вкладывается в  $f_i$ . В состав  $V$  включена вершина  $v_r$ , соответствующая верхней границе НЯО. Вершина  $v_r$  связана со всеми вершинами  $V \setminus v_r$ , т.к. любой фрагмент вкладывается в НЯО, ограниченную верхней границей. На рис. 5 представлена схема, а на рис. 6 – соответствующий ей граф  $G$ .

Обозначим через  $B_i$  множество дочерних вершин вершины  $v_i$  графа  $G$ , Ребра, связывающие  $v_i$  с  $B_i$ , исходят из  $v_i$ . Если  $v_j \in B_i$ , то это значит, что  $f_j = \Gamma(v_j)$  вкладывается в  $f_i = \Gamma(v_i)$ . В этом случае будем считать, что  $v_j \in B_i$  вкладывается в  $v_i$ .

В множестве  $F_i$  фрагментов, вкладываемых в  $f_i$ , можно выделить подмножество  $F_{is} \subset F_i$ ,  $\Gamma^{-1}(F_{is}) = B_{is}$ , такое, что никакие два фрагмента  $f_k \in F_{is}$  и  $f_j \in F_{is}$  не пересекаются. Назовем такое подмножество  $F_{is}$   $s$ -м вложением фрагмента  $f_i$  а подмножество  $B_{is}$   $s$ -м вложением вершины  $v_i$ . Отметим, что таких множеств как  $F_{is}$  может быть несколько,  $\cup F_{is} = F_i$ .

Выделим в графе  $G$  подграф  $T_w=(V_w, U_w)$ ,  $T_w \in G$ , обладающий следующими свойствами:

–  $T_w$  является деревом с корневой вершиной  $v_r$ ;

- $B_{iw}$  – множество дочерних вершин вершины  $v_i \in V_w$ .  $B_{iw} \subset V_w$ .
- для любой  $v_i \in V_w$  множество  $B_{iw}$  является вложением вершины  $v_i$ . Другими словами множество фрагментов  $F_{iw} = \Gamma(B_{iw})$  является непересекающимся и вкладываемым в фрагмент  $f_i$ .

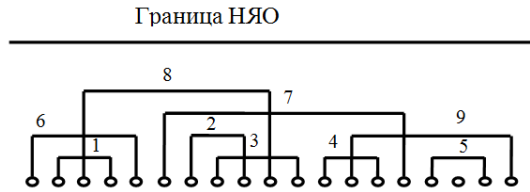


Рис. 5. Пример схемы

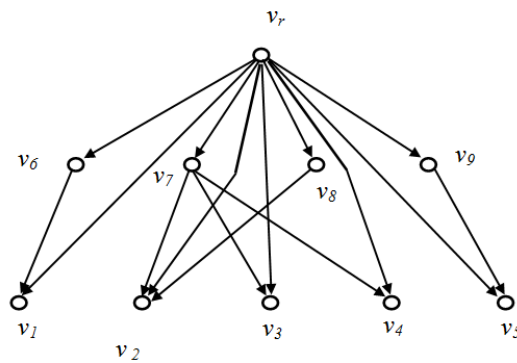


Рис. 6. Граф решений

Дерево  $T_w \in G$  с вышеперечисленными свойствами фактически задает способ планарной укладки в НЯО фрагментов, соответствующих вершинам множества  $V_w \subset V$ . Назовем такой подграф решающим деревом.

На рис. 7 приведены варианты решающих деревьев  $T_1$  и  $T_2$  с соответствующими им планарными укладками фрагментов.

**2. Постановка задачи трассировки в надъязыечной области.** Пусть для каждого фрагмента  $f_i$  задана его стоимость  $\varphi_i$ . Тогда решающему дереву  $T_w = (V_w, U_w)$  соответствует суммарная стоимость

$$\psi_w = \sum_{i/v_i \in V_w} \varphi_i.$$

Величина  $\psi_w$  принимается в качестве критерия оптимизации.

Введем рекурсивное определение уровня  $\alpha_{wi}$  фрагмента  $f_i$  (вершины  $v_i$ ), входящего в состав решающего дерева  $T_w$  [5].

Фрагмент  $f_i$ , у которого нет вложенных в него фрагментов, имеет уровень, равный нулю ( $\alpha_{wi}=0$ ).

Пусть  $\beta_{wi}$  – это максимальный уровень среди фрагментов, вложенных в  $f_i$ , тогда уровень  $f_i$  равен  $\beta_{wi} + 1$ , ( $\alpha_{wi} = \beta_{wi} + 1$ ).

Вершина  $v_i$  (фрагмент  $f_i$ ) в различных решающих деревьях может иметь различный уровень  $\alpha_i$ . На рис. 7,а  $\alpha_2 = \alpha_5 = 0$ ;  $\alpha_8 = \alpha_9 = 1$ . На рис. 7,б  $\alpha_1 = \alpha_3 = \alpha_4 = \alpha_5 = 0$ ;  $\alpha_6 = \alpha_7 = 1$ .

Назовем блоком  $t_{wi}$  поддерево решающего дерева  $T_w$ , имеющее своим корнем вершину  $v_i \in V_w$ . Уровень блока  $t_{wi}$  равен уровню вершины  $v_i \in V_w$ , входящей в решающее дерево  $T_w$ . Будем считать, что блок  $t_{wi}$  входит в состав блока  $t_{wj}$ , если в решающем дереве  $T_w$  корневая вершина  $v_i$  блока  $t_{wi}$  является дочерней вершиной корневой вершины  $v_j$  блока  $t_{wj}$  и  $\alpha_{wj} \geq \alpha_{wi} + 1$ .

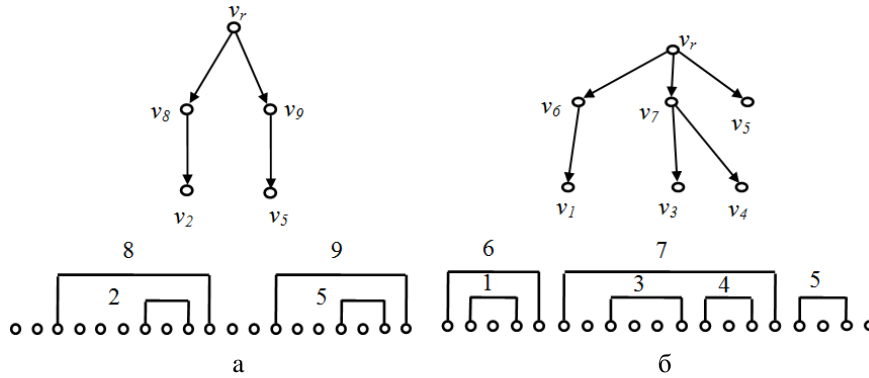


Рис. 7. Примеры решающих деревьев

Назовем два блока  $t_{wi}$  и  $t_{wj}$  непересекающимися, если не пересекаются  $f_i$  и  $f_j$ .

Введем рекурсивное определение стоимости  $\gamma_{wi}$  блока  $t_{wi}$ :

– любая вершина  $v_i \in V_w$  нулевого уровня представляет собой блок нулевого уровня, стоимость которого равна стоимости вершины  $v_i$ , т.е.

$$\gamma_{wi} = \varphi_i;$$

– стоимость блока  $t_{wi}$  ненулевого уровня ( $\alpha_{wi} \neq 0$ ) определяется как сумма стоимостей блоков, входящих в состав  $t_{wi}$ , плюс стоимость корневой вершины  $v_i$  блока  $t_{wi}$ , т.е.

$$\gamma_{wi} = \varphi_i + \sum_{j | v_j \in B_{wi}} \gamma_{wj}.$$

Тогда  $\psi_w = \gamma_{wr}$ , т.е. стоимость блока  $t_{wr}$ , имеющего корнем вершину  $v_r$ , равна стоимости всех вершин решающего дерева  $T_w$ , а  $t_{wr}$  фактически есть  $T_w$ .

Для НЯО типа Т1 и Т2 фрагмент  $f_i$  независимо от места его расположения представляет собой один горизонтальный отрезок, к которому от контактов подходят вертикальные отрезки.

Число магистралей  $\mu_w$ , необходимых для укладки решающего дерева  $T_w$ , определяется значением уровня  $\alpha_{wr}$  корневой вершины, т.е.  $\mu_w = \alpha_{wr}$ .

Для НЯО типа Т1 задача надъязыечной трассировки формулируется следующим образом.

Выделить в графе  $G$  решающее дерево  $T_{max}$  с максимальным значением  $\psi_{max}$  при выполнении ограничения

$$\mu_{max} = \alpha_{max,r} \leq \mu_0,$$

где  $\mu_0$  – заданное число магистралей в НЯО.

Номер магистралей  $m_{wi}$  для размещения каждого  $f_i$  определяется как

$$m_{wi} = \alpha_{wi} + 1.$$

Для НЯО типа Т2 для каждого  $f_i$  вводятся дополнительные ограничения. Это  $\delta_i$  – минимальный номер магистрали, на которой может быть размещен  $f_i$ . В этом случае номер магистрали для размещения  $f_i$  определится как

$$m_{wi} = \alpha_{wi} + \delta_i.$$

А для каждой вершины  $v_i$  решающего дерева введем ограничение

$$\alpha_{wi} + \delta_i \leq \mu_0.$$

При трассировке соединений различной ширины используется “бессеточный” метод трассировки, при котором опорные магистрали отсутствуют. В этом случае размещение фрагментов в НЯО сводится к фактической привязке координат опорных точек фрагментов к системе координат НЯО.

Каждый фрагмент  $f_i$  характеризуется параметром  $S_i$ , интегральным образом учитывающим ширину цепи и допустимое расстояние между цепями.

Введем рекурсивное определение ширины  $d_{wi}$  блока  $t_{wi}$ :

– если блок  $t_{wi}$  нулевого ранга, т.е. фактически он состоит из одного фрагмента  $f_i$ , то

$$d_{wi} = S_i;$$

– если блок  $t_{wi}$  ненулевого ранга, то его ширина равна самой максимальной ширине  $\max(d_{wj})$  блоков, входящих в состав  $t_{wi}$ , плюс ширина  $S_i$  фрагмента  $f_i$ :

$$d_{wi} = S_i + \max_{j | v_j \in B_{wi}}(d_{wj}).$$

Пусть ширина НЯО типа Т1 равна  $S_{max}$ , тогда для НЯО типа Т1 задача формулируется следующим образом.

Выделить в графе  $G$  решающее дерево  $T_w$ , для которого  $d_{wr} \leq S_{max}$ .

Отметим, что  $S_r = 0$ .

Для НЯО типа Т2 на основе анализа его геометрической структуры для каждого  $f_i$  определяется максимально возможная толщина  $d_i$ , построенного на нем блока  $t_{wi}$ .

В этом случае вводится ограничение:

$$(\forall i) [d_{wi} \leq d_i].$$

Размещение  $f_i$  в соответствии с решающим деревом  $T_w$  заключается в последовательной укладке фрагментов с прижиманием в сторону нижней границы НЯО, начиная с фрагментов нижнего уровня, и так до тех пор, пока не будут размещены фрагменты самого высокого уровня.

**3. Построение решающего дерева на основе механизмов муравьиного дерева.** Приведенный в [1, 5] обзор, сравнение и анализ разработанных алгоритмов однослойной трассировки (последовательных, итерационных и т.д.) показывает, что для создания эффективного алгоритма, отвечающего современным требованиям, необходимы новые технологии и подходы. В последние годы интенсивно разрабатывается научное направление, объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений [6–9]. Одним из таких направлений являются мультиагентные методы интеллектуальной оптимизации, базирующиеся на моделировании коллективного интеллекта [10–13].

Для построения решающего дерева используется новая парадигма комбинаторной оптимизации **муравьиное дерево** (*trees ant colony optimization* (Т-АСО)) [14], основанная на идеях адаптивного поведения муравьиной колонии [7, 9] и в первую очередь на идее непрямого обмена - *стигмержи* (*stigmergy*), позволяющая осуществлять синтез дерева. Такой подход является эффективным способом поиска рациональных решений для задач оптимизации, допускающих интерпретацию решений в виде деревьев.

Представление оптимизационной задачи в виде парадигмы Т-АСО опирается на два ключевых момента: формирование графа поиска решений (ГПР) и построения допустимых альтернативных решений деревьев на графе поиска решений [9].

В качестве ГПР используется модифицированный граф решений. Построение модифицированного графа решений  $G=(V,U)$  осуществляется путем рекурсивного определения возможных вложений фрагментов друг в друга.

Процесс построения графа решений начинается от корневой вершины  $v_r$ , соответствующей верхней границе НЯО. На первом шаге на основе анализа схемы формируется множество вершин  $V_1$ , каждая из которых может быть вложена только в корневую вершину  $v_r$ . Для нашего примера на рис. 5 это будет  $V_1 = \{v_6, v_8, v_7, v_9\}$ . Вершина  $v_r$  связана ребрами со всеми вершинами множества  $V_1$ . На втором шаге на основе анализа схемы формируется множество вершин  $V_2$ , каждая из которых может быть вложена только в некоторые вершины множества  $V_1$ . Для нашего примера на рис. 5 это будет  $V_2 = \{v_1, v_2, v_3, v_4, v_5\}$ . Между парами вершин  $v_i \in V_1$  и  $v_j \in V_2$  вводится ребро, направленное от  $v_i$  к  $v_j$ , тогда и только тогда, когда  $f_j$  вкладывается в  $f_i$ . На шаге  $n$  на основе анализа схемы формируется множество вершин  $V_n$ , каждая из которых может быть вложена только в некоторые вершины множества  $V_{n-1}$ . И так далее, пока в графе  $G$  не будут отражены все фрагменты схемы. Построенный для нашего примера граф решений представлен на рис. 8.

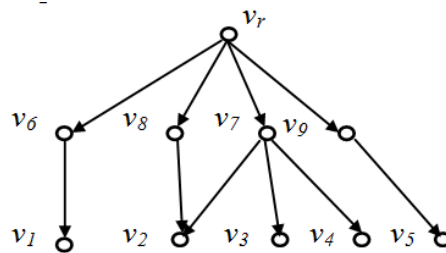


Рис. 8. Граф поиска решений – граф решений

Работа алгоритма муравьиного дерева Т-АСО заключается в нахождении на графе решений решающего дерева с лучшей оценкой. Поисковая процедура базируется на использовании графа решений  $GF$  с фиксированной структурой и графа решений  $G$ , структура которого трансформируется в процессе поиска в искомое решающее дерево. Исходная структура графа  $G$  идентична структуре графа  $GF$ .

В общем случае поиск решения задачи осуществляется коллективом муравьев  $Z = \{z_k / k = 1, 2, \dots, n_k\}$ . Моделирование поведения муравьев связано с распределением феромона на ребрах графа  $GF$ . На начальном этапе на всех ребрах графа  $GF$  откладывается одинаковое (небольшое) количество феромона  $Q/v$ , где  $v = |U|$ . Параметр  $Q$  задается априори.

Вначале муравьи размещаются в стартовой вершине  $v_r$  графа решений  $G$ . Процесс поиска решений итерационный. На каждой итерации алгоритма муравьиного дерева каждый муравей  $z_k$  строит свое конкретное решение задачи на своем конкретном графе решений  $G$ . Решением является решающее дерево  $T_k$ , которому соответствует эскиз планарной трассировки в надъязыечной области. Каждая итерация  $l$  включает три этапа. На первом этапе каждый муравей находит свое решение, являющееся результатом последовательной трансформации собственного исходного графа решений  $G = (V, U)$ , на втором этапе муравьи откладывают феромон на ребрах графа решений  $GF$ , на третьем этапе осуществляется испарение феромона на ребрах графа решений  $GF$ . В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромон откладывается агентом на ребрах графа решений  $GF$  после полного формирования решения.

Поиск решения организован в процессе последовательного выбора вершин дерева решений. На каждом шаге муравей формирует на графе решений  $G$  для выбранной вершины  $v_i$  подмножество вкладываемых в нее непересекающихся вершин  $D_{ik}$  и удаляет из графа решений множество вершин  $P_{ik}$ , конфликтующих с вершинами  $D_{ik}$ .



Для организации последовательного выбора вершин используются два вспомогательных подмножества  $VV_1$  и  $VV_2$ . На первом шаге  $VV_1=\{v_r\}$ ,  $VV_2=\emptyset$ . Вершины выбираются из подмножества  $VV_1$ , содержимое которого периодически обновляется. Для очередной выбранной вершины  $v_i \in VV_1$  определяется множество ее дочерних вершин  $B_{ik}$ . На базе  $B_{ik}$  муравей последовательно формирует подмножество вершин  $D_{ik} \subset B_{ik}$ , являющееся непересекающимся и вкладываемым в вершину  $v_i$ . После этого:

- $D_{ik}$  включается в  $VV_2=\emptyset$ ;
- формируется множество вершин  $P_{ik}$ , пересекающихся с вершинами подмножества  $D_{ik}$ . Вершины множества  $P_{ik}$ , удаляются из графа решений  $G=(V,U)$ .

После этого осуществляется выбор очередной вершины множества  $VV_1$ . Если просмотрены все вершины множества  $VV_1$ , то:

- из  $VV_1$  удаляются все элементы;
- в  $VV_1$  включаются все элементы множества  $VV_2$ , т.е.  $VV_1=VV_2$ ;
- из  $VV_2$  удаляются все элементы и снова  $VV_2=\emptyset$ .

Процесс поиска продолжается, начиная с выбора первой вершины подмножества  $VV_1$ . Завершение процесса поиска наступает при достижении  $VV_1=VV_2=\emptyset$ . Таким образом, конечный результат формируется путем последовательной трансформации исходного графа решений. На каждом шаге муравей формирует для выбранной вершины  $v_i$  подмножество вкладываемых в нее непересекающихся вершин  $D_{ik} \subset B_{ik}$ , формирует множество вершин  $P_{ik}$ , пересекающихся с вершинами подмножества  $D_{ik}$ , и удаляет  $P_{ik}$  из графа решений  $G=(V,U)$ .

Для нашего примера вначале  $VV_1=\{v_r\}$ ,  $VV_2=\emptyset$ . Просматривая  $VV_1$  выбираем  $v_r$ . Определяется в исходном графе решений  $G$  множество  $B_{rk}$  дочерних вершин вершины  $v_r$ .  $B_{rk}=\{v_6, v_8, v_7, v_9\}$ . На базе  $B_{rk}$  муравей  $z_k$  последовательно формирует подмножество непересекающихся вершин  $D_{rk} \subset B_{rk}$ , вкладываемых в вершину  $v_r$ .  $D_{rk}=\{v_6, v_7\}$ . Вершины  $\{v_6, v_7\}$  заносятся в  $VV_2$ . Формируется множество вершин  $P_{rk}$ , пересекающихся с вершинами подмножества  $D_{rk}$ ,  $P_{rk}=\{v_8, v_9\}$ . Вершины  $\{v_8, v_9\}$  удаляются из графа решений, представленного на рис.8. Новый вид графа решений  $G$  представлен на рис. 9.

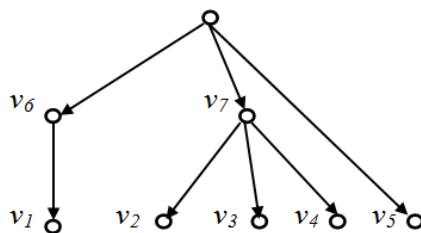


Рис. 9. Граф решений  $G$  после первой трансформации

Поскольку список  $VV_1$  исчерпан, то вершины из  $VV_2$  переносятся в  $VV_1$ . Теперь  $VV_1=\{v_6, v_7\}$ ,  $VV_2=\emptyset$ . Сначала из  $VV_1$  выбираем  $v_6$ .  $D_{6k}=\{v_1\}$ .  $P_{6k}=\emptyset$ . Граф решений остается без изменений. Из  $VV_1$  выбираем  $v_7$ .  $B_{7k}=\{v_2, v_3, v_4\}$ . Пусть муравьем сформировано  $D_{7k}=\{v_2, v_4\}$ .  $P_{7k}=\{v_3\}$ . Удаляем  $v_3$  из ГПР и получаем дерево решений, представленное на рис. 10.

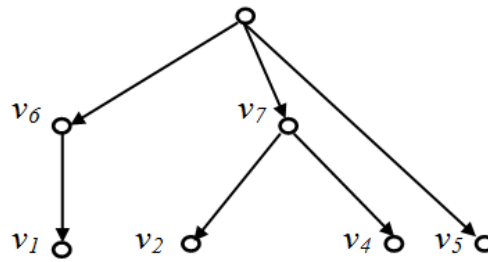


Рис. 10. Дерево решений

Рассмотрим процедуру последовательного формирования муравьем  $z_k$  подмножества непересекающихся вершин  $D_{ik} \subset B_{ik}$ , вкладываемых в вершину  $v_i$ . Множество  $D_{ik}$  формируется путем последовательного включения в него вершин множества  $B_{ik}$ . Пусть  $DI_{ik}$  множество уже включенных в  $D_{ik}$  вершин множества  $B_{ik}$ . Определяется множество вершин  $BI_{ik} \subset B_{ik}$ , каждая из которых может быть добавлена к  $DI_{ik}$  с соблюдением условий и ограничений. Для каждой вершины  $v_j \in BI_{ik}$  определяется параметр  $f_{ijk}$  – суммарный уровень феромона на ребре, связывающем  $v_j$  с  $v_i$ .

Вероятность  $P_{ijk}$  включения вершины  $v_j$  в формируемое множество  $DI_{ik}$  определяется следующим соотношением

$$P_{ijk} = f_{ij} / \sum_i (f_{ijk}).$$

Агент с вероятностью  $P_{ijk}$  выбирает одну из вершин, которая включается в подмножество непересекающихся вершин  $D_{ik} \subset B_{ik}$ . Процесс продолжается пока ни одна из вершин множества  $BI_{ik}$  не может быть назначена в  $DI_{ik}$ .

На втором этапе итерации, каждый муравей откладывает феромон на ребрах построенного дерева решений в графе решений.

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

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

где  $l$  – номер итерации,  $Q$  – общее количество феромона, откладываемое муравьем на ребрах дере  $D_k$ ,  $F_k(l)$  – целевая функция для решения, полученного муравьем  $z_k$  на  $l$ -ой итерации. Чем меньше  $F_k(l)$ , тем больше феромона откладывается на ребрах построенного дерева и, следовательно, тем больше вероятность выбора этих ребер при построении деревьев на следующей итерации.

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

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

где  $\rho$  – 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  $c$  и числа агентов  $m$ , и определяется как  $O(l * c^2 * m)$ .

Если НЯО относится к типу Т3 и Т4, то для рационального использования ресурсов НЯО конфигурация фрагмента  $f_i$  может представлять собой ломаную линию.

После построения  $T_m$  производится укладка фрагментов с прижиманием в сторону нижней границы НЯО, сопровождающаяся модификацией конфигураций фрагментов, входящих в состав  $T_m$ . Укладка производится начиная с конфигурации нулевого уровня.

Используя ломаные границы НЯО и процедуру укладки с “прилипанием” к границе, можно организовать оптимизацию использования ресурсов всех НЯО в целом. НЯО нумеруются сверху вниз. Процесс трассировки в НЯО реализуется последовательно, в порядке возрастания их номеров. В пределах НЯО сначала трассируется верхняя часть, а затем нижняя. После трассировки в верхней части НЯО и использования процедуры “прилипания” образуется ломаная граница между верхней и нижней частями НЯО. Это позволяет рационально использовать ресурсы НЯО.

Программа трассировки в НЯО была реализована на языке C++ для IBM/PC. Во всех перечисленных постановках был получен результат, близкий к оптимальному. Трудоемкость программы имеет вид  $O(n^2)$ , где  $n$  – число фрагментов. При использовании “ломаных” границ, процедуры “прилипания” и последовательной трассировки НЯО удавалось дополнительно снизить общую плотность каналов до 40 %.

Отдельный момент исследования был связан с выбором функции стоимости фрагментов. Исследовались два варианта. В первом варианте в качестве стоимости фрагмента использовалась длина фрагмента, а во втором – интегральная оценка. Стоимость фрагмента равна сумме плотностей сечений в канале, которые пересекают фрагменты.

Эксперименты показали, что общая плотность при использовании второй функции в среднем меньше на 4%, чем при использовании первой функции. Однако в отдельных конкретных примерах результаты были противоположными.

**Заключение.** В работе был представлен НЯО-трассировщик, позволяющий выполнять однослойную трассировку в различных постановках. Для построения эскиза однослойной трассировки используется новая парадигма комбинаторной оптимизации **муравьиное дерево** (*trees ant colony optimization* (Т-АСО)), основанная на идеях адаптивного поведения муравьиной колонии и в первую очередь на идее непрямого обмена – *стигмержи* (*stigmergy*), позволяющая осуществлять синтез дерева. В отличие от канонической парадигмы муравьиного алгоритма муравьем на графе поиска решений  $G=(X,U)$  строится дерево.

Предложенный трассировщик полностью применим для “бессеточной” трассировки соединений разной ширины.

Источником усовершенствования может стать более адекватный подбор функции стоимости фрагментов. Другой проблемой является выбор очередности рассматриваемых НЯО и способ их разделения на верхнюю и нижнюю части, что может способствовать снижению суммарной плотности каналов. Повышения качества можно добиться с помощью интеграции различных методов роевого интеллекта [15].

Предложены новые механизмы решения задачи трассировки, использующие математические методы, в которых заложены принципы природных механизмов принятия решений. Экспериментальные исследования проводились на IBM PC. Временная сложность алгоритма (ВСА), полученная экспериментальным путем, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет (ВСА  $\approx O(n^2)$ ).

Для проведения объективных экспериментов были использованы известные тестовые задачи, представленные в литературе и Интернет. Для составления достоверных выводов был проведен не один, а серия опытов-экспериментов. По сравнению с существующими алгоритмами достигнуто улучшение результатов на 2–3 %.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Charles J. Alpert, Dinesh P. Mehta, Sachin S. Sapatnekar. Handbook of algorithms for physical design automation. CRC Press, New York, USA, 2009.
2. Roy J.A. High-Performance routing at the nanometer scale // IEEE Trans.Comput.-Aided Design Integr. – Syst. June 2008. – Vol. 27, issue. 6. – P. 1066-1077.

3. Лебедев Б.К., Лебедев В.Б. Поискные процедуры канальной трассировки базирующиеся на моделировании адаптивного поведения роя частиц в пространстве решений с неупорядоченным лингвистическим шкалированием // Известия ЮФУ. Технические науки. – 2009. – № 12 (101). – С. 15-22.
4. Лебедев О.Б. Трассировка в канале методом муравьиной колонии // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 46-52.
5. Лебедев Б.К. Интеллектуальные процедуры синтеза топологии СБИС. – Таганрог: Изд-во ТРТУ, 2003.
6. Dorigo M. and Stützle T. Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.
7. Dorigo M., Stützle T. Ant Colony Optimization: Overview and Recent Advances. M. Gendreau and Y. Potvin, editors, Handbook of Metaheuristics, 2nd edition. Vol. 146 in International Series in Operations Research & Management Science. Springer, Verlag, New York, 2010. – P. 227-263.
8. Лебедев Б.К., Лебедев В.Б. Оптимизация методом кристаллизации россыпи альтернатив // Известия ЮФУ. Технические науки. – 2013. – № 7 (144). – С. 11-17.
9. Лебедев О.Б. Модели адаптивного поведения муравьиной колонии в задачах проектирования. – Таганрог: Изд-во ЮФУ, 2013.
10. Лебедев О.Б. Покрывание методом муравьиной колонии // Двенадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2010. Труды конференции. Т. 2. – М.: Физматлит, 2010. – С. 423-431.
11. Лебедев Б.К., Лебедев В.Б. Построение кратчайших связывающих сетей на основе метода муравьиной колонии // Нечеткие системы и мягкие вычисления: сб. ст. Третьей Всероссийской научной конференции. В 2 т. Т. II. Волгоградский гос. техн. университет. – Волгоград, 2009. – С. 42-50.
12. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Разбиение на основе моделирования адаптивного поведения биологических систем // Нейрокомпьютеры: разработка, применение. – 2010. – № 2. – С. 28-34.
13. Лебедев В.Б. Построение кратчайших связывающих сетей на основе роевого интеллекта // Известия ЮФУ. Технические науки. – 2011. – № 7 (120). – С. 37-44.
14. Лебедев Б.К., Лебедев О.Б. Моделирование адаптивного поведения муравьиной колонии при поиске решений, интерпретируемых деревьями // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 27-35.
15. Лебедев В.Б., Лебедев О.Б. Роевой интеллект на основе интеграции моделей адаптивного поведения муравьиной и пчелиной колоний // Известия ЮФУ. Технические науки. – 2013. – № 7 (144). – С. 41-47.

## REFERENCES

1. Charles J. Alpert, Dinesh P. Mehta, Sachin S. Sapatnekar. Handbook of algorithms for physical design automation. CRC Press, New York, USA, 2009.
2. Roy J.A. High-Performance routing at the nanometer scale, *IEEE Trans. Comput.-Aided Design Integr. Syst.*, – June 2008, Vol. 27, issue 6, pp. 1066-1077.
3. Lebedev B.K., Lebedev V.B. Poiskovye protsedury kanal'noy trassirovki baziruyushchiesya na modelirovanii adaptivnogo povedeniya roya chastits v prostranstve resheniy s neuporyadochennym lingvisticheskim shkalirovaniem [Search procedures of channel trace in space of decisions with disorder linguistic scales, based on modelling of adaptive behaviour of a plenty of particles], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2009, No. 12 (101), pp. 15-22.
4. Lebedev O.B. Trassirovka v kanale metodom murav'inoy kolonii [Chanel routing bases on method of ant colony optimization], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2009, No. 4 (93), pp. 46-52.
5. Lebedev B.K. Intellektual'nye protsedury sinteza topologii SBIS [Intellectual procedure of synthesis of VLSI topology]. Taganrog: Izd-vo TRTU, 2003.
6. Dorigo M. and Stützle T. Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.
7. Dorigo M., Stützle T. Ant Colony Optimization: Overview and Recent Advances. M. Gendreau and Y. Potvin, editors, Handbook of Metaheuristics, 2nd edition. Vol. 146 in International Series in Operations Research & Management Science. Springer, Verlag, New York, 2010, pp. 227-263.

8. *Lebedev B.K., Lebedev V.B.* Optimizatsiya metodom kristallizatsii rossypi al'ternativ [Optimization by the crystallization of alternatives field (CAF) Method], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 11-17.
9. *Lebedev O.B.* Modeli adaptivnogo povedeniya murav'inoi koloniiv v zadachakh proektirovaniya [Models of adaptive behaviour formic colonies in the task of designing]. Taganrog: Izd-vo YuFU, 2013.
10. *Lebedev O.B.* Pokrytie metodom murav'inoi kolonii [The coating method of the ant colony], *Dvenadsataya natsional'naya konferentsiya po iskusstvennomu intellektu s mezhdunarodnym uchastiem KII-2010. Trudy konferentsii* [Twelfth national conference on artificial intelligence with international participation KII-2010. The conference proceedings]. Vol. 2. Moscow: Fizmatlit, 2010, pp. 423-431.
11. *Lebedev B.K., Lebedev V.B.* Postroenie kratchayshikh svyazyvayushchikh setey na osnove metoda murav'inoi kolonii [Build shortest connecting networks based on ant colony method], *Nechetkie sistemy i myagkie vychisleniya: sb. st. Tre'tey Vserossiyskoy nauchnoy konferentsii* [Fuzzy systems and soft computing: a collection of articles of the Third all-Russian scientific conference]. In 2 volumes. Vol. II. Volgograd state technical University. Volgograd, 2009, pp. 42-50.
12. *Kureychik V.M., Lebedev B.K., Lebedev O.B.* Razbienie na osnove modelirovaniya adaptivnogo povedeniya biologicheskikh sistem [Partitioning based on simulation of adaptive behavior of biological systems], *Neyrokomp'yutery: razrabotka, primeneniye* [Neurocomputers: Development, Application], 2010, No. 2, pp. 28-34.
13. *Lebedev V.B.* Postroenie kratchayshikh svyazyvayushchikh setey na osnove roevogo intellekta [Construction of the shortest connecting networks on the basis of swarm intelligence], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2011, No. 7 (120), pp. 37-44.
14. *Lebedev B.K., Lebedev O.B.* Modelirovanie adaptivnogo povedeniya murav'inoi kolonii pri poiske resheniy, interpretiruemykh derev'yami [Modelling of an ant colony adaptive behaviour by search of the decisions interpreted by trees], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 7 (132), pp. 27-35.
15. *Lebedev V.B., Lebedev O.B.* Roeffoy intellekt na osnove integratsii modeley adaptivnogo povedeniya murav'inoi i pchelinoy kolonii [Swarm intelligence on the basis of the adaptive behaviour models integration of the ant and beer colonies], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 41-47.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.А. Гатчин.

**Лебедев Олег Борисович** – Южный федеральный университет; e-mail: lbk@tsure.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371743; кафедра систем автоматизированного проектирования; доцент.

**Пурчина Ольга Андреевна** – Донской государственный технический университет; e-mail: knagna\_olga@inbox.ru; 344000, г. Ростов-на-Дону, пл. Гагарина, 1; тел.: 8952581299; кафедра автоматизации процессов и производств в нефтегазовом комплексе; ассистент.

**Lebedev Oleg Borisovich** – Southern Federal University; e-mail: lbk@tsure.ru; 44, Nekrasovsky, Taganrog, 347928, Russia; phone: +78634371743; the department of computer aided design; associate professor.

**Purchina Olga Andreevna** – Don State Technical University; e-mail: knagna\_olga@inbox.ru; 1, Gagarin sq., Rostov-on-Don, 344000, Russia; phone: +7952581299; the department of processes automation and manufactures in an oil and gas complex; the assistant.