

17. Kureychik V.M., Lebedev B.K., Lebedev O.B. Reshenie zadachi razmeshcheniya na osnove evolyutsionnogo modelirovaniya [The solution of the location problem based on evolutionary modeling], *Izvestiya RAN. Teoriya i sistemy upravleniya* [Journal of Computer and Systems Sciences International], 2007, No. 4, pp. 78-91.
18. Kureychik V.V., Kureychik V.V. Bionicheskiy poisk pri proektirovanii i upravlenii [Search inspired by natural systems, for the design and management], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 178-183.
19. Kureychik V.V., Kureychik V.M., Rodzin S.I. Teoriya evolyutsionnykh vychisleniy [The theory of evolutionary computing]. M.: Fizmatlit, 2012, 260 p.
20. Kuliev E.V., Lezhebokov A.A. O gibridnom algoritme razmeshcheniya komponentov SBIS [On the hybrid algorithm of component placement VLSI], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 188-192.
21. Kuliev E.V., Lezhebokov A.A. Issledovanie 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.

**Кулиев Эльмар Валерьевич** – Южный федеральный университет; e-mail: ekuliev@sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; доцент

**Щеглов Сергей Николаевич** – e-mail: sch@sfedu.ru; кафедра систем автоматизированного проектирования; доцент.

**Пантелюк Елизавета Александровна** – e-mail: pantelyk@sfedu.ru; кафедра систем автоматизированного проектирования; магистр.

**Кульева Нина Владимировна** – e-mail: xolopova@sfedu.ru; кафедра систем автоматизированного проектирования; аспирант.

**Kuliev Elmar Valerievich** – Southern Federal University; e-mail: ekuliev@sfedu.ru; 44, Nekrasovskiy lane, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; assistant professor.

**Sheglov Sergey Nikolaevich** – e-mail: sch@sfedu.ru; the department of computer aided design; assistant professor.

**Pantelyuk Elizaveta Alexandrovna** – e-mail: pantelyk@sfedu.ru; the department of computer aided design; master's degree.

**Kulieva Nina Vladimirovna** – e-mail: xolopova@sfedu.ru; the department of computer aided design; graduate student.

УДК 004.896

**Б.К. Лебедев, О.Б. Лебедев, Е.О. Лебедева**

### **МОДИФИЦИРОВАННЫЙ МУРАВЬИНЫЙ АЛГОРИТМ ПОСТРОЕНИЯ МИНИМАЛЬНОГО ДЕРЕВА ШТЕЙНЕРА\***

*Предлагаются новые технологии, принципы и механизмы решения задачи построения дерева Штейнера, использующие математические методы, в которых заложены принципы природных механизмов принятия решений. Задача построения дерева Штейнера представляется в виде адаптивной системы, на основе интеграции принципов самоорганизации и муравьиного подхода к поиску решения. Известная проблема Штейнера состоит в следующем. Дано множество  $P$  точек на плоскости. Формируется ортогональная сетка пу-*

\* Работа выполнена при финансовой поддержке РФФИ (проект № 15–01–05297).

тем проведения через множество точек  $P$ , расположенных на плоскости, горизонтальных и вертикальных линий. Образуется ортогональная сетка  $G=(V, E)$ , где  $V$  множество точек (узлов) пересечений линий сетки. Исходная задача эквивалентна задаче отыскания в графе  $G=(V, E)$  дерева  $G^*=(V^*, E^*)$ , имеющего минимальный суммарный вес  $F$  ребер и включающего заданный набор вершин  $P$  графа  $G$ , т.е.  $P \subset V^* \subset V$ ,  $E^* \subset E$ . Построение минимального дерева Штейнера сводится к задаче построения и выбора  $(n-1)$   $s$ -маршрутов, связывающих  $n$  основных вершин. Задача решается в два этапа. На первом этапе формируется набор альтернативных вариантов  $s$ -маршрутов, заведомо большей размерности, чем  $(n-1)$ , покрывающий минимальное дерево Штейнера. На втором этапе из сформированного набора выбирается  $(n-1)$   $s$ -маршрутов, покрывающих минимальное дерево Штейнера. В работе рассматриваются комбинаторный и параллельно последовательный подходы к построению минимального дерева Штейнера: каждый из  $s$ -маршрутов строится последовательно, но все  $(n-1)$   $s$ -маршрутов, покрывающих минимальное дерево Штейнера, строятся параллельно. Оба рассматриваемых подхода базируются на методе муравьиной колонии. В общем случае поиск решения задачи построения минимального дерева Штейнера осуществляется популяцией кластеров агентов  $A=\{A_\sigma | \sigma=1, 2, \dots, n_\sigma\}$ . На каждой итерации агенты каждого кластера  $A_\sigma=\{a_{\sigma k} | k=1, 2, \dots, n-1\}$  строят свое конкретное дерево Штейнера. Другими словами число решений, формируемых агентами на каждой итерации равно числу кластеров агентов. В работе впервые используется метод двойного непрямого обмена информацией. Метод построен на основе композитной структуры феромона и дифференцированного способа его отложения. Каждый агент в кластере  $A_\sigma$  метит путь (ребра маршрута) двумя видами феромона: феромон-1 и феромон-2. Тестирование производилось на бенчмарках. По сравнению с существующими алгоритмами достигнуто улучшение результатов на 6–7 %. Вероятность получения глобального оптимума составила 0.94. Временная сложность алгоритма при фиксированных значениях  $M$  и  $T$  лежит в пределах  $O(n)$ .

*Дерево Штейнера; новые технологии; принципы; природные механизмы; принятие решений; адаптивная система; кластеры агентов; муравьиная колония; алгоритм; оптимизация.*

**В.К. Lebedev, O.B. Lebedev, E.O. Lebedeva**

### **MODIFIED ANT COLONY ALGORITHM FOR BUILDING THE STEINER MINIMAL TREES**

*We propose new technologies, principles and mechanisms for solving the problem of constructing a Minimal Steiner Tree using mathematical methods, which lay down the principles of natural decision-making mechanisms. The problem of constructing a Steiner tree is represented in the form of an adaptive system, based on the integration of the principles of self-organization and the ant approach to finding a solution. The well-known Steiner problem consists in the following. Given is a set of  $P$  points in the plane. An orthogonal grid is formed by conducting a horizontal and vertical line through a set of points  $P$  located on a plane. An orthogonal grid  $G=(V, E)$  is formed, where  $V$  is the set of points (nodes) of intersections of grid lines. The initial problem is equivalent to the problem of finding a tree  $G^*=(V^*, E^*)$  in the graph  $G=(V, E)$  having the minimal total weight  $F$  of edges and including a given set of vertices  $P$  of  $G$ ;  $P \subset V^* \subset V$ ,  $E^* \subset E$ . The construction of the Minimal Steiner Tree is reduced to the problem of constructing and selecting  $(n-1)$   $s$ -routes connecting the  $n$  major vertices. The problem is solved in two stages. At the first stage, a set of alternative variants of  $s$ -routes is formed, of course, of greater dimension than  $(n-1)$  covering the Minimal Steiner Tree. In the second step, the  $(n-1)$   $s$ -routes covering the Minimal Steiner Tree are selected from the generated set. The paper discusses combinatorial and parallel sequential approaches to the construction of the Minimal Steiner Tree: each of the  $s$ -routes is constructed sequentially, but all  $(n-1)$   $s$ -routes covering the Minimal Steiner Tree are built in parallel. Both approaches are based on the method of the ant colony. In the general case, the search for a solution to the problem of constructing a Minimal Steiner Tree is performed by a population of clusters of agents  $A=\{A_\sigma | \sigma = 1, 2, \dots, n_\sigma\}$ . At each iteration, the agents of each cluster  $A_\sigma = \{a_{\sigma k} | k = 1, 2, \dots, n-1\}$  construct their concrete Steiner tree. In other words, the number of decisions generated by agents at each iteration is equal to the number of agent clusters. The method of double indirect communication is used for the first time. The method is based on the composite structure of pheromone and its differentiated deposition*

*method. Each agent in cluster  $A_c$  marks the path (route edges) with two kinds of pheromone: pheromone-1 and pheromone-2. The testing was performed on benchmarks. Compared with the existing algorithms, the improvement of results is achieved by 6–7 %. The probability of obtaining the global optimum was 0.94. The time complexity of the algorithm for fixed values  $M$  and  $T$  of the iterations number and population size lies in the range  $O(n)$ .*

*Steiner tree; new technologies; principles; natural mechanisms; decision making; adaptive system; clusters of agents; ant colony; algorithm; optimization.*

**Введение.** Задача Штейнера является фундаментальной задачей в проектировании сверхбольших интегральных схем (Very Large Scale Integration, VLSI). Также широко применяется в программируемых логических матрицах (Field-Programmable Gate Arrays, FPGA). FPGA представляют собой многократно используемые прикладные интегральные схемы, конфигурацию которых пользователь может достаточно легко изменить. Есть несколько доступных технологий FPGA, но чаще всего они состоят из симметричной матрицы реконфигурируемых логических блоков.

Проблема построения кратчайших связывающих сетей одна из наиболее сложных при автоматизированном проектировании внутрисхемных соединений электронных устройств. Поскольку при машинной трассировке соединений используется ортогональная опорная сетка, интерес представляет построение связывающих деревьев Штейнера в ортогональной метрике [1–6].

Существует множество различных способов, которыми можно решить задачу Штейнера.

Одним из наиболее широко распространенных методов точного решения является алгоритм, основанный на методе динамического программирования и уравнении Беллмана [2]. Другой точный метод использует приведение задачи к задаче целочисленного программирования и решения ее с помощью имеющихся в теории линейного программирования методов (метод ветвей и границ, отсечений Гомори и т.д.). Алгоритмы с применением метода ветвей и границ [7] основаны на идее последовательного разбиения множества на заданное число подмножеств и формировании оценок, позволяющих рассматривать только те подмножества, которые могут содержать решение задачи (т.е. отбрасываются заведомо не перспективные варианты). Основной недостаток этих методов заключается в необходимости корректного выбора оценок, в противном случае (при неверном выборе) данный метод будет осуществлять полный перебор. Точные алгоритмы, такие как метод ветвей и границ, не позволяют находить оптимальные решения в течение разумного периода времени, за исключением небольших проблем. В связи с этим при решении практических задач большую актуальность приобретают эвристические способы сокращения перебора.

Для решения задачи Штейнера в литературе предложены различные эвристические подходы, в основе которых лежат алгоритмы последовательной маршрутизации, алгоритмы динамического перестроения дерева Штейнера алгоритмы трассировки с разрывом связей и поиском новых соединений, иерархические методы

Основной недостаток многих эвристических алгоритмов трассировки, заключается в том, что проведение новых соединений зависит от ранее выполненных, которые рассматриваются как преграды и ограничивают проведение новых трасс.

В [8] рассматривается алгоритм построения дерева Штейнера основанный на методе горизонтальных (вертикальных) столбов Штейнера. Существенным недостатком этого метода является тот факт, что в результате получается дерево, сумма ребер которого далека от минимально возможной. В статье [9] описан эвристический алгоритм поиска приближённого решения задачи Штейнера, основанный на физических аналогиях. Суть алгоритма Z-RST [8] сводится к изменению конфигу-

рации ребер полученного на первом шаге дерева. Следующим известным методом является алгоритм WRST взвешенных деревьев Штейнера [8]. В алгоритме каждая цепь представляется в виде связывающего дерева, построенного алгоритмом Прима, на основе которого последовательно строится дерево Штейнера с учетом веса ребер и динамически изменяющейся в процессе построения среды (коммутационного поля). Алгоритмы [10, 11] алгоритмы последовательного введения дополнительных точек в дерево Прима-Краскала. К недостаткам алгоритма [12] следует отнести:

- ◆ возможное появление решений, длина которых больше, чем длина кратчайшей связывающей сети;
- ◆ корректность работы лишь для евклидовой метрики.

Существенным средством повышения эффективности алгоритмов решения задачи Штейнера стало использование методов эволюционного моделирования.

В настоящее время используются различные эволюционные парадигмы, которыми можно решить задачу Штейнера: генетический алгоритм, алгоритм роя, пчелиный алгоритм, муравьиный и другие [13–18].

Результаты применения эволюционного моделирования к задаче построения деревьев Штейнера были достаточно успешными. Однако предложенные структуры фактически являются “слепыми” поисковыми структурами с присущими им недостатками: генерация большого количества подобных решений; генерация большого количества “плохих” решений; генерация решений с нарушениями, что требует дополнительного контроля.

Резкое повышение функциональной сложности СБИС стимулирует разработку новых эффективных методов и средств их проектирования.

Предлагаются новые технологии, принципы и механизмы решения задачи построения дерева Штейнера, использующие математические методы, в которых заложены принципы природных механизмов принятия решений. Рассмотрен подход, в котором интегрированы механизмы коллективной адаптации и адаптивного поведения муравьиной колонии. Построение МДШ осуществляется популяцией кластеров муравьев. Впервые используется двухканальная система непрямого обмена информации. Это позволило организовать пространство решений, в рамках которого организован поисковый процесс, базирующийся на моделировании адаптивного поведения муравьиной колонии. Задача построения дерева Штейнера представляется в виде адаптивной системы, на основе интеграции принципов самоорганизации и муравьиного подхода к поиску решения. По сравнению с существующими алгоритмами достигнуто улучшение результатов.

Известная проблема Штейнера состоит в следующем [1,6]. Дано множество точек на плоскости:  $P = \{p_i | i = 1, 2, \dots, n\}$ . Требуется найти дерево  $S = (X, U)$ , с множеством вершин  $X$  и множеством ребер  $U$  таких, что  $P \subset X$ , и суммарная длина ребер  $U$  минимальна. В случае ортогональной метрики расстояние между точками с координатами  $(x_i, y_i)$  и  $(x_j, y_j)$  определяется по формуле  $d_{ij} = |x_i - x_j| + |y_i - y_j|$ .

Отличительной особенностью дерева Штейнера является то, что оно кроме основных вершин  $P$  содержит некоторое множество дополнительных вершин  $Q = X \setminus P$ .

Проведем через множество точек  $P = \{p_i | i = 1, 2, \dots, n\}$ , расположенных на плоскости горизонтальные и вертикальные линии (рис. 1). Образуется ортогональная сетка. Обозначим через  $V$  множество точек пересечений линий сетки и назовем их узлами сетки. Сопоставим ортогональной сетке граф  $G = (V, E)$ ,  $|V| = n_v$ ,  $|E| = n_e$ . Ряд теорем [8, 9] доказывает, что существует дерево Штейнера, все дополнительные вершины которого принадлежат  $V$ . Другими словами, при отыскании дополнительных точек можно ограничиться узлами ортогональной сетки  $G = (V, E)$ , а само дерево Штейнера полностью укладывается в ортогональной сети. При этом число этих точек  $Q \leq n - 2$ .

Исходная задача эквивалентна задаче отыскания в графе  $G=(V,E)$  дерева  $G^*=(V^*,E^*)$ , имеющего минимальный суммарный вес  $F$  ребер и включающего заданный набор вершин  $P$  графа  $G$ , т.е.  $P \subset V^* \subset V$ ,  $E^* \subset E$ .

**Подходы к построению дерева Штейнера.** Назовем цепь в графе  $G=(V, E)$ , связывающую две основные вершины,  $s$ -маршрутом. Задача построения МДШ сводится к задаче построения и выбора  $(n-1)$   $s$ -маршрутов, связывающих  $n$  основных вершин. К настоящему времени сложились два основных подхода к построению МДШ [6, 8]. Первый подход базируется на идеях волнового алгоритма Ли [2] и заключается в последовательной реализации  $s$ -маршрутов.

Второй подход базируется на комбинаторных принципах. Задача решается в два этапа. На первом этапе формируется набор альтернативных вариантов  $s$ -маршрутов, заведомо большей размерности, чем  $(n-1)$ , покрывающий МДШ. На втором этапе из сформированного набора выбирается  $(n-1)$   $s$ -маршрутов, покрывающих МДШ.

Рассмотрим процесс формирования альтернативных вариантов  $s$ -маршрутов [5,6]. На множестве вершин  $P$  алгоритмом Прима строится минимальное связывающее дерево (МСД)  $R=\{R_k/k=1,2,\dots,n-1\}$ ,  $R_k=(p_i,p_j)$  – ребро МСД (рис. 1). Для каждого ребра  $R_k$  на графе  $G$  строится набор  $S_k=\Gamma(R_k)$  альтернативных вариантов  $s$ -маршрута, равный  $S_k=\{s_{kz}/z=1,2,\dots,m\}$ .  $S=\{S_k/k=1,2,\dots,n-1\}$  – множество наборов. Формирование  $S_k$  осуществляется исходя из следующих посылок: длина  $s_{kz}$  должна быть минимальна; варианты различных  $s$ -маршрутов должны обеспечивать максимально возможное совпадение друг с другом.

В общем случае в набор  $S_k$  для ребра  $R_k=(p_i, p_j)$  включаются все маршруты минимальной длины, соединяющие  $p_i$  и  $p_j$  в ортогональном графе  $G$ . На рис. 1 для ребра  $R_k$ , связывающего  $p_i$  и  $p_j$ , существует 10 альтернативных  $s$ -маршрутов, проходящих через узлы с номерами 1 – 12:

$$s_{k1}=(3,6,9,12,11,10), s_{k2}=(3,2,1,4,7,10), s_{k3}=(3,2,5,4,7,10) \text{ и т.д.}$$

При таком подходе все варианты  $s$ -маршрута, связывающего две вершины  $p_i$  и  $p_j$ , имеют одну и ту же длину, но различаются конфигурациями (составом ребер, входящих в маршрут).

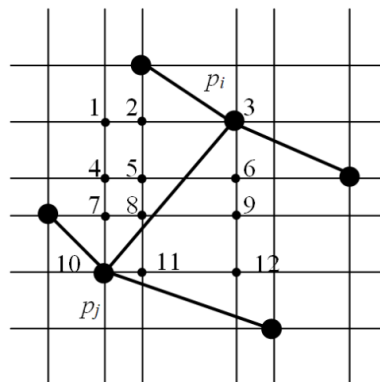


Рис. 1. Формирование набора альтернативных вариантов  $s$ -ребер

**Механизмы построения дерева Штейнера на основе муравьиной колонии.** Для поиска решения задачи используется ортогональный граф  $G=(V,E)$ , построенный по выше приведенной методике. Поиск МДШ осуществляется на базе предварительно построенного алгоритмом Прима МСД  $R=\{R_k/k=1,2,\dots,n-1\}$ . В общем случае поиск решения задачи построения МДШ осуществляется популяцией кластеров агентов  $A=\{A_\sigma/\sigma=1,2,\dots,n_\sigma\}$ . На каждой итерации агенты каждого

кластера  $A_\sigma = \{a_{\sigma k} | k=1, 2, \dots, n-1\}$  строят свое конкретное дерево Штейнера. Другими словами число решений, формируемых агентами на каждой итерации равно числу кластеров агентов.

Задача построения агентами каждого кластера  $A_\sigma$  дерева Штейнера сводится к задаче построения и выбора в графе  $G$  совокупности маршрутов  $M_\sigma$   $s$ -маршрутов, покрывающих МДШ, причем каждый маршрут  $s_k \in S_\sigma$  соответствует ребру  $R_k = (p_i, p_j)$ .

При использовании параллельно-последовательного подхода к построению МДШ за каждым ребром  $R_k = (p_i, p_j)$  закрепляются либо один, либо два агента –  $a_{k1}$  и  $a_{k2}$ . В случае двух агентов агент  $a_{k1}$  помещается в вершину  $p_i$ , а агент  $a_{k2}$  в вершину  $p_j$  и построение МДШ осуществляется кластером, включающем  $2(n-1)$  агентов. Задачей каждого агента  $a_{k1}$  или  $a_{k2}$  является построение на графе  $G$   $s$ -маршрута, связывающего вершины  $p_i$  и  $p_j$ . Задачей кластера агентов является построение МДШ. Формирование каждым агентом  $a_k$   $s$ -маршрута осуществляется последовательно (пошагово). Все маршруты кластера строятся агентами параллельно и независимо друг от друга анализируя количество и качество феромона, отложенного на ребрах графа  $G = (V, E)$ .

Для каждого агента, расположенного в вершине  $p_i$ , задаются координаты вершины  $p_j$  к которой прокладывается  $s$ -маршрут и определяются возможные направления его перемещения по ребрам ортогонального графа  $G = (V, E)$ , обеспечивающие построение  $s$ -маршрута минимальной длины. Пусть, например, вершина  $p_j$  расположена ниже и левее вершины  $p_i$ , тогда для агента, прокладывающего  $s$ -маршрут из вершины  $p_i$  в  $p_j$  возможные направления его перемещения – вниз и влево. А для агента, прокладывающего  $s$ -маршрут из вершины  $p_j$  в  $p_i$  возможные направления его перемещения – вверх и вправо. При этом агент не должен выходить за границы прямоугольника, диагональ которого связывает вершины  $p_j$  и  $p_i$ .

При таком подходе все варианты  $s$ -маршрута, связывающего две вершины  $p_i$  и  $p_j$ , имеют одну и ту же длину, но различаются конфигурациями (составом ребер, входящих в маршрут).

При использовании комбинаторного подхода к построению МДШ изначально для каждого ребра  $R_k$ , строится набор  $S_k = I(R_k)$  альтернативных вариантов  $s$ -маршрутов  $S_k = \{s_{kz} | z=1, 2, \dots, m\}$ . Методика формирования вариантов  $s$ -маршрутов изложена ниже. За каждым ребром  $R_k = (p_i, p_j)$  закрепляется один агент. В общем случае за каждым ребром может быть закреплено несколько агентов. На каждой итерации каждый агент  $a_k$  выбирает вариант  $s$ -маршрута из набора  $S_k = \{s_{kz} | z=1, 2, \dots, m\}$ .

Процесс поиска решений итерационный. Предварительно на всех ребрах графа  $G$  откладывается одинаковое (небольшое) количество феромона, равное  $Q/m$ , где  $m = |E|$ . Каждая итерация  $l$  включает три этапа.

На первом этапе каждой итерации каждый кластер  $A_\sigma$  популяции  $A$  строит свое дерево Штейнера. При этом каждый агент  $a_k$  кластера  $A_\sigma$  выбирает свой собственный маршрут  $s_k(l)$ .

На втором этапе на ребрах графа  $G$ , откладывается феромон.

В работе используется циклический (ant-cycle) метод муравьиных систем. Феромон откладывается каждым кластером  $A_\sigma$  на ребрах сформированного агентами дерева Штейнера, после построения (или выбора) всеми кластерами агентов деревьев Штейнера.

Количество феромона откладываемого на ребрах МДШ, построенного кластером агентов  $A_\sigma$ , пропорционально оценке качества МДШ. Чем больше феромона будет отложено на ребрах графа  $G$ , тем большее количество агентов будет включать эти ребра в собственные маршруты.

На третьем этапе осуществляется испарение феромона.

При параллельно-последовательном подходе процесс построения агентом маршрута  $s_k$  пошаговый. Пусть к данному моменту  $t$  сформирована часть маршрута  $s_k(t)$ . Агент  $a_k$  формирует список  $E_k(t+1) \in E$  ребер – кандидатов для включения в  $s$ -маршрут, в соответствии с ограничениями и возможными направлениями перемещения из вершины, в которой он расположен. В  $s$ -маршрут включается одно из ребер списка  $E_k(t+1)$ . Выбор следующего ребра для включения его в формируемый маршрут  $s_k$  выполняется с помощью вероятностного правила. Для каждого ребра  $e_j \in E_k(t+1)$  определяется количество  $h_j$  размещенного на нем феромона. Вероятность  $P_j$  включения на шаге  $t$  ребра  $e_j \in E_k(t+1)$  в формируемый агентом  $a_k$  маршрут  $s_k(t)$ , пропорциональна количеству феромона на ребре  $e_j$  и определяется следующим соотношением

$$P_j = h_j / H_j, \text{ где } H_j = \sum_i h_i \text{ для } (i|e_i \in E_k(t+1)). \quad (1)$$

Агент с вероятностью  $P_j$  выбирает одно из ребер  $e_j \in E_k(t+1)$ , которое включается в маршрут  $s_k(t)$ , маршрут после этого обозначается как  $s_k(t+1)$ . На конечном шаге  $t=n_k$  агентом  $a_k$  будет сформирован  $s$ -маршрут  $s_k$ .

При комбинаторном подходе - агентом на каждой итерации не строится, а выбирается из набора  $S = \{s_{kz} | z=1, 2, \dots, m\}$  маршрут  $s_{kz}$ . Пусть  $E(s_{kz})$  – множество ребер графа  $G$ , входящих в  $s_{kz}$ .

Вероятность  $P_{kz}$  выбора маршрута  $s_{kz}$ , определяется следующим соотношением

$$P_{kz} = C_{kz} / \sum_z C_{kz}, \quad (2)$$

где  $C_{kz}$  – стоимость маршрута  $s_{kz}$ ,  $C_{kz} = \sum_j h_j$  (для всех  $j|e_j \in E(s_{kz})$ ).

На основе совокупности  $s$ -маршрутов, построенных или выбранных агентами одного кластера  $A_\sigma$ , формируется дерево Штейнера  $G_\sigma^*$

Обозначим как  $F_\sigma(l) = \gamma / W(G_\sigma^*)$  оценку качества дерева Штейнера, построенного агентами одного кластера  $A_\sigma$  на  $l$ -ой итерации, где  $W(G_\sigma^*)$  – суммарный вес ребер, построенного дерева Штейнера, а  $\gamma$  – управляющий параметр, который подбирается экспериментально.

На ребрах сети (дерева Штейнера), построенного агентами одного кластера  $A_\sigma$  на  $l$ -ой итерации откладывается феромон. Количество  $\tau_\sigma(l)$  феромона, откладываемое кластером  $A_\sigma$  на каждом ребре построенного им дерева Штейнера, определяется следующим соотношением:

$$\tau_\sigma(l) = \mu \cdot Q \cdot F_\sigma(l), \quad (3)$$

где  $\mu$  – управляющий параметр, а общее количество феромона на ребре  $e_j$  определяется как:

$$h_j(l+1) = h_j(l) + \sum_\sigma \tau_\sigma(l). \quad (4)$$

На третьем этапе происходит общее испарение феромона на ребрах полного графа  $G$  в соответствии с формулой:

$$h_j = h_j (1 - \rho), \quad (5)$$

где  $\rho$  – коэффициент обновления.

Как указывалось выше при использовании параллельно-последовательного подхода к построению МДШ за каждым ребром  $R_k = (p_i, p_j)$  закрепляются два агента  $a_{k1}$  и  $a_{k2}$ . В связи с этим после построения этими агентами маршрутов выбирается лучший из них. Лучшим является маршрут, для которого оценка  $C_{kz}(l)$  имеет большее значение.

На основе совокупности  $s$ -маршрутов, построенных агентами одного кластера, формируется МДШ.

После выполнения всех действий на итерации запоминается лучшее дерево Штейнера, построенное на данной итерации. Далее осуществляется переход на следующую итерацию. Временная сложность этого алгоритма зависит от времени жизни колонии  $l$  (число итераций), количества вершин графа  $n$  и числа кластеров агентов  $n_\sigma$ .

**Комбинированный подход к формированию оценки (стоимости) ребра в процессе построения маршрута.** Концепция коллективной адаптации (адаптивного поведения) кластера агентов, используемая в рассмотренном выше подходе, заключается в следующем.

Глобальная цель коллектива (кластера агентов) заключается в построении минимального дерева Штейнера.

Локальная цель каждого агента заключается в построении маршрута максимальной стоимости, то есть маршрута, максимально совпадающего с маршрутами, построенными другими агентами, что косвенно способствует достижению глобальной цели коллектива.

Отметим, что в соответствии с подходом, описанным выше, все маршруты, которые может построить агент, будут иметь одну и ту же длину. Качество маршрута, построенного агентом  $a_k$ , будет определяться наличием общих участков с маршрутами, построенными другими агентами. Чем больше в составе маршрута общих участков с маршрутами, построенными другими агентами кластера, тем меньше суммарная длина ребер МДШ.

В работе впервые используется метод двойного непрямого обмена информации. Метод построен на основе композитной структуры феромона и дифференцированного способа его отложения. Каждый агент в кластере  $A_\sigma$  метит путь (ребра маршрута) двумя видами феромона: феромон-1 и феромон-2.

Для отражения коллективной эволюционной памяти в течение жизни популяции кластеров агентов используются граф поиска решений  $G=(V,E)$  и счетчик числа включений ребер (СВР)  $D=\{d_j|j=1,2,\dots,n_E\}$ .

На ребрах графа  $G$ , входящих в состав дерева Штейнера, построенного кластером агентов  $A_\sigma$ , откладывается феромон в количестве, пропорциональном оценке решения.

Для учета среднего количества включений каждого ребра  $e_j \in E$  графа поиска решений  $G=(V,E)$  в состав маршрутов, с сформированных в процессе построения деревьев Штейнера, используется счетчик включений ребер (СВР)  $D=\{d_j|j=1,2,\dots,n_E\}$ . Значением показателя  $d_j$  является среднее количество включений ребра  $e_j$  в состав маршрутов, сформированных всеми агентами коллектива кластера  $A$ . Изначально счетчик  $D$  обнуляется, т.е. всем показателям  $d_j$  присваивается нулевое значение. Пересчет значений показателей счетчика  $D$  выполняется каждый раз после выполнения очередной итерации.

Для расчета показателей счетчика  $D$  используются три счетчика  $D1=\{d1_j|j=1,2,\dots,n_E\}$ ,  $D2=\{d2_j|j=1,2,\dots,n_E\}$ ,  $D3=\{d3_j|j=1,2,\dots,n_E\}$ . Изначально все счетчики  $D1$ ,  $D2$ ,  $D3$ ,  $D$  обнуляются. Счетчики  $D1$ ,  $D2$  обнуляются вначале каждой итерации. Счетчики  $D1$ ,  $D2$ ,  $D3$ ,  $D$  формируются на каждой итерации последовательно один за другим. Значения показателей счетчиков формируются путем последовательного наращивания.

Счетчик  $D1$  используется для учета количества  $d1_j$  включений каждого ребра  $e_j \in E$  графа поиска решений  $G=(V,E)$ , входящего в состав совокупности  $M_\sigma$  маршрутов, сформированных в процессе построения дерева Штейнера агентами кластера  $A_\sigma$ .



Каждый раз после построения очередным агентом кластера  $E_\sigma$  соединения (маршрута)  $s_{kz}$  определяется совокупность всех ребер  $E(s_{kz})$  графа  $G$ , входящих в соединение (маршрут)  $s_{kz}$ . Затем значение каждого показателя  $d1_j$ , такого, что  $e_j \in E(s_{kz})$  увеличивается на единицу. После построения совокупности маршрутов  $M_\sigma$  агентами кластера  $A_\sigma$  определяется совокупность всех ребер  $E(M_\sigma)$  графа  $G$ , входящих в эти соединения (маршруты).

Счетчик  $D2$  используется для учета количества  $d2_j$  включений каждого ребра  $e_j \in E$  графа  $G$  в состав совокупности маршрутов  $M = \cup M_\sigma$ , сформированных в процессе построения деревьев Штейнера агентами множества кластеров  $A$  на одной итерации.

Каждый раз после построения очередным кластером агентов  $A_\sigma$  дерева Штейнера (совокупности маршрутов  $M_\sigma$ ) значение каждого показателя  $d2_j$ , такого, что  $e_j \in E(M_\sigma)$  увеличивается на значение показателя  $d1_j$ , т.е.  $d2_j = d2_j + d1_j$ . После сложения значений показателей счетчика  $D1$  со значениями соответствующих показателей счетчика  $D2$  значения показателей счетчика  $D1$  обнуляются.

Счетчик  $D3$  используется для учета количества включений каждого ребра  $e_j \in E$  графа  $G$  в состав совокупности маршрутов, сформированных в процессе построения деревьев Штейнера агентами множества кластеров  $A$  на всех выполненных итерациях.

Каждый раз после выполнения очередной итерации  $l$  формируются значения показателей счетчиков  $D3$  и  $D4$ . Значение каждого показателя  $d3_j \in D3$  увеличивается на значение показателя  $d2_j$ , т.е.  $d3_j = d3_j + d2_j$ . Значение каждого показателя  $d_j \in D$  равно среднему значению показателя  $d3_j$ , сформированному за  $l$  итераций, т.е.  $d_j = d3_j / l$ .

После обновления показателей счетчиков  $D3$  и  $D4$  показатели счетчика  $D2$  обнуляются.

Состояние каждого ребра  $e_j \in E$  графа поиска решений  $G$  описывается двумя показателями  $h_j$  и  $d_j$ . Значения показателей  $h_j$  и  $d_j$  обновляются путем наращивания на каждой итерации после построения всеми кластерами агентов деревьев Штейнера. Показатель  $h_j$  соответствует феромону-1, а  $d_j$  – феромону-2.

С учетом комбинированной оценки ( $h_j$  и  $d_j$ ) состояния ребер графа поиска решений  $G$  пошаговый процесс построения маршрута  $s_k$  агентом  $a_k$  при параллельно-последовательном подходе осуществляется следующим образом. Пусть к данному моменту  $t$  сформирована часть маршрута  $s_k(t)$ . Агент  $a_k$  формирует список  $E_k(t+1) \in E$  ребер - кандидатов для включения в  $s$ -маршрут.

Для каждого ребра  $e_j \in E_k(t+1)$  рассчитывается параметр  $w_j = d_j \cdot r_j$ , где  $r_j$  – длина ребра  $e_j$ .

Показатель  $w_j$  используется исходя из следующих соображений. Рассмотрим два маршрута  $s_1$  и  $s_2$ . Пусть  $l_1$  длина маршрута  $s_1$ , а  $l_2$  длина маршрута  $s_2$ . В худшем случае общая длина сети  $l = l_1 + l_2$ . Если же маршруты  $s_1$  и  $s_2$  используют общие ребра, то общая длина сети будет меньше на суммарную длину общих ребер. Таким образом, чем больше общих ребер используемых множеством маршрутов, построенных агентами одного кластера, и суммарная длина общих ребер, тем меньше общая длина сети.

По формуле (6) – при мультипликативной свертке, либо по формуле (7) – при аддитивной свертке определяется потенциальная стоимость  $C_j$  каждого ребра  $e_j \in E_k(t+1)$ .

$$C_j = (h_j)^\alpha \cdot (w_j)^\beta \quad (6)$$

$$C_j = \alpha \cdot (h_j) + \beta \cdot (w_j), \quad (7)$$

где  $\alpha, \beta$  – управляющие параметры, которые подбираются экспериментально.

Вероятность  $P_j$  включения ребра  $e_j \in E_k(t+1)$  в формируемый маршрут  $s_k(t+1)$ , определяется следующим соотношением:

$$P_j = C_j / \sum_i C_i \quad \text{для } (i/e_i \in E_k(t+1)). \quad (8)$$

При комбинаторном подходе агентом  $a_k$  на каждой итерации не строится, а выбирается из набора  $S_k = \{s_{kz}/z=1,2,\dots,m\}$  маршрут  $s_{kz}$ . Пусть  $E(s_{kz})$  – множество ребер графа  $G$ , входящих в  $s_{kz}$ .

Вероятность  $P_{kz}$  выбора маршрута  $s_{kz}$  определяется следующим соотношением:

$$P_{kz} = C_{kz} / \sum_i C_{ki}, \quad \text{для } (i/e_i \in E_k(t+1)). \quad (9)$$

где  $C_{kz}$  – потенциальная стоимость маршрута  $s_{kz}$ ,  $C_{kz} = \sum_j C_j$  для  $(j/e_j \in E(s_{kz}))$ .

После формирования (или выбора) агентами одного кластера  $A_\sigma$  маршрутов (каждый агент  $a_k$  – свой маршрут) определяется оценка  $F_\sigma(l)$  сети (дерева Штейнера), построенного агентами кластера  $A_\sigma$  на  $l$ -ой итерации.

После построения всеми кластерами агентов деревьев Штейнера на втором этапе итерации, для каждого ребра  $e_j \in E$  графа  $G$  обновляются значения показателей  $h_j$  и  $d_j$ .

Общее испарение феромона на ребрах графа  $G$  производится в соответствии с формулой (5).

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

Алгоритм построения дерева Штейнера на основе метода муравьиной колонии для модернизированного варианта концепции коллективной адаптации формулируется следующим образом.

1. В соответствии с вышеприведенной методикой на базе исходного множества точек  $P = \{p_i/i=1,2,\dots,n\}$ , формируется ортогональный граф поиска решений  $G=(V,E)$ .

2. На базе исходного множества точек  $P = \{p_i/i=1,2,\dots,n\}$ , с помощью алгоритма Прима строится минимальное связывающее дерево  $R = \{R_k/k=1,2,\dots,n-1\}$ ,  $R_k=(p_i, p_j)$  – ребро МСД.

3. На всех ребрах графа поиска решений  $G=(V,E)$ , откладывается начальное количество  $h(1)$  феромона-1, равное  $Q/m$ , где  $m=|E|$ .

4. Все счетчики  $D1, D2, D3, D$  обнуляются.

5. Выбирается формула (6) или (7) для вычисления потенциальной стоимости ребра  $C_j$ . Задаются значения параметров:  $\alpha, \beta$ .

6. Задается число кластеров агентов –  $N_\sigma$ .

7. Задается число итераций –  $N_l$ .

8.  $l=1$ . ( $l$  – номер итерации).

(На первом этапе каждой итерации, на графе поиска решений  $G=(V,E)$  каждым кластером агентов  $A_\sigma$  строится МДШ).

9.  $\sigma=1$ . ( $\sigma$  – номер кластера).

10. За каждым ребром  $R_k=(p_i, p_j)$  закрепляется агент  $a_k \in A_\sigma$ , который размещается в одной из вершин графа поиска решений  $G=(V,E)$ .

(Далее каждым агентом  $a_k \in A_\sigma$  на графе поиска решений  $G=(V,E)$  строится маршрут  $s_k(l)$ ).

11.  $k=1$ . ( $k$  – номер агента).

12.  $t=0$ . ( $t$  – номер шага).

13. Агент  $a_k$  определяет множество ребер  $E_k(t+1)$  кандидатов для включения в свой маршрут  $s_k(t+1)$ .
14. Рассчитывается стоимость  $C_j$  каждого ребра  $e_j \in E_k(t+1)$ .
15. По формуле (8) для каждого ребра  $e_j \in E_k(t+1)$  рассчитывается вероятность  $P_j$  его включения в формируемый маршрут  $s_k(t+1)$ .
16. Случайным образом в соответствии с рассчитанными вероятностями выбирается ребро  $e_j \in E_k(t+1)$ , которое включается в маршрут  $s_k(t+1)$ .
17. Если маршрут  $s_k(t+1)$  построен, то переход к п.18, иначе  $t=t+1$  и переход к п.13.

18. (*Наращивание показателей счетчика D1*).

Определяется совокупность всех ребер  $E(s_{kz})$  графа  $G$ , входящих в соединение (маршрут)  $s_{kz}$ . Затем значение каждого показателя  $dI_j$ , такого, что  $e_j \in E(s_{kz})$  увеличивается на единицу.

19. Если  $k < (n-1)$ , то  $k=k+1$  и переход к п.12, иначе переход к п.20.

20. (*Наращивание показателей счетчика D2*).

$$D2 = D2 + D1.$$

21. Определяется оценка качества  $F_\sigma(l)$  сети (дерева Штейнера), построенного агентами кластера  $A_\sigma$  на  $l$ -ой итерации.

22. Обнуление счетчика  $D1$ .

22. Если  $\sigma < N\sigma$ , то  $\sigma=\sigma+1$  и переход к п.10, иначе переход к п.23.

23. (*Наращивание показателей счетчика D3*).

$$D3 = D3 + D2$$

24. Обнуление счетчика  $D2$ .

25. Формирование счетчика  $D$ . ( $\forall j$ ) [ $d_j = d3_j/l$ ].

(*Второй этап итерации*)

26. Каждый кластер  $A_\sigma$  агентов откладывает феромон на ребрах построенного им на  $l$ -ой итерации дерева Штейнера в количестве  $\tau_\sigma(l) = \mu \cdot Q \cdot F_\sigma(l)$ , а общее количество феромона на ребре  $e_j$  определится как:

$$h_j(l+1) = h_j(l) + \sum_{\sigma} \tau_{\sigma}(l)$$

(*Третий этап итерации*)

27. В соответствии с формулой (5) производится общее испарение феромона на ребрах графа  $G=(V,E)$ .

28. Выбор лучшего решения, полученного на протяжении всех выполненных итераций.

29. Если  $l < N_l$ , то  $l=l+1$  и переход к п.9, иначе конец работы алгоритма.

Временная сложность этого алгоритма на одной итерации определяется как  $O(n^2)$ , где  $n$  – количество вершин графа  $G=(V,E)$ . В общем случае время работы этого алгоритма зависит от времени жизни колонии  $l$  (число итераций), количества вершин графа  $n$  и числа кластеров агентов  $m$ .

**Экспериментальные исследования.** Для написания программы построения дерева Штейнера на основе метода муравьиной колонии был использован язык C++ в среде Microsoft Visual Studio 2010 для ОС Windows. Тестирование проводилось на ЭВМ с процессором Intel Core 2 Duo T6600 2200 МГц, 4 Гб ОЗУ под управлением операционной системы Windows 7.

Экспериментальные исследования программы построения дерева Штейнера на основе метода муравьиной колонии (AMST) показали, что начальное количество феромона  $Q$  должно быть в 14 раз больше среднего количества значения феромона  $\tau_k(l)$ , откладываемого агентами на каждой итерации.

Для нахождения наилучшего сочетания таких параметров, как объем популяции  $M$  и количество итераций  $T$ , использовались точные решения задачи Штейнера в прямоугольной метрике для тестовых наборов estein OR-library [19], приведенные в

работе D. Warme [20]. Данный тестовый набор представляет собой совокупность задач с размерностью 10, 20, 30, 40, 50, 60, 70, 90, 100, 250 и 500 точек (по 15 задач на каждую размерность). Результаты исследований приведены в табл. 1.

Таблица 1

Результаты исследований

Число вершин	Качество решения			Число генераций T		
	Опт.	< 0,1%	> 0,1%	Min	Max	Средн.
10	8	1	1	20	100	80
20	14	4	2	20	115	90
30	15	5	0	30	200	110
40	5	4	1	32	150	120
50	6	4	0	40	175	150
60	6	4	0	70	200	150
80	68	6	6	60	150	145
100	82	10	8	55	135	120
250	200	20	30	50	140	125
500	460	400	60	60	135	110

На основе обработки экспериментальных исследований была построена средняя зависимость степени качества от числа итераций (рис. 2). В результате экспериментов установлено, что при объеме популяции кластеров агентов  $M=100$  алгоритм сходится в среднем на 130 итерации. В среднем решение близко к оптимальному. В среднем результат, полученный предложенным алгоритмом на 130-ой итерации, отличается от точного на 0.15 процента. Временная сложность адаптивной процедуры на одном итерации –  $O(n)$ .

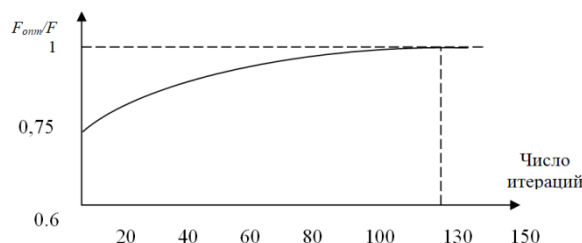


Рис. 2. Зависимость степени качества алгоритма AMST от числа итераций

В качестве аналога для сравнения выбраны пять алгоритмов: algorithm (RMST) [21], Refined Single Trunk Tree (RST-T) [22], the spanning graph based RSMT algorithm (SPAN)[23], the near-optimal Batched Iterated 1-Steiner (BIIS) heuristic [24], and FLUTE [25]. В качестве оценки качества рассматривалась суммарная длина ребер построенного МДШ. Сравнительный анализ с другими алгоритмами производился на стандартных тестовых примерах и схемах (бенчмарках). Генерация оптимальных решений МДШ была произведена программой GeoSteiner 3.1 [26]. Исходные значения тестов были взяты из [27].

Сравнение производилось по величине отклонения в % значения суммарной стоимости ребер МДШ от оптимального по методике, изложенной в [27].

Сравнение отклонений показано в табл. 2.

Таблица 2

## Сравнение отклонений

Число вершин МДШ	Отклонение суммарной длины МДШ от оптимального значения в %					
	RMST	RST-T	SPAN	BIS	FLUTE	AMST
10	11.920	5.016	0.950	0.381	0.126	0.071
20	12.373	14.471	1.124	0.484	0.572	0.327
30	12.526	22.215	1.235	0.533	1.100	0.648
40	12.630	29.130	1.316	0.551	1.633	0.971
50	12.742	35.598	1.391	0.563	2.081	1.540
60	12.766	41.832	1.436	0.566	2.475	1.471
70	12.802	47.781	1.497	0.575	2.837	1.687
80	12.872	53.967	1.554	0.588	3.195	1.910
90	12.892	59.430	1.600	0.587	3.508	2.101
100	12.881	64.722	1.658	0.590	3.798	2.267

Анализируя данные можно сделать вывод, что разработанный алгоритм **AMST** находит решения, превосходящие своих аналогов. Среднее отклонение по всем примерам составляет всего 0,15 %. Из табл. 2 видно, что отклонение критерия от оптимального линейно зависит от числа вершин. Отклонение RST-T очень быстро возрастает с ростом числа вершин и очень существенно даже для сетей среднего размера. **AMST** очень точен для сетей с малым числом вершин, и достаточно точен для сетей с числом вершин до 100. Таким образом, он подходит для приложений СБИС. Временная сложность алгоритма при фиксированных значениях  $M$  и  $T$  лежит в пределах  $O(n)$ .

**Заключение.** Для решения задачи однослойной глобальной трассировки разработан алгоритм построения минимального дерева Штейнера на основе метода МК. Алгоритм базируется на общем подходе, заключающемся в декомпозиции связывающей сети и представлении ее в виде совокупности двутерминальных соединений.

Отличительной особенностью представленного алгоритма построения минимального дерева Штейнера, является то, муравьиная колония разбита на кластеры и поиск конкретного решения задачи осуществляется популяцией кластеров муравьев.

В работе рассматривается комбинаторный и параллельно последовательный подходы к построению МДШ. Впервые используется композитная структура феромона и дифференцированный метод его отложения. Каждый агент в кластере метит путь (ребра маршрута) двумя видами феромона: феромон-1 и феромон-2. Декомпозиция задачи в рамках параллельно-последовательного подхода позволили избежать проблемы очередности прокладки маршрутов и организовать систему коллективной адаптации с высокой степенью целесообразного поведения и сходимости.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Норенков И.П.* Основы автоматизированного проектирования. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2006.
2. *Деньдобренко Б.П., Малика А.С.* Автоматизация проектирования радиоэлектронной аппаратуры. – М.: Высш. шк., 2002. – 384 с.
3. *Promel H.J. and Steger A.* The Steiner Tree Problem: A Tour Through Graphs, Algorithms, and Complexity. Friedrich Vieweg and Son, Braunschweig, 2002.
4. *Zachariasen M. and Rohe A.* Rectilinear group Steiner trees and applications in VLSI design // *Mathematical Programming.* – 2003. – P. 407-433.

5. *Лебедев Б.К., Лебедев О.Б., Чернышев Ю.О.* Основные задачи синтеза топологии СБИС. – Ростов-на-Дону: Изд-во РГАСХМ, 2006. – 184 с.
6. *Лебедев Б.К.* Интеллектуальные процедуры синтеза топологии СБИС. – Таганрог: Изд-во ТРТУ, 2003. – 107 с.
7. *Курейчик В.М., Лебедев Б.К.* Построение минимального связывающего дерева Штейнера на основе метода ветвей и границ // Методы и программы решения оптимизационных задач на графах и сетях. Ч. 2. Материалы 2 всесоюзного совещания. – Новосибирск, 1982. – С. 82-85.
8. *Щемелинин В.М.* Автоматизация топологического проектирования БИС. – М.: МИЭТ, 2001. – 132 с.
9. *Лисин А.В., Файзуллин Р.Т.* Эвристический алгоритм поиска приближённого решения задачи Штейнера, основанный на физических аналогиях // Компьютерная оптика. – 2013. – Т. 37, № 4. – С. 503-510.
10. *Рыженко Н.В.* Задача построения дерева Штейнера для этапа глобальной трассировки. «Высокопроизводительные вычислительные системы и микропроцессоры» // Труды ИМВС РАН. – М., 2003. – Вып. 4. – С. 96-105.
11. *Заглядин Г.Г., Сырцов И.А., Школа А.В.* Алгоритм синтеза множества остовных деревьев для выполнения глобальной трассировки заказных СБИС // Известия высших учебных заведений. Электроника. – 2010. – № 5 (85). – С. 36-40.
12. *Yang, Z.-X.* Geometry Experiment Algorithm Steiner Minimal Tree Problem // Journal of Applied Mathematics. – 2013. – P. 1-10.
13. *Rabkin M.* Efficient use of Genetic Algorithms for the Minimal Steiner Tree and Arborescence Problems with applications to VLSI Physical Design // Genetic Algorithms and Genetic Programming at Stanford. – 2002. – No. 1. – P. 195-202.
14. *Калашиников П.С.* Экспериментальные исследования комбинированного эвристического алгоритма построения дерева Штейнера основанного на методе эволюционного поиска для этапа глобальной трассировки // Известия ТРТУ. – 2005. – № 3.
15. *Лебедев В.Б.* Построение связывающих сетей на основе роевого интеллекта и генетической эволюции // Сборник научных трудов XIV Всероссийской научно-технической конференции "Нейроинформатика-2012". Ч. 2. – М.: Изд-во Физматлит, 2012. – С. 93-103.
16. *Tao Huang, Evangeline F.Y.* Obstacle-avoiding rectilinear Steiner minimum tree construction: an optimal approach // Proceeding ICCAD '10 Proceedings of the International Conference on Computer-Aided Design San Jose, California, 2010. – P. 610-613.
17. *Полежаев П.Н., Миронов А.П., Поляк Р.И.* Применение алгоритмов муравьиной колонии в решении задачи Штейнера // Философские проблемы информационных технологий и киберпространства. – 2015. – Вып. № 1. – С. 96-105.
18. *Patel M.K., Kabat M.R., Tripathy C.R.* A hybrid ACO/PSO based algorithm for QoS multicast routing problem // Ain Shams Engineering Journal. – 2014. – Vol. 5, Issue 1. – P. 113-120.
19. OR-Library is collection of test data for a variety of OR problem. – <http://mscmga.ms.ic.ac.uk>.
20. *Warne D.M.* A new exact algorithm for rectilinear Steiner trees. INFORMS Conf., San-Diego, California, 1997.
21. *Andrew B.K., Mandoiu I.* RMST-Pack: Rectilinear minimum spanning tree algorithms. <http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/RSMT/RMST/>.
22. *Chen H., Qiao C., Zhou F., and Cheng C.-K.* Refined single trunk tree: A rectilinear Steiner tree generator for interconnect prediction // In Proc. ACM Intl. Workshop on System Level Interconnect Prediction. – 2002. – P. 85-89.
23. *Hai Zhou.* Efficient Steiner tree construction based on spanning graphs // In Proc. Intl. Symp. on Physical Design. – 2003. – P. 152-157.
24. *Griffith J., Robins G., Salowe J.S., and Zhang T.* Closing the gap: Near-optimal Steiner trees in polynomial time // IEEE Trans. Computer-Aided Design. – 1994. – No. 13 (11). – P. 1351-1365.
25. *Chris Chu.* FLUTE: Fast lookup table based wirelength estimation technique // In Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design. – 2004. – P. 696-701.
26. GeoSteiner – software for computing Steiner trees. – <http://www.diku.dk/geosteiner/>.
27. *Chu C. and Wong Y.-C.* Fast and accurate rectilinear steiner minimal tree algorithm for VLSI design // In Proc. International Symposium on Physical Design. – ACM Press, 2005. – P. 28-35.

## REFERENCES

1. *Norenkov I.P.* Osnovy avtomatizirovannogo proektirovaniya [Fundamentals of computer-aided design]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2006.
2. *Den'dobrenko B.P., Malika A.S.* Avtomatizatsiya proektirovaniya radioelektronnoy apparatury [Automation of designing of radio-electronic equipment]. Moscow: Vyssh. shk., 2002, 384 p.
3. *Promel H.J. and Steger A.* The Steiner Tree Problem: A Tour Through Graphs, Algorithms, and Complexity. Friedrich Vieweg and Son, Braunschweig, 2002.
4. *Zachariasen M. and Rohe A.* Rectilinear group Steiner trees and applications in VLSI design, *Mathematical Programming*, 2003, pp. 407-433.
5. *Lebedev B.K., Lebedev O.B., Chernyshev Yu.O.* Osnovnye zadachi sinteza topologii SBIS [The main tasks of synthesis of VLSI topology]. Rostov-on-Don: Izd-vo RGASKhM, 2006, 184 p.
6. *Lebedev B.K.* Intellektual'nye protsedury sinteza topologii SBIS [Intellectual procedures for the synthesis of VLSI topology]. Taganrog: Izd-vo TRTU, 2003, 107 p.
7. *Kureychik V.M., Lebedev B.K.* Postroenie minimal'nogo svyazyvayushchego dereva Shteynera na osnove metoda vetvey i granits [Construction of a minimal Steiner binding tree on the basis of the branch and boundary method], *Metody i programmy resheniya optimizatsionnykh zadach na grafakh i setyakh. Ch. 2. Materialy 2 vsesoyuznogo soveshchaniya* [Methods and programs for solving optimization problems on graphs and networks. Part 2. Materials of the 2nd All-Union Conference]. Novosibirsk, 1982, pp. 82-85.
8. *Shchemelinin V.M.* Avtomatizatsiya topologicheskogo proektirovaniya BIS [Automation of topological design of BIS]. Moscow: MIET, 2001, 132 p.
9. *Lisin A.V., Fayzullin R.T.* Evristicheskiy algoritm poiska priblizhennogo resheniya zadachi Shteynera, osnovanny na fizicheskikh analogiyakh [Heuristic algorithm for the search for an approximate solution of the Steiner problem, based on physical analogies], *Komp'yuternaya optika* [Computer optics], 2013, Vol. 37, No. 4, pp. 503-510.
10. *Ryzhenko N.V.* Zadacha postroeniya dereva Shteynera dlya etapa global'noy trassirovki. «Vysokoproizvoditel'nye vychislitel'nye sistemy i mikroprotsessory» [The task of constructing a Steiner tree for the global trace phase. "High-performance computing systems and microprocessors"], *Trudy IMVS RAN* [Proceedings of IMVS RAS]. Moscow, 2003, Issue 4, pp. 96-105.
11. *Zaglyadin G.G., Syrtsov I.A., Shkola A.V.* Algoritm sinteza mnozhestva ostovnykh derev'ev dlya vypolneniya global'noy trassirovki zakaznykh SBIS [Algorithm for the synthesis of a set of spanning trees for performing a global trace of custom VLSI], *Izvestiya vysshikh uchebnykh zavedeniy. Elektronika* [Izvestiya Vysshikh Uchebnykh Zavedenii. Electronics], 2010, No. 5 (85), pp. 36-40.
12. *Yang, Z.-X.* Geometry Experiment Algorithm Steiner Minimal Tree Problem, *Journal of Applied Mathematics*, 2013, pp. 1-10.
13. *Rabkin M.* Efficient use of Genetic Algorithms for the Minimal Steiner Tree and Arborescence Problems with applications to VLSI Physical Design, *Genetic Algorithms and Genetic Programming at Stanford*, 2002, No. 1, pp. 195-202.
14. *Kalashnikov R.S.* Eksperimental'nye issledovaniya kombinirovannogo evristicheskogo algoritma postroeniya dereva Shteynera osnovannogo na metode evolyutsionnogo poiska dlya etapa global'noy trassirovki [Experimental studies of the combined heuristic algorithm for constructing a Steiner tree based on the method of evolutionary search for the stage of global tracing], *Izvestiya TRTU* [Izvestiya TSURE], 2005, No. 3.
15. *Lebedev V.B.* Postroenie svyazyvayushchikh setey na osnove roevogo intellekta i geneticheskoy evolyutsii [Construction of connecting networks based on swarm intelligence and genetic evolution], *Sbornik nauchnykh trudov XIV Vserossiyskoy nauchno-tekhnicheskoy konferentsii "Neuroinformatika-2012"* [Collection of scientific papers of the XIV All-Russian Scientific and Technical Conference "Neuroinformatics-2012."]. Part 2. Moscow: Izd-vo Fizmatlit, 2012, pp. 93-103.
16. *Tao Huang, Evangeline F.Y.* Obstacle-avoiding rectilinear Steiner minimum tree construction: an optimal approach, *Proceeding ICCAD '10 Proceedings of the International Conference on Computer-Aided Design San Jose, California, 2010*, pp. 610-613.
17. *Polezhaev P.N., Mironov A.P., Polyak R.I.* Primenenie algoritmov murav'inoi kolonii v reshenii zadachi Shteynera [Application of the algorithms of the ant colony in solving the Steiner problem], *Filosofskie problemy informatsionnykh tekhnologiy i kiberprostranstva* [Philosophical Problems of Information Technologies and Cyberspace], 2015, Issue No. 1, pp. 96-105.

18. Patel M.K., Kabat M.R., Tripathy C.R. A hybrid ACO/PSO based algorithm for QoS multicast routing problem, *Ain Shams Engineering Journal*, 2014, Vol. 5, Issue 1, pp. 113-120.
19. OR-Library is collection of test data for a variety of OR problem. Available at: <http://mscmga.ms.ic.ac.uk>.
20. Warme D.M. A new exact algorithm for rectilinear Steiner trees. INFORMS Conf., San-Diego, California, 1997.
21. Andrew B.K., Mandoiu I. RMST-Pack: Rectilinear minimum spanning tree algorithms. Available at: <http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/RSMT/RMST/>.
22. Chen H., Qiao C., Zhou F., and Cheng C.-K. Refined single trunk tree: A rectilinear Steiner tree generator for interconnect prediction, *In Proc. ACM Intl. Workshop on System Level Interconnect Prediction*, 2002, pp. 85-89.
23. Hai Zhou. Efficient Steiner tree construction based on spanning graphs, *In Proc. Intl. Symp. on Physical Design*, 2003, pp. 152-157.
24. Griffith J., Robins G., Salowe J.S., and Zhang T. Closing the gap: Near-optimal Steiner trees in polynomial time, *IEEE Trans. Computer-Aided Design*, 1994, No. 13 (11), pp. 1351-1365.
25. Chris Chu. FLUTE: Fast lookup table based wirelength estimation technique, *In Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design*, 2004, pp. 696-701.
26. GeoSteiner – software for computing Steiner trees <http://www.diku.dk/geosteiner/>.
27. Chu C. and Wong Y.-C. Fast and accurate rectilinear steiner minimal tree algorithm for VLSI design, *In Proc. International Symposium on Physical Design*. ACM Press, 2005, pp. 28-35.

Статью рекомендовала к опубликованию д.т.н., профессор А.Г. Коробейников.

**Лебедев Борис Константинович** – Южный федеральный университет; e-mail: lebedev.b.k@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 89282897933; кафедра систем автоматизированного проектирования; профессор.

**Лебедев Олег Борисович** – e-mail: lebedev.ob@mail.ru; тел.: 89085135512; кафедра систем автоматизированного проектирования; доцент.

**Лебедева Екатерина Олеговна** – e-mail: lebedev.ob@mail.ru; тел.: 89289591426; кафедра систем автоматизированного проектирования; студентка.

**Lebedev Boris Konstantinovich** – Southern Federal University; e-mail: lebedev.b.k@gmail.com; 44, Nekrasovsky, Taganrog, 347928, Russia; phone: 89282897933; the department of computer aided design; professor.

**Lebedev Oleg Borisovich** – e-mail: lebedev.ob@mail.ru; phone: 89085135512; the department of computer aided design; associate professor.

**Lebedeva Ekaterina Olegovna** – e-mail: lebedev.ob@mail.ru; phone: 89289591426; the department of computer aided design; student.

УДК 519.712.2

**Л.А. Гладков, Н.В. Гладкова, С.Н. Лейба**

### **РАЗРАБОТКА И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ГИБРИДНОГО АЛГОРИТМА РЕШЕНИЯ ЗАДАЧ РАЗМЕЩЕНИЯ И ТРАССИРОВКИ\***

*Рассматривается гибридный алгоритм решения задач размещения и трассировки элементов схем цифровой электронно-вычислительной аппаратуры. Отмечена актуальность и важность рассматриваемой задачи и многообразие существующих подходов к решению подобного рода задач. Приведена постановка задачи, выбраны ограничения области допустимых решений и сформулирован критерий оценки качества получаемых решений. Предложен новый гибридный подход к решению рассматриваемой задачи на основе*

---

\* Работа выполнена при финансовой поддержке РФФИ (проект № 17-01-00627).