

Гладков Леонид Анатольевич – Южный федеральный университет; e-mail: leo_gladkov@mail.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371625; кафедра САПР; доцент.

Гладкова Надежда Викторовна – e-mail: nadyusha.gladkova77@mail.ru; кафедра САПР; старший преподаватель.

Лейба Сергей Николаевич – e-mail: lejba.sergej@mail.ru; кафедра САПР; аспирант.

Gladkov Leonid Anatol'evich – Southern Federal University; e-mail: leo_gladkov@mail.ru; 44, Nekrasovskiy lane, Taganrog, 347928, Russia; phone: +78634371625; CAD department; associated professor.

Gladkova Nadezhda Viktorovna – e-mail: nadyusha.gladkova77@mail.ru; CAD department; senior teacher.

Leiba Sergey Nikolaevich – e-mail: lejba.sergej@mail.ru; CAD department; postgraduate student.

УДК 004.896

DOI 10.23683/2311-3103-2019-4-174-186

Б.К. Лебедев, О.Б. Лебедев, Е.О. Лебедева**ПАРАЛЛЕЛЬНО ПОСЛЕДОВАТЕЛЬНЫЙ БИОИНСПИРИРОВАННЫЙ
АЛГОРИТМ ПОСТРОЕНИЯ МИНИМАЛЬНОГО ДЕРЕВА ШТЕЙНЕРА***

Рассматривается параллельно последовательный подход к построению минимального дерева Штейнера. Разработанный алгоритм базируется на общем подходе, заключающемся в декомпозиции связывающей сети и представлении ее в виде совокупности двухтерминальных соединений. Для цепи t_i на множестве вершин X_i , $|X_i|=n_i$ графа G с помощью алгоритма Прима строится минимальное связывающее дерево $R_i=\{r_{ik}|i=1, 2, \dots, n_i\}$. Для каждого ребра $r_{ik} \in R_i$ формируется маршрут s_{ik} , связывающий на графе G пару вершин, соответствующих ребру r_{ik} . Каждому маршруту s_{ik} соответствует множество $\Gamma(s_{ik})$ ребер графа G . Задача построения минимального дерева Штейнера сводится к задаче построения и выбора s -маршрута s_{ik} на графе G , для каждого ребра r_{ik} . 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. Качество маршрута, построенного агентом, будет определяться наличием общих участков с маршрутами, построенными другими агентами кластера. Чем больше в составе маршрута общих участков с маршрутами, построенными другими агентами кластера, тем меньше суммарная длина ребер минимального дерева Штейнера. 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. Особенностью представленного муравьиного алгоритма построения минимального дерева Штейнера является то, что муравьиная колония разбита на кластеры и поиск конкретного решения задачи осуществляется популяцией кластеров муравьев. Задача построения муравьями каждого кластера A_σ дерева Штейнера сводится к задаче построения и выбора в графе G совокупности M_σ s -маршрутов, покрывающих минимальное дерево Штейнера. В отличие от канонической парадигмы муравьиной колонии, предложена модифицированная жадная стратегия построения ориентированного маршрута на модели представления решения. Концепция коллективной адаптации (адаптивного поведения) кластера агентов, используемая в

* Работа выполнена при финансовой поддержке гранта РФФИ № 18-07-00737 А.

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 кластера, что косвенно способствует достижению глобальной цели коллектива. Состояние каждого ребра $e_j \in E$ графа поиска G решений описывается двумя параметрами 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. В работе впервые используется композитная структура феромона и дифференцированный метод его отложения. Каждый муравей в группе метит путь (ребра маршрута) двумя видами феромона: феромон-1 и феромон-2. Это обеспечивает более высокую вероятность локализации глобального экстремума задачи. Временная сложность этого алгоритма на одной итерации определяется как $O(n^2)$. После выполнения всех действий на итерации находится агент с лучшим решением, которое запоминается. Далее осуществляется переход на следующую итерацию.

Минимальное дерево Штейнера; СБИС; муравьиная колония; кластер; феромон; композитная структура; глобальный экстремум; роевой интеллект; бионический поиск.

В.К. Lebedev, О.В. Lebedev, Е.О. Lebedeva

PARALLEL AND CONSISTENT BIOINSPIRED ALGORITHM FOR CONSTRUCTION OF A STEINER MINIMAL TREE

The paper considers a parallel sequential approach to constructing a minimal Steiner tree. The developed algorithm is based on a general approach, consisting in the decomposition of a connecting network and its presentation in the form of a combination of two-terminal connections. For the chain t_i on the set of vertices X_i , $|X_i|=n_i$ of the graph G , using the Prim algorithm, we construct the minimal connecting tree $R_i=\{r_{ik}|i=1, 2, \dots, n_{i-1}\}$. For each edge $r_{ik} \in R_i$, a route s_{ik} is formed that connects on the graph G a pair of vertices corresponding to the edge r_{ik} . Each route s_{ik} corresponds to the set $\Gamma(s_{ik})$ of edges of G . The problem of constructing a minimal Steiner tree reduces to the problem of constructing and choosing an s -route s_{ik} on the graph G , for each edge r_{ik} . For each agent located at the vertex p_i , the coordinates of the vertex p_j to which the s -route is laid are determined and the possible directions of its movement along the edges of the orthogonal graph $G=(V,E)$ are determined, which ensure the construction of the s -route of minimum length. The quality of the route built by the agent will be determined by the presence of common sections with routes built by other agents of the cluster. The more common sections of the route with routes constructed by other cluster agents, the smaller the total length of the Steiner tree ribs. The decomposition of the problem in the framework of a parallel-sequential approach allowed us to avoid the problem of the sequence of routing and organize a collective adaptation system with a high degree of appropriate behavior and convergence. A feature of the presented ant algorithm for constructing a minimal Steiner tree is that the ant colony is divided into clusters and the search for a specific solution to the problem is carried out by a population of ant clusters. The task of ants constructing each cluster $A\sigma$ of the Steiner tree is reduced to the problem of constructing and selecting in the graph G the set of $M\sigma$ s -routes covering the minimal Steiner tree. In contrast to the canonical paradigm of the ant colony, a modified greedy strategy is proposed for constructing an oriented route on the solution representation model. The concept of collective adaptation (adaptive behavior) of an agent cluster used in the above approach is as follows. The global goal of the team (agent cluster) is to build a minimal Steiner tree. The local goal of each agent is to build a route of maximum cost, that is, a route that maximally matches the routes built by other agents of the cluster, which indirectly contributes to the achievement of the global goal of the team. The state of each edge $e_j \in E$ of the graph of the search for G solutions is described by two parameters h_j and d_j . The values of the indicators h_j and d_j are updated by increasing at each iteration after all the clusters of agents build Steiner trees. For the first time, the composite structure of pheromone and the differentiated method of its deposition are used. Each ant in the group marks the path (edges of the route) with two types of pheromone: pheromone-1 and pheromone-2. This provides a

higher probability of localization of the global extremum of the problem. The time complexity of this algorithm at one iteration is defined as $O(n^2)$. After all the actions are completed, the agent with the best solution is found in the iteration, which is remembered. Next, the transition to the next iteration.

Steiner minimal tree; VLSI; ant colony; clusters; pheromone; composite structure; global extremum; swarm intelligence; bionic search.

Введение. Задача Штейнера является фундаментальной задачей в проектировании сверхбольших интегральных схем (Very Large Scale Integration, VLSI). 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.

Построение кратчайших связывающих сетей одна из наиболее сложных проблем при проектировании внутрисхемных соединений электронных устройств. Поскольку при машинной трассировке соединений используется ортогональная опорная сетка, интерес представляет построение связывающих деревьев Штейнера в ортогональной метрике [1–3].

Существует множество различных способов, которыми можно решить задачу Штейнера. Основной недостаток многих эвристических алгоритмов решения задач Штейнера, заключается в том, что проведение новых соединений зависит от ранее выполненных, которые рассматриваются как преграды и ограничивают проведение новых трасс. Существенным средством повышения эффективности алгоритмов построения минимального дерева Штейнера (МДШ) 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 [4–10]. Результаты применения эволюционного моделирования к задаче построения МДШ 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.

1. Постановка задачи построения МДШ. Известная проблема Штейнера состоит в следующем [2, 3]. 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 $d_{ij}=\lvert x_i-x_j \rvert + \lvert y_i-y_j \rvert$.

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 сло-

вами, при отыскании дополнительных точек можно ограничиться узлами ортогональной сетки $G=(V,E)$, а само дерево Штейнера полностью укладывается в ортогональную сеть. 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. $P \subset V^* \subset V, E^* \subset E$.

Рассмотрим концепцию построения МДШ, используемую в данной работе. На множестве P основных вершин графа $G=(V,E)$, 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). Назовем цепь в графе $G=(V,E)$, связывающую две основные вершины, s -маршрутом. Для каждого ребра R_k на графе G существует набор $S_k=\Gamma(R_k)$ альтернативных вариантов s -маршрута, равный $S_k=\{s_{kz}|z=1, 2, \dots, n_z\}$. $S=\{S_k|k=1, 2, \dots, n-1\}$ – множество наборов.

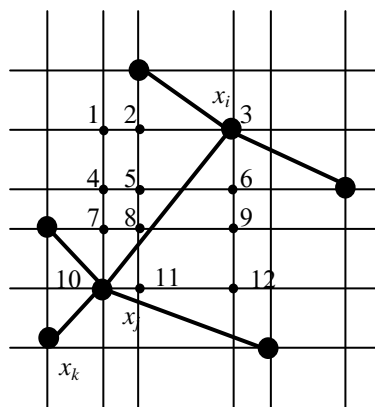


Рис. 1. Формирование набора альтернативных вариантов s -ребер

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 [11,12]. Первый подход базируется на идеях волнового алгоритма Ли [11] 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.

Формирование множества S_k вариантов s -маршрутов 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 альтернативных вариантов s -маршрутов для ребра $R_k=(p_i, p_j)$ включаются 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: $s_{k1}=(3,6,9,12,11,10)$, $s_{k2}=(3,2,1,4,7,10)$, $s_{k3}=(3,2,5,4,7,10)$ 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).

В работе рассматривается параллельно последовательный подход к построению МДШ: каждый из s -маршрутов строится последовательно, но все $(n-1)$ s -маршрутов, покрывающих МДШ, строятся параллельно. Подход базируется на методе муравьиной колонии.

2. Коллективная адаптация кластера агентов муравьиной колонии, использующая композитную структуру феромона и дифференцированный метод его отложения. Для поиска решения задачи используется ортогональный граф $G=(V, E)$, построенный по выше приведенной методике.

Процесс поиска МДШ итерационный. Предварительно на всех ребрах графа G откладывается одинаковое (небольшое) количество феромона, равное Q/m , где $m=|E|$. Каждая итерация l включает три этапа. Отличительной особенностью муравьиного алгоритма является то, что в его состав входит встроенный конструктивный алгоритм решения основной задачи.

На первом этапе каждой итерации муравьиного алгоритма с помощью встроенного (конструктивного) алгоритма формируется популяция решений МДШ. Поиск МДШ осуществляется на базе предварительно построенного алгоритмом Прима минимального связывающего дерева (МСД) $R=\{R_k|k=1, 2, \dots, n-1\}$.

В общем случае формирование популяции решений МДШ осуществляется популяцией кластеров агентов $A=\{A_\sigma|\sigma=1, 2, \dots, n_\sigma\}$ [3]. 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_σ дерева Штейнера сводится к задаче построения и выбора в графе G совокупности M_σ s -маршрутов, покрывающих МДШ, причем каждый маршрут $m_{\sigma k} \in M_\sigma$ соответствует ребру $R_k=(p_i, p_j)$.

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).

Таким образом, на первом этапе каждой итерации каждый кластер A_σ популяции A строит свое дерево Штейнера. При этом каждый агент $a_{\sigma k}$ кластера A_σ строит свой вариант маршрута $s_{kz}(l)$.

Второй этап выполняется после построения всеми кластерами агентов популяции A независимо друг от друга деревьев Штейнера.

На втором этапе на ребрах графа G , откладывается феромон. В работе используется циклический (ant-cycle) метод муравьиных систем. Феромон откладывается каждым кластером A_σ на ребрах сформированного агентами дерева Штейнера, после построения всеми кластерами агентов деревьев Штейнера.

Количество феромона откладываемого на ребрах МДШ, построенного кластером агентов A_σ , пропорционально оценке качества МДШ.

Чем больше феромона будет отложено на рёбрах графа G , тем большее количество агентов будет включать эти рёбра в собственные маршруты.

На третьем этапе осуществляется испарение феромона.

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 кластера, 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 $a_{\sigma k}$, будет определяться наличием общих участков с маршрутами, построенными другими агентами кластера. 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 $a_{\sigma k}$ в кластере A_{σ} 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_{σ} , откладывается феромон (феромон-1) в количестве, пропорциональном оценке решения и феромон-2, отражающий значение счетчика числа включений.

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 кластеров 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 промежуточных 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 в начале 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_{σ} маршрутов, сформированных в процессе построения дерева Штейнера агентами кластера A_{σ} .

Каждый раз после построения очередным агентом $a_{\sigma k}$ кластера A_{σ} соединения (маршрута) $m_{\sigma k}$ определяется совокупность всех ребер $E(m_{\sigma k})$ графа G , входящих в соединение (маршрут) $m_{\sigma k}$. Затем значение каждого показателя $d1_j$, такого, что $e_j \in E(m_{\sigma k})$ увеличивается на единицу. После построения совокупности маршрутов M_{σ} агентами кластера A_{σ} определяется совокупность всех ребер $E(M_{\sigma})$ графа G , входящих в эти соединения (маршруты), т.е. в состав построенного дерева Штейнера.

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 всех кластеров A на одной итерации.

Каждый раз после построения очередным кластером агентов A_σ дерева Штейнера (совокупности маршрутов M_σ) 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 D . 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 D показатели счетчика $D2$ обнуляются.

Состояние каждого ребра $e_j \in E$ графа поиска G решений описывается двумя параметрами 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. В работе используется композитный подход к формированию оценки (стоимости) ребра в процессе построения маршрута.

С учетом композитной оценки (h_j и d_j) состояния ребер графа поиска решений G пошаговый процесс построения маршрута m_{ok} агентом a_{ok} при параллельно-последовательном подходе осуществляется следующим образом. Пусть к данному моменту t сформирована часть маршрута $m_{ok}(t)$. Агент a_{ok} 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.

При мультипликативной свертке по формуле:

$$C_j = (h_j)^\alpha \cdot (w_j)^\beta, \quad (1)$$

либо при аддитивной свертке по формуле:

$$C_j = \alpha \cdot (h_j) + \beta \cdot (w_j), \quad (2)$$

определяется потенциальная стоимость C_j каждого ребра $e_j \in E_k(t+1)$, где α, β – управляющие параметры, которые подбираются экспериментально.

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

$$P_j = C_j / \sum_i (C_i), \text{ для } (i/e_i \in E_k(t+1)). \quad (3)$$

После формирования агентами одного кластера A_σ маршрутов (каждый агент $a_{\sigma k}$ – свой маршрут) определяется оценка $F_\sigma(l)$ сети (дерева Штейнера), построенного агентами кластера A_σ 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:

$$h_j = h_j (1 - \rho), \quad (4)$$

где ρ – коэффициент обновления.

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 МСД $R = \{R_k | k = 1, 2, \dots, n-1\}$, $R_k = (p_i, p_j)$ – 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 (1) или (2) для вычисления потенциальной стоимости ребра C_j . Задаются значения параметров: α, β .

6. Задается число кластеров агентов – N_σ .

7. Задается число итераций – N_l .

8. $l = 1$. (l – номер итерации).

(Первый этап итерации)

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

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

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

(Далее каждым агентом $a_{\sigma k} \in A_\sigma$ 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 $a_{\sigma k}$ определяет множество ребер $E_k(t+1)$ кандидатов для включения в свой маршрут $m_{\sigma k}(t+1)$.

14. Рассчитывается стоимость C_j каждого ребра $e_j \in E_k(t+1)$.

15. По формуле (3) для каждого ребра $e_j \in E_k(t+1)$ рассчитывается вероятность P_j его включения в формируемый маршрут $m_{\sigma k}(t+1)$.

16. Случайным образом в соответствии с рассчитанными вероятностями выбирается ребро $e_j \in E_k(t+1)$, которое включается в маршрут $m_{\sigma k}(t+1)$.

17. Если маршрут $m_{\sigma k}(t+1)$ построен, то переход к п.18, иначе $t = t + 1$ и переход к п.13.

18. *(Наращивание показателей счетчика $D1$).*

Определяется совокупность всех ребер $U(m_{\sigma k})$ графа G , входящих в соединение (маршрут) $m_{\sigma k}$. Затем значение каждого показателя dI_j , такого, что $e_j \in E(m_{\sigma k})$ 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 длина $F_{\sigma}(l)$ сети (дерева Штейнера), построенного агентами кластера A_{σ} 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_{σ} агентов откладывает феромон на ребрах построенного им на l -ой итерации дерева Штейнера в количестве:

$$\tau_{\sigma}(l) = Q \setminus F_{\sigma}(l),$$

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 (4) производится общее испарение феромона на ребрах графа $G=(V, E)$.

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

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

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_{σ} .

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 агентов m , помещаемых в каждую вершину графа поиска решений $G=(V, E)$.

3. Экспериментальные исследования. Целью экспериментальных исследований является выявление зависимостей и влияния на качество МДШ различных комбинаций управляющих параметров и структур.

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 [13], приведенные в работе D.Warme [14]. 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.

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.

Таблица 1

Результат 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	40	60	60	135	110

В качестве аналога для сравнения выбраны пять алгоритмов: algorithm (RMST) [15], Refined Single Trunk Tree (RST-T) [16], the spanning graph based RSMT algorithm (SPAN) [17], the near-optimal Batched Iterated 1-Steiner (BIIS) heuristic [18], and FLUTE [19]. 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 [20]. Исходные значения тестов были взяты из [21].

Анализируя данные можно сделать вывод, что разработанный алгоритм **AMST** находит решения, превосходящие своих аналогов. Среднее отклонение по всем примерам составляет всего 0,15 %. 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	BIIS	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

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. В работе рассмотрен параллельно последовательный подход к построению МДШ и глобальной трассировки. 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.

Разработана модификация метода муравьиной колонии для решения задачи построения МДШ, позволяющая находить приемлемые решения при меньших машинных затратах.

Особенностью представленного алгоритма построения МДШ является то, что муравьиная колония разбита на кластеры и поиск конкретного решения задачи осуществляется популяцией кластеров муравьев, что позволяет использовать представление решений в виде совокупности элементов. В отличие от канонической парадигмы МК, предложена модифицированная жадная стратегия построения ориентированного маршрута на графовой модели представления решения.

В работе впервые используется композитная структура феромона и дифференцированный метод его отложения. Каждый муравей в группе метит путь (ребра маршрута) двумя видами феромона: феромон-1 и феромон-2. Отличительные черты процесса формирования маршрута отдельным муравьем a_k заключаются в использовании комбинированного подхода к формированию оценки, определяющей выбор ребра для включения в формируемый маршрут. Это обеспечивает более высокую вероятность локализации глобального экстремума задачи. 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 агентов m , помещаемых в каждую вершину графа поиска решений $G=(V, E)$. Сравнительный анализ с другими алгоритмами глобальной трассировки производился на стандартных тестовых примерах и схемах (бенчмарках). На каждой из бенчмарок в среднем результаты были лучше на 2–5 %.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Лебедев Б.К., Лебедев О.Б., Чернышев Ю.О.* Основные задачи синтеза топологии СБИС. – Ростов-на-Дону: Изд-во РГАСХМ, 2006. – 154 с.
2. *Лебедев Б.К.* Интеллектуальные процедуры синтеза топологии СБИС. – Таганрог: Изд-во ТРТУ, 2003. – 132 с.
3. *Лебедев О.Б.* Построение дерева Штейнера на основе метода муравьиной колонии // Тр. конгресса по интеллектуальным системам и информационным технологиям «AIS–IT'09». Научное издание в 4-х т. Т. 1. – М.: Физматлит, 2009. – С. 58-65.
4. *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.
5. *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. – № 1. – P. 195-202.
6. *Калашиников Р.С.* Экспериментальные исследования комбинированного эвристического алгоритм построения дерева Штейнера основанного на методе эволюционного поиска для этапа глобальной трассировки // Известия ТРТУ. – 2005. – № 3 (47). – С. 34-41.
7. *Лебедев В.Б.* Построение связывающих сетей на основе роевого интеллекта и генетической эволюции // Сб. научных трудов XIV Всероссийской научно-технической конференции «НЕЙРОИНФОРМАТИКА-2012». Ч. 2. – М.: Изд-во Физматлит, 2012. – С. 93-103.

8. *Tao Huang, Evangeline F.Y.* Obstacle-avoiding rectilinear Steiner minimum tree construction: an optimal approach // *Proceedings of the International Conference on Computer-Aided Design San Jose, California*. 2010. – P. 610-613.
9. *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.
10. *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.
11. *Норенков И.П.* Основы автоматизированного проектирования. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2006. – 288 с.
12. *Щемелинин В.М.* Автоматизация топологического проектирования БИС. – М.: МИЭТ, 2001. – 188 с.
13. OR-Library is collection of test data for a variety of OR problem. – <http://mscmga.ms.ic.ac.uk>.
14. *Warne D.M.* A new exact algorithm for rectilinear Steiner trees. *INFORMS Conf., San-Diego, California*, 1997. – P. 455-472.
15. *Andrew B. Kahng and Ion Mandoiu.* RMST-Pack: Rectilinear minimum spanning tree algorithms. – <http://vlscad.ucsd.edu/GSRC/bookshelf/Slots/RSMT/RMST/>.
16. *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.
17. *Hai Zhou.* Efficient Steiner tree construction based on spanning graphs // *In Proc. Intl. Symp. on Physical Design*. – 2003. – P. 152-157.
18. *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.
19. *Chris Chu.* FLUTE: Fast lookup table based wirelength estimation technique // *In Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design*. – 2004. – P. 696-701.
20. GeoSteiner – software for computing Steiner trees. – <http://www.diku.dk/geosteiner/>.
21. *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. *Lebedev B.K., Lebedev O.B., Chernyshev Yu.O.* Osnovnye zadachi sinteza topologii SBIS [The main tasks of the synthesis of VLSI topology]. Rostov-on-Don: Izd-vo RGASKHM, 2006, 154 p.
2. *Lebedev B.K.* Intellektual'nye protsedury sinteza topologii SBIS [Intelligent synthesis procedures for VLSI topology]. Taganrog: Izd-vo TRTU, 2003, 132 p.
3. *Lebedev O.B.* Postroenie dereva SHteynera na osnove metoda murav'inoy kolonii [Building a Steiner tree based on the ant-colony method], *Tr. kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam «AIS-IT'09»* [Proceedings of the congress on intellectual systems and information technologies "AIS-IT'09"]. Scientific publication in 4 vol. Vol. 1. Moscow: Fizmatlit, 2009, pp. 58-65.
4. *Lebedev V.B., Lebedev O.B.* Roeffvoy intellekt na osnove integratsii modeley adaptivnogo povedeniya murav'inoy i pchelinoy koloniy [Swarm intelligence based on the integration of models of adaptive behavior of formic and bee colonies], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 41-47.
5. *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.
6. *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 evolutionary search method for the global trace stage], *Izvestiya TRTU* [Izvestiya TSURE], 2005, No. 3 (47), pp. 34-41.
7. *Lebedev V.B.* Postroenie svyazyvayushchikh setey na osnove roevogo intellekta i geneticheskoy evolyutsii [Building connecting networks based on swarm intelligence and genetic evolution], *Sb. nauchnykh trudov XIV Vserossiyskoy nauchno-tekhnicheskoy konferentsii «NEYROINFORMATIKA-2012»* [Collection of scientific papers of the XIV All-Russian Scientific-Technical Conference "NEUROINFORMATICS-2012"]. Part 2. Moscow: Izd-vo Fizmatlit, 2012, pp. 93-103.

8. *Tao Huang, Evangeline F.Y.* Obstacle-avoiding rectilinear Steiner minimum tree construction: an optimal approach, *Proceedings of the International Conference on Computer-Aided Design San Jose, California. 2010*, pp. 610-613.
9. *Polezhaev P.N., Mironov A.P., Polyak R.I.* Primenenie algoritmov murav'inoi kolonii v reshenii zadachi Shteynera [Application of algorithms of the ant colony in solving the Steiner problem], *Filosofskie problemy informatsionnykh tekhnologiy i kiberprostranstva* [Philosophical problems of information technology and cyberspace], 2015, Issue No. 1, pp. 96-105.
10. *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.
11. *Norenkov I.P.* Osnovy avtomatizirovannogo proektirovaniya [Basics of computer-aided design]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2006, 288 p.
12. *Shchemelinin V.M.* Avtomatizatsiya topologicheskogo proektirovaniya BIS [Automation of topological design of LSI]. Moscow: MIET, 2001, 188 p.
13. OR-Library is collection of test data for a variety of OR problem. Available at: <http://mscmga.ms.ic.ac.uk>.
14. *Warne D.M.* A new exact algorithm for rectilinear Steiner trees. *INFORMS Conf., San-Diego, California, 1997*, pp. 455-472.
15. *Andrew B. Kahng and Ion Mandoiu.* RMST-Pack: Rectilinear minimum spanning tree algorithms. Available at: <http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/RSMT/RMST/>.
16. *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.
17. *Hai Zhou.* Efficient Steiner tree construction based on spanning graphs, *In Proc. Intl. Symp. on Physical Design*, 2003, pp. 152-157.
18. *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.
19. *Chris Chu.* FLUTE: Fast lookup table based wirelength estimation technique, *In Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design*, 2004, pp. 696-701.
20. GeoSteiner – software for computing Steiner trees. Available at: <http://www.diku.dk/geosteiner/>.
21. *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 рекомендовал к опубликованию д.т.н., профессор А.В. Чернов.

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: lbedevakate@mail.ru; тел.: 89289591426; кафедра систем автоматизированного проектирования; аспирант.

Lebedev Boris Konstantinovich – Southern Federal University; e-mail: lebedev.b.k@gmail.com; 44, Nekrasovsky, Taganrog, 347928, Russia; phone: +79282897933; the department of computer aided design; professor.

Lebedev Oleg Borisovich – e-mail: lebedev.ob@mail.ru; phone: +79085135512; the department of computer aided design; associate professor.

Lebedeva Ekaterina Olegovna – e-mail: lbedevakate@mail.ru; phone: +79289591426; the department of computer aided design; graduate student.