

Построение кратчайших связывающих соединений методом кристаллизации россыпи альтернатив

В.Б. Лебедев, Б.К. Лебедев

Инженерно-технологическая академия ЮФУ, lebedev.b.k@gmail.com

Аннотация — Предлагается метод построения кратчайших связывающих соединений на основе новой, предложенной авторами, парадигмы коллективного интеллекта - кристаллизации россыпи альтернатив (КРА). Проведены экспериментальные исследования, подтвердившие эффективность предложенной парадигмы.

Ключевые слова — автоматизация проектирования, дерево Штейнера, метод кристаллизации россыпи альтернатив, коллективный интеллект, адаптивное поведение, самоорганизация, оптимизация.

I. ВВЕДЕНИЕ

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]. Алгоритмы построения дерева Штейнера можно разбить на два класса: последовательный и комбинаторный [1,2].

При последовательном подходе соединения строятся последовательно. В основе большинства из них лежит волновой алгоритм Ли и его модификации [1-3]. Качество решения во многом определяется порядком трассируемых соединений. Анализ существующих методов упорядочения показывает, что не существует радикального метода, гарантирующего построения дерева Штейнера.

Сущность комбинаторных алгоритмов заключается в том, что для каждого соединения t_i формируется набор вариантов его реализации [4, 5]. Цель задачи заключается в нахождении на заданном наборе таких вариантов, которые обеспечивают наилучшее решение. В последние годы интенсивно разрабатывается научное направление с названием «Природные вычисления» (Natural Computing), объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений [6-10]. Одним из новых направлений таких методов являются мультиагентные методы интеллектуальной

оптимизации, базирующиеся на моделировании коллективного интеллекта [5-8].

Разработка новых алгоритмов заключается в изучении и использовании метаэвристик, заложенных в природных механизмах принятия решений [10-14]. В работе наряду с метаэвристиками, на которых построены роевые алгоритмы [15-18], используется метаэвристика, имеющая тенденцию к использованию альтернатив (вариантов компонентов) из наилучших найденных решений. В процессе эволюционной коллективной адаптации методами дискриминантного анализа формируются оценки приспособленности альтернатив. Приспособленность альтернатив рассматривается как вероятность ее использования в формируемом решении. Совокупность данных об альтернативах и их оценках составляет **россыпь альтернатив**. Дискриминантный анализ альтернатив в процессе эволюционной коллективной адаптации назван по аналогии с процессами вычленения объектов (формирования кристаллов) кристаллизацией. Другими словами в процессе эволюционной коллективной адаптации производится вычленение из множества вариантов наиболее приспособленных альтернатив. Отсюда название метода оптимизации - **метод кристаллизации россыпи альтернатив (КРА), (Crystallization of alternatives field (CAF))** [18].

В работе излагается алгоритм построения дерева Штейнера на основе метода кристаллизации россыпи альтернатив. С этой целью разработана единая структура данных. С учетом особенностей единой структуры данных разработан поисковый процесс, базирующийся на моделировании коллективной адаптации.

II. ФОРМУЛИРОВКА ПРОБЛЕМЫ ШТЕЙНЕРА

Пусть задано множество точек $P = \{p_i | i = 1, 2, \dots, n\}$, расположенных на плоскости, на которых нужно построить дерево Штейнера. Отличительной особенностью дерева Штейнера является то, что оно кроме основных вершин P содержит некоторое множество дополнительных вершин. Проведем через точки P линии, параллельные осям X и Y (рис. 1). Образуется ортогональная сетка. Обозначим через V множество точек пересечений линий сетки и назовем их узлами сетки. Сопоставим ортогональной сетке

граф $G=(V,E)$. Множеству точек P соответствует множество вершин $V^* \subset V$. Задача решается в два этапа. На первом этапе, на множестве вершин V^* алгоритмом Прима строится минимальное связывающее дерево (МСД) $D=\{D_l \mid l=1,2,\dots,n-1\}$; $D_l=(v_i, v_j)$ - ребро МСД. Назовем цепь в графе $G=(V,E)$, связывающую две основные вершины, s -маршрутом. Для каждого ребра D_l на графе $G=(V,E)$ формируется набор $S_l=\Gamma(D_l)$ альтернативных вариантов маршрутов. $S_l=\{s_{lk} \mid k=1,2,\dots,m\}$. $S=\{S_l \mid l=1,2,\dots,n-1\}$ - множество наборов.

Формирование S_l осуществляется исходя из следующих посылок: длина s_{lk} должна быть минимальна; варианты различных s -маршрутов должны обеспечивать максимально возможное совпадение друг с другом [1]. Например: s -маршруты, связывающие основные вершины v_i и v_j , имеют вид: $(v_i, 2, 1, 4, 7, v_j)$, $(v_i, 2, 5, 8, 11, v_j)$, $(v_i, 6, 9, 12, 11, v_j)$, $(v_i, 6, 9, 8, 7, v_j)$.

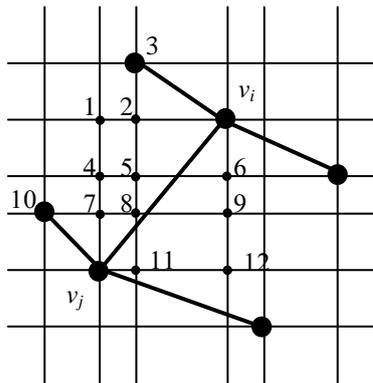


Рис. 1. Формирование вариантов маршрутов

Задача построения МДШ сводится к задаче выбора $(n-1)$ -го s -маршрута, причем каждый маршрут s_{lk} соответствует ребру $D_l=(p_i, p_j)$.

При использовании комбинаторного подхода к построению МДШ изначально для каждого ребра D_l строится набор $S_l=\Gamma(R_k)$ альтернативных вариантов s -маршрутов $S_k=\{s_{kz} \mid z=1, 2, \dots, m\}$.

III. ПАРАДИГМА МЕТОДА КРИСТАЛЛИЗАЦИИ РОССЫПИ АЛЬТЕРНАТИВ

В методе кристаллизации россыпи альтернатив (**Crystallization of alternatives field (CAF)**) [18] каждое решение формируется (представляется) множеством агентов $A=\{a_i \mid i=1, 2, \dots, n_a\}$, где n_a – число агентов. Каждому агенту a_i соответствует множество альтернативных состояний $S_i=\{s_{ij} \mid j=1, 2, \dots, n_{si}\}$, где n_{si} – число состояний агента a_i . Каждый агент a_i может находиться в одном из альтернативных состояний. Решение R_k определяется совокупностью альтернативных состояний множества агентов. Обозначим как s^k_i альтернативное состояние агента a_i в решении R_k . Тогда $R_k=\{s^k_i \mid i=1, 2, \dots, n_a\}$.

Примером такой задачи может служить задача построения дерева Штейнера [1-4].

На втором этапе построения МДШ в каждом наборе S_l выбирается вариант маршрута, обеспечивающий построение МДШ.

В рассмотренной задаче агентами являются ребра, а альтернативами – варианты маршрутов.

Обозначим оценку решения R_k , как f_k . Под *россыпью альтернатив* (РА) решения в работе называется структура данных, несущая информацию об альтернативах агентов в данном решении и об оценке этого решения. Алгоритм оперирует с множествами решений и реализует эволюционную стратегию случайного направленного поиска решения. В процессе поиска множество *оценки решений* трансформируется в *интегральные оценки альтернатив*. На каждом шаге в соответствии с интегральными оценками альтернатив производится генерация новых решений и пересчет интегральных оценок. При этом происходит рост оценок лучших альтернатив и снижение оценок худших альтернатив. Происходит процесс, аналогичный кристаллизации. Лучшие альтернативы, обеспечивающие лучшие решения, как бы выкристаллизовываются (вычлняются) в процессе эволюционного поиска.

Представим решение R_k в виде совокупности векторов $X_k=\{X_{ik} \mid i=1, 2, \dots, n_a\}$, которую назовем россыпью альтернатив (РА). Каждый вектор $X_{ik}=\{x_{ijk} \mid j=1, 2, \dots, n_{si}\}$ соответствует агенту a_i . Размерность вектора X_{ik} определяется числом возможных состояний агента a_i . В векторе X_{ik} только один элемент x_{ijk} , соответствующий состоянию s^k_i , в котором находится агент a_i , имеет значение, отличное от нуля, и это значение равно оценке f_k этого решения. Остальные элементы вектора X_{ik} имеют нулевые значения.

Таким образом, в векторе X_{ik} хранится информация о состоянии, реализованном агентом a_i в решении R_k , и об оценке f_k этого решения.

Пример. Решение R_k формируется четырьмя агентами. Агенты имеют 5 возможных состояний. В решении R_k агентами реализованы следующие состояния: a_1-2 , a_2-5 , a_3-4 , a_4-1 . Значение целевой функции $f_k=7$. Тогда россыпь альтернатив для решения R_k имеет вид, представленный на рис. 2.

X_{1k}	0	7	0	0	0
X_{2k}	0	0	0	0	7
X_{3k}	0	0	0	7	0
X_{4k}	7	0	0	0	0

Рис. 2. Россыпь альтернатив решения R_k

Отметим, что в случае равенства числа вариантов у всех альтернатив россыпь альтернатив удобно представлять в виде матрицы, строка которой соответствует альтернативе, а столбец номеру варианта.

Алгоритм построения дерева Штейнера методом кристаллизации россыпи альтернатив.

0. Задано множество точек $P=\{p_i|i=1, 2, \dots, n\}$. На множестве точек P формируется ортогональная сеть.

1. На множестве точек P строится минимальное связывающее дерево (МСД) $D=\{D_l|l=1, 2, \dots, n-1\}$.

2. Для каждого ребра D_l формируются наборы $S_l=\Gamma(D_l)$ альтернативных вариантов маршрутов. $S_l=\{s_{lk}|k=1, 2, \dots, m\}$. $S=\{S_l|l=1, 2, \dots, n-1\}$ - множество наборов.

3. Генерация начального множества решений $|R|=n_0$ путем выбора агентами в каждом S_l случайным образом альтернативы s_{lk} . Расчет оценок для всех решений множества R .

4. Сужение сгенерированного множества решений R до заданного размера $RI=\{R_k|k=1, 2, \dots, n_k\}$ путем отбрасывания худших решений. Определение у сформированного множества решений RI решения с лучшим значением оценки $f^\#$ и решения с худшим значением - f^\ominus . Формирование россыпи альтернатив X_k для каждого решения множества RI .

5. Формирование интегральной россыпи альтернатив X^*I путем объединения всех россыпей альтернатив X_k .

5.1. Пусть агентом a_i альтернатива s_{ij} была выбрана в n_{ij} решениях множества RI .

Если $n_{ij} \neq 0$, то элементы x^*_{ij} интегральной россыпи альтернатив $X^*=\{X^*_i|i=1,2,\dots,n_a\}$, $X^*_i=\{x^*_{ij}|j=1,2,\dots,n_{si}\}$ принимают значения, вычисляемые по формуле

$$x^*_{ij} = \gamma \cdot (\sum_k x_{ijk}) / n_{ij}, \quad (1)$$

где γ - управляющий параметр, который подбирается экспериментально.

Фактически x^*_{ij} является средним значением оценок решений, в которых агентом a_i была реализована альтернатива s_{ij} .

5.2. Если $n_{ij}=0$, то производится коррекция (дополнение) интегральной россыпи альтернатив. (Обоснование необходимости коррекции будет приведено ниже). Выбирается параметр Q , лежащий в границах $f^\ominus \leq Q \leq f^\#$. Сущность коррекции заключается в том, что всем элементам x^*_{ij} интегральной россыпи альтернатив соответствующим альтернативам s_{ij} с нулевым значением n_{ij} присваивается значение $\gamma \cdot Q$.

Пример. Произведем интеграцию решений и построим интегральную россыпь альтернатив множества решений $RI = \{R_1, R_2, R_3\}$. Агентами в каждом из решений реализованы следующие альтернативы. $R_1: a_1-2, a_2-5, a_3-4, a_4-1$. $R_2: a_1-4, a_2-3, a_3-2, a_4-1$. $R_3: a_1-2, a_2-5, a_3-4, a_4-3$. $f_1=7, f_2=6, f_3=11$.

Россыпи альтернатив множества решений $\{R_1, R_2, R_3\}$ представлены на рисунке 3. Лучшее значение оценки решения $f^\#=11$, худшее значение $f^\ominus=6$.

Выбирается параметр Q , лежащий в границах $f^\ominus \leq Q \leq f^\#$. Пусть $Q=8$.

Произведем расчет значений элементов x^*_{ij} интегральной россыпи альтернатив по формуле 1. $n_{11}=0, x^*_{11}=0$. $n_{12}=2, x^*_{12}=(7+0+11)/2=9$. $n_{13}=0, x^*_{13}=0$. $n_{14}=1, x^*_{14}=6$. $n_{15}=0, x^*_{15}=0$.

X_{11}	0	7	0	0	0	X_{12}	0	0	0	6	0
X_{21}	0	0	0	0	7	X_{22}	0	0	6	0	0
X_{31}	0	0	0	7	0	X_{32}	0	6	0	0	0
X_{41}	7	0	0	0	0	X_{42}	6	0	0	0	0
						X_{13}	0	11	0	0	0
						X_{23}	0	0	0	0	11
						X_{33}	0	0	0	11	0
						X_{43}	0	0	11	0	0

Рис. 3. Россыпи альтернатив решений $\{R_1, R_2, R_3\}$

Интегральная россыпь альтернатив представлена на рис. 4.

Производим коррекцию интегральной россыпи альтернатив. Всем элементам с нулевым значением присваиваем значение $Q=8$. (Рис. 5).

Первые пять пунктов составляют подготовительный этап работы алгоритма

Начиная с пункта 6, выполняется итерационная процедура эволюционного поиска решения.

X^*_{1k}	0	9	0	6	0
X^*_{2k}	0	0	6	0	9
X^*_{3k}	0	6	0	9	0
X^*_{4k}	6,5	0	11	0	0

Рис. 4. Интегральная россыпь альтернатив

6. Формирование распределения вероятностей выбора альтернатив агентами. Сущность этой операции заключается в том, что всем элементам x^*_{ij} интегральной россыпи альтернатив ставится в соответствие значение вероятности p_{ij} выбора агентом a_i состояния s_{ij} . Расчет вероятностей осуществляется по формуле

$$p_{ij} = x^*_{ij} / (\sum_j x^*_{ij}). \quad (2)$$

Рассмотрим расчет вероятностей альтернатив для первого агента

$$p_{11}=p_{13}=p_{15}=8/(8+9+8+6+8)=8/39, p_{12}=9/39, p_{14}=6/39.$$

После расчета вероятностей альтернатив для остальных агентов нашего примера интегральная россыпь вероятностей альтернатив X^p примет вид. (рис. 6).

Каждый вектор X^p_i является распределением вероятности выбора альтернативы агентом a_i .

X^*_{1k}	8	9	8	6	8
X^*_{2k}	8	8	6	8	9
X^*_{3k}	8	6	8	9	8
X^*_{4k}	6,5	8	11	8	8

Рис. 5. ИРА после коррекции

Вернемся к пояснению действий, производимых в пункте 5.2. После построения начальной интегральной россыпи альтернатив отдельные элементы могут иметь нулевые значения. Это значит, что соответствующие им вероятности, рассчитываемые по формуле (2), будут иметь нулевые значения и соответствующие альтернативы будут исключены из рассмотрения в самом начале процесса поиска. Чтобы не допустить исключения альтернатив из рассмотрения производится коррекция начальной интегральной россыпи альтернатив, т.е. всем элементам с нулевым значением присваиваем значение Q , лежащее в границах $f^0 \leq Q \leq f^{\#}$.

X^p_1	8/39	9/39	8/39	6/39	8/39
X^p_2	8/39	8/39	6/39	8/39	9/39
X^p_3	8/39	6/39	8/39	9/39	8/39
X^p_4	6,5/41,5	8/41,5	11/41,5	8/41,5	8/41,5

Рис. 6. Интегральная россыпь вероятностей альтернатив

7. На базе интегральной россыпи вероятностей альтернатив X^p формируется множество решений $R2$. Агентами выбираются альтернативы случайным образом, но в соответствии с распределениями вероятностей, задаваемыми россыпью вероятностей альтернатив X^p . Будем называть множество решений $R2$ популяцией.

8. Сужение сгенерированного множества решений до заданного размера путем отбрасывания худших решений. Выбор лучшего решения среди множеств $R1$ и $R2$.

Если выполнено заданное число итераций, то переход к пункту 12.

9. Формирование россыпи альтернатив X_k для каждого решения множества $R2$.

Формирование интегральной россыпи альтернатив X^*2 путем объединения всех россыпей альтернатив, соответствующих множеству решений $R2$.

10. Объединение интегральной россыпи альтернатив X^*1 с интегральной россыпью альтернатив X^*2 . $X^*1 = X^*1 \cup X^*2$. Объединение производится по следующему правилу

$$(x^*_{ij})I = ((x^*_{ij})I + (x^*_{ij})2)/2.$$

Таким образом, формируется среднее значение параметра x^*_{ij} .

11. Производится уменьшение значений элементов интегральной россыпи альтернатив по формуле

$$x^*_{ij} = \rho \cdot x^*_{ij},$$

где ρ – коэффициент обновления (0.93–0.99).

Этот пункт (прием) выполняется исходя из следующих соображений.

Поскольку худшие альтернативы выбираются реже и их оценки меньше лучших, то интегральные оценки лучших альтернатив растут быстрее худших. Периодическое уменьшение значений элементов интегральной россыпи альтернатив приводит к ускоренному снижению оценок худших альтернатив, фактическому обнулению худших интегральных оценок и, следовательно, к уменьшению вероятности выбора соответствующих им альтернатив.

Переход к пункту 6.

12. Завершение работы алгоритма. Фиксация и вывод лучшего решения.

IV. МОДИФИКАЦИИ АЛГОРИТМА

Платформой для организации эволюционной процедуры поиска является интегральная россыпь альтернатив (ИРА) на базе сгенерированного множества решений $R1 = \{R_k / k=1, 2, \dots, n_k\}$, где $R_k = \{s^k_i / i=1, \dots, n_a\}$. Поэтому нет необходимости в построении для каждого решения индивидуальной россыпи альтернатив. В работе индивидуальная россыпь альтернатив решения используется как виртуальное описание для изложения сущности предложенного метода оптимизации. Процесс генерации каждого решения заключается в генерации вектора $R_k = \{s^k_i / i=1, 2, \dots, n_a\}$ и расчете оценки f_k решения. А затем на базе этой информации строится интегральная россыпь альтернатив для множества решений.

Для усиления различия вероятностей выбора альтернатив предлагается модифицированная формула расчета вероятностей. В интегральной россыпи альтернатив отыскивается элемент с минимальным значением $(x^*_{ij})_{min}$. Выбирается параметр $q < (x^*_{ij})_{min}$. Модифицированная формула имеет вид

$$p_{ij} = (x^*_{ij} - q) / (\sum_j (x^*_{ij} - q)).$$

Рассмотренный алгоритм в полной мере применим для пересекających множеств альтернатив агентов, то есть

$$(\forall i, t) [S_i \cap S_t = \emptyset, S_i \neq \emptyset, S_t \neq \emptyset].$$

К задачам подобного вида можно отнести задачи построения дерева Штейнера на основе комбинаторного подхода [14], трассировки [13], разнесения соединений по слоям [12], планирования кристалла СБИС [14] и др.

Существует довольно обширный класс задач, у которых $(\exists i, t) [S_i \cap S_t \neq \emptyset]$. Частным является случай, когда все агенты имеют один и тот же набор альтернатив $(\forall i, t) [S_i = S_t]$. При этом возможны три случая. Первый: любая альтернатива может быть выбрана только одним агентом. Например, задача размещения (в общем случае задача о назначениях). Второй: для каждой альтернативы задается число агентов, которые могут ее реализовать. Например, задача разбиения. Третий: пусть множество агентов разбито на подмножества; в пределах каждого подмножества агентов альтернатива может быть реализована только одним из агентов.

Для такого рода задач при формировании решения используется модификация рассмотренного алгоритма, суть которой заключается в том, что при последовательном выборе агентами альтернатив каждый последующий агент учитывает результаты выбора альтернатив предыдущими агентами.

Пусть при формировании решения, заключающегося в выборе агентами альтернатив, множеством агентов A_1 альтернативы выбраны, а множеством агентов A_2 - нет. Выбор альтернатив агентами осуществляется случайным образом на основе интегральной россыпи вероятностей альтернатив. Пусть осуществляется выбор альтернативы агентом $a_i \in A_2$. При этом оказывается, что некоторое множество альтернатив S_j , выбранное агентами A_1 , не может быть использовано агентом a_i . В этом случае эти альтернативы исключаются из рассмотрения агентом a_i . Пусть S_{2i} множество альтернатив, доступных агенту a_i . Пусть PI_i сумма вероятностей выбора альтернатив множества S_{2i} агентом a_i , хранящиеся в интегральной россыпи вероятностей альтернатив X^p . Поскольку ряд альтернатив был исключен, то эта сумма будет меньше единицы. В связи с этим значения вероятностей выбора агентом a_i доступных альтернатив множества S_{2i} нормируется так, чтобы их сумма была равна единице. Пересчет осуществляется по формуле

$$p_{ij} = p_{ij} / PI_i. \quad (3)$$

V. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

Алгоритм построения минимального дерева Штейнера реализован на языке C++ в среде Windows. Экспериментальные исследования проводились на ЭВМ типа IBM PC. Проведение экспериментов преследовало две цели: исследование эффективности и

качества механизмов кристаллизации россыпи альтернатив для решения задачи построения минимального дерева Штейнера. Для этих целей была использована процедура синтеза **контрольных примеров с известным оптимумом**.

Первой целью являлось исследование влияния управляющих операторов, таких как: размер начального множества решений $|R|=n_0$, количество итераций, сужение сгенерированного множества решений Ω и т.д. Исследованию подвергались примеры, содержащие до 1000 вершин.

Временная сложность алгоритма имеет оценку $O(n^2)$, где n - число связываемых вершин. Эксперименты показали, что в 95% случаев пространство решений на базе сформированного набора вариантов ребер включает глобальное оптимальное решение. При исследовании сходимости алгоритма для каждого эксперимента запоминался номер генерации, после которой не наблюдалось улучшения оценки. В каждой серии из 50 испытаний определялись минимальный и максимальный номера генерации. Кроме этого рассчитывалось среднее значение числа генераций, после которого не наблюдалось улучшения оценки. Для каждой серии испытаний определялось лучшее решение, которое фактически являлось оптимальным. В результате экспериментов установлено, что при формировании на каждой итерации популяции решений R_2 в объеме 80 алгоритм сходится в среднем на 120 итерации. При этом отклонения в сторону увеличения этой оценки составляли до 10%, а в сторону уменьшения - до 35%.

Сравнение значений критерия, полученных алгоритмом кристаллизации россыпи альтернатив дерева на тестовых примерах, с известным оптимумом показало, что у 80% примеров полученное решение было оптимальным, у 15% примеров решения были на 3% хуже оптимального, а у 5% примеров решения были хуже не более, чем на 2%.

Для сравнения качества решений задачи построения применялись стандартные тесты для оценки разработанных алгоритмов. В качестве аналога для сравнения выбран алгоритм WRST [4], основанный на последовательном построении взвешенных деревьев Штейнера. В таблице 2 представлены результаты сравнения экспериментальных данных с аналогом для пяти примеров. В колонке $\sum L$ приводится суммарная длина, а в колонке t - время работы алгоритма.

Как видно из таблицы, разработанный алгоритм построения минимального дерева Штейнера обеспечивает более качественное решение, причем время решения на 10-40% меньше.

Сравнение результатов

№	ΣL		t (сек)	
	KPA	WRST	KPA	WRST
1	4212	4570	33	37
2	3870	4610	31	34
3	5180	5900	57	69
4	4256	4570	59	61
5	3650	4220	29	34

VI. ЗАКЛЮЧЕНИЕ

Разработан новый алгоритм построения дерева Штейнера, базирующийся на парадигме мультиагентного метода интеллектуальной оптимизации. Рассмотрены ключевые моменты анализа альтернатив в процессе эволюционной коллективной адаптации, названной по аналогии с процессами вычленения объектов (формирования кристаллов) кристаллизацией.

Представленный алгоритм находит решения, превосходящие результаты существующих алгоритмов. Экспериментальная временная сложность алгоритма составляет $O(n^2)$. При больших размерностях временные показатели разработанного алгоритма превосходят показатели сравниваемых алгоритмов при лучших значениях целевой функции.

Важным этапом синтеза топологии субмикронных СБИС является глобальная трассировка. На этом этапе решается задача планировки цепей, каждая цепь описывается соответствующим ей деревом Штейнера. Основными критериями глобальной трассировки являются рациональное использование ресурсов при распределении проводников на кристалле и площадь кристалла СБИС. Рассмотренный подход к построению дерева Штейнера безусловно является базовым для глобальной трассировки и заключается в предварительном синтезе альтернативных вариантов двухтерминальных соединений на основе анализа структуры коммутационного поля и цепей, и последующем выборе совокупности вариантов, обеспечивающем рациональное распределение ресурсов коммутационного поля.

С ними тесно связаны критерии задержки прохождения сигнала, перекоса сигнала. Дополнительными критериями могут быть: число поворотов проводников, надежность и выход годных. Перспективными путями улучшения алгоритмов построения кратчайших связывающих сетей являются исследования принципов формирования базового множества альтернатив, ориентированного на оптимизацию рассмотренных выше критериев.

- [1] Лебедев Б.К. Интеллектуальные процедуры синтеза топологии СБИС. - Таганрог: Изд-во ТРТУ, 2003.
- [2] Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поискковая адаптация: Теория и практика. М.: Физматлит, 2006.
- [3] Andrew B. Kahng, Gabriel Robins "On optimal interconnection for VLSI" Kluwer 1995.
- [4] C. Chiang, M.Sarrafraden, C.K.Wong. A Weighted - Steiner - Tree - Based Global Router, Manuscript, 1992.
- [5] M. Clerc. Particle Swarm Optimization. ISTE, London, UK, 2006.
- [6] О. Лебедев, В. Курейчик, Б. Лебедев. Адаптация в задачах проектирования топологии. LAP LAMBERT Academic Publishing GmbH@Co/KG. 2012.
- [7] R. Poli. Analysis of the publications on the applications of particle swarm optimisation. Journal of Artificial Evolution and Applications, Article ID 685175, 10 pages, 2008.
- [8] Lučić P., Teodorović D. Computing with Bees: Attacking Complex Transportation Engineering Problems // International Journal on Artificial Intelligence Tools. – 2003. – №12. – P. 375–394.
- [9] M. Dorigo and T. Stützle. Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.
- [10] Курейчик В.М., Лебедев Б.К., Лебедев В.Б. Адаптация в задачах проектирования топологии // Проблемы разработки перспективных микро- и нанoeлектронных систем – 2010 //Сборник научных трудов / под ред. А.Л. Стемповского. – М.: ИПИМ РАН, 2010. С.170-177.
- [11] Лебедев Б.К., Лебедев О.Б. Моделирование адаптивного поведения муравьиной колонии при поиске решений, интерпретируемых деревьями // Известия ЮФУ. Изд-во ТТИ ЮФУ, 2012, №7. С. 27-35.
- [12] Лебедев О.Б. Покрытие методом муравьиной колонии // Двенадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2010. Труды конференции. Т. 2. М.: Физматлит, 2010. С. 423-431.
- [13] Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Гибридный алгоритм разбиения на основе природных механизмов принятия решений //Искусственный интеллект и принятие решений. – Москва: Изд-во Институт системного анализа РАН, 2012. С. 3-15.
- [14] Лебедев О.Б. Трассировка в канале методом муравьиной колонии // Известия ЮФУ. Изд-во ТТИ ЮФУ, 2009, №2. С. 46-52.
- [15] Лебедев О.Б. Построение дерева Штейнера на основе метода муравьиной колонии //Труды конгресса по интеллектуальным системам и информационным технологиям «AIS– IT'09». Научное издание в 4-х томах. – М.: Физматлит, 2009, Т.1. – С. 58-65.
- [16] Лебедев В.Б. Построение кратчайших связывающих сетей на основе роевого интеллекта // Известия ЮФУ. Изд-во ТТИ ЮФУ, 2011, №7. С. 37-44.
- [17] 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.
- [18] Лебедев Б.К., Лебедев В.Б. Оптимизация методом кристаллизации россыпи альтернатив //Известия ЮФУ. Изд-во ТТИ ЮФУ, 2013, №7. С. 11-17.