

УДК 681.3.001.63

**В.Б. Лебедев**

### **ПОСТРОЕНИЕ КРАТЧАЙШИХ СВЯЗЫВАЮЩИХ СЕТЕЙ НА ОСНОВЕ РОЕВОГО ИНТЕЛЛЕКТА\***

*Излагается метод построения дерева Штейнера на основе роевого интеллекта и генетической эволюции. Связующим звеном такого подхода является структура данных, описывающая в виде хромосомы решение задачи. Это позволило организовать пространство решений, в рамках которого организован поисковый процесс, базирующийся на моделировании адаптивного поведения роя частиц. В отличие от канонического метода роя частиц, в работе в качестве скорости  $v_i(t+1)$  выступает процедура перемещения в новую позицию, суть которой заключается в изменении значений элементов в векторе  $H_i(t)$ , соответствующем позиции  $x_i(t)$ . С этой целью на базе расширенной метрики Хемминга разработаны методы измерения расстояния между позициями в пространстве решений. По сравнению с существующими алгоритмами достигнуто улучшение результатов.*

*Роевой интеллект; генетическая эволюция; гибридизация; оптимизация; дерево Штейнера*

**V.B. Lebedev**

### **CONSTRUCTION OF THE SHORTEST CONNECTING NETWORKS ON THE BASIS OF SWARM INTELLIGENCE**

*In work the method of the decision of a steiner tree problem on a basis swarm intelligence and genetic evolution is stated. A link of such approach is the structure of data describing in the form of a chromosome the decision of a problem. It has allowed to form the space of decisions in which frameworks the search process which is based on modelling of adaptive behaviour of a swarm of particles is organized. Unlike an initial method of particles swarm, in work as speed  $v_i(t+1)$  procedure of moving to the new position which essence consists in changes of values of elements in vector  $H_i(t)$ , corresponding to a position  $x_i(t)$  acts. With that aim on the basis of expanded Hamming metrics are developed methods of distance measurement between positions in space of decisions.*

*In comparison with existing algorithms improvement of results is reached.*

*Swarm intelligence; genetic evolution; hybridization; optimization; steiner tree.*

**Введение.** Алгоритмы построения дерева Штейнера разбиты на два класса: последовательный и комбинаторный [1].

При последовательном подходе цепи распределяются по областям последовательно. В основе большинства из них лежит волновой алгоритм Ли и его модификации [1,2]. 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. Цель задачи заключается в нахождении на заданном наборе таких вариантов, которые обеспечивают наилучшее решение. Большинство алгоритмов [1,2] используют традиционные итерационные улучшающие структуры, основанные на слепом случайном поиске. Основным недостатком, присущим этому подходу, является вхождение алгорит-

\* Работа выполнена при частичной поддержке РФФИ (проекты: № 09-01-00509, № 10-01-00115, № 10-07-00055).

мов в локальный оптимум, часто далекий от глобального оптимума. В последнее время для решения различных «сложных» задач, к которым относятся и задачи построения дерева Штейнера, всё чаще используются способы, основанные на применении методов искусственного интеллекта. Особенно наблюдается стремительный рост интереса к разработке алгоритмов, инспирированных природными системами [3–5]. Одним из новых направлений таких методов являются мультиагентные методы интеллектуальной оптимизации, базирующиеся на моделировании коллективного интеллекта [5–7]. Оптимизация с использованием роя частиц (Particle Swarm Optimization, PSO) – это метод поиска, который базируется на понятии популяции, и моделирует поведение птиц в стае и косяков рыб [7–9]. Рой частиц рассматривается как многоагентная система, в которой каждый агент (частица) функционирует автономно по очень простым правилам. В таких случаях говорят о роевом интеллекте (Swarm intelligence).

В работе излагается метод построения дерева Штейнера на основе интеграции роевого интеллекта [6] и генетической эволюции [10,11]. С этой целью разработана единая для обоих подходов структура данных. С учетом особенностей единой структуры данных разработаны модифицированные механизмы поведения роя частиц и структура пространства решений, в рамках которого организован поисковый процесс, базирующийся на моделировании адаптивного поведения роя частиц.

**Формулировка проблемы Штейнера.** 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  $Q=XP$ .

Проведем через множество точек  $P=\{p_i|i=1,2,\dots,n\}$ , расположенных на плоскости горизонтальные и вертикальные линии. Образуется ортогональная сетка. Обозначим через  $V$  множество точек пересечений линий сетки и назовем их узлами сетки. Сопоставим ортогональной сетке граф  $G=(V,E)$ . Ряд теорем [1,6] 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 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 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 решается в два этапа. На первом этапе формируется набор альтернативных вариантов  $s$ -маршрутов. На втором этапе из сформированного набора выбирается  $(n-1)$   $s$ -маршрутов, покрывающих МДШ.

Рассмотрим процесс формирования альтернативных вариантов  $s$ -маршрутов [9]. 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).

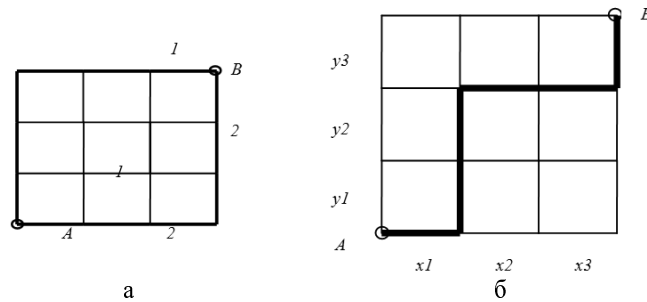


Рис. 1. Способы формирования маршрутов на опорной сетке

Для каждого ребра  $R_k$  на графе  $G$  строится набор  $S_k = \Gamma(R_k)$  альтернативных вариантов  $s$ -маршрутов, 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.

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

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

**Общая структура представления решений в алгоритмах построения дерева Штейнера на основе роевого интеллекта и генетического поиска.** В эвристических алгоритмах роевого интеллекта многомерное пространство поиска населается роем частиц [9]. Каждая частица представляет некоторое решение. В нашем случае решение задачи глобальной трассировки. Процесс поиска решений заключается в последовательном перемещении частиц в пространстве поиска. Обозначим позицию частицы  $i$  в пространстве решений в момент времени  $t$  ( $t$  имеет дискретные значения) как  $x_i(t)$ .

По аналогии с эволюционными стратегиями, рой можно трактовать как популяцию, а частицу как индивида (хромосому). Это дает возможность построения гибридной структуры поиска решения, основанную на сочетании генетического поиска с методами роевого интеллекта. Связующим звеном такого подхода является структура данных, описывающая в виде хромосомы решение задачи. Если в качестве частицы используется хромосома, то число параметров, определяющих положение частицы в пространстве решений должно быть равно числу генов в хромосоме. Значение каждого гена откладывается на соответствующей оси пространства решений. В этом случае возникают некоторые требования к структуре хромосомы и значениям генов.

Значения генов должны быть независимыми друг от друга, то есть хромосомы должны быть гомологичными. В работе предлагается подход к построению структур и принципов кодирования хромосом, обеспечивающих их гомологичность и возможность одновременного использования в генетическом алгоритме, и в алгоритме на основе роя частиц.

Каждое двухтерминальное соединение представляет собой маршрут, состоящий из горизонтальных и вертикальных отрезков. В работе рассматриваются два подхода к формированию маршрутов. При первом подходе конфигурация маршрутов формируется априори. В простейшем случае формируются два варианта. На рисунке (см. рис. 1,а) показаны два варианта маршрута минимальной длины, связывающего вершины  $A$  и  $B$ .

В работе [6] каждое двухтерминальное соединение представляет собой маршрут, состоящий из горизонтальных и вертикальных отрезков. Обозначим через  $X = \{x_i / i = 1, 2, \dots, n_x\}$  и  $Y = \{y_j / j = 1, 2, \dots, n_y\}$  соответственно базовые множества горизонтальных и вертикальных отрезков, составляющих некоторое d-соединение (маршрут). На рисунке (см. рис. 1,б) любой маршрут  $M$  на опорной сетке, связывающий точки  $A$  и  $B$ , состоит из трех вертикальных и трех горизонтальных отрезков:  $X = \{x_1, x_2, x_3\}$ ,  $Y = \{y_1, y_2, y_3\}$ . Отметим, что в любом маршруте число и размеры отрезков не изменяются.

Различные маршруты отличаются различными вариантами чередования горизонтальных и вертикальных отрезков, причем отрезки одного вида (горизонтальные или вертикальные) имеют строго фиксированное взаимное расположение.

Представленный на рисунке (см. рис. 1,б) маршрут описывается следующим списком отрезков:  $M = \langle x_1, y_1, y_2, x_2, x_3, y_3 \rangle$ . Поставим в соответствие горизонтальному отрезку ноль, а вертикальному – единицу. Тогда маршрут  $M$  (рис. 1) в модифицированном виде имеет вид  $M^* = \langle 0, 1, 1, 0, 0, 1 \rangle$ . Число нулей равно  $n_x$ , а число единиц равно  $n_y$ .

Для обратного перехода необходимо для каждого элемента списка  $M^*$  определить порядковый номер  $l$  в списке среди элементов своего вида. Тогда, если это был ноль, то ему соответствует  $x_l$ , а если это была единица, то ей соответствует  $y_l$ .

На основе выше изложенного хромосома, описывающая дерево Штейнера, имеет следующую структуру:  $H = \{h_i / i = 1, 2, \dots, n_i\}$ .  $H$  представляет собой объединение множества частей  $h_i$ , число которых равно числу ребер дерева  $R$ . Каждая часть  $h_i$  соответствует ребру  $R_i$  и несет информацию о проложенном  $s$ -маршруте на графе  $G = (V, E)$ . В свою очередь,  $h_i$  состоит из генов  $g_{ik}$ . При первом подходе каждая часть  $h_i$  состоит из одного гена  $g_{il}$ , который несет информацию о варианте двухтерминального соединения (маршрута), реализующего ребро  $R_i$ . Если  $g_{il} = 1$ , то выбран первый вариант маршрута, если  $g_{il} = 0$ , то выбран второй вариант маршрута.

При втором подходе каждая часть  $h_i$  соответствует списку отрезков  $M_i$ , который несет информацию о распределении  $s$ -маршрута по областям.

Будем использовать множество нулей в качестве опорного множества для кодирования списка  $M^*$ . Сформируем на базе опорного множества нулей множество позиций  $P$  (рис. 2), в которых могут располагаться единицы, образующие совместно с нулями список  $M^*$  некоторого маршрута  $M$ . Если число нулей  $n_x$ , то число позиций  $P$  равно  $n_x + 1$ . Каждая из  $n_y$  единиц может быть расположена в любой из позиций  $p_i \in P = \{p_i / i = 1, 2, \dots, (n_x + 1)\}$ .

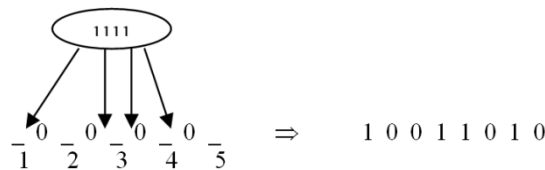


Рис. 2. Способ формирования маршрутов на опорной сетке

Всевозможным вариантам размещения единиц в позициях  $P$  совместно с опорным множеством нулей соответствуют всевозможные списки  $M^*$ , а следовательно, и всевозможные маршруты  $M$ , соединяющие две точки.

Распределение единиц по позициям будем задавать с помощью списка  $Q$ , размер которого равен  $n_y$ , а значением элемента списка является число, лежащее в диапазоне  $1 \div (n_x + 1)$ . Таким образом,  $Q$  является кодом маршрута, соединяющего две точки на опорной сетке.

Например, пусть  $n_y = 4$ ,  $n_x = 4$ ,  $Q = 5, 3, 3, 1$ .

Тогда (см. рис. 2):

$M^* = \langle 1, 0, 0, 1, 1, 0, 0, 1 \rangle$ , а  $M = \langle y_1, x_1, x_2, y_2, y_3, x_3, x_4, y_4 \rangle$ .

В общем случае хромосома имеет вид:  $H = \{g_{ik} | i=1, 2, \dots, n_i; k=1, 2, \dots, n_k\}$ .

При первом подходе максимальное значение гена  $g_{il}$  равно числу вариантов  $s$ -маршрутов, сформированных априори для ребра  $R_i$ .

При втором подходе каждая часть  $h_i$  состоит из генов  $g_{ik}$ , образующих код  $Q_i$  маршрута, соединяющего две точки на опорной сетке. Значение гена  $g_{ik}$  для фиксированного  $i$  лежит в диапазоне  $1 \div (n_x + 1)$ , где  $n_x$  – число нулей в списке  $M_i^*$ .

**Механизмы построения дерева Штейнера на основе роевого интеллекта.** В методе оптимизации роем частиц агентами являются частицы в пространстве параметров задачи оптимизации. Основу поведения роя частиц составляет самоорганизация, обеспечивающая достижение общих целей роя на основе низкоуровневого взаимодействия. Каждая частица связана со всем роем, может взаимодействовать со всем роем и она тяготеет к лучшему решению роя. Процесс поиска решений итерационный. На каждой итерации каждая частица перемещается в новую позицию. В каждый момент времени (на каждой итерации) частицы имеют в этом пространстве некоторое положение и вектор скорости. Для каждого положения частицы вычисляется соответствующее значение целевой функции. При определении следующего положения частицы учитывается информация о наилучшей частице из числа «соседей» данной частицы, а также информация о данной частице на той итерации, когда этой частице соответствовало наилучшее значение целевой функции, и на этой основе по определенным правилам частица меняет свое положение и скорость в пространстве поиска [8].

В каноническом методе роя частиц [9], использующем непрерывные метрические шкалы, новая позиция частицы  $i$  определяется как

$$x_i(t+1) = x_i(t) + v_i(t+1),$$

где  $v_i(t+1)$  скорость перемещения частицы  $i$  из позиции  $x_i(t)$  в позицию  $x_i(t+1)$ . Начальное состояние определяется, как  $x_i(0)$ ,  $v_i(0)$ . Приведенная формула представлена в векторной форме. Для отдельного измерения  $k$  пространства поиска формула примет вид

$$x_{ik}(t+1) = x_{ik}(t) + v_{ik}(t+1), \quad (3)$$

где  $x_{ik}(t)$  – позиция частицы  $i$  в измерении  $k$ ,  $v_{ik}(t+1)$  – скорость частицы  $i$  в измерении  $k$ .

Введем обозначения:

- ◆  $f_i(t)$  – текущее значение целевой функции частицы  $i$  в позиции  $x_i(t)$ ;
- ◆  $f^*(t)$  – лучшее значение целевой функции среди частиц роя в момент времени  $t$ , а  $x^*(t)$  – позиция с этим значением;
- ◆  $x_i^*(t)$  – лучшая позиция частицы  $i$ , которую она посещала с начала первой итерации, а  $f_i^*(t)$  – значение целевой функции в этой позиции – лучшее значение частицы  $i$ .

Лучшие частицы с точки зрения целевой функции объявляются «центром притяжения». Векторы скоростей всех частиц устремляются к этим центрам.

В работе в отличие от стандартной парадигмы метода роя частиц вместо непрерывной метрической (числовой) шкалы используется *дискретная*.

Для того, чтобы избежать при описании популяции (роя) путаницы будем в дальнейшем каждую хромосому, описывающую  $i$ -е решение популяции обозначать как  $H_i(t)$ . Причем, каждая  $H_i(t)$  имеет структуру аналогичную структуре  $R$ . В нашем случае позиция  $x_i(t)$  соответствует  $i$ -му решению, задаваемому хромосомой  $H_i(t)$ , т.е.  $x_i(t) = H_i(t)$ . Аналогично  $x_i^*(t) = H_i^*(t)$ ,  $x^*(t) = H^*(t)$ . Число осей в про-

странстве решений равно числу генов в хромосоме  $H_i(t)$ . Точками отсчета на каждой оси  $k$  являются значения генов.

Скорость  $v_i(t+1)$  рассматривается как средство изменения решения. В отличие от канонического метода роя частиц, в нашем случае скорость  $v_i(t+1)$  не может быть представленной в виде аналитического выражения. В качестве скорости  $v_i(t+1)$  выступает процедура перемещения, суть которой заключается в изменении значения элементов в векторе  $H_i(t)$ . Частица  $i$  перемещается в новую позицию  $H_i(t+1)$  с новыми значениями переменных  $g_{il}$ .

По аналогии с каноническим методом роя частиц позицию  $x_i^*(t)$  будем называть когнитивным центром притяжения, а позицию  $x_i(t)$  – социальным центром притяжения. Когнитивный центр выступает в роли индивидуальной памяти о наиболее оптимальных позициях данной частицы. Благодаря социальному центру частица имеет возможность передвигаться в оптимальные позиции, найденные соседними частицами.

Для учета одновременного тяготения частицы  $i$  к позициям  $x_i(t)$  и  $x_i^*(t)$  формируется центр притяжения  $x_i^c(t)$  этой частицы. Формирование  $x_i^c(t)$  осуществляется путем применения процедуры перемещения к позициям  $x_i(t)$  и  $x_i^*(t)$ . После определения центра притяжения  $x_i^c(t)$  частица  $i$  с помощью процедуры перемещения перемещается в направлении  $x_i^c(t)$  из позиции  $x_i(t)$  в позицию  $x_i(t+1)$ .

При первом подходе к кодированию решения гены имеют только два значения – 0 или 1. Таким образом, решение  $H_i(t)$  можно рассматривать как вектор  $H_i(t) = \{g_{il}(t) | l=1, 2, \dots, n_i\}$  элементы которого имеют только два значения – 0 или 1. Аналогичную структуру имеют решения  $H_i^*(t) = \{g_{il}^*(t) | l=1, 2, \dots, n_i\}$ ,  $H_i^c(t) = \{g_{il}^c(t) | l=1, 2, \dots, n_i\}$ .

При втором подходе каждая часть  $H_i$  состоит из генов  $g_{ik}$ , образующих код  $Q_i$  маршрута, соединяющего две точки на опорной сетке. Значение гена  $g_{ik}$  для фиксированного  $i$  лежит в диапазоне  $1 \div (n_x + 1)$ , где  $n_x$  – число нулей в списке  $M_i^*$ .

В качестве оценки отличия между двумя позициями будем использовать расстояние Хемминга. Расстояние Хемминга между двумя бинарными векторами одинаковой длины – это число несовпадающих бит в этих векторах. Пусть  $s_i(t)$  – расстояние Хемминга между позициями  $x_i(t)$  и  $x_i^*(t)$ ,  $s_i^c(t)$  – расстояние Хемминга между позициями  $x_i(t)$  и  $x_i^c(t)$ . Целью процедуры перемещения является уменьшение расстояния Хемминга (отличия) между позициями. Введем определение лотереи. Простой лотереей  $L(x_1, p, x_2)$  назовем вероятностное событие, имеющее два возможных исхода  $x_1$  и  $x_2$ , вероятности наступления которых обозначим соответственно через  $p$  и  $(1-p)$ . Ожидаемая (или средняя) цена лотереи определяется по формуле  $px_1 + (1-p)x_2$ . Как уже указывалось выше, позиции задаются хромосомами. Позициям  $x_i(t)$ ,  $x_i^*(t)$ ,  $x_i^c(t)$  соответствуют  $H_i(t) = \{g_{il}(t) | l=1, 2, \dots, n_i\}$ ,  $H_i^*(t) = \{g_{il}^*(t) | l=1, 2, \dots, n_i\}$ ,  $H_i^c(t) = \{g_{il}^c(t) | l=1, 2, \dots, n_i\}$ .

Рассмотрим процедуру перемещения. Для формирования  $H_i^c(t)$  (позиции  $x_i^c(t)$ ) последовательно просматриваются (начиная с первого) локусы хромосом  $H_i(t)$  и  $H_i^*(t)$  и сравниваются соответствующие им гены.

Если на шаге  $l$  значения генов равны, т.е.  $g_{il}(t) = g_{il}^*(t)$ , то гену  $g_{il}^c(t)$  присваивается их значение,  $g_{il}^c(t) = g_{il}(t) = g_{il}^*(t)$ .

Если же  $g_{il}(t) \neq g_{il}^*(t)$ , то гену  $g_{il}^c(t)$  присваивается значение, определяемое лотереей  $L$ , т.е.  $g_{il}^c(t) = L(g_{il}(t), p, g_{il}^*(t))$ . Путем изменения параметра  $p$  осуществляется управление степенью приближения  $H_i^c(t)$  к  $H_i(t)$  или  $H_i^*(t)$ . При  $p=0,5$  степень приближения нейтральна. При  $p < 0,5$  позиция  $H_i^c(t)$  ближе к  $H_i(t)$ . При  $p > 0,5$  позиция  $H_i^c(t)$  ближе к  $H_i^*(t)$ .

Аналогичным образом осуществляется перемещение частицы из позиции  $H_i(t)$  в позицию  $H_i(t+1)$  под воздействием притяжения к позиции  $H_i^c(t)$ . Для фор-

мирования  $H_i(t+1)$  (позиции  $x_i(t+1)$ ) последовательно просматриваются (начиная с первого) локусы хромосом  $H(t)$  и  $H^c_i(t)$ , и сравниваются соответствующие им гены.

Если на шаге  $l$  значения генов равны, т.е.  $g_{il}(t)=g^c_{il}(t)$ , то значение гена  $g_{il}(t+1) = g_{il}(t)$ .

Если же  $g_{il}(t)$  не равно  $g^c_{il}(t)$ , то гену  $g_{il}(t+1)$  присваивается значение, определяемое лотереей  $L$ , т.е.  $g_{il}(t+1)=L(g^c_{il}(t), p, g_{il}(t))$ . Параметр  $p$  зависит от расстояния Хемминга  $s^c_i(t)$  между позициями  $H(t)$  и  $H^c_i(t)$ , и определяется следующим образом

$$p=\alpha \cdot s^c_i(t)/n_i,$$

где  $\alpha$  коэффициент,  $n_i$  – длина хромосомы в битах. Таким образом, чем больше расстояния Хемминга  $s^c_i(t)$  между позициями  $H(t)$  и  $H^c_i(t)$ , тем больше вероятность, что значением  $g_{il}(t+1)$  будет значение  $g^c_{il}(t)$ .

При реализации второго подхода для определения отличия между хромосомами (расстояния Хемминга) формируется бинарное представление сравниваемых хромосом. С этой целью в хромосоме  $H(t)$  каждая группа  $h_i$  генов  $g_{il}$ , образующих код  $Q_i$  некоторого маршрута, заменяется на соответствующий коду  $Q_i$  бинарный список  $M^*$ . В результате сравниваемые хромосомы преобразуются в бинарные вектора, между которыми определяется расстояние Хемминга  $s^c_i$  и рассчитывается вероятность изменения значения гена  $p=\alpha \cdot s^c_i(t)/n_i$ , где  $n_i$  – длина бинарного представления сравниваемых хромосом в битах.

Процедура перемещения реализуется следующим образом. Если на шаге  $l$  при просмотре (начиная с первого) локусов хромосом  $H(t)$  и  $H^c_i(t)$  значения генов равны, т.е.  $g_{il}(t)=g^c_{il}(t)$ , то значение гена  $g_{il}(t+1) = g_{il}(t)$ .

Если  $g^c_{il}(t)>g_{il}(t)$ , то с вероятностью  $p$  значение  $g_{il}(t)$  увеличивается на единицу, т.е.  $g_{il}(t+1) = g_{il}(t)+1$ , если новое значение  $g_{il}(t+1)$  не выходит за пределы диапазона возможных значений.

Если  $g^c_{il}(t)<g_{il}(t)$ , то с вероятностью  $p$  значение  $g_{il}(t)$  уменьшается на единицу, т.е.  $g_{il}(t+1) = g_{il}(t)-1$ , если новое значение  $g_{il}(t+1)$  не выходит за пределы диапазона возможных значений.

Роевой алгоритм построения дерева Штейнера включает следующие шаги:

1. В соответствии с постановкой задачи построения дерева Штейнера и исходными данными на множестве вершин  $X_i$  графа  $G$  с помощью алгоритма Прима строится минимальное связывающее дерево  $R=\{R_k|k=1,2,\dots,n-1\}$ . Для каждого ребра  $R_k$  на графе  $G$  строится набор  $S_k=\Gamma(R_k)$  альтернативных вариантов  $s$ -маршрутов, равный  $S_k=\{s_{kz}|z=1,2,\dots,m\}$ .

2. Создается исходная «случайная» популяция частиц  $H=\{H_i(t)|i=1, 2, \dots, n_i\}$ ,  $t=0$ . Для каждой частицы случайным образом задается начальная позиция  $x_i(0)$ .

3. Шаги 4-10 итерационно повторяются заданное число раз.

4. Рассчитывается целевая функция  $f_i(t)$  для каждой частицы.

5. Определяются лучшая позиция роя  $x^*(t)$  на шаге  $t$  и значение целевой функции  $f^*(t)$  в этой позиции.

6. Для каждой частицы определяются лучшая позиция  $x^*_i(t)$ , которую она посещала с начала первой итерации, и значение целевой функции  $f^*_i(t)$  в этой позиции.

7. Для каждой частицы определяется центр притяжения  $x^c_i(t)$  этой частицы.

8. Рассчитываются новые позиции частиц в пространстве решений. С помощью процедуры перемещения каждая частица перемещается из позиции  $x_i(t)$  в позицию  $x_i(t+1)$ .

9.  $t=t+1$ .

10. Запоминается лучшее решение, найденное роем к моменту  $t$ .

**Заключение.** В отличие от канонического метода роя частиц, в работе в качестве скорости  $v_i(t+1)$  выступает процедура перемещения в новую позицию, суть которой заключается в изменении значений элементов в векторе  $H_i(t)$ , соответствующем позиции  $x_i(t)$ . С этой целью на базе расширенной метрики Хэмминга разработаны методы измерения расстояния (отличия) между позициями в пространстве решений. Частица  $i$  перемещается в новую позицию  $x_i(t+1)$  (вектор  $H_i(t+1)$ ) с новыми значениями переменных  $g_{il} \in H_i(t+1)$ .

Экспериментальные исследования проводились на IBM PC. Для анализа точности получаемых решений был синтезирован ряд примеров с априори известным оптимальным значением целевой функции. Исследованию подвергались примеры, содержащие до 1000 вершин. При совместной работе алгоритмов в рамках второго подхода вероятность получения оптимального решения составила 0,9.

Сравнительный анализ с другими алгоритмами трассировки производился на стандартных тестовых примерах и схемах. Представленный роевой алгоритм построения дерева Штейнера в сочетании с генетическим алгоритмом находит решения для задач большой размерности, не уступающие по качеству, а иногда и превосходящие своих аналогов с меньшими временными затратами.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дендобренко Б.П., Малика А.С. Автоматизация проектирования радиоэлектронной аппаратуры. – М.: Высш. шк., 2002.
2. Alpert C.J., Mehta D.P., and Sapatnekar S.S. Handbook of Algorithms for Physical Design Automation. Boston, MA: Auerbach, 2009.
3. G. Di Caro, F. Ducatelle, L. M. Gambardella. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks // European Transactions on Telecommunications. – 2005. – Vol. 16 (5). – С. 443-455.
4. Курейчик В.В., Полупанова Е.Е. Эволюционная оптимизация на основе алгоритма колонии пчел // Известия ЮФУ. Технические науки. – 2009. – № 12 (101). – С. 41-46.
5. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поисковая адаптация: теория и практика. – М.: Физматлит, 2006.
6. Лебедев Б.К. Методы поисковой адаптации в задачах автоматизированного проектирования СБИС: Монография. – Таганрог: Изд-во ТРТУ, 2000.
7. Лебедев О.Б. Трассировка в канале методом муравьиной колонии // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 46-52.
8. Engelbrecht A.P. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.
9. Clerc M. Particle Swarm Optimization. ISTE, London, UK, 2006.
10. Курейчик В.М. Модифицированные генетические операторы // Известия ЮФУ. Технические науки. – 2009. – № 12 (101). – С. 7-15.
11. Курейчик В.В., Курейчик В.М., Родзин С.И. Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 16-24.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.О. Чернышев.

**Лебедев Владимир Борисович**

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: lbk@tsure.ru.

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: 88634371743.

Кафедра системного анализа и телекоммуникаций; доцент.



**Lebedev Vladimir Borisovich**

Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: lbk@tsure.ru.

44, Nekrasovsky, Taganrog, 347928, Russia.

Phone: +78634371743.

The Department of System Analysis and Telecommunications; Associate Professor.

УДК 658.512

**И.Н. Ерошенко**

**ОБЗОР СОВРЕМЕННЫХ МОДЕЛЕЙ ЭВОЛЮЦИОННЫХ ВЫЧИСЛЕНИЙ  
ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПЛАНИРОВАНИЯ СБИС\***

*Рассматривается задача планирования сверхбольших интегральных схем. Планирование является одним из ранних этапов конструкторского проектирования СБИС. Данная задача относится к классу NP. В работе указаны недостатки традиционных алгоритмов поиска субоптимального плана СБИС. Приведены описания более современных моделей эволюционных вычислений: дифференциальной эволюции, муравьиного алгоритма, метода роя частиц, меметического алгоритма, который является гибридным генетическим алгоритмом. Для перечисленных алгоритмов приведены описания представления плана в виде польской записи, O-дерева, B\*-дерева.*

*Планирование СБИС; эволюционные вычисления; оптимизация.*

**I.N. Yeroshenko**

**REVIEW OF MODERN EVOLUTIONARY COMPUTATION MODELS  
FOR VLSI FLOORPLANNING**

*The article is dedicated to the problem of VLSI floorplanning. Floorplanning refers to the early stage of physical VLSI design. VLSI floorplanning is an NP-hard problem. In this paper the disadvantages of the traditional floorplanning algorithms for searching suboptimal solution are mentioned. The description of the more modern evolutionary computation models is given (differential evolution, ant colony optimization, swarm optimization, memetic algorithm which is a hybrid genetic algorithm). Floorplan representations Polish expression, O-tree, B\*-tree are briefly described for the mentioned algorithms.*

*VLSI floorplanning; evolutionary computation; optimization.*

**Введение.** Планирование является важным этапом конструкторского проектирования сверхбольших интегральных схем. Оно влияет на производительность, размеры чипа.

Планирование заключается в размещении модулей таким образом, чтобы они не пересекались, при этом целевая функция минимизировалась. Целью оптимизации могут быть минимизация площади кристалла, длины межсоединений и др. При качественном размещении не только уменьшается площадь микросхемы, но и уменьшается стоимость внутренних соединений. Задача планирования относится к классу NP.

**Планирование СБИС и представления плана.** При планировании СБИС важно выбрать соответствующее представление плана. Обычно выделяют два типа

---

\* Работа выполнена при частичной поддержке РФФИ (проект № 09-01-00492).