

Galaly Valentin Gavrilovich – Southern Federal University; e-mail: v.galalu@mail.ru; Taganrog, Russia; phone: +78634371638; the department of information technology and measuring systems; cand. of eng. sc.; associate professor.

Al-Karawi Hussein Sh. Mogheer – Southern Federal University; e-mail: alkaravi@sfned.ru; Taganrog, Russia; phone: +79612936340; the department of fundamental of radioengineering; postgraduate student.

Turulin Igor' P'ich – e-mail: iiturulin@sfned.ru; phone: +78634371632; the department of fundamental of radioengineering; dr. of eng. sc.; professor.

Kirakosyan Stepan Ayrapetovich – Radioavtomatika LLC; e-mail: ksa_say@mail.ru; Moscow, Russia; cand. of eng. sc.; head of the department for the development of technological equipment.

УДК 004.896

DOI 10.18522/2311-3103-2023-2-18-30

Б.К. Лебедев, О.Б. Лебедев, М.А. Ганжур**ОПТИМИЗАЦИЯ НА ОСНОВЕ ОБЪЕДИНЕНИЯ МОДЕЛЕЙ
АДАПТИВНОГО ПОВЕДЕНИЯ РОЯ АГЕНТОВ**

Разработана архитектура бионического поиска для решения задачи размещения элементов СБИС на основе гибридизации алгоритмов пчелиной колонии и роя хромосом, что позволяет выходить из «локальных ям» и увеличивает сходимость алгоритма размещения. Начальные итерации реализует пчелиный алгоритм, чтобы обеспечить широкий обзор области поиска, а завершающие – алгоритм роя хромосом, обеспечивающий точную локализацию экстремума, найденного пчелиным алгоритмом. Агенты представляются в виде популяции хромосом, являющихся генотипами решения задачи размещения. В работе описывается модифицированная парадигма роя хромосом, обеспечивающая, в отличие от канонического метода, возможность поиска решений в аффинном пространстве позиций с целочисленными значениями параметров. В поисковом популяционном методе оптимизации роем хромосом агентами популяция являются хромосомы. Хромосома является генотипом объекта оптимизации. Суть поисковой процедуры заключается в последовательной смене оператором направленной мутации состояний объекта оптимизации (хромосомы) и поиске оптимального состояния. Предложена аффинно-релаксационная модель (АРМ) роя хромосом – это граф вершины которого соответствуют хромосомам, а дуги соответствуют аффинным связям между ними. Переход хромосомы в новое состояние осуществляется с помощью релаксационной процедуры. В качестве средства изменения решения выступает оператор направленной мутации (ОНМ), суть которого заключается в изменении целочисленных значений генов в хромосоме. Целью перехода является сокращение веса аффинной связи между хромосомами. Описаны механизмы ОНМ. Предложена модифицированная структура алгоритма пчел. Для каждой базовой хромосомы реализуется вероятностный выбор набора хромосом, расположенных в окрестности базовой хромосомы. Улучшить качество работы разработанного алгоритма можно при помощи настройки значений управляющих параметров. Временная сложность алгоритма при фиксированных значениях размера популяции и количества генераций составляет $O(n)$. В общем зависимость времени работы гибридного алгоритма составляет $O(n^2) - O(n^3)$.

СБИС; размещение; роевой интеллект; пчелиный алгоритм; рой хромосом; гибридизация; аффинное пространство поиска; оператор направленной мутации; бионический поиск.

В.К. Lebedev, O.B. Lebedev, M.A. Ganzhur**OPTIMIZATION BASED ON COMBINING MODELS OF ADAPTIVE
BEHAVIOR OF A SWARM OF AGENTS**

A bionic search architecture has been developed to solve the problem of placing VLSI elements based on the hybridization of the algorithms of a bee colony and a swarm of chromosomes, which allows you to get out of "local holes" and increases the convergence of the placement algo-

rithm. The initial iterations are implemented by the bee algorithm to provide a broad overview of the search area, and the final iterations are implemented by the chromosome swarm algorithm, which ensures the exact localization of the extremum found by the bee algorithm. Agents are represented as a population of chromosomes, which are genotypes for solving the placement problem. The paper describes a modified paradigm of a swarm of chromosomes, which, in contrast to the canonical method, provides the possibility of searching for solutions in the affine space of positions with integer values of the parameters. In the search population method of optimization by a swarm of chromosomes, the agents of the population are chromosomes. The chromosome is the genotype of the optimization object. The essence of the search procedure is the successive change of the states of the object of optimization (chromosome) by the directed mutation operator and the search for the optimal state. An affine-relaxation model (ARM) of a swarm of chromosomes is proposed - this is a graph whose vertices correspond to chromosomes, and arcs correspond to affine bonds between them. The transition of the chromosome to a new state is carried out using a relaxation procedure. In the work, the directed mutation operator acts as a means of changing the solution, the essence of which is to change the integer values of genes in the chromosome. The purpose of the transition is to reduce the weight of the affine bond between chromosomes. The mechanisms of the directed mutation operator are described. A modified structure of the bee algorithm is proposed. For each base chromosome, a probabilistic choice of a set of chromosomes located in the vicinity of the base chromosome is implemented. It is possible to improve the quality of the developed algorithm by adjusting the values of the control parameters. The time complexity of the algorithm for fixed values of the population size and the number of generations is $O(n)$. In general, the dependence of the running time of the hybrid algorithm is $O(n^2) - O(n^3)$.

VLSI; placement; swarm intelligence; bee algorithm; chromosome swarm; hybridization; affine search space; directed mutation operator; bionic search.

Введение. С переходом на уровень нанотехнологий происходит смещение акцентов в проектировании СБИС [1, 2]. Так, с уменьшением линейных размеров элементов (транзисторов) и проводников, задержка в них масштабируется по-разному. Дефицит трассировочных ресурсов приводит к проблемам трассируемости схем, когда становится невозможно развести все необходимые соединения между элементами по коммутационным слоям [3, 4].

В момент размещения неизвестно расположение трасс и их характеристики, но от того, как будет выполнена трассировка межсоединений, зависят такие характеристики схемы, которые должны в итоге удовлетворить заданным ограничениям на быстродействие и трассируемость. Поэтому, на этапе размещения необходимо уметь более точно оценивать физические характеристики межсоединений так, чтобы эти оценки соответствовали результатам последующей трассировки.

Задача размещения относится к классу NP [5]. В течение последних лет были предложены различные подходы к решению этих проблем. Эти подходы могут быть классифицированы следующим образом: линейное и квадратичное программирование; имитация отжига; основанные на ограничениях; сила направленная парадигма; основанные на геометрической дуализации списков связей; иерархические методы сверху-вниз и снизу-вверх; метод кластеризации; генетические алгоритмы (ГА), и на основ поисковой адаптации биоинспирированные [1, 3–6] и др. Тем не менее, в последнее время для решения различных «сложных» задач, к которым относятся и задачи размещения всё чаще используются способы, основанные на применении методов искусственного интеллекта. Особенно наблюдается стремительный рост интереса к разработке алгоритмов, инспирированных природными системами [7]. В основе большинства этих алгоритмов лежат идеи, заимствованные в природе [8].

Новые требования к задаче размещения требуют новых подходов к ее решению. Проанализировав известные подходы, применяемые для решения «сложных» задач, можно сделать вывод, что использование какого-либо одного алгоритма не дает гарантии получения качественного решения. В связи с этим, в настоящее

время один из способов улучшения эффективности методов нахождения глобального оптимума решаемых задач, состоит в гибридизации алгоритмов [9]. Особенностью гибридного алгоритма является то, что достоинства одного алгоритма могут возместить недостатки другого. Объединение различных методик поисковых алгоритмов, дает возможность определения большей области допустимых решений и нахождение оптимального решения.

В работе описан разработанный алгоритм решения задачи размещения элементов СБИС, объединяющий процедуры поведения коллектива пчел и роя хромосом.

В данной работе предлагаются усовершенствованные методы размещения для решения упомянутых проблем, связанных с увеличением относительной задержки в проводниках и уменьшением трассировочных ресурсов.

1. Постановка задачи размещения. Основные проблемы, решаемые при разработке алгоритмов размещения, и сравнительные характеристики существующих алгоритмов размещения могут быть даны с учетом следующих показателей [3–5]:

- 1) модель представления конструктивных элементов, как геометрических объектов;
- 2) модель монтажного пространства (пространство позиций);
- 3) модель представления электрической принципиальной схемы;
- 4) метод поиска решений, включающий:
 - ◆ способ выбора позиции для установки элементов;
 - ◆ характер целевой функции выбора позиции;
 - ◆ характер целевой функции выбора элемента-претендента из множества неразмещенных;
 - ◆ характер целевой функции для оценки размещения [5].

Необходимо разместить элементы на коммутационном поле с оптимизацией некоторых критериев качества.

Простой и в то же время эффективной оценкой длины проводника является полупериметр охватывающего цепь прямоугольника. Известно, что для двух- и трех- контактных цепей полупериметр охватывающего прямоугольника совпадает с длиной минимального дерева Штейнера, которое используется во многих алгоритмах трассировки. Поэтому сумма полупериметров охватывающих прямоугольников по всем цепям является наиболее распространенной целевой функцией в алгоритмах размещения.

В настоящий момент обычно используют один из двух вариантов оценки длины соединений:

- а) метод полупериметра – для каждой цепи подсчитывается полупериметр минимального охватывающего прямоугольника;
- б) метод полного графа – вычисляется нормированная сумма расстояний между всеми парами узлов цепи.

Пусть дано множество элементов $A = \{a_j | j = 1, 2, \dots, n\}$ и множество позиций $P = \{p_i | i = 1, 2, \dots, c\}$ на КП.

Для размещения всех элементов необходимо выполнение условия $c \geq n$.

Произвольное размещение элементов в позициях представляет собой функциональное (однозначное) соответствие между множествами P и A , обозначается как $f: P \rightarrow A$. Каждому элементу p_i из области определения функция f ставит в соответствие единственный элемент a_j из области значений. Это обозначается $f(p_i) = a_j$. Элемент p_i – аргумент функции, элемент a_j – значение функции на p_i . В зависимости от выбранного критерия для оценки результатов размещения вводится целевая функция $F(f)$.

Задача размещения состоит в отыскании оптимального значения функции $F(f)$.

В качестве модели схемы используется граф $G=(X, Y)$ или гиперграф $H=(X, E)$, где $X = \{x_i | i=1, 2, \dots, n\}$ – множество вершин, моделирующих элементы, а $U = \{u_j | j=1, 2, \dots, l\}$ – множество ребер. Вершина x_i связана с x_j ребром, если соответствующие элементы связаны соединением.

$E = \{e_j | e_j \subset X, j=1, 2, \dots, m\}$ – множество гиперребер, моделирующих цепи, связывающие элементы. Граф адекватно моделирует двутерминальные соединения, а гиперграф – многотерминальные.

Расстояние между двумя вершинами с координатами (x_i, y_i) и (x_j, y_j) определяется по формуле:

$$d_{ij} = |x_i - y_i| + |x_j - y_j|.$$

В настоящий момент основными известными критериями при размещении [1, 3, 5] являются: минимальная суммарная длина связи, минимальная длина самой длинной связи, минимум числа возможных пересечений, минимум числа изгибов соединений, минимальная площадь кристалла.

В качестве оценки l_j длины цепи t_j , моделируемой гиперребром e_j , используются: длина минимального связывающего дерева, построенного на множестве вершин $e_j \subset X$; длина звездного графа, ребра которого инцидентны вершинам $e_j \subset X$, а корневая вершина помещена в центре “тяжести” множества вершин e_j ; длина полупериметра прямоугольника, описывающего множество вершин e_j ; суммарная длина ребер полного графа, построенного на множестве e_j . С учетом этого, критерий оптимизации имеет вид:

$$F = \sum_{j=1}^m l_j.$$

Для более полного учета связей между задачами размещения и трассировки более эффективны алгоритмы, основанные на оценках числа цепей, пересекающих заданные линии КП. Эти линии могут быть либо прямыми, пересекающими все КП, либо замкнутыми и ограничивающими некоторую область [5].

Пусть на КП наложена опорная сеть $G(X, U)$, разбивающая КП на области $O = \{o_k | k=1, 2, \dots, n_k\}$, где $X = \{x_i | i=1, 2, \dots, n\}$ – множество опорных узлов сети, $U = \{u_j | j=1, 2, \dots, m\}$ множество ребер, связывающих узлы. Позиции располагаются внутри областей. В качестве исходных данных для КП задается число позиций p_k внутри каждой области o_k и для каждого ребра u_i $D = \{d_j | j=1, 2, \dots, m\}$, где – задается пропускная способность d_j , т.е. число цепей (трасс), которые могут ее пересечь. Значения $\sum d_j$ определяются размерами ребра и ограничениями на прокладку соединений.

Назовем цикл L_k , составленный из ребер сетки и ограничивающий некоторую область, границей области. Под пропускной способностью PS_k границы L_k будем понимать суммарную пропускную способность ребер сетки, входящих в состав L_k , т.е. $PS_k = \sum_j d_j$ ($\forall j | u_j \in L_k$),

Пусть H_k – число цепей, связывающих элементы, расположенные внутри области, ограниченной L_k , с элементами, расположенными вне этой области. Введем характеристику границы:

$$\gamma_k = \frac{PS_k - H_k}{PS_k}.$$

Чем большее значение имеет γ_k , тем легче осуществить прокладку связей через границу L_k .

Пусть задано некоторое множество областей $O = \{o_k | k=1, 2, \dots, n_k\}$, для которых определено множество границ $L = \{L_k | k=1, 2, \dots, k_L\}$.

Пусть задано некоторое размещение элементов. Найдем среди характеристик границ наименьшую γ_{min} , т.е. $\forall k [(PS_k - H_k) / PS_k \geq \gamma_{min}]$.

Величина $F = \gamma_{min}$ используется в качестве критерия оптимизации. Задача оптимизации – максимизация γ_{min} .

2. Подход к представлению решений в алгоритме на основе роевого интеллекта. В работе представлена архитектура многоагентной системы на основе природных вычислений, выполняющая размещение компонентов сверхбольших интегральных схем, используя объединенные модели роевого интеллекта: роя хромосом (chromosomeswarmoptimization – CSO) и пчелиного роя [10, 11].

Решение задачи размещения представляется в виде вектора $A = \{a_i | i = 1, 2, \dots, n\}$, где a_i – номер элемента, размещаемого в i -й позиции. Если число размещаемых элементов на m меньше числа позиций, то m элементов p_i в векторе A имеет нулевое значение [5].

Хромосома $H_i = \{g_{il} | l = 1, 2, \dots, n_i\}$ представляет собой совокупность n генов g_{il} . Значение гена g_{il} равно значению соответствующего элемента, размещаемого в i -й позиции.

В поисковом популяционном методе оптимизации роем хромосом агентами популяция являются хромосомы. Хромосома является генотипом объекта оптимизации. Миграция агентов реализуется с помощью операторов направленного поиска. Каждая хромосома H_i связана со всем роем $H = \{H_{ij} | i = 1, 2, \dots, n_o\}$, может взаимодействовать со всем роем и она тяготеет к лучшему решению H^* роя. В каждый момент времени t (на каждой итерации) хромосомы находятся в некотором состоянии, определяемом значениями генов. Для каждого состояния хромосомы $H_i(t)$ вычисляется соответствующее значение целевой функции $f_i(t)$ [5, 12].

Суть поисковой процедуры заключается в последовательной смене оператором направленного поиска состояний объекта оптимизации (хромосомы) и поиске оптимального состояния [13].

При определении следующего состояния хромосомы учитывается информация о «наилучших» хромосомах $H^*(t)$ из числа «соседей» данной хромосомы, а также информация о данной хромосоме на той итерации, когда этой хромосоме $H_i^*(t)$ соответствовало наилучшее значение целевой функции $f_i^*(t)$, и на этой основе по определенным правилам хромосома меняет свое состояние. [14].

Аффинно-релаксационная модель (АРМ) роя хромосом $H = \{H_i | i = 1, 2, \dots, n_o\}$ – это граф вершины которого соответствуют хромосомам, а дуги соответствуют аффинным связям между хромосомами. Аффинность – мера близости двух хромосом агентов. На каждой итерации каждая хромосома $H_i \in H$ переходит в аффинном графе в новое состояние $H_i(t+1)$, при котором вес аффинной связи между хромосомой H_i и базовой (лучшей) H_i^* уменьшается. Переход хромосомы $H_i(t)$ в новое состояние $H_i(t+1)$ осуществляется с помощью релаксационной процедуры, зависящей от вида структуры данных (хромосомы): вектор, матрица, дерево и их совокупности, являющейся интерпретацией решений.

В работе в качестве средства изменения решения выступает оператор направленной мутации (ОНМ), суть которого заключается в изменения целочисленных значений генов h_{il} в хромосоме $H_i(t) = \{h_{il}(t) | l = 1, 2, \dots, n\}$. Переход хромосомы в новое состояние означает переход от хромосомы $H_i(t)$ к новой $H_i(t+1)$ – с новыми целочисленными значениями генов, полученными после применения ОНМ.

В качестве веса аффинной связи между хромосомами $H_i(t)$ и $H_j(t)$ используются оценки степени близости S_{ij} между двумя хромосомами.

Целью перехода $H_i(t)$ в $H_i(t+1)$ является сокращения веса аффинной связи между хромосомой $H_i(t)$ и лучшей хромосомой $H^*(t)$ популяции.

Аналитическое выражение для расчета оценки степени близости S_{ij} , зависящей от вида структуры хромосомы: вектор, матрица, дерево и их совокупности, являющейся интерпретацией решений [14].

Пример. В качестве степени близости между гомологичными хромосомами $H_i(t)$ и $H_j(t)$ будем использовать величину S_i :

$$S_{ij} = \sum_l |h_{il} - h_{jl}|,$$

где $h_{il} \in H_i(t)$, $h_{jl} \in H_j(t)$, S_{ij} – степень близости (вес аффинной связи) между двумя хромосомами $H_i(t)$ и $H_j(t)$.

Суть процедуры перехода, реализуемой ОНМ, заключается в изменениях разности между значениями каждой пары генов (h_{il}, h_{jl}) двух хромосом, $l=1, 2, \dots, n$. Если вес аффинной связи между хромосомами рассматривать как длину, то в процессе поисковой процедуры происходит релаксационное сжатие аффинного графа.

Рассмотрим *процедуру перехода*, которая была разработана авторами и заключается в изменениях взаимного расположения элементов в хромосоме. Хромосома $H_i(t)$ переходит в новое состояние $H_i(t+1)$ с новым взаимным расположением элементов в списке.

Пример работы процедуры перехода (ОНМ).

Пусть $H_i(t)$ и $H_j(t)$ имеют вид: $H_i(t) = \{1, 3, 2, 10, 8\}$, $H_j(t) = \{1, 10, 2, 3, 8\}$.

На первом такте в $H_i(t)$ формируется множество пар $D_1 = (1, 3), (2, 10)$. Взаимное расположение элементов пары $(1, 3)$ в $H_i(t)$ и $H_j(t)$ совпадают, пары $(2, 10)$ – нет. В $H_i(t)$ переставляются местами элементы пары $(2, 10)$. $H_i(t+1) = \{1, 3, 10, 2, 8\}$.

На втором такте в $H_i(t+1)$ формируется множество пар $D_2 = (3, 10), (2, 8)$. Взаимное расположение элементов пары $(2, 8)$ в $H_i(t+1)$ и $H_j(t)$ совпадают, пары $(3, 10)$ – нет. В $H_i(t+1)$ переставляются местами элементы пары $(3, 10)$. $H_i(t+2) = \{1, 10, 3, 2, 8\}$.

Алгоритм оптимизации – рой хромосом имеет вид:

1. Формирование исходного роя хромосом $H = \{H_i | i=1, 2, \dots, n_o\}$. $t = 1$.
2. Расчет целевой функции $f_i(t)$ для всех H_i .
3. Выбор лучших хромосом $H^*(t)$ и $H^*(t)$.
4. Если $t < T$, то $t = t + 1$ и переход к пункту 5, иначе переход к пункту 7.
5. Переход всех хромосом роя под действием ОНМ в новые состояния.
6. Переход к пункту 2.
7. Конец работы алгоритма.

3. Адаптивное поведение пчелиной колонии. Пусть имеется популяция пчел $P = \{p_i | i=1, 2, \dots, n_i\}$. На первой итерации ($t=1$) пчелы-разведчики случайным образом размещаются в области поиска решений. Каждая пчела разведчик p_i на шаге t выбирает позицию $X_i(t)$. Каждая позиция является моделью решения задачи размещения. Объем нектара является значением критерия в этой точке [15].

Позиция $X_i(t)$, выбранная пчелой, представляется в виде хромосомы $H_i(t) = \{h_{ij} | j=1, 2, \dots, n_o\}$.

Таким образом первая операция ($t=1$) выполняемая пчелами заключается в генерации случайным образом исходного роя отличающихся друг от друга хромосом $H = \{H_i | i=1, 2, \dots, n_o\}$. Для каждой хромосомы $H_i(t)$ вычисляется соответствующее значение целевой функции $f_i(t)$.

Среди роя хромосом популяции H выбираются n_θ лучших решений, которые включаются в базовое множество $H^\theta(t)$ хромосом, $H^\theta(t) \subset H$. Обычно реализуется вероятностный способ формирования $H^\theta(t)$. Вероятность $p(H^\theta(t))$ выбора агентом фуражиром базовой $H^\theta_i \in H(t)$ для включения в $H^\theta(t)$ пропорциональна значению целевой функции $f_i(t)$ этой хромосомы и определяется как:

$$p(H^\theta(t)) = f_i(t)^\theta / \sum_i f_i(t)^\theta.$$

Далее в окрестности каждой базовой хромосомы $H^\theta_i(t)$ формируется соответствующий ей набор хромосом $O^\theta_i(t+1) = \{o_{ij}(t+1) | j=1, 2, \dots, n_o\}$, $F: H^\theta_i(t) \rightarrow O^\theta_i(t+1)$.

Формирование новой хромосомы $o_{ij}(t+1)$, лежащей в θ -окрестности базовой позиции $H^\theta_i(t)$ производится путём δ выборочных (случайных) парных перестановок соседних элементов в векторе $H^\theta_i(t)$. Будем считать, что решение $o_{ij}(t+1)$ лежит в θ -окрестности решения $H^\theta_i(t)$, если $o_{ij}(t+1)$ получено путем θ случайных парных перестановок соседних элементов в списке $H^\theta_i(t)$ [16].

Обозначим множество хромосом, вошедших в θ -окрестность хромосомы $H_i^\theta(t)$, как $O_k^\theta(t+1)$. Рассчитывается оценка каждой хромосомы множества $O_k^\theta(t+1)$. В каждой θ -окрестности $O_k^\theta(t+1)$ выбирается лучшая хромосома $o_{ij}^*(t+1)$. Лучшие хромосомы θ -окрестностей образуют новое множество базовых хромосом $H^*(t)$.

Лучшее решение (хромосома) среди множества $H^*(t+1)$ сохраняется, а затем происходит переход к следующей итерации. В начале второй и на последующих итерациях, прежде всего, формируется множество базовых хромосом $H^\theta(t)$ ($t=2,3,\dots,L$), составленное из двух частей $H^{\theta 1}(t)$ и $H^{\theta 2}(t)$, $H^{\theta 1}(t) \cup H^{\theta 2}(t) = H^\theta(t)$. В первую часть $H^{\theta 1}(t)$ включаются $n_{\theta 1}$ лучших хромосом, среди $H^*(t+1)$, найденных агентами в каждой из областей, сформированных на предыдущей итерации. Вторая часть $H^{\theta 2}(t)$ формируется агентами разведчиками также, как и на первой итерации. Далее выполняются действия, аналогичные действиям, рассмотренным на первой итерации.

Алгоритм оптимизации пчелиным роем имеет вид:

1. Формирование исходного роя хромосом $H = \{H_i | i=1,2,\dots,n_o\}$. $t=1$.
2. Расчет целевой функции $f_i(t)$ для всех H_i .
3. Среди хромосом роя H выбираются n_θ лучших, которые включаются в базовое множество $H^\theta(t)$ хромосом, $H^\theta(t) \subset H$.
4. В окрестности каждой базовой хромосомы $H_i^\theta(t) \in H^\theta(t)$ формируется соответствующий ей набор хромосом $O_i^\theta(t+1) = \{o_{ij}(t+1) | j=1,2,\dots,n_o\}$, $(\forall i)F: H_i^\theta(t) \rightarrow O_i^\theta(t+1)$. $|H_i^\theta(t)|=1$, $|O_i^\theta(t+1)|=n_o$.
5. В каждой окрестности $O_i^\theta(t+1)$ отыскивается лучшая хромосома $o_{ij}^*(t+1)$. $(\forall i)F: O_i^\theta(t+1) \rightarrow o_{ij}^*(t+1)$. $|O_i^\theta(t+1)|=n_o$, $|o_{ij}^*(t+1)|=1$.
6. Лучшие хромосомы θ -окрестностей образуют новое базовое множество хромосом $H^\theta(t)$.
7. Переход к следующей итерации.

4. Гибридизация структуры роевого интеллекта. Разработанный алгоритм решения задачи размещения элементов СБИС, использует архитектуру бионического поиска, и состоит из объединенных процедур алгоритмов пчелиной колонии и роя хромосом, что позволяет выходить из «локальных ям» и увеличивает сходимость алгоритма размещения. Данные представляются в матричном или векторном виде.

При движении в области допустимых значений популяция агентов по очередности представляется как рой пчел или рой хромосом, с определенными свойствами адаптивного поведения. Агенты представляются в виде популяции хромосом $H_i(t) = \{g_{il} | l=1,2,\dots,n_l\}$, являющихся генотипами решения задачи размещения

В работе применяется гибридизация типа препроцессор/поспроцессор. Начальные итерации реализует пчелиный алгоритм [16], чтобы обеспечить широкий обзор области поиска, а завершающие итерации при помощи алгоритма роя хромосом, обеспечивающего точную локализацию экстремума, найденного пчелиным алгоритмом.

Первая операция ($t=1$) выполняемая пчелами заключается в генерации случайным образом исходного роя отличающихся друг от друг хромосом $H = \{H_i | i=1,2,\dots,n_o\}$.

Далее, в соответствии с механизмами пчелиной колонии пчелами фуражирами формируются θ -окрестности каждой из базовых хромосом множества $H_i^\theta(t)$.

Процедурой ОКРЕСТНОСТЬ в окрестности (на базе) каждой хромосомы H_i^θ генерируется легитимный рой хромосом $O_i^\theta = \{o_{ij} | j=1,2,\dots,n_o\}$, в состав которого входит H_i^θ .

Далее путем многократного применения алгоритма роя хромосом к каждому O_i^θ хромосомы каждого рома переходят в новые состояния.

Ключевой операцией пчелиного алгоритма является исследование перспективных хромосом и их θ окрестностей в пространстве решений. В каждом обновленном рое $O_i^\delta(t+1)$ выбирается лучшая хромосома o_{ij}^* . Лучшие хромосомы o_{ij}^* включаются в базовое множество хромосом $H^\delta(t+1)$. На последующей $(t+1)$ -ой итерации гибридного алгоритма множество хромосом $H^\delta(t+1)$ рассматривается как базовый рой хромосом.

Гибридный алгоритм размещения имеет вид:

1. Формирование исходного базового роя хромосом $H = \{H_i | i=1, 2, \dots, n_o\}$ с заданной исходной мощностью n_o .

2. Формирование начальных значений генов у хромосом $H_i \in H$.

Если хромосомы гомологичные, то допускается совпадение значений генов в составе хромосомы. В этом случае генам задаются значения в пределах заданных диапазонов.

Если хромосомы негомологичные, то не допускается совпадение значений генов в составе хромосомы. Хромосомы различаются порядком расположения значений генов в хромосоме.

3. Для каждой хромосомы H_i рассчитывается оценка f_i целевой функции.

4. РХ H сужается до H^δ заданного размера $n < n_o$ путем отбрасывания хромосом с худшими оценками. $|H^\delta|=n$. $i=0$.

5. $i=i+1$.

6. Процедурой ОКРЕСТНОСТЬ в окрестности (на базе) каждой хромосомы H_i^δ генерируется легитимный рой хромосом $O_i^\delta = \{o_{ij} | j=1, 2, \dots, n_o\}$, в состав которого входит H_i^δ .

В случае гомологичных хромосом допускается совпадение значений генов. У заданного числа генов в составе хромосомы H_i^δ значения изменяются в пределах заданных диапазонов.

В случае негомологичных хромосом задается число пар генов, порядок расположения которых в новых хромосомах друг относительно друга подвергается инверсии.

8. У каждого роя O_i^δ для каждой хромосомы $o_{ij} \in O_i^\delta$ рассчитывается оценка f_{ij} целевой функции.

9. Каждый РХ O_i^δ сужается до заданного размера $n < n_o$ путем отбрасывания хромосом с худшими оценками. $|O_i^\delta|=n$.

10. На каждой из заданного числа итераций, хромосомы каждого роя O_i^δ , работанным алгоритмом роя хромосом последовательно k раз переходят в новые состояния:

$$O_i^\delta(t) \rightarrow O_i^\delta(t+1) \rightarrow O_i^\delta(t+2) \dots O_i^\delta(t+3) \rightarrow$$

11. В каждом рое O_i^δ выбирается хромосома $o_{ij}^* \in O_i^\delta$ с максимальным значением оценки f_{ij}^* . o_{ij}^* включается в множество O_i^* .

12. Формирование исходного базового роя хромосом $H^\delta = \{H_i^\delta | i=1, 2, \dots, n\}$. Так как $|O_i^*|=|H^\delta|=n$, множество O_i^* переименуется в множество H^δ , т.е. $(\forall i) [H_i^\delta = o_{ij}^*]$.

Результатом работы алгоритма является рой O_i^* , включающий хромосому с лучшим значением целевой функции.

Для получения более качественного решения предусмотрен расширенный вариант гибридного алгоритма, отличающийся дополнительным использованием алгоритма роя хромосом.

Ниже приведены основные этапы расширенного варианта гибридного алгоритма.

1. Формирование базового роя хромосом $H^\delta(t) = \{H_i^\delta(t) | i=1, 2, \dots, n_o\}$. $t=1$.

2. Расчет целевой функции $f_i(t)$ для всех $H_i^\delta(t)$.

3. В окрестности каждой базовой хромосомы $H_i^\delta(t) \in H^\delta(t)$ формируется соответствующий ей набор хромосом $O_i^\delta(t+1) = \{o_{ij}(t+1) | j=1, 2, \dots, n_o\}$, $(\forall i) F: H_i^\delta(t) \rightarrow O_i^\delta(t+1)$. $|H_i^\delta(t)|=1$, $|O_i^\delta(t+1)|=n_o$.

4. К хромосомам каждого роя O_i^{δ} , $i=[1;n_{\delta}]$ последовательно k раз применяется алгоритм роя хромосом. После каждого применения алгоритма РХ хромосомы роя переходят в новые состояния:

$$O_i^{\delta}(1) \rightarrow O_i^{\delta}(2) \dots \rightarrow O_i^{\delta}(k).$$

5. В каждом рое O_i^{δ} выбирается хромосома $o_{ij}^* \in O_i^{\delta}$ с максимальным значением оценки $f_{ij}^* \cdot o_{ij}^*$ включается в множество O_i^* .

6. К рою O_i^* , применяется алгоритм роя хромосом. Хромосомы роя переходят в новые состояния: $O_i^*(vx) \rightarrow O_i^*(vix)$.

7. Формирование исходного базового роя хромосом $H^{\delta}=\{H_{ij}^{\delta}, i=1,2,\dots,n\}$. Так как $|O_i^*|=|H^{\delta}|=n$, множество O_i^* переименуется в множество H^{δ} , т.е. $(\forall i)[H_{ij}^{\delta}=o_{ij}^*]$

8. Если $t < T$, то $t = t + 1$ и переход к пункту 1, иначе конец алгоритма.

5. Экспериментальные исследования. Исследования гибридного роевого алгоритма размещения (ГРАР) состоит в формировании тестовых заданий для задачи размещения с имеющимся оптимальным результатом (РЕКО) [17]. Оптимальные результаты РЕКО имеются в обоих форматах GSRC BookShelf и LEF/DEF, и они доступны в сети [18].

Все цепи в РЕКО являются локальными, т.е. длина проводников каждой цепи имеет минимально возможное значение. Схемы набора РЕКУ состоят из локальных цепей в стиле РЕКО. Для экспериментальных исследований разработанной программы размещения были применены схемы набора РЕКУ с известным оптимумом F_{opt} : Ex.1 на 30 блоков, Ex.2 – 60, Ex.3 – 90, Ex.4 – 120, Ex.5 – 150. Для сравнения были выбраны современные алгоритмы размещения: Dragon v.2.20 [19], Caro v.8. [20], mPL v.2.0 [21], mPG v.1.0 [22] и QPlace v.5.1. [23].

Для определения оптимальности достигнутых значений, рассчитывался параметр: наилучшая длина соединений F_{opt} к полученной длин соединения F (для РЕКО) или (для G-РЕКУ и РЕКУ). Это отношение называется «степенью качества».

В табл. 1. приведены полученные значения показателя степень качества F_{opt}/F ряда известных алгоритмов и алгоритма ГРАР.

Таблица 1

Значения показателя «степень качества»

Тест	Dragon	Caro	mPL	mPG	Qplace	ГРАР
Ex.1	0.71	0.73	0.81	0.71	0.73	0.92
Ex.2	0.72	0.72	0.84	0.72	0.78	0.9
Ex.3	0.81	0.8	0.83	0.81	0.82	0.9
Ex.4	0.83	0.84	0.85	0.83	0.81	0.92
Ex.5	0.82	0.81	0.86	0.82	0.84	0.92

Степень качества у разработанной программы РАР на 10% больше, чем у программ Dragon, Caro, mPL, mPG и Qplace. ВСА $O(n^2)$.

На основе анализа проведенных исследований можно сделать следующие выводы:

1. Ни один из алгоритмов размещения при исследовании не достиг коэффициента качества близкого к 1

2. Показатель качества для одного и того же алгоритма размещения может значительно варьироваться для схем схожего размера, но с различными характеристиками. Ни один из алгоритмов не получает неизменно лучшие результаты, чем другой.

3. Различные алгоритмы размещения демонстрируют различную масштабируемость в показателях времени выполнения и качества решения. Ни один из них не может успешно решить все схемы набора РЕКО, из-за ограниченного времени решения (например Dragon), или расхода памяти (например Caro, mPL v.2.0.).

Анализ сходимости алгоритма PАР осуществлялся следующим образом. Каждое тестовое задание запускалось на выполнение 10 раз. Для каждого теста определялся номер итерации, после которой далее не происходило улучшения критерия.

В результате экспериментов установлено, что при объеме популяции $M=100$ алгоритм сходится в среднем на 115 итерации. При этом отклонения в сторону увеличения этой оценки составлял до 10%, а в сторону уменьшения до 35%. Оценкой качества служит величина $F_{\text{опт}}/F$, где F – оценка полученного решения.

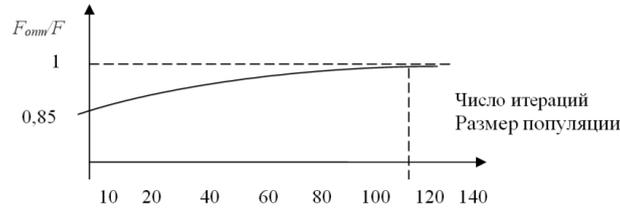


Рис. 1. Зависимость качества решений от числа итераций

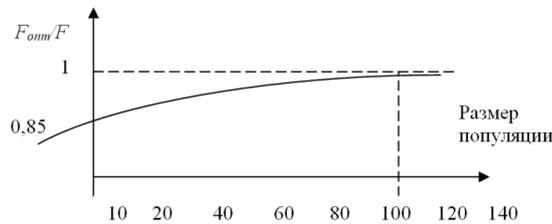


Рис. 2. Зависимость качества решений от размера популяции

Сравнение значений критерия, полученных муравьиным алгоритмом на тестовых примерах с известным оптимумом показало, что у 70% примеров полученное решение было оптимальным, у 5% примеров решения были на 5% хуже, а у 25% примеров решения были хуже не более, чем на 2%. Сравнение с известными алгоритмами показало, что при меньшем времени работы у полученных с помощью разработанного алгоритма решений отклонение целевой функции от оптимального значения меньше в среднем на 6%.

В настоящее время Большое количество примеров содержится в электронных библиотеках OR-Library (<http://mscmga.ms.ic.ac.uk/mfo.html>) и DIMACS Challenge II (<ftp://dimacs.rutgers.edu/pub/challenge/graph/>).

Временная сложность алгоритма при фиксированных значениях размера популяции и количества генераций составляет $O(n)$. Общая временная сложность гибридного алгоритма составляет $O(n^2) - O(n^3)$.

Заключение. С переходом на уровень нанотехнологий происходит смещение акцентов в проектировании СБИС. Дефицит трассировочных ресурсов приводит к проблемам трассируемости схем, когда становится невозможно развести все необходимые соединения между элементами по коммутационным слоям. В работе решение задачи размещения направлено на повышение трассируемости посредством минимизации ресурсов, требуемых для реализации соединений. Для более полного учета связей между задачами размещения и трассировки более эффективны алгоритмы, основанные на оценках числа цепей, пересекающих заданные линии КП [2, 3].

Разработана архитектура бионического поиска для решения задачи размещения элементов СБИС на основе гибридизации алгоритмов пчелиной колонии и роя хромосом, что позволяет выходить из «локальных ям» и увеличивает сходимость

алгоритма размещения. Начальные итерации реализует пчелиный алгоритм, чтобы обеспечить широкий обзор области поиска, а завершающие – алгоритм роя хромосом, обеспечивающий точную локализацию экстремума, найденного пчелиным алгоритмом. Агенты представляются в виде популяции хромосом, являющихся генотипами решения задачи размещения

В работе описывается модифицированная парадигма роя хромосом, обеспечивающая, в отличие от канонического метода, возможность поиска решений в аффинном пространстве позиций с целочисленными значениями параметров.

Суть поисковой процедуры заключается в последовательной смене оператором направленной мутации состояний объекта оптимизации (хромосомы) и поиске оптимального состояния. В поисковом популяционном методе оптимизации роем хромосом агентами популяция являются хромосомы. Хромосома является генотипом объекта оптимизации.

Предложена аффинно-релаксационная модель (АРМ) роя хромосом – это граф вершины которого соответствуют хромосомам, а дуги соответствуют аффинным связям между ними. Переход хромосомы в новое состояние осуществляется с помощью релаксационной процедуры.

В работе в качестве средства изменения решения выступает оператор направленной мутации (ОНМ), суть которого заключается в изменении целочисленных значений генов в хромосоме. Целью перехода является сокращении веса аффинной связи между хромосомами. Описаны механизмы

оператора направленной мутации.

Предложена модифицированная структура алгоритма пчел. Для каждой базовой позиции реализуется вероятностный выбор набора позиций, расположенных в окрестности базовой позиции.

Улучшить качество работы разработанного алгоритма можно при помощи настройки значений управляющих параметров. Временная сложность алгоритма при фиксированных значениях размера популяции и количества генераций составляет $O(n)$. В общем зависимость времени работы гибридного алгоритма составляет $O(n^2) - O(n^3)$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Казеннов Г. Основы проектирования интегральных схем и систем. – М.: Бином. Лаборатория знаний, 20105.
2. Тучин А.В., Бормонтов Е.Н., Пономарев К.Г. Введение в системы автоматизированного проектирования интегральных микросхем. – Воронеж: Издательский дом ВГУ, 2017.
3. Норенков И.П. Основы автоматизированного проектирования. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2006.
4. Alpert C.J., Mehta D.P., and Sapatnekar S.S. Handbook of Algorithms for Physical Design Automation. – Boston, MA: Auerbach, 2009.
5. Лебедев Б.К., Лебедев В.Б., Лебедев О.Б. Методы, модели и алгоритмы размещения. – Ростов-на-Дону: Изд-во ЮФУ, 2015.
6. Jason Cong, Joseph R., Shinnerl u Min Xie (UCLA Computer Science), Tim Kong (Magma Design Automation) u Xin Yuan (IBM Corporation, Microelectronics Division). Large-Scale Circuit Placement // ACM Transactions on Design Automation of Electronic Systems. – April 2005. – Vol. 10, No. 2.
7. Poli R. Analysis of the publications on the applications of particle swarm optimization // Journal of Artificial Evolution and Applications, Article ID 685175, 2008.
8. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014.
9. Wang X. Hybrid nature-inspired computation method for optimization: Doctoral Dissertation. – Helsinki University of Technology, TKK Dissertations, Espoo 2009. – 161 p.
10. Clerc M. Particle Swarm Optimization. – ISTE, London, UK, 2006.

11. *Kennedy J., Eberhart R.C.* Particle swarm optimization // In Proceedings of IEEE International Conference on Neural Networks. – 1995. – P. 1942-1948.
12. *Лебедев Б.К., Лебедев О.Б., Лебедев В.Б.* Гибридный метод стохастической оптимизации на основе интеграции моделей эволюции и роевого (стадного) поведения животных в аффинных пространствах поиска // Сб. трудов Шестнадцатой национальной конференции по искусственному интеллекту с международным участием КИИ-2018. В 2 т. Т. 2. – М.: ФГП ИТАР-ТАСС филиал РКП, 2018. – С. 148-156.
13. *Лебедев Б.К., Лебедев О.Б., Лебедева Е.О., Нагабедян А.А.* Гибридный роевой алгоритм глобальной оптимизации в аффинном пространстве поиска // Электронный журнал «Программные продукты, системы и алгоритмы». – 2019. – № 1.
14. *Lučić P., Teodorović D.* Computing with Bees: Attacking Complex Transportation Engineering Problems // International Journal on Artificial Intelligence Tools. – 2003. – No. 12. – P. 375-394.
15. *Quijano N., Passino K.M.* Honey Bee Social Foraging Algorithms for Resource Allocation: Theory and Application. – Columbus: Publishing house of the Ohio State University, 2007. – 39 p.
16. *Лебедев Б.К., Лебедев В.Б.* Размещение на основе метода пчелиной колонии // Известия ЮФУ. Технические науки. – 2010. – № 12. – С. 12-18.
17. *Cong J., Romesis M., and Xie M.* Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms // Proc. of the International Symposium on Physical Design, Monterey, CA, April 2004. – P. 88-94.
18. MCNC. Electronic and Information Technologies (Online). Available: www.mcnc.org.
19. *Wang M., Yang X. and Sarrafzadeh M.* Dragon2000: Standard-cell Placement Tool for Large Industry Circuits // ICCAD 2000. – P. 260-263.
20. *Caldwell A.E., Kahng A.B. and Markov I.L.* Can Recursive Bisection Alone Produce Routable Placements? // DAC 2000. – P. 477-482.
21. *Chan T., Cong J., Kong T., Shinnerl J. u Sze K.* An enhanced multilevel algorithm for circuit placement // In Proceedings of the IEEE International Conference on Computer Aided Design (San Jose, Calif.). – IEEE Computer Society Press, Los Alamitos, Calif. 2003.
22. *Yang X., Choi B.-K. and Sarrafzadeh M.* Routability-driven white space allocation for fixed-die standard-cell placement // IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems. – 2003. – 22 (4). – P. 410-419.
23. CADENCE DESIGN SYSTEMS, INC. QPlace version 5.1.55, compiled on 10/25/1999. Envisia ultra-placer reference.

REFERENCES

1. *Kazennov G.* Osnovy proektirovaniya integral'nykh skhem i system [Fundamentals of designing integrated circuits and systems]. Moscow: Binom. Laboratoriya znaniy, 20105.
2. *Tuchin A.V., Bormontov E.N., Ponomarev K.G.* Vvedenie v sistemy avtomatizirovannogo proektirovaniya integral'nykh mikroskhem [Introduction to computer-aided design of integrated circuits]. Voronezh: Izdatel'skiy dom VGU, 2017.
3. *Norenkov I.P.* Osnovy avtomatizirovannogo proektirovaniya [Fundamentals of computer-aided design]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2006.
4. *Alpert C.J., Mehta D.P., and Sapatnekar S.S.* Handbook of Algorithms for Physical Design Automation. Boston, MA: Auerbach, 2009.
5. *Lebedev B.K., Lebedev V.B., Lebedev O.B.* Metody, modeli i algoritmy razmeshcheniya [Methods, models and placement algorithms]. Rostov-on-Donu: Izd-vo YuFU, 2015.
6. *Jason Cong, Joseph R., Shinnerl u Min Xie (UCLA Computer Science), Tim Kong (Magma Design Automation) u Xin Yuan (IBM Corporation, Microelectronics Division).* Large-Scale Circuit Placement, *ACM Transactions on Design Automation of Electronic Systems*. April 2005, Vol. 10, No. 2.
7. *Poli R.* Analysis of the publications on the applications of particle swarm optimization, *Journal of Artificial Evolution and Applications*, Article ID 685175, 2008.
8. *Karpenko A.P.* Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature: textbook]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2014.
9. *Wang X.* Hybrid nature-inspired computation method for optimization: Doctoral Dissertation. Helsinki University of Technology, TKK Dissertations, Espoo 2009, 161 p.
10. *Clerc M.* Particle Swarm Optimization. ISTE, London, UK, 2006.

11. Kennedy J., Eberhart R.C. Particle swarm optimization, *In Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.
12. Lebedev B.K., Lebedev O.B., Lebedev V.B. Gibrinnyy metod stokhasticheskoy optimizatsii na osnove integratsii modeley evolyutsii i roevogo (staynogo) povedeniya zhivotnykh v affinnykh prostranstvakh poiska [A hybrid method of stochastic optimization based on the integration of models of evolution and swarm (packing) behavior of animals in affine search spaces], *Sb. trudov Shestnadsatoy natsional'noy konferentsii po iskusstvennomu intellektu s mezhdunarodnym uchastiem KII-2018* [Proceedings of the Sixteenth National Conference on Artificial Intelligence with International Participation KII-2018]. In 2 vol. Vol. 2. Moscow: FGP ITAR-TASS filial RKP, 2018, pp. 148-156.
13. Lebedev B.K., Lebedev O.B., Lebedeva E.O., Nagabedian A.A. Gibrinnyy roevoy algoritm global'noy optimizatsii v affinnom prostranstve poiska [Hybrid swarm algorithm of global optimization in affine search space], *Elektronnyy zhurnal "Programmnye produkty, sistemy i algoritmy"* [Electronic journal "Software products, systems and algorithms"], 2019, No. 1.
14. Lučić P., Teodorović D. Computing with Bees: Attacking Complex Transportation Engineering Problems, *International Journal on Artificial Intelligence Tools*, 2003, No. 12, pp. 375-394.
15. Quijano N., Passino K.M. Honey Bee Social Foraging Algorithms for Resource Allocation: Theory and Application. Columbus: Publishing house of the Ohio State University, 2007, 39 p.
16. Lebedev B.K., Lebedev V.B. Razmeshchenie na osnove metoda pchelinoy kolonii [Placement based on the bee colony method], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 12, pp. 12-18.
17. Cong J., Romesis M., and Xie M. Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms, *Proc. of the International Symposium on Physical Design, Monterey, CA, April 2004*, pp. 88-94.
18. MCNC. Electronic and Information Technologies (Online). Available: www.mcnc.org.
19. Wang M., Yang X. and Sarrafzadeh M. Dragon2000: Standard-cell Placement Tool for Large Industry Circuits, *ICCAD 2000*, pp. 260-263.
20. Caldwell A.E., Kahng A.B. and Markov I.L. Can Recursive Bisection Alone Produce Routable Placements?, *DAC 2000*, pp. 477-482.
21. Chan T., Cong J., Kong T., Shinnerl J. u Sze K. An enhanced multilevel algorithm for circuit placement, *In Proceedings of the IEEE International Conference on Computer Aided Design (San Jose, Calif.)*. IEEE Computer Society Press, Los Alamitos, Calif. 2003.
22. Yang X., Choi B.-K. and Sarrafzadeh M. Routability-driven white space allocation for fixed-die standard-cell placement, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2003, 22 (4), pp. 410-419.
23. CADENCE DESIGN SYSTEMS, INC. QPlace version 5.1.55, compiled on 10/25/1999. Envisia ultra-placer reference.

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

Лебедев Борис Константинович – Южный федеральный университет; e-mail: lebedev.ob@mail.ru; г. Таганрог, Россия; тел.: 89085135512; кафедра систем автоматизированного проектирования; д.т.н.; профессор.

Лебедев Олег Борисович – e-mail: lebedev.ob@mail.ru; тел.: 89085135512; д.т.н.; доцент.

Ганжур Марина Александровна – Донской государственный технический университет; e-mail: mganzhur@yandex.ru; г. Ростов-на-Дону, Россия; тел.: +79081819111; кафедра вычислительных систем и информационной безопасности.

Lebedev Boris Konstantinovich – Southern Federal University; e-mail: lebedev.ob@mail.ru; Taganrog, Russia; phone: +79085135512; the department of computer aided design; dr. of eng. sc.; professor.

Lebedev Oleg Borisovich – e-mail: lebedev.ob@mail.ru; phone: +79085135512; dr. of eng. sc.; associate professor.

Ganzhur Marina Aleksandrovna – Don State Technical University; e-mail: mganzhur@yandex.ru; Rostov-on-Don, Russia; phone: +79081819111; the department of computer systems and information security.