

---

## Решение задачи покрытия на основе интеграции моделей эволюции и стайного поведения животных в аффинных пространствах поиска

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

*Южный федеральный университет, Таганрог*

**Аннотация:** В работе представлена композитная архитектура многоагентной системы бионического поиска на основе роевого интеллекта и генетической эволюции для решения задачи покрытия множествами. Описывается модифицированная парадигма роя частиц, обеспечивающая, в отличие от канонического метода, возможность использования в аффинном пространстве позиций с целочисленными значениями параметров. Рассмотрены механизмы перемещения частиц в аффинном пространстве для уменьшения веса аффинных связей. Разработанные структуры позиций (хромосом) ориентированы на интеграцию роевого интеллекта и генетической эволюции. Проведенные теоретические разработки были проверены на тестовых примерах (бенчмарках) и на основе полученных экспериментально результатах построена временная зависимость работы алгоритма. Она составляет  $O(n^2)$ - $O(n^3)$ .

**Ключевые слова:** покрытие множествами, рой частиц, генетическая эволюция, аффинное пространство, целочисленные параметры, интеграция.

### Введение

В работе объектами исследования являются системы бионического поиска на основе роевого интеллекта и генетической эволюции для решения задачи покрытия множествами.

К задаче о покрытии можно свести многие существующие задачи дискретной оптимизации: задачи стандартизации, упаковки и разбиения множества [1], задача о наибольшей клике [2], задача минимизации полинома от булевых переменных [3,4] и др.

Начальными данными задачи о покрытии множествами являются конечное множество  $X$ , а также семейство его подмножеств  $F = \{X_i | i = 1, 2, \dots, n\}$  таких, что  $X_i \subseteq X$  и  $\bigcup_i X_i = X$ . Задача о минимальном покрытии множествами заключается в отыскании набора  $P \subseteq F$  с минимальным числом подмножеств  $X_i \in P$  и  $X_i \in F$  такого, чтобы  $\bigcup_i X_i = X$ .

Для решения задачи покрытия могут применяться различные методы и алгоритмы. В основном, это хорошо известные алгоритмы, содержащие разного рода эвристики, применение которых достаточно часто дает возможность получить приемлемый результат за полиномиальное время [5,6]. Следует отметить, что вычислительная сложность рассматриваемых задач все время возрастает. Это автоматически приводит к увеличению требований к получаемым в процессе решения задач покрытия результатам. Поэтому можно сказать, что на сегодняшний день, разработка новых алгоритмов, обладающих большей эффективностью, является актуальной и востребованной задачей. В литературе приведено множество методов и алгоритмов для решения задачи покрытия (последовательных, итерационных и т.д.). Проанализировав эти существующие методы можно утверждать, что разработка нового эффективного алгоритма покрытия, который будет удовлетворять требованиям сегодняшнего дня, возможна лишь в случае использования новых технологий, подходов и методов.

В результате постоянного поиска для выявления более эффективных способов решения данной задачи, стало применение бионических методов и алгоритмов интеллектуальной оптимизации, базирующихся на моделировании коллективного интеллекта [7]. Анализ методов решения сложных прикладных задач говорит о том, что использование, какого либо одного алгоритма оптимизации (как классического, так и популяционного) далеко не всегда однозначно позволяет получить оптимальное решение. В связи с этим, в настоящее время одним из используемых путей позволяющих улучшить эффективность методов для решения задач глобального поиска является разработка гибридных алгоритмов [8]. Гибридные алгоритмы обладают одним важнейшим качеством – недостатки одного алгоритма могут быть компенсированы достоинствами другого алгоритма.

---

Интеграция метаэвристик популяционных алгоритмов обеспечивает более широкий обзор области поиска и более большую вероятность локализации глобального экстремума задачи [9]. Предлагается композитная архитектура многоагентной системы бионического поиска на основе роевого интеллекта и генетической эволюции [10].

Самый простой способ к гибридизации выглядит следующим образом. В начале для нахождения решения применяется генетический алгоритм. Далее на основе полученных результатов (популяции полученной на конечной генерации генетического поиска) формируются данные (популяция) для роевого алгоритма. Построение популяции осуществляется включением в ее состав наилучших хромосом, но отличных друг от друга. Если существует такая необходимость, то в сформированную популяцию добавляют новых индивидуумов. Далее поиск решения проводится роевым алгоритмом.

Второй способ гибридизации выглядит так. Генетический алгоритм выполняет поиск решения. При этом метод роя частиц используется внутри генетического алгоритма и применяется в виде генетических операторов для получения новых решений. Такой подход дает возможность синтеза новых хромосом на каждой генерации генетического алгоритма двумя путями:

- при помощи генетических операторов кроссовера и мутации;
- при помощи операторов направленной мутации (ОНМ) методами роя частиц.

Метод роя частиц (Particle Swarm Optimization, PSO) относится к классу методов стохастической оптимизации имеющих нечто общее с эволюционными алгоритмами.

Моделирование этого метода заключается в построении модели стайного поведения животных, а не эволюции в целом [9-11]. Основное отличие Particle Swarm Optimization, PSO, по сравнению с популяционными

---

методами состоит в том, что он использует одну статическую популяцию. В этой популяции происходит постепенное улучшение хромосом. Это достигается тем, что появляется информация о пространстве поиска. Данный метод представляет собой вид направленной мутации (*directed mutation*). Решения в PSO мутируют в направлении наилучших найденных решений. Частицы никогда не умирают (т.к. нет селекции).

В работе предлагается подход к разработке модифицированной парадигмы роя частиц, которая сможет выполнять действия с хромосомами с целочисленными значениями параметров в генетическом алгоритме и в алгоритме на основе роя частиц.

Отличительной чертой такой парадигмы является использование такой структуры данных, которая применяет описание в виде хромосомы для решения задачи.

В сравнении с алгоритмами, использующими эволюционную стратегию можно провести следующую аналогию. Рой частиц представляется как популяция, а частица роя - индивидуум. Такое представление дает возможность при поиске решения сформировать гибридный алгоритм, основой которого является интеграция метода роя частиц и генетического поиска [8]. В таком методе структура данных, которая описывает решение задачи, является ключевым связывающим звеном [9]. В генетических алгоритмах генотип решения представляется в виде хромосомы. При решении комбинаторных задач гены в хромосомах имеют, как правило, целочисленные значения [7].

В заданном пространстве поиска, позиция частицы (т.е. ее положение) является эквивалентом генотипа в эволюционных алгоритмах. Т.е. если необходимо определить частицу в некотором пространстве решений, то ее состояние описывается при помощи хромосомы. Отметим, что число параметров, которые необходимы для описания частицы равно числу генов в

---

хромосоме. Значение каждого гена фиксируется на соответствующей оси пространства решений. При этом возможно возникновение различных требований к общей структуре хромосомы, значениям ее генов, операторам задающим перемещения частиц и пространству поиска.

Анализ существующих методов и алгоритмов решения комбинаторных задач показал, что в качестве моделей для представления решений используют широкий диапазон графовых структур (маршрут, дерево, двудольный граф, паросочетание, внутреннеустойчивое множество и т.д.) [12-15]. Такое представление является наиболее эффективным способом нахождения «наилучших» результатов оптимизационных задач, допускающих интерпретацию решений в виде различного рода графовых структур.

Каноническая парадигма роя частиц предусматривает использование вещественных значений параметров в многомерных, вещественных, метрических пространствах. Однако в большинстве генетических алгоритмов гены в хромосомах имеют дискретные целочисленные значения. В свою очередь хромосомы являются некоторыми интерпретациями решений, которые трансформируются в решения путем декодирования хромосом. Это не позволяет напрямую использовать каноническую парадигму роя частиц (например, задача направленной мутации одного дерева в направлении другого, с формальной точки зрения, является весьма нетривиальной).

В связи с этим актуальной является разработка модернизированной структуры пространства поиска, структуры данных для представления решений и позиций, модернизированных механизмов перемещения частиц в пространстве поиска. Исследования показали, что структуры данных для представления позиций и механизмы перемещения частиц в пространстве поиска связаны, но зависят от структуры данных для представления интерпретаций решений.

---

В работе предлагается подход к разработке модифицированной парадигмы роя частиц, позволяющей применять возможность совместного использования хромосом с дискретными целочисленными значениями параметров в генетическом алгоритме и в алгоритме на основе роя частиц.

## 1. Аффинное пространство поиска

Пусть имеется линейное векторное пространство (ЛВП), элементами которого являются  $n$ -мерные точки (позиции). Каждым любым двум точкам  $p$  и  $q$  этого пространства однозначным образом сопоставим единственную упорядоченную пару этих точек, которую в дальнейшем будем называть геометрическим вектором  $p, q \in V(p, q)$  – геометрический вектор (упорядоченная пара). Совокупность всех точек ЛВП, пополненная геометрическими векторами, называют точечно-векторным или аффинным пространством. Аффинное пространство является  $n$ -мерным, если соответствующее ЛВП так же является  $n$ - мерным.

Аффинно-релаксационная модель (АРМ) роя частиц – это граф вершины которого соответствуют позициям роя частиц, а дуги соответствуют аффинным связям между позициями (точками) в аффинном пространстве. Аффинность – мера близости двух агентов (частиц). На каждой итерации каждый агент  $p_k$  переходит в аффинном пространстве в новое состояние (позицию), при котором вес аффинной связи между агентом  $p_i$  и базовым (лучшим) агентом  $p^*$  уменьшается. Переход агента  $p_i$  в новую позицию  $x_i(t+1)$  из  $x_i(t)$  осуществляется с помощью релаксационной процедуры.

Специальная релаксационная процедура перехода зависит от вида структуры данных (хромосомы): вектор, матрица, дерево и их совокупности, являющейся интерпретацией решений.

Позиции лучших частиц с точки зрения целевой функции объявляются «центром притяжения». Векторы перемещения всех частиц в аффинном пространстве устремляются к этим центрам.

Переход возможен с учетом степени близости к одной базовой позиции либо к группе соседних позиций с учетом вероятности перехода в новое состояние.

Обозначим каждую хромосому, описывающую  $i$ -е решение популяции как  $H_i(t) = \{g_{il} | l=1, 2, \dots, n\}$ . Такое обозначение необходимо для того, чтобы в дальнейшем не возникло некорректного описания популяции (роя). Отметим, что каждая  $H_i(t)$  содержит в себе представленную структуру. В нашем случае позиция  $x_i(t)$  соответствует решению, задаваемому хромосомой  $H_i(t)$ , т.е.  $x_i(t) = H_i(t) = \{g_{il} | l=1, 2, \dots, n\}$ . Аналогично  $x_i^*(t) = H_i^*(t) = \{g_{il}^* | l=1, 2, \dots, n\}$ ,  $x^*(t) = H^*(t) = \{g_{il}^* | l=1, 2, \dots, n\}$ . Число осей в пространстве решений равно числу  $n$  генов в хромосомах  $H_i(t)$ ,  $H_i^*(t)$ ,  $H_i^*(t)$ . Точками отсчета на каждой оси  $l$  являются целочисленные значения генов.

## 2. Поиск методом роя частиц

По определению, эвристические алгоритмы, относящиеся к классу алгоритмов роевого интеллекта, имеют многомерное пространство поиска, которое заполняется, роем частиц [14-16]. В процессе поиска этим методом каждая частица из роя стремится попасть в новую позицию. Новая позиция в канонической парадигме роя частиц определяется как:

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

где  $v_i(t+1)$  скорость перемещения частицы из позиции  $x_i(t)$  в позицию  $x_i(t+1)$ . Начальное состояние определяется, как  $x_i(0)$ ,  $v_i(0)$  [14-16].

Представленная формула описывается в форме вектора. Для отдельного измерения  $j$  пространства поиска формула примет вид:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (2)$$

где  $x_{ij}(t)$ - позиция частицы  $p_i$  в измерении  $j$ ,  $v_{ij}(t+1)$ - скорость частицы  $p_i$  в измерении  $j$ .

Введем обозначения:

-  $x_i(t)$ - текущая позиция частицы,  $f_i(t)$  значение целевой функции частицы  $p_i$  в позиции  $x_i(t)$ ;

-  $x_i^*(t)$ - лучшая позиция частицы  $p_i$ , которую она посещала с начала первой итерации, а  $f_i^*(t)$  - значение целевой функции частицы  $p_i$  в этой позиции (лучшее значение с момента старта жизненного цикла частицы  $p_i$ );

-  $x^*(t)$ - позиция частицы роя с лучшим значением целевой функции  $f^*(t)$  среди частиц роя в момент времени  $t$ .

Тогда скорость частицы  $p_i$  на шаге  $(t+1)$  в измерении  $j$  вычисляется как:

$$v_{ij}(t+1) = w \cdot v_{ij}(t) + k_1 \cdot \text{rnd}(0,1) \cdot (x_{ij}^*(t) - x_{ij}(t)) + k_2 \cdot \text{rnd}(0,1) \cdot (x_j^*(t) - x_{ij}(t)), \quad (3)$$

где  $\text{rnd}(0,1)$  – случайное число на интервале  $(0,1)$ ,  $(w, k_1, k_2)$  – некоторые коэффициенты.

Скорость  $v_i(t+1)$  представляется, как инструмент позволяющий изменить решение.

В данном алгоритме скорость  $v_i(t+1)$  не может быть представленной в виде аналитического выражения с вещественными значениями переменных. Этим рассматриваемый алгоритм отличается от канонического метода роя частиц. Здесь в роли позиции выступает хромосома. При этом в процессе определения местоположения частицы в области поиска решений, необходимо, чтобы количество параметров, на основе которых происходит определение этого места, было равным количеству генов в хромосоме. Значение каждого гена откладывается на соответствующей оси пространства решений. При этом дополнительно рассматриваются возникающие новые требования к структуре хромосомы и значениям генов. Значения генов должны быть дискретными и независимыми друг от друга. В качестве аналога скорости  $v_i(t+1)$  выступает оператор направленной мутации (ОНМ),

суть которого заключается в изменения целочисленных значений генов в хромосоме  $H_i(t)$ . Перемещение частицы  $p_i$  в новую позицию означает переход от хромосомы  $H_i(t)$  к новой –  $H_i(t+1)$  с новыми целочисленными значениями генов  $g_{il}$ , полученными после применения ОНМ.

По аналогии с каноническим методом роя частиц позицию  $x^*_i(t)$  будем называть когнитивным центром притяжения, а позицию  $x^*(t)$  – 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.

Как уже указывалось выше, позиции задаются хромосомами. Позициям  $x(t)$ ,  $x_i(t)$ ,  $x^*_i(t)$ ,  $x^c_i(t)$  соответствуют хромосомы  $H^*(t)=\{g^*_{il}(t)|l=1,2,\dots,n_l\}$ ,  $H_i(t)=\{g_{il}(t)|l=1,2,\dots,n_l\}$ ,  $H^*_i(t)=\{g^*_{il}(t)|l=1,2,\dots,n_l\}$ ,  $H^c_i(t)=\{g^c_{il}(t)|l=1,2,\dots,n_l\}$ .

В качестве оценки степени близости между двумя позициями  $x_i(t)$  и  $x_z(t)$  будем использовать величину  $S_{iz}$  расстояния (вес аффинной связи) между хромосомами  $H_i(t)$  и  $H_z(t)$ .

Целью перемещения хромосомы  $H_i(t)$  в направлении хромосомы  $H_z(t)$  является сокращении расстояния между ними.

Для учета одновременного тяготения частицы  $p_i$  к лучшей к позиции  $x^*(t)$ , среди частиц роя в момент времени  $t$ , и к лучшей позиции  $x^*_i(t)$  частицы  $p_i$ , которую она посещала с начала первой итерации, формируется *виртуальный центр* (позиция) притяжения  $x^c_i(t)$  частицы  $p_i$ .

Синтез *виртуальной позиции*  $x^c_i(t)$  происходит при использовании действий *виртуального* перехода из позиции  $x^*_i(t)$  в *виртуальную позицию*  $x^c_i(t)$  в направлении к позиции  $x^*(t)$ . После того, как определен центр притяжения  $x^c_i(t)$  частица  $p_i$  движется в направлении *виртуальной позиции*  $x^c_i(t)$  из позиции  $x_i(t)$  в позицию  $x_i(t+1)$ .

После перемещения частицы  $p_i$  в новую позицию  $x_i(t+1)$  виртуальная позиция  $x_i^c(t)$  исключается.

В качестве локальной цели, получаемой, в процессе передвижения частицы  $p_i$  выступает ее попадание в позицию с наилучшим показателем критерия качества. В качестве глобальной цели получаемой, в процессе передвижения частицы  $p_i$  является формирование оптимального решения задачи.

### 3. Постановка задачи покрытия

Существующие и описанные в литературных источниках выполняющие покрытие схем библиотечными наборами ячеек алгоритмы [1-5], в качестве критериев, в основном, используют следующие: общее количество элементов из которых состоит покрывающий набор ячеек; количество типов применяемых ячеек; общее количество ячеек требуемых для реализации схемы; количество межъячеечных соединений; суммарную стоимость ячеек, покрывающих схему.

Пусть задано множество  $E=\{e_i|i=1,2,\dots,n\}$  типов элементов, входящих в состав покрываемой функциональной схемы. Количественный состав схемы по типам элементов описывается вектором  $B=\{b_i|i=1,2,\dots,n\}$ , где  $b_i$  – число элементов типа  $e_i$ , входящих в состав схемы.

Пусть задан набор ячеек  $T=\{t_j|j=1,2,\dots,m\}$ . Каждая ячейка включает свой набор элементов из  $E$ . Элементы внутри ячейки между собой не соединены. Количественный состав ячеек описывается с помощью матрицы  $A=||a_{ij}||_{n \times m}$ , где  $a_{ij}$  – число элементов типа  $e_i$  в ячейке типа  $t_j$ . С помощью вектора  $C=\{c_j|j=1,2,\dots,m\}$  задается для каждой ячейки  $t_j$  её стоимость  $c_j$ .

Покрытие схемы библиотечными наборами ячеек будет выполнено, если в покрываемой схеме каждый элемент сможет быть реализован элементами, входящими в состав выбранных библиотечных ячеек.

---

Для определения количества ячеек типа  $t_j$ , из которых состоит библиотечный покрывающий набор, введём целочисленную переменную  $x_j$ .

Задача покрытия формулируется следующим образом:

$$\text{минимизировать } F = \sum_{j=1}^m x_j \cdot c_j$$

$$\text{при ограничениях } \sum_{j=1}^m a_{ij} \cdot x_j \geq b_i, i = 1, 2, \dots, n; \quad (4)$$

$$x_j \geq 0, j = 1, 2, \dots, m;$$

$$a_{ij} \geq 0.$$

Таким образом, решением задачи является набор значений параметров  $x_j, j=1, 2, \dots, m$ , при которых функция  $F$  (суммарная стоимость ячеек покрывающего набора) имеет минимальное значение.

Если за критерий  $c_j$  принять все количество элементов, из которых состоят ячейки  $t_j$ , т.е.  $c_j = \sum_{i=1}^n a_{ij}$ , то целевая функция  $F$  учитывает общее количество элементов, из которых состоит покрывающий набор ячеек.

#### 4. Формирование пространства решений

Для более детального представления, процесс формирования области решений рассмотрим на примере [11-13].

Пусть покрываемая схема составлена из элементов 3-х типов:  $E = \{e_1, e_2, e_3\}$ . Количественный состав схемы описывается вектором  $V = \{30, 10, 21\}$ . Набор покрывающих ячеек включает 5 типов:  $T = \{t_1, t_2, t_3, t_4, t_5\}$ . Матрица  $A$ , описывающая количественный состав ячеек, имеет вид, представленный на рис.1.

Введём матрицу  $B$ , задающую граничные требования по количественному составу элементов, покрываемых ячейками каждого типа.  $B = \|b_{ij}\|_{n \times m}$ , где  $b_{ij}$  – минимальное число элементов типа  $e_i$ , которое обязательно должно быть покрыто ячейками типа  $t_j$ ,  $b_{ij} \geq 0$ ,  $b_{ij}$  – целое число.

При этом для реализации полного покрытия всех элементов, в соответствии с требованиями матрицы  $B$ , необходимо выполнение ограничений:

$$\sum_{j=1}^m b_{ij} = \delta_i, i = 1, 2, \dots, n. \quad (5)$$

$A =$

Рис.1 - Матрица  $A$ , описывающая количественный состав ячеек

Для выше примера одним вариантов матрицы  $B$  представленный на рис. 2.

Типы элементов	Типы покрывающих ячеек				
	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$
$e_1$	2	1	2	3	2
$e_2$	3	2	2	1	2
$e_3$	1	2	3	1	2

представленного из ВОЗМОЖНЫХ имеет вид,

$B =$

Типы элементов	Типы покрывающих ячеек					$\delta_i = \sum b_{ij}$
	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	
$e_1$	8	4	4	9	5	$\delta_1=30$
$e_2$	3	2	3	1	1	$\delta_2=10$
$e_3$	5	2	7	6	1	$\delta_3=21$

Рис. 2 - Символьное представление решения - матрица  $B$

Матрица  $B$  однозначно соответствует покрывающему набору ячеек, который определяется следующим образом.

Вначале строится матрица  $D = \|d_{ij}\|_{n \times m}$ ; элемент матрицы  $d_{ij} = \lceil b_{ij}/a_{ij} \rceil$  это потолок числа  $b_{ij}/a_{ij}$  - наименьшее целое число, не меньшее  $b_{ij}/a_{ij}$  и фактически равное минимальному числу ячеек типа  $t_j$ , необходимых для покрытия  $b_{ij}$  элементов типа  $e_i$ .

Затем в пределах каждого  $j$ -го столбца матрицы  $D$  отыскивается максимальное число  $d_{jmax}$ ,  $(\forall i)[d_{jmax} \geq d_{ij}]$ , которое является минимальным числом  $x_j = d_{jmax}$  ячеек типа  $t_j$ , обеспечивающих покрытие  $\delta_1$  элементов типа  $e_1$ ;  $\delta_2$  элементов типа  $e_2, \dots$ ;  $\delta_n$  элементов типа  $e_n$  в соответствии с

требованиями матрицы  $B$ . Причём, кроме элементов типа  $e_i$ , для которых  $d_{ij}=d_{jmax}$ , остальные типы элементов будут покрыты с избытком.

Для нашего случая матрица  $D$  и покрывающий набор  $X$  имеет вид, представленный на рис.3.

$D =$

Типы элементов	Типы покрывающих ячеек				
	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$
$e_1$	$\lceil 8/2 \rceil = 4$	$\lceil 4/1 \rceil = 4$	$\lceil 4/2 \rceil = 2$	$\lceil 9/3 \rceil = 3$	$\lceil 5/2 \rceil = 3$
$e_2$	$\lceil 3/3 \rceil = 1$	$\lceil 2/2 \rceil = 1$	$\lceil 3/2 \rceil = 2$	$\lceil 1/1 \rceil = 1$	$\lceil 1/2 \rceil = 1$
$e_3$	$\lceil 5/1 \rceil = 5$	$\lceil 2/2 \rceil = 1$	$\lceil 7/3 \rceil = 3$	$\lceil 6/1 \rceil = 6$	$\lceil 1/2 \rceil = 1$
$d_{jmax}$	5	4	3	6	3

$X =$

Рис. 3 - Матрица  $D$  и покрывающий набор  $X$

Общее число элементов типа  $e_i$ , входящее в состав покрывающего набора ячеек, определится как:

$$N_i = \sum_{j=1}^m a_{ij} \cdot x_j$$

Общее число  $N_{я}$  ячеек в покрывающем наборе определится как:

$$N_{я} = \sum_{j=1}^m x_j$$

Для нашего случая:  $N_{я}=5+4+3+6+3=21$ ;  $N_1=5 \cdot 2+4 \cdot 1+3 \cdot 2+6 \cdot 3+3 \cdot 2=44$ ;  $N_2=5 \cdot 3+4 \cdot 2+3 \cdot 2+6 \cdot 1+3 \cdot 2=41$ ;  $N_3=5 \cdot 1+4 \cdot 2+3 \cdot 3+6 \cdot 1+3 \cdot 2=34$ . Общее число элементов входящих в покрывающий набор  $N_3=N_1+N_2+N_3=119$ . Итак, матрице  $B$ , значения элементов которой удовлетворяет ограничениям (5), соответствует одно решение.

При комбинировании значениями  $b_{ij}$ , входящими в матрицу  $B$ , возможно получение других, отличных друг от друга решений. Все эти решения будут удовлетворять ограничениям (5). В данной работе область решений описывается множеством  $R$  матриц  $B$ .

Найти решение – это значит отыскать матрицу  $B^* \in R$ . Т.е. найти в матрице  $B^*$  совокупности таких значений элементов  $b_{ij}^*$ , которые смогут оптимизировать целевую функцию.

## 5. Структура хромосом и принципы декодирования

Как уже указывалось выше, в качестве решения рассматривается матрица  $P$ . В процессе формирования структуры хромосомы использовалось одно ключевое требование: гены в одних и тех же локусах хромосом должны быть гомологичными. В хромосоме с такой структурой будет гораздо проще использование генетических операторов кроссинговера и мутации и оператора направленной мутации в рое частиц. Представим матрицу  $B = \|b_{ij}\|_{n \times m}$  в виде набора строк  $B_i$ .  $B = \{B_i | i=1, 2, 3, \dots, n\}$ , где  $B_i = \{b_{ij} | j=1, 2, \dots, m\}$ . Каждой строке  $B_i$  матрицы  $B$  соответствует хромосома  $H_i$  и наоборот. Каждая  $H_i$  представляет собой совокупность  $(m-1)$  генов, значения которых могут изменяться в пределах диапазона, определяемого параметром  $\delta_i$ :

$$0 < g_{il} < b_i, \quad g_{il} \in H_i \quad (6)$$

В свою очередь матрица  $B$  представляется совокупностью  $n$  хромосом  $H_i$ .

Рассмотрим механизмы кодирования и декодирования на примере одной строки.

Пусть имеется список  $B_i = \langle b_{ij} | j=1, 2, \dots, m \rangle$  с фиксированной суммой значений элементов  $\delta_i = \sum_j (b_{ij})$ .

Подобные списки лежат в основе интерпретаций решений в задачах покрытия [14], распределения ресурсов [15,16] и др.

Хромосома  $H_i = \{g_{il} | l=1, 2, \dots, (n-1)\}$  представляет собой совокупность  $(n-1)$  генов  $g_l$ , значения которых могут изменяться в пределах диапазона, определяемого параметром  $\delta_i$ :

$$g_{il} \in H_i, \quad 0 \leq g_{il} \leq \delta_i.$$

Рассмотрим подробнее переход от хромосомы  $H_i$  к списку. Вначале происходит упорядочивание генов в порядке увеличения их значений, т.е. если  $g_{il} \in H_i$  и  $g_{il+1} \in H_i$ , то  $g_{il+1} \geq g_{il}$ . Значения  $g_{il}$  в заданных пределах от 0 до  $\delta_i$ , выступают в роли координат опорных точек на отрезке длиной  $\delta_i$  (от 0 до  $\delta_i$ ), разделяющих отрезок на интервалы. Длина интервала между двумя соседними опорными точками и есть величина соответствующего элемента  $b_{ij}$  списка  $B_i$ , при этом сумма значений элементов списка  $B_i$  равна  $\delta_i$ .

Например. Пусть имеется хромосома  $H_i = \{8, 12, 16, 25\}$ ,  $b_i = 30$ ,  $(m-1) = 4$ . На отрезке длиной  $b_i = 30$  наносятся опорные точки с координатами  $\langle 0, 8, 12, 16, 25, 30 \rangle$ , разбивающими отрезок на интервалы длиной  $\langle 8, 4, 4, 9, 5 \rangle$ . Эти значения являются значениями списка  $B_i$ .  $B_i = \langle 8, 4, 4, 9, 5 \rangle$ ,  $b_i = 30$ .

## 6. Перемещение частиц в пространстве поиска

Пусть имеется рой частиц  $P = \{p_k | k = 1, 2, \dots, K\}$ . Каждая частица  $p_k$  на шаге  $t$  размещена в позиции  $X_k(t)$ . Поскольку матрица  $B$  включает  $n$  строк, соответствующая, частице  $p_k$ , определяется набором из  $n$  хромосом, соответствующим  $n$  строкам матрицы  $B$ ,  $X_k(t) = \{H_{ki}(t) | i = 1, 2, \dots, n\}$ .

Хромосома  $H_{ki}(t) = \{g_{kil} | l = 1, 2, \dots, (n-1)\}$  представляет собой совокупность  $(n-1)$  генов  $g_{kil}$ , значения которых могут изменяться в пределах диапазона, определяемого параметром  $\delta_i$ :

$$g_{kil} \in H_{ki}, 0 \leq g_{kil} \leq \delta_{ki}.$$

Пространство поиска включает число осей равное числу генов в хромосоме  $H_{ki}(t)$ . Каждому гену в хромосоме  $H_{ki}(t)$  соответствует ось (номер оси). Точками отсчета на каждой оси  $X_i$  являются целые числа в интервале от 0 до  $\delta_{ki}$ .

Перемещение частицы  $p_k$  из позиции  $X_k(t)$  в позицию  $X_k(t+1)$  под воздействием притяжения к позиции  $X_z(t)$  осуществляется путем применения

оператора направленной мутации (ОНМ) к хромосомам позиции  $X_k(t)$  следующим образом.

В качестве оценки степени близости между двумя хромосомами  $H_{ki}(t) \in X_k(t)$  и  $H_{zi}(t) \in X_z(t)$  будем использовать величину  $S_{kzi}$  расстояния между хромосомами  $H_{ki}(t)$  и  $H_{zi}(t)$ :

$$S_{kzi} = \sum_{l=1}^{m-1} |g_{kil} - g_{zil}|. \quad (7)$$

В качестве оценки степени близости между двумя позициями  $X_k(t)$  и  $X_z(t)$  используется величина расстояния между  $X_k(t)$  и  $X_z(t)$ .

$$Q_{kz} = \sum_{i=1}^n S_{kzi}. \quad (8)$$

Суть *процедуры перемещения*, реализуемой ОНМ, заключается в изменении разности между значениями каждой пары генов ( $g_{kil}$ ,  $g_{zil}$ ) двух позиций  $X_k(t)$  и  $X_z(t)$ :  $i=1,2,\dots,n$ ;  $l=1,2,\dots,(m-1)$ .

Это достигается путем просмотра локусов хромосом  $H_{ki}(t)$  и  $H_{zi}(t)$  (с первого последовательным образом), и проводится сравнение соответствующих этим локусам генов. Если в ходе последовательного просмотра локусов в текущем локусе  $l$  выпадает с вероятностью  $P$  событие «мутация», то ген  $g_{kil} \in H_{ki}(t)$  мутирует.

Пусть  $R_{kzi}(t)$ —число локусов в хромосомах  $H_{ki}(t)$  и  $H_{zi}(t)$ , в которых значения генов  $g_{kil}(t) \in H_{ki}(t)$  и  $g_{zil}(t) \in H_{zi}(t)$  не совпадают.

Вероятность мутации  $\pi$  зависит от числа  $R_{kzi}(t)$  несовпадений между позициями, и определяется следующим образом:

$$\pi = \alpha \cdot R_{kzi}(t) / (m-1), \quad (9)$$

где  $\alpha$  коэффициент,  $(m-1)$  – длина хромосомы. Таким образом, чем больше число  $R_{kzi}(t)$  несовпадающих генов между хромосомами  $H_{ki}(t)$  и  $H_{zi}(t)$ , тем больше вероятность, что значение  $g_{kil}(t) \in H_{ki}(t)$  будет изменено.

Пусть имеется некое вероятностное событие, в результате которого возможно наступление двух вариантов событий  $1$  и  $0$ . Вероятности наступления этих событий обозначим соответственно через  $\pi$  и  $(1-\pi)$ . Такое вероятностное событие назовем *простой лотереей*  $L(1, \pi, 0)$ . Другими словами с вероятностью  $\pi$  лотерея  $L(1, \pi, 0)=1$ , а с вероятностью  $(1-\pi)$  лотерея  $L(1, \pi, 0)=0$ . Значения генов новой позиции частицы  $p_i$  в результате мутации определяются как:

$$\begin{aligned} g_{kil}(t+1) &= g_{kil}(t), \text{ если } (g_{kil}(t) = g_{zil}(t)); \\ g_{kil}(t+1) &= g_{kil}(t) + L(1, \pi, 0), \text{ если } (g_{kil} < g_{zil}(t)); \\ g_{kil}(t+1) &= g_{kil}(t) - L(1, \pi, 0), \text{ если } (g_{kil} > g_{zil}(t)). \end{aligned} \quad (10)$$

*Пример.* Пусть  $b_{ki}=23$ ,  $m=6$ ,  $H_{ki}(t)=\langle 1, 2, 4, 8, 8 \rangle$ ,  $H_{ki}(t) (t)=\langle 1, 1, 2, 4, 0, 15 \rangle$ ,  
 $H_{zi}(t) = \langle 3, 3, 4, 6, 7 \rangle$ ,  $P_{zi}(t) = \langle 3, 0, 1, 2, 1, 16 \rangle$ ,  
 $S_{kzi}(t) = 6$ .

Пусть с некоторой вероятностью в  $H_{ki}(t)$  мутировали гены в 1 и 4 локусах. В результате направленной мутации

$$\begin{aligned} H_{ki}(t+1) &= \langle 2, 2, 4, 7, 8 \rangle, P_{ki}(t+1) = \langle 2, 0, 2, 3, 1, 15 \rangle, \\ S_{kzi}(t+1) &= 4. \end{aligned}$$

Перемещение частиц в пространстве поиска может быть описано в виде последовательности следующих шагов.

1. Создать рой частиц  $P = \{p_k | k=1, 2, \dots, K\}$ . Задать число итераций  $T$ .
2. На основе исходной постановки задачи покрытия сформировать множество матриц  $B = \{B_k | k=1, 2, \dots, K\}$ , каждая из которых является символьным представлением решения задачи покрытия,  $B_k = ||b_{kij}||_{n \times m}$ .
3. Сформировать  $(m-1)$  – мерное пространство поиска. Установить:  $t=1$ .
4. Сформировать  $K$  позиций  $X_k(t)$  для множества частиц  $P = \{p_k | k=1, 2, \dots, K\}$  на базе множества матриц  $B = \{B_k | k=1, 2, \dots, K\}$ .
5. Вычислить значение критерия  $f_k(t)$  для каждой позиции  $X_k(t)$ .
6. Найти лучшую позицию  $X^*(t)$  роя на шаге  $t$ .

7. Для каждой частицы  $p_k$ , определить лучшую позицию  $X_k^*(t)$ , которую она посещала с начала первой итерации.
8. Для каждой частицы  $p_k$ , определить оценку степени близости между позицией  $X_k(t)$  и лучшей позицией  $X_k^*(t)$  роя.
9. Перемещение каждой частицы  $p_k$  из занимаемой позиции  $X_k(t)$  в позицию  $X_k(t+1)$  под воздействием притяжения к лучшей позиции  $X_k^*(t)$  роя (социальному центру притяжения) путем применения оператора направленной мутации (ОНМ) к хромосомам позиции  $X_k(t)$ .
10. Если  $t < T$ , то  $t = t + 1$  и переход к пункту 5, иначе переход к пункту 11.
11. Конец работы алгоритма.

## 7. Механизмы генетического поиска

На первом этапе гибридного алгоритма выполняется генетическая эволюция популяции позиций  $X = \{X_k | k = 1, 2, \dots, K\}$ , в которых помещен рой частиц  $P = \{p_k | k = 1, 2, \dots, K\}$ . Позиция  $X_k(t)$ , соответствующая, частице  $p_k$ , определяется набором из  $n$  хромосом, соответствующим  $n$  строкам матрицы  $B$ ,  $X_k(t) = \{H_{ki}(t) | i = 1, 2, \dots, n\}$ .

Далее на каждой итерации алгоритма выполняют работу генетические операторы кроссовера и мутации. После применения этих операторов популяция увеличивается (происходит ее расширение) и требуется ее редуцировать (уменьшить) применяя операторы селекции до исходного размера.

Основными генетическими операторами являются кроссинговер и мутация [11]. Структура кроссинговера в основном определяет эффективность ГА.

После выбора пары родительских позиций  $X_1(t)$  и  $X_2(t)$ , формируются пары соответствующих одной строке матрицы  $B$  хромосом ( $H_{1i}(t)$ ,  $H_{2i}(t)$ ),  $H_{1i}(t) \in X_1(t)$ ,  $H_{2i}(t) \in X_2(t)$ ,  $i = 1, 2, \dots, n$ . Кроссинговер выполняется для каждой

пары хромосом  $(H_{1i}(t), H_{2i}(t))$ . Поскольку хромосомы гомологичны, кроссинговер выполняется путем взаимного обмена генами между парой хромосом в случайно выбранных локусах.

Каждая хромосома позиции подвергается мутации.

Оператор мутации выполняется путем изменения взаимного расположения двух (случайно выбранных) генов в хромосоме.

При этом можно сказать, что те алгоритмы в которых происходит реализация кроссовера и мутации, в пределах одной хромосомы имеют линейную зависимость оценки временной сложности.

## 8. Экспериментальные исследования

На основе рассмотренной парадигмы разработана программа покрытия **SWARM COVERING**. Для проведения тестовых исследований была применена процедура формирования тестовых примеров с уже полученным оптимумом  $F_{\text{опт}}$  по аналогии с существующим методом **BEKU** (Partitioning Examples with Tight Upper Bound of Optimal Solution) [18]. Оценкой качества служит величина  $F_{\text{опт}}/F$  – «**степень качества**», где  $F$  – критерий качества решения. После проведения тестовых испытаний, полученные результаты были обработаны и на их основе получена зависимость значения качества от количества генераций (рис. 4).

Исследования показали, что число итераций, при которых алгоритм находил лучшее решение лежит в пределах 110-130.

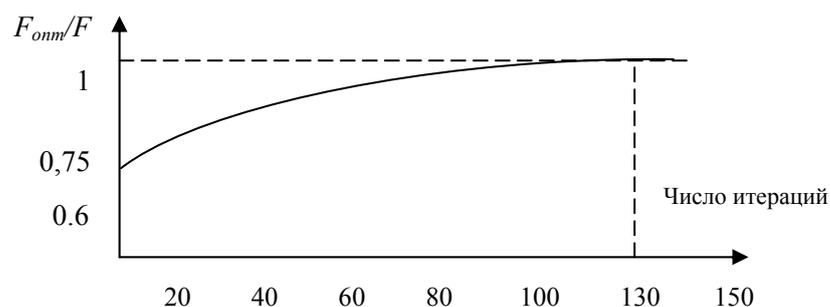


Рис. 4 - Зависимость степени качества алгоритма от числа итераций

Алгоритм сходится в среднем на 125 итерации. Тестовые испытания выявили, что при расширении популяции  $M$  больше чем до 120 не происходит заметное улучшение критерия. В результате проведенных исследований установлено, что целевая функция у гибридного алгоритма имеет значение на 10–15% лучше целевой функции генетического и роевого алгоритмов выполняющий поиск решения по отдельности.

Проведя сравнение значений целевой функции полученных гибридным алгоритмом при решении выбранных тестов имеющих известное лучшее значение, можно сделать вывод, что вероятность получения глобального оптимума составила 0,95. Полученный разработанным гибридным алгоритмом результат (был получен на 130 генерации), отличен от точного на 0.15 процента. Было установлено, что в среднем работа программы позволяет определять результаты имеющие отличие по сравнению с оптимальным менее, чем на 2%. Проанализировав существующие алгоритмы [15-21], можно сделать следующий вывод. Разработанный алгоритм обладает меньшим временем работы. При этом полученные результаты имеют различие значения критерия по сравнению с оптимальным значением менее чем на 4-6%.

Разработанный гибридный алгоритм был исследован на бенчмарках, расположенных в библиотеке OR-объектов (<http://www.ms.ic.ac.uk/info.html>). Зависимость времени работы гибридного алгоритма покрытия при фиксированных значениях  $M$  и  $T$  лежит в пределах  $O(n)$ . Общая зависимость времени работы гибридного алгоритма покрытия, в общем подтверждает теоретические исследования и для использованных экспериментов лежит в пределах  $O(n^2)$ -  $O(n^3)$ .

---

## Заключение

В работе используется символьное представление решения задачи покрытия в виде компактной матрицы граничных требований (МГТ). Это дает возможность сформировать компактное пространство решений, внутри которого происходит поиск оптимального решения. В основу этого поискового процесса положен принцип моделирования адаптивного поведения биологических систем.

Предложена композитная архитектура многоагентной системы бионического поиска на основе интеграции роевого интеллекта и генетической эволюции.

Уникальность этого метода состоит в том, что ключевым моментом выступает общая структура данных, которая описывает и представляет решение задачи в виде хромосомы.

Интеграция метаэвристик популяционных алгоритмов обеспечивает более широкий обзор области поиска и более высокую вероятность локализации глобального экстремума задачи.

Ключевая проблема, которая была решена в данной работе, связана с разработкой структуры аффинного пространства позиций, позволяющей отображать и осуществлять поиск интерпретаций решений с целочисленными значениями параметров. В отличие от канонического метода роя частиц, для уменьшения веса аффинных связей, путем перемещения частицы  $p_i$  в новую позицию аффинного пространства решений разработан оператор направленной мутации, суть которого заключается в изменении целочисленных значений генов в хромосоме. Разработаны новые структуры хромосом для представления решений.

Представленный и описанный в работе алгоритм покрытия схем библиотечными наборами ячеек является универсальным. В этой связи он может быть применен при решении широкого круга задач линейного

---

целочисленного программирования. Эксперименты показали, что качество решений, полученных гибридным алгоритмом на 1-15% лучше, чем у генетического и роевого алгоритмов. Вероятность получения глобального оптимума составила 0.9. Общая зависимость времени работы гибридного алгоритма покрытия, какой бы не использовался способ гибридизации, не больше зависимости времени работы генетического алгоритма и лежит в пределах  $O(n^2)$ - $O(n^3)$ .

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

### Литература

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: 2004. 226 с.
2. Андерсон Д. Дискретная математика и комбинаторика. М.: Вильямс, 2003. 187 с.
3. Корниенко В.П. Методы оптимизации. М.: Высш. шк., 2007. 176 с.
4. Galinier P. and Heutz A. Solution techniques for the large set covering problem // Discrete applied Mathematics. 2007. Vol. 155. Issue 3. 96 p.
5. Еремеев А.В., Заозерская Л.А., Колоколов А.А. Задача о покрытии множества: сложность, алгоритмы, экспериментальные исследования // Дискрет. анализ и исслед. операций. Сер. 2. 2000. Т. 7, № 2. С. 22–46.
6. Забиняко Г.И. Реализация алгоритмов решения задачи о покрытии множеств и анализ их эффективности // Вычислительные технологии. Том 12, № 6, 2007. С. 50-58.
7. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учебное пособие. М: Издательство МГТУ им. Н.Э. Баумана, 2014. 446 с.

8. Raidl G.R. A Unified View On Hybrid Metaheuristics. In: Lecture Notes In Computer Science, 2006. Springer, Verlag. pp. 1-12.

9. Wang X. Hybrid nature-inspired computation method for optimization. Doctoral Dissertation. Helsinki University of Technology, TKK Dissertations, Espoo 2009, 161 p.

10. Blum C., Roli A. Metaheuristics in combinatorial optimization: overview and conceptual comparison // ACM computing surveys. № 35, 2003. pp. 268-308.

11. M. Clerc. Particle Swarm Optimization. ISTE, London, UK, 2006. 191 p.

12. Лебедев Б.К., Лебедев О.Б. Гибридный биоинспирированный алгоритм решения задачи символьной регрессии // Известия ЮФУ. Технические науки. Ростов-на-Дону: Изд-во ЮФУ, 2015. № 6 (167). С. 28-41.

13. Лебедев Б.К., Лебедев В.Б. Покрытие методом роя частиц // Материалы VI-й Международной научно-практической конференции “Интегрированные модели и мягкие вычисления в искусственном интеллекте”. М.: Физматлит, 2011. С. 611-619.

14. Лебедев Б.К., Лебедев О.Б., Лебедева Е.М. Распределение ресурсов на основе гибридных моделей роевого интеллекта // Научно-технический вестник информационных технологий, механики и оптики. 2017. Т. 17. № 6. С. 1063–1073.

15. Лебедев Б.К., Лебедев В.Б. Трассировка на основе метода роя частиц. Двенадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2010. Труды конференции. Т.2. М.: Физматлит, 2010. С. 414-422.

16. J. Kennedy, R.C. Eberhart. Particle swarm optimization // In Proceedings of IEEE International Conference on Neural Networks, 1995. pp. 1942–1948.

17. J. Cong, M. Romesis, and M. Xie "Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms". Proc. of the International Symposium on Physical Design, Monterey, CA, April 2003. pp. 88-94.

---

18. Нгуен Минь Ханг. Применение генетического алгоритма для задачи нахождения покрытия множества // Труды института системного анализа РАН, Т. 33, 2008. С. 206-211.

19. Коновалов И.С., Фатхи В.А., Кобак В.Г. Применение генетического алгоритма для решения задачи покрытия множеств // Вестник Донского государственного технического университета №3(86), 2016. С. 125-132.

20. Коновалов И.С., Фатхи В.А., Кобак В.Г. Сравнительный анализ работы жадного алгоритма Хватала и модифицированной модели Голдберга при решении взвешенной задачи нахождения минимального покрытия множеств // Труды СКФ МТУСИ. Часть I, 2015. С. 366–370.

21. Есипов Б.А., Муравьев В.В. Исследование алгоритмов решения обобщенной задачи о минимальном покрытии // Известия Самарского научного центра Российской академии наук, Т. 16. №4(2), 2014. С. 256–260.

22. Засядко Г.Е. Представление графовых моделей данных в виде парных деревьев // Инженерный вестник Дона, 2017, №2 URL: [ivdon.ru/ru/magazine/archive/N2y2017/4166](http://ivdon.ru/ru/magazine/archive/N2y2017/4166).

23. Лебедев Б.К., Лебедев О.Б., Лебедева Е.О. Роевой алгоритм планирования работы многопроцессорных вычислительных систем // Инженерный вестник Дона, 2017, №3 URL: [ivdon.ru/ru/magazine/archive/N3y2017/4362](http://ivdon.ru/ru/magazine/archive/N3y2017/4362).

### References

1. Kormen T., Leyzerson C.H., Rivest R. Algoritmy: postroyeniye i analiz [Algorithms: construction and analysis]. М.: 2004. 226 p.

2. Anderson D. Diskretnaya matematika i kombinatorika [Discrete mathematics and combinatorics]. М.: Vil'yams, 2003. 187 p.

3. Korniyenko V.P. Metody optimizatsii [Optimization methods]. М.: Vysshaya shkola, 2007. 176 p.

4. Galinier P. and Heztz A. Discrete applied Mathematics. 2007. Vol. 155. Issue 3. 96 p.
  5. Yeremeyev A.V., Zaozerskaya L.A., Kolokolov A.A. Diskret. analiz i issled. operatsiy. Ser. 2. 2000. Tom 7, № 2. pp. 22–46.
  6. Zabinyako G.I. Vychislitel'nyye tekhnologii. Tom 12, № 6, 2007. pp. 50-58.
  7. Karpenko A.P. Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy [Modern algorithms of search optimization. Algorithms inspired by nature]. Moscow: MSTU. N.E. Bauman. 2014. 446 p.
  8. Raidl G.R. In: Lecture Notes In Computer Science, 2006. Springer, Verlag. pp. 1-12.
  9. Wang X. Hybrid nature-inspired computation method for optimization. Doctoral Dissertation. Helsinki University of Technology, TKK Dissertations, Espoo 2009. 161 p.
  10. Blum C., Roli A. ACM computing surveys. № 35, 2003. pp. 268-308.
  11. M. Clerc. Particle Swarm Optimization. ISTE, London, UK, 2006. 191 p.
  12. Lebedev B.K., Lebedev O.B. Izvestiya YUFU. Tekhnicheskiye nauki. Rostov-na-Donu: Izd-vo SFU, 2015. № 6 (167). pp. 28-41.
  13. Lebedev B.K., Lebedev V.B. Materialy VI Mezhdunarodnoy nauchno-prakticheskoy konferentsii “Integrirovannyye modeli i myagkiye vychisleniya v iskusstvennom intellekte”. M.: Fizmatlit, 2011. pp. 611-619.
  14. Lebedev B.K., Lebedev O.B., Lebedeva E.M. Nauchno-tekhnicheskiy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki. 2017. Tom 17. № 6. pp. 1063–1073.
  15. Lebedev B.K., Lebedev V.B. Dvenadtsataya natsional'naya konferentsiya po iskusstvennomu intellektu s mezhdunarodnym uchastiyem KII-2010. Trudy konferentsii. Tom 2. M.: Fizmatlit, 2010. pp. 414-422.
-



16. J. Kennedy, R.C. Eberhart. In Proceedings of IEEE International Conference on Neural Networks, 1995. pp. 1942–1948.
17. Cong J., Romesis M. and Xie M. Proc. of the International Symposium on Physical Design, Monterey, CA, April 2003. pp. 88-94.
18. Nguyen Min' Khang. Trudy instituta sistemnogo analiza. RAN, T. 33, 2008. pp. 206-211.
19. Konovalov I.S., Fatkhi V.A., Kobak V.G. Vestnik Donskogo gosudarstvennogo tekhnicheskogo universiteta №3(86), 2016. pp. 125-132.
20. Konovalov I.S., Fatkhi V.A., Kobak V.G. Trudy SKF MTUSI. Chast' I, 2015. pp. 366–370.
21. Yesipov B.A., Murav'yev V.V. Izvestiya Samarskogo nauchnogo tsentra Rossiyskoy akademii nauk, T. 16. №4(2), 2014. pp. 256–260.
22. Zasyadko G.Ye. Inženernyj vestnik Dona (Rus), 2017, №2. URL: [ivdon.ru/ru/magazine/archive/N2y2017/4166](http://ivdon.ru/ru/magazine/archive/N2y2017/4166).
23. Lebedev B.K., Lebedev O.B., Lebedeva E.O. Inženernyj vestnik Dona (Rus), 2017, №3. URL: [ivdon.ru/ru/magazine/archive/N3y2017/4362](http://ivdon.ru/ru/magazine/archive/N3y2017/4362).