

Раздел III. Эволюционное моделирование и биоинспирированные алгоритмы

УДК 004.896

DOI 10.23683/2311-3103-2018-4-121-136

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

МОДИФИЦИРОВАННЫЕ МЕХАНИЗМЫ ПЕРЕМЕЩЕНИЯ РОЯ ЧАСТИЦ (АГЕНТОВ) В АФФИННОМ ПРОСТРАНСТВЕ С ЦЕЛОЧИСЛЕННЫМИ ПАРАМЕТРАМИ*

Предложена композитная архитектура многоагентной системы бионического поиска для решения комбинаторных задачи на основе интеграции роевого интеллекта и генетической эволюции. Интеграция метаэвристик популяционных алгоритмов обеспечивает более широкий обзор пространства поиска и более высокую вероятность локализации глобального экстремума задачи. Связующим звеном такого подхода является единая структура данных, описывающая в виде хромосомы решение задачи. В отличие от канонической парадигмы роя частиц гибридные алгоритмы в качестве моделей для представления решений используют широкий диапазон графовых структур (маршрут, дерево, двудольный граф, паросочетание, внутреннеустойчивое множество и т.д.). Такой подход является эффективным способом поиска рациональных решений для задач оптимизации, допускающих интерпретацию решений в виде различного рода графовых структур. В работе описывается модифицированная парадигма роя частиц, обеспечивающая, в отличие от канонического метода, возможность поиска решений в аффинном пространстве позиций с целочисленными значениями параметров. Рассмотрены механизмы перемещения частиц в аффинном пространстве для уменьшения веса аффинных связей. Описываются операторы направленной мутации, суть которых заключается в изменения целочисленных значений генов в хромосоме. Анализ существующих методов и алгоритмов решения комбинаторных задач показал, что в качестве структуры данных, несущих информацию о решении, чаще всего используются списки, фактически являющимися интерпретациями решений. Разработанные структуры пространства поиска и позиций позволяют отображать: Списки, элементы которых могут иметь два значения 0 или 1; Списки, содержащие фиксированные числа нулей и единиц; Списки с фиксированной суммой значений элементов; Списки, описывающие структуру бинарного дерева; Списки, задающие последовательность элементов. Разработанные структуры позиций (хромосом) ориентированы на интеграцию роевого интеллекта и генетической эволюции. В ряде алгоритмов качестве структуры данных используется кодированное представление списков. Переход от кодированного представления к списку производится с помощью декодера. Разработаны новые структуры хромосом для представления решений и методы декодирования. Эксперименты показали, что качество решений, полученных гибридным алгоритмом на 10–15 % лучше, чем у генетического и роевого алгоритмов. Вероятность получения глобального оптимума составила 0.9. Общая оценка временной сложности при любом подходе к гибридизации не превышает оценки временной сложности генетического алгоритма и лежит в пределах $O(n^2)$ - $O(n^3)$.

Рой частиц; генетическая эволюция; аффинное пространство; целочисленные параметры; структуры позиций; оператор направленной мутации; механизмы перемещения частиц; интеграция; гибридизация.

* Работа выполнена при финансовой поддержке РФФИ (проект № 18-07-00737).

B.K. Lebedev, O.B. Lebedev, E.O. Lebedeva

MODIFIED MECHANISMS OF MOVEMENT OF PARTICLE SWARM (AGENTS) IN AFFINE SPACE WITH INTEGRATED PARAMETERS

The composite architecture of the multi-agent bionic search system is proposed to solve combinatorial problems based on integration of swarm intelligence and genetic evolution. Integration of metaheuristics of population algorithms provides a broader overview of the search space and a higher probability of localization of the problem's global extremum. The connecting link of this approach is a unified data structure describing the solution of the problem in the form of a chromosome. Unlike the canonical paradigm of a swarm of particles, hybrid algorithms use a wide range of graph structures (routes, tree, bipartite graph, matching, maximal independent set, etc.) as models for representing solutions. Such an approach is an effective way of finding rational solutions for optimization problems that allow the interpretation of solutions in the form of various graph structures. The paper describes a modified paradigm of particle swarm that provides, unlike the canonical method, the possibility of finding solutions in the affine space of positions with integer parameter values. Mechanisms for moving particles in affine space to reduce the weight of affine bonds are considered. Operators of directed mutation are described, the essence of which is to change the integer values of genes in the chromosome. An analysis of existing methods and algorithms for solving combinatorial problems has shown that the list of data that are actually interpretations of solutions is used most often as the data structure carrying information about the solution. The developed structures of search space and positions allow displaying: 1. Lists, the elements of which can have two values 0 or 1; 2. Lists containing fixed numbers of zeros and ones; 3. Lists with a fixed sum of the values of the elements; 4. Lists describing the structure of a binary tree; 5. Lists specifying the sequence of elements. The developed position structures (chromosomes) are focused on the integration of swarm intelligence and genetic evolution. In a number of algorithms, the coded representation of lists is used as the data structure. The transition from the encoded view to the list is performed using the decoder. New chromosome structures have been developed to represent solutions and decoding methods. Experiments have shown that the quality of the solutions obtained by the hybrid algorithm is 10 to 15 % better than the genetic and swarm algorithms. The probability of obtaining a global optimum is 0.9. The overall estimate of time complexity for any hybridization approach does not exceed the estimate of the time complexity of the genetic algorithm and lies within the range $O(n^2) - O(n^3)$.

Particle swarm; genetic evolution; affine space; integer parameters; position structures; directed mutation operator; particle transport mechanisms; integration, hybridization.

Введение. В последние годы интенсивно разрабатывается научное направление, объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений. Результатом непрекращающегося поиска наиболее эффективных методов стало использование бионических методов и алгоритмов интеллектуальной оптимизации, базирующихся на моделировании коллективного интеллекта [1, 2]. Анализ методов решения сложных прикладных задач показывает, что применение любого одного алгоритма оптимизации (как классического, так и популяционного) далеко не всегда приводит к успеху. В связи с этим, в настоящее время одним из основных путей повышения эффективности методов для решения задач глобального поиска является разработка гибридных алгоритмов [3]. В гибридных алгоритмах преимущества одного алгоритма могут компенсировать недостатки другого. Интеграция метаэвристик популяционных алгоритмов обеспечивает более широкий обзор пространства поиска и более высокую вероятность локализации глобального экстремума задачи [4].

Предлагается композитная архитектура многоагентной системы бионического поиска на основе роевого интеллекта и генетической эволюции [5]. Первый и наиболее простой подход к гибридизации заключается в следующем. С начала поиска решения осуществляется генетическим алгоритмом. Затем на основе популяции, полученной на последней итерации генетического поиска, формируется по-

пуляция для роевого алгоритма. В формируемую популяцию включаются лучшие, но отличные друг от друга хромосомы. При необходимости полученная популяция доукомплектовывается новыми индивидами. После этого дальнейший поиск решения осуществляется роевым алгоритмом.

При втором подходе метод роя частиц используется в процессе генетического поиска и играет роль аналогичную генетическим операторам. В этом случае на каждой итерации генетического алгоритма синтез новых хромосом с одной стороны осуществляется с помощью кроссинговера и мутации, а с другой стороны, с помощью операторов направленной мутации методами роя частиц.

Метод роя частиц (Particle Swarm Optimization, PSO) является методом стохастической оптимизации в чем-то схожим с эволюционными алгоритмами. Этот метод моделирует не эволюцию, а ройное и стайное поведение животных [6]. В отличие от популяционных методов PSO работает с одной статической популяцией, члены которой постепенно улучшаются с появлением информации 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).

Рассмотрим каноническую парадигму метода PSO, разработанную Джеймсом Кеннеди (*James Kennedy*) и Расселом Эберхартом (*Russell Eberhart*) [7]. Метод PSO работает в многомерных, вещественных, метрических пространствах. Многомерное пространство поиска населяется роем частиц. Каждой частице соответствует некоторое решение. Основу поведения роя $P = \{p_i | i = 1, 2, \dots, n\}$ частиц составляет самоорганизация, обеспечивающая достижение общих целей роя на основе низкоуровневого взаимодействия. Каждая частица p_i размещается в точке (позиции) x_i многомерного пространства поиска, связана и может взаимодействовать со всеми частицами роя, она тяготеет к лучшему решению роя. Процесс поиска решений итерационный и заключается в перемещении частиц на каждой итерации в новые позиции.

По аналогии с эволюционными стратегиями, рой частиц можно трактовать как популяцию, а частицу как индивида. Это дает возможность построения гибридной структуры поиска решения, основанную на сочетании генетического поиска с методами роя частиц [8]. Связующим звеном такого подхода является структура данных, описывающая решение задачи. В генетических алгоритмах генотип решения представляется в виде хромосомы. При решении комбинаторных задач гены в хромосомах имеют, как правило, целочисленные значения. [5]. Положение частицы в пространстве поиска (позиция) является эквивалентом генотипа в эволюционных алгоритмах. Если в качестве описания состояния 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, значениям генов, операторам перемещения частиц и пространству поиска.

Каноническая парадигма роя частиц предусматривает использование вещественных значений параметров в многомерных, вещественных, метрических пространствах. Однако в большинстве генетических алгоритмов гены в хромосомах имеют целочисленные значения. В свою очередь хромосомы являются некоторыми интерпретациями решений, которые трансформируются в решения путем декодирования хромосом.

Анализ существующих методов и алгоритмов решения комбинаторных задач показал, что в качестве структуры данных, несущих информацию о решении, чаще всего используются списки, фактически являющимися интерпретациями решений. Данное представление удобно для его использования в различных метаэвристиках (генетические алгоритмы, муравьиные алгоритмы).

На списки, используемые в качестве интерпретаций решений, в зависимости от специфики задачи, накладываются ограничения. Существенную роль в общем процессе нахождения решения играет декодер, осуществляющий трансформацию от списка к решению. Важной характеристикой декодера является его способность получить оптимальное решение по заранее известному оптимальному коду (приоритетному списку), т.е. способность правильно декодировать.

В отличие от канонической парадигмы роя частиц гибридные алгоритмы в качестве моделей для представления решений используют широкий диапазон графовых структур (маршрут, дерево, двудольный граф, паросочетание, внутренне-устойчивое множество и т.д.) [8–11]. Такой подход является эффективным способом поиска рациональных решений для задач оптимизации, допускающих интерпретацию решений в виде различного рода графовых структур.

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 построению модифицированной парадигмы роя частиц, обеспечивающей возможность одновременного использования хромосом с целочисленными значениями параметров в генетическом алгоритме и в алгоритме на основе роя частиц. Рассматриваются методы представления графовых интерпретаций решений в виде хромосом, используемых без трансформаций, как в роевом, так и в генетическом алгоритмах. Для каждого типа представления описываются механизмы перемещения агентов (частиц) в пространстве поиска.

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 (вектором). $p, q \in V(p, q)$ – геометрический вектор (упорядоченная пара).

Совокупность всех точек ЛВП, дополненная 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_i 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.

Лучшие частицы 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 одному базовому элементу либо к группе соседних элементов и с учетом вероятности перехода в новое состояние.

Для того, чтобы избежать при описании популяции (роя) путаницы будем в дальнейшем каждую хромосому, описывающую i -е решение популяции обозначать как $H_i(t) = \{g_{il} | l=1, 2, \dots, n\}$. Причем каждая $H_i(t)$ имеет описанную выше 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.

Поиск методом роя частиц. В эвристических алгоритмах роевого интеллекта многомерное пространство поиска населяется роем частиц [7]. В процессе поиска методом роя частиц каждая частица перемещается 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 $x_i(0)$, $v_i(0)$ [1].

Приведенная формула представлена в векторной форме. 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 измерения 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 $v_i(t+1)$ 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 с вещественными значениями переменных. Если в качестве позиции используется хромосома, то число параметров, определяющих положение частицы в пространстве решений (позицию) 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 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 $x(t)$, $x_i(t)$, $x_i^*(t)$, $x_i^c(t)$ соответствуют хромосомы $H^*(t) = \{g_{il}^*(t) | l=1, 2, \dots, n_l\}$, $H_i(t) = \{g_{il}(t) | l=1, 2, \dots, n_l\}$, $H_i^*(t) = \{g_{il}^*(t) | l=1, 2, \dots, n_l\}$, $H_i^c(t) = \{g_{il}^c(t) | l=1, 2, \dots, n_l\}$.

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 сокращение 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 $x_i^c(t)$ частицы p_i . Формирование *виртуальной позиции* $x_i^c(t)$ осуществляется путем применения процедуры *виртуального* перемещения из позиции $x_i^*(t)$ в *виртуальную позицию* $x_i^c(t)$ по направлению к позиции $x^*(t)$. После определения центра притяжения $x_i^c(t)$ частица p_i перемещается в направлении *виртуальной позиции* $x_i^c(t)$ 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.

Локальная цель перемещения частицы p_i – достижение ею позиции с наилучшим значением целевой функции. Глобальная цель роя частиц – формирование оптимального решения задачи.

Механизмы перемещения роя частиц в аффинном пространстве. Рассмотрим модификации аффинного пространства поиска, структуры позиций и механизмы перемещения частиц для различных типов списков.

1. Списки, элементы которых могут иметь два значения 0 или 1.

В ряде задач решение представляется в виде совокупности фрагментов, каждый из которых может быть реализован в одном из двух вариантов. В этом случае решение полностью определяется списком, элементы которого могут иметь два значения 0 или 1.

Пространство поиска включает число осей равное числу элементов списка (число генов в хромосоме). Каждой позиции элемента в списке соответствует ось (номер оси). На каждой оси X_i две точки отсчета – $x_{i1} = 0$, $x_{i2} = 1$.

Хромосома, описывающая решение, имеет следующую структуру: $H_i(t) = \{g_{il}(t) | l=1, 2, \dots, n_l\}$. Каждый ген $g_{il}(t)$ несет информацию о варианте фрагмента. Если $g_{il} = 1$, то выбран первый вариант соответствующего фрагмента. Если $g_{il}(t) = 0$, то выбран второй вариант соответствующего фрагмента.

Например: Список $M = \langle 1, 0, 0, 1, 0, 0, 1 \rangle$ представляется в виде позиции в пространстве поиска

$$H = \{x_{12}, x_{21}, x_{31}, x_{42}, x_{51}, x_{61}, x_{72}\}.$$

Подобные списки лежат в основе интерпретаций решений в задачах символьной регрессии [9], трассировки [10] и в ряде других задач.

В качестве веса аффинной связи (оценки расстояния) между двумя позициями будем использовать расстояние Хемминга. Расстояние Хемминга между двумя бинарными векторами одинаковой длины – это число несовпадающих бит в этих векторах. Пусть $S_{iz}(t)$ – расстояние Хемминга между позициями $H_i(t)$, и $H_z(t)$. Частица p_i перемещается из позиции $H_i(t)$ в направлении позиции $H_z(t)$.

Введем определение лотереи. *Простой лотереей* $L(y_1, p, y_2)$ назовем вероятностное событие, имеющее два возможных исхода y_1 и y_2 , вероятности наступления которых обозначим соответственно через p и $(1-p)$.

Для формирования новой позиции $H_i(t+1)$ (позиции $x_i(t+1)$) последовательно просматриваются (начиная с первого) локусы хромосом $H_i(t)$ и $H_z(t)$ и сравниваются соответствующие им гены.

Если на шаге l значения генов равны, т.е.

$$g_{il}(t) = g_{zl}(t), \text{ то } g_{il}(t+1) = g_{il}(t). \quad (4)$$

Если же $g_{il}(t)$ не равно $g_{zl}(t)$, то гену $g_{il}(t+1)$ присваивается значение, определяемое лотереей L , т.е.

$$g_{il}(t+1) = L(g_{zl}(t), p, g_{il}(t)). \quad (5)$$

Параметр p зависит от расстояния Хемминга $s_{iz}(t)$ между позициями $H_i(t)$ и $H_z(t)$ и определяется как

$$p = \alpha \cdot S_{iz}(t) / n, \quad (6)$$

где α – коэффициент, n – число генов в хромосоме. Таким образом, чем больше расстояния Хемминга $S_{iz}(t)$ между позициями $H_i(t)$ и $H_z(t)$, тем больше вероятность, что значением $g_{il}(t+1)$ будет значение $g_{zl}(t)$.

Пример. Пусть $H_z(t) = \langle 1, 0, 0, 1, 1, 0, 0, 0, 1 \rangle$,

$$H_i(t) = \langle 0, 1, 0, 0, 1, 0, 1, 0, 0 \rangle.$$

Значения генов не совпадают в 1, 2, 4, 7, 9 локусах, $S_{iz}(t) = 5$. Пусть с некоторой вероятностью в $H_i(t)$ мутировали 2, 4 и 7 гены. В результате направленной мутации

$$H_i(t+1) = \langle 0, 0, 0, 1, 1, 0, 0, 0, 0 \rangle. \quad S_{iz}(t+1) = 2.$$

2. Списки, содержащие фиксированные числа нулей и единиц.

Для некоторых задач решение полностью определяется списком $M = \langle 0, 1, 1, 0, 0, 1 \rangle$, составленным из нулей и единиц с фиксированным числом нулей – n_0 и числом единиц – n_1 . С помощью списка M можно для некоторого множества A , мощность которого $|A| = n_0 + n_1$, разбить на два подмножества A_1 и A_2 такие, что $|A_1| = n_0$, а $|A_2| = n_1$.

Подобные списки лежат в основе используются как интерпретации решений в задачах построения кратчайших связывающих сетей [11], разбиения [12], кластеризации [13] и др.

Для кодирования списка M используется опорный список нулей $O = \{o_i \mid i = 1, 2, \dots, n_0\}$. На базе опорного списка нулей O формируется множество позиций $D = \{d_i \mid i = 1, 2, \dots, (n_0 + 1)\}$ (рис. 1), в которых могут располагаться единицы, образующие совместно с нулями список M .

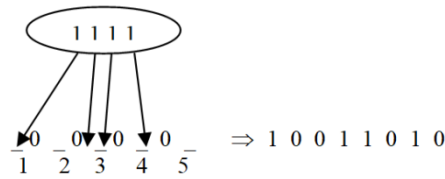
Если число нулей n_0 , то число позиций списка D равно $n_0 + 1$ (см. рис.1).

Каждая из n_1 единиц может быть расположена в любой из позиций $d_i \in D$.

Допустимому варианту распределения единиц по позициям D опорного списка нулей соответствует список M . Распределение единиц по позициям опорного списка нулей задается с помощью списка Q , размер которого равен n_1 , а значением элемента списка Q является число, лежащее в диапазоне от 1 до $(n_0 + 1)$. Таким образом, список Q является кодом (интерпретацией) списка M .

Например:

Пусть $n_1 = 4$, $n_0 = 4$, $Q = \langle 4, 3, 3, 1 \rangle$. Тогда $D = \langle 0, 0, 0, 0 \rangle$. $M^* = \langle 1, 0, 0, 1, 1, 0, 1, 0 \rangle$ (см. рис. 1).

Рис. 1. Декодирование списка Q

Отметим, что порядок взаимного расположения элементов в списке Q не имеет значения, важно только, какие значения имеют элементы списка Q .

Хромосома, описывающая решение, имеет следующую структуру: $H_i(t) = \{g_{il}(t) | l=1, 2, \dots, n_l\}$. Значение гена $g_{il}(t)$ равно значению соответствующего элемента списка Q .

Пространство поиска включает число осей, равное числу элементов списка Q (числу генов в хромосоме $H_i(t)$) и соответственно числу единиц в списке M . Каждой позиции элемента в списке Q соответствует ось (номер оси). Точками отсчета на каждой оси X_i являются целые числа в интервале от 1 до $(n_0 + 1)$.

Например:

Пусть список $Q = \langle 4, 3, 3, 1 \rangle$ является интерпретацией (кодом) списка $M = \langle 1 0 0 1 1 0 1 0 \rangle$.

Список Q представляется в виде позиции $H = \{x_{14}, x_{23}, x_{33}, x_{41}\}$.

Перемещение частицы p_i из позиции $H_i(t)$ в позицию $H_i(t+1)$ под воздействием притяжения к позиции $H_z(t)$ осуществляется путем применения оператора направленной мутации (ОНМ) к $H_i(t)$ следующим образом.

В качестве оценки степени близости между двумя позициями $H_i(t)$ и $H_z(t)$ будем использовать величину S_{iz} расстояния между хромосомами $H_i(t)$ и $H_z(t)$:

$$S_{iz} = \sum_{l=1}^n |g_{il} - g_{zl}|. \quad (7)$$

Суть процедуры перемещения, реализуемой ОНМ, заключается в изменении разности между значениями каждой пары генов (g_{il}, g_{zl}) двух хромосом, $l=1, 2, \dots, n$.

Последовательно просматриваются (начиная с первого) локусы хромосом $H_i(t)$ и $H_z(t)$, и сравниваются соответствующие им гены. Если в ходе последовательного просмотра локусов в текущем локусе l выпадает с вероятностью P событие «мутация», то ген $g_{il}(t) \in H_i(t)$ мутирует.

Пусть $R_{iz}(t)$ – число локусов в хромосомах $H_i(t)$ и $H_z(t)$, в которых значения генов $g_{il}(t) \in H_i(t)$ и $g_{zl} \in H_z(t)$ не совпадают.

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

$$P = \alpha \cdot R_{iz}(t) / n, \quad (8)$$

где α коэффициент, n – длина хромосомы. Таким образом, чем больше число $R_{iz}(t)$ несовпадающих генов между позициями $H_i(t)$ и $H_z(t)$, тем больше вероятность, что значение $g_{il}(t)$ будет изменено.

Простой лотереей $L(1, p, 0)$ назовем вероятностное событие, имеющее два возможных исхода 1 и 0, вероятности наступления которых обозначим соответственно через p и $(1-p)$. Другими словами с вероятностью p лотерея $L(1, p, 0) = 1$, а с вероятностью $(1-p)$ лотерея $L(1, p, 0) = 0$. Значения генов новой позиции частицы p_i в результате мутации определяются как:

$$g_{il}(t+1) = g_{il}(t), \text{ если } (g_{il}(t) = g_{zl}(t));$$

$$\begin{aligned} g_{il}(t+1) &= g_{il}(t) + L(1, p, 0), \text{ если } (g_{il}(t) < g_{zl}(t)); \\ g_{il}(t+1) &= g_{il}(t) - L(1, p, 0), \text{ если } (g_{il}(t) > g_{zl}(t)). \end{aligned} \quad (9)$$

Пример. Пусть, $n_1 = 5$, $n_0 = 7$, $H_i(t) = \langle 1, 2, 4, 8, 8 \rangle$, $M_i(t) = \langle 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1 \rangle$,
 $H_z(t) = \langle 3, 3, 4, 7, 8 \rangle$, $M_z(t) = \langle 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1 \rangle$,
 $S_{iz}(t) = 4$.

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

$$\begin{aligned} H_i(t+1) &= \langle 2, 2, 4, 7, 8 \rangle, \quad M_i(t+1) = \langle 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1 \rangle, \\ S_{iz}(t+1) &= 2. \end{aligned}$$

3. Списки, с фиксированной суммой значений элементов.

Пусть имеется список $M = \{m_j | j = 1, 2, \dots, n_m\}$, с фиксированной суммой значений элементов $W = \sum_j m_j$.

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

Хромосома $H = \{g_l | l = 1, 2, \dots, (n_m - 1)\}$ представляет собой совокупность $(n_m - 1)$ генов g_l , значения которых могут изменяться в пределах диапазона, определяемого параметром W :

$$g_l \in H, 0 \leq g_l \leq S.$$

Процесс перехода от хромосомы H к списку осуществляется следующим образом. Сначала гены упорядочиваются по возрастанию их значений, то есть, если $g_l \in H$ и $g_{l+1} \in H$, то $g_{l+1} \geq g_l$. Значения g_l в пределах от 0 до W рассматриваются в качестве координат опорных точек на отрезке длиной W (от 0 до W), разбивающих отрезок на интервалы. Длина интервала между двумя соседними опорными точками и есть величина соответствующего элемента m_j списка M , при этом сумма значений элементов списка M равна W .

Например.

Пусть имеется хромосома $H = \{8, 12, 16, 25\}$, $S = 30$. На отрезке длиной W 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 M .

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

Механизмы перемещение частиц в пространстве поиска полностью совпадают с описанными выше механизмами перемещение частиц - списков, содержащих фиксированные числа нулей и единиц.

4. Списки, описывающие структуру бинарного дерева.

Структуру бинарного дерева можно задать, используя “польскую запись” [17] – список M , элементы которого могут иметь два значения 0 или 1, где элемент типа 0 соответствует листьям дерева, а элемент типа 1 – соответствует внутренним вершинам дерева (точкам ветвления). Список M , содержит фиксированное число нулей и единиц. Обозначим через n_0 – число элементов “польской записи” типа 0, а через n_1 – число элементов типа 1. Для “польской записи”, описывающей бинарное дерево, всегда выполняется равенство $n_0 = n_1 + 1$, $n_0 + n_1 = |M|$.

“Польская запись” формируется аналогично рассмотренному выше списку, содержащему фиксированное число нулей и единиц. На первом этапе формируется опорный список нулей O . Пронумеруем множество позиций $B = \langle b_i | i = 1, 2, \dots, n_1 \rangle$, размещенных между элементами 0 в опорном списке нулей O , как показано на рис. 2.

$$00 \underline{1} 0 \underline{2} 0 \underline{3} 0 \underline{4} 0 \dots (n_l - 1) 0 \underline{n_l}$$

Рис. 2. Формирование множества позиций в опорном списке нулей

Допустимым вариантам распределения n_l единиц по позициям B опорного списка нулей O соответствуют всевозможные списки M . Распределение единиц по позициям множества B задается с помощью списка $Q = \langle q_i | i = 1, 2, \dots, n_l \rangle$. q_i – номер, в диапазоне от i до n_l , позиции в списке B , в которую помещается единица. Таким образом, список Q является кодом (интерпретацией) списка M .

Отметим основные свойства “польской записи”. На размещение единиц в позициях опорного множества нулей накладываются ограничения.

Если в “польской записи” провести справа от элемента 1 сечение, то слева от сечения число элементов типа 0 больше числа элементов типа 1, по крайней мере, на единицу.

Первый элемент типа 1 в “польской записи” (при просмотре слева направо) может появиться только после двух элементов типа 0.

Максимальное число элементов типа 1, которое может появиться в позиции, равно номеру позиции. Если польское выражение соответствует вышеперечисленным свойствам, то ему соответствует бинарное дерево.

Процесс восстановления дерева по польскому выражению достаточно прост. Последовательно слева направо просматривается “польская запись”, и отыскиваются элементы типа 1, соответствующие внутренним вершинам дерева. Каждая такая вершина объединяет два ближайших подграфа, расположенных в “польской записи” слева от элемента типа 1 и образованных на предыдущих шагах.

Подобные списки лежат в основе интерпретаций решений в задачах планирования [17, 18].

“Польской записи” $M = \langle 001000111 \rangle$ соответствует дерево, представленное на рис. 3.

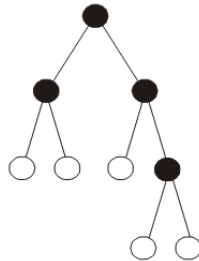


Рис. 3. Дерево, соответствующее “польской записи” $M = \langle 001000111 \rangle$

Список Q представляется в виде хромосомы $H = \{g_i | i = 1, 2, \dots, n_l\}$. В результате декодирования хромосомы H строится “польская запись”. Число генов в хромосоме равно n_l , т.е. числу единиц в “польской записи”, $\sum_i(g_i) = n_l$. Значение гена g_i колеблется в пределах от i до n_l , т.е. $i \leq g_i \leq n_l$. Значение гена указывает номер позиции в “польской записи”, в которую необходимо поместить единицу.

Например: пусть для $n_0 = 5$, $n_l = 4$ имеется хромосома $H = \langle 2, 2, 4, 4 \rangle$. “Польская запись”, соответствующая этой хромосоме, имеет вид: $M = \langle 000110011 \rangle$.

Пространство поиска включает число осей равное числу элементов списка Q (число генов в хромосоме $H_i(t)$). Число элементов списка Q равно числу единиц в списке M . Каждой позиции элемента в списке Q соответствует ось (номер оси). Точками отсчета на каждой оси X_i являются целые числа в интервале от i до n_l .

Например:

Пусть $n_0=5, n_1=4$. $X_1=\langle x_{11}, x_{12}, x_{13}, x_{14} \rangle = \langle 1, 2, 3, 4 \rangle$, $X_2=\langle x_{21}, x_{22}, x_{23} \rangle = \langle 2, 3, 4 \rangle$,
 $X_3=\langle x_{31}, x_{32} \rangle = \langle 3, 4 \rangle$, $X_4=\langle x_{41} \rangle = \langle 4 \rangle$.

Пусть список $Q = \langle 1, 3, 3, 4 \rangle$ является кодом списка $M = \langle 0 0 1 0 0 1 1 0 1 \rangle$.

Список Q представляется в виде позиции $H = \{x_{11}, x_{22}, x_{31}, x_{41}\}$.

Механизмы перемещение частиц в пространстве поиска полностью совпадают с описанными выше механизмами перемещение частиц - списков, содержащих фиксированные числа нулей и единиц.

5. Списки, задающие последовательность элементов.

В ряде алгоритмов качестве структуры данных, несущей информацию о решении, используется последовательность, представляющую собой очередность формирования решения, которая называется приоритетным списком. Приоритетный список – это кодированное решение, в терминах генетического алгоритма – «хромосома». Приоритетный список является косвенной схемой кодирования решения. Переход от приоритетного списка к решению производится с помощью декодера. Декодер – оператор, позволяющий перейти от косвенной (числовой) схемы кодирования решения задачи к фенотипу. Например, в задаче упаковки приоритетный список задает последовательность укладки блоков, в задаче размещения – последовательность размещения элементов в позициях и т.п. Фактически приоритетный список является интерпретацией решения в конкретной предметной области.

Пространство поиска включает число осей равное числу элементов списка (число генов в хромосоме). Каждая ось (номер оси) соответствует номеру позиции элемента в списке. Точками отсчета на оси являются значения элементов списка. Расположены точки счета на оси в одном заранее выбранном порядке, принятом в качестве базового.

Хромосома, описывающая решение, имеет следующую структуру: $H_i(t) = \{g_{il}(t) | l=1, 2, \dots, n\}$. Значение гена $g_{il}(t)$ равно значению соответствующего элемента списка M_i .

Пусть в качестве базового списка используется список $B = \langle 1, 7, 21, 4, 8, 18 \rangle$. $|B|=6$. Пространство поиска X включает 6 осей: $X_1 - X_6$, в соответствии с числом элементов списка. Каждая ось соответствует позиции списка. Точками отсчета x_{ij} на оси

$X_i = \langle x_{ij} | j=1, 2, \dots, 6 \rangle$ являются упорядоченные значения элементов списка B ,

$X_i = \langle x_{i1}=1, x_{i2}=7, x_{i3}=21, x_{i4}=4, x_{i5}=8, x_{i6}=18 \rangle = \langle 1, 7, 21, 4, 8, 18 \rangle$.

Например: Список $M_i = \langle 21, 8, 7, 1, 8, 4, 18 \rangle$ представляется в виде позиции $H_i = \{x_{13}, x_{25}, x_{32}, x_{41}, x_{54}, x_{66}\}$.

Подобные списки лежат в основе интерпретаций решений в задачах упаковки, разбиения, размещения, планирования работы многопроцессорных вычислительных систем [19] и в ряде других задач.

Суть *процедуры перемещения*, реализуемой ОНМ, заключается в изменения взаимного расположения элементов в списке. Частица p_i перемещается из позиции $H_i(t)$ в направлении позиции в новую позицию $H_i(t+1)$ с новым взаимным расположением элементов в списке.

Введем характеристику, отражающую степень различия между позициями $H_i(t)$ и $H_z(t)$. Для этого произведем сравнение взаимного расположения элементов у всех возможных пар в сравниваемых позициях $H_i(t)$ и $H_z(t)$. Пусть $S_{iz}(t)$ – число пар, у которых взаимное расположение элементов в сравниваемых позициях $H_i(t)$ и $H_z(t)$ не совпадает.

Чем больше $S_{iz}(t)$, тем больше различие между $H_i(t)$ и $H_z(t)$, и наоборот, чем меньше $S_{iz}(t)$, тем различие меньше. Тяготение частицы p_i в позиции $H_i(t)$ к лучшей позиции $H_z(t)$, выражается в стремлении уменьшения различия между $H_i(t)$ и $H_z(t)$, то есть уменьшения показателя $S_{iz}(t)$. Будем считать, чем меньше различие между $H_i(t)$ и $H_z(t)$, тем ближе позиция $H_i(t)$, к лучшей позиции $H_z(t)$.

Модификация позиции $H_i(t)$, то есть переход к позиции $H_i(t+1)$, производится путём выборочных групповых парных перестановок соседних элементов в позиции $H_i(t)$. Модификация позиции $H_i(t)$ выполняется за два такта.

На первом такте формируется множество D_1 непересекающихся пар элементов в позиции $H_i(t)$ таких у которых индекс l -нечетное число: $P_1=\{(g_{il}(t), g_{i,l+1}(t))|l=1,3,5,\dots\}$. Подсчитывается число пар $S_{iz}(t)$ в множестве D_1 , у которых взаимное расположение элементов пары в позиции $H_i(t)$ не совпадает с взаимным расположением этих элементов в позиции $H_z(t)$. Принимается решение о перестановке элементов каждой такой пары в позиции $H_i(t)$, вычисляются показатели $S_{ik}(t)$ и осуществляется модификация $H_i(t)$.

На втором такте формируется множество D_2 непересекающихся пар элементов в позиции $H_i(t)$ у которых индекс l – четное число: $D_2=\{(g_{il}(t), g_{i,l+1}(t))|l=2,4,6,\dots\}$. Подсчитывается число пар $S_{iz}(t)$ в множестве D_2 , у которых взаимное расположение элементов пары в позиции $H_i(t)$ не совпадает с взаимным расположением этих элементов в позиции $H_z(t)$. Принимается решение о перестановке элементов каждой такой пары в позиции $H_i(t)$, вычисляются показатели $S_{ik}(t)$ и формируется позиция $H_i(t+1)$.

Перестановка выполняется с вероятностью

$$P=\alpha \cdot S_{iz}(t) / n,$$

где n – число пар в множестве D .

Отметим, что изменение взаимного расположения пары соседних элементов в позиции не приводит к изменению относительного расположения каждого элемента пары с остальными элементами позиции, а также к изменению взаимного расположения остальных элементов друг относительно друга.

Пример работы процедуры перемещения.

Пусть позиции $H_i(t)$ и $H_z(t)$ имеют вид:

$$H_z(t)=\{1,10,2,3,8\}, H_i(t)=\{1,3,2,10,8\}.$$

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

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

Экспериментальные исследования. На основе рассмотренной гибридной парадигмы разработана программа разбиения **ГЕН-РОЙ РАЗБ**. Для написания программы **ГЕН-РОЙ РАЗБ** был использован язык C++ в среде Microsoft Visual Studio 2010 для ОС Windows. Тестирование проводилось на ЭВМ с процессором Intel Core 2 Duo T6600 2200 МГц, 4 Гб ОЗУ под управлением операционной системы Windows 7.

Для проведения экспериментов программы **ГЕН-РОЙ РАЗБ** была использована процедура синтеза контрольных примеров с известным оптимумом F_{opt} по аналогии с известным методом **ВКУ** (Partitioning Examples with Tight Upper Bound of Optimal Solution) [20]. Оценкой качества служит величина F_{opt}/F – **«степень качества»**, где F – оценка полученного решения.

На основе результатов экспериментальных исследований была построена средняя зависимость степени качества от числа итераций (рис. 4).

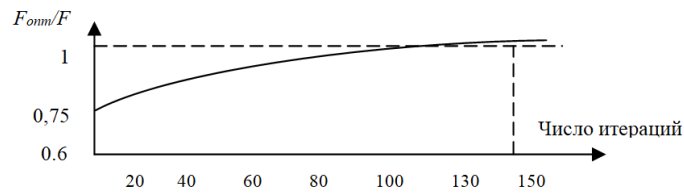


Рис. 4. Зависимость степени качества алгоритма **ГЕН-РОЙ РАЗБ** от числа итераций

Исследования показали, что число итераций, при которых алгоритм нашел лучшее решение лежит в пределах 120–135. Алгоритм сходится в среднем на 130 итерации. В результате проведенных исследований установлено, что качество решений у гибридного алгоритма на 10–12 % лучше качества решений генетического и роевого алгоритмов по отдельности. Эксперименты показали, что увеличение популяции M больше 120 нецелесообразно, т.к. это не приводит к заметному изменению качества. Вероятность получения глобального оптимума составила 0,96. В среднем результат, полученный предложенным алгоритмом на 130-й итерации, отличается от точного на 0.15 процента. В среднем запуск программы обеспечивает нахождения решения, отличающегося от оптимального менее, чем на 2 %. Временная сложность алгоритма при фиксированных значениях M – размер популяции и T – число итераций лежит в пределах $O(n)$. Общая оценка временной сложности лежит в пределах $O(n^2)$ - $O(n^3)$.

Заключение. Предложена композитная архитектура многоагентной системы бионического поиска на основе интеграции роевого интеллекта и генетической эволюции. Связующим звеном такого подхода является единая структура данных, описывающая решение задачи, в виде хромосомы. Интеграция метаэвристик популяционных алгоритмов обеспечивает более широкий обзор пространства 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. В качестве структуры данных для описания позиции используется хромосома, что упрощает интеграцию роевого интеллекта и генетической эволюции. Разработаны новые структуры хромосом для представления решений. В отличие от канонической парадигмы роя частиц для представления решений доступен широкий диапазон графовых структур (маршрут, дерево, двудольный граф, паросочетание, внутренне-устойчивое множество и т.д.). Такой подход является эффективным способом поиска рациональных решений для задач оптимизации, допускающих интерпретацию решений в виде различного рода графовых структур.

В качестве основных структур данных, несущих информацию о решении, используются списки, фактически являющимися интерпретациями решений. В работе используется кодированное представление списков. Переход от кодированного представления к списку производится с помощью декодера.

Разработанные структуры пространства поиска и позиций позволяют отображать: Списки, элементы которых могут иметь два значения 0 или 1; 2. Списки, содержащие фиксированные числа нулей и единиц; Списки с фиксированной суммой значений элементов; Списки, описывающие структуру бинарного дерева;

Списки, задающие последовательность элементов. Эксперименты показали, что качество решений, полученных гибридным алгоритмом на 10–15 % лучше, чем у генетического и роевого алгоритмов. Вероятность получения глобального оптимума составила 0.9. Общая оценка временной сложности при любом подходе к гибридизации не превышает оценки временной сложности генетического алгоритма и лежит в пределах $O(n^2)$ - $O(n^3)$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 448 с.
2. *Wang X.* Hybrid nature-inspired computation method for optimization // Doctoral Dissertation. – Helsinki University of Technology, TKK Dissertations, Espoo 2009. – 161 p.
3. *Raidl G.R.* A Unified View On Hybrid Metaheuristics. In: Lecture Notes In Computer Science, 2006. – Springer, Verlag. – P. 1-12.
4. *Лебедев Б.К., Лебедев О.Б.* Распределение ресурсов на основе гибридных моделей роевого интеллекта // Научно-технический вестник информационных технологий, механики и оптики. – 2017. – Т. 17, № 6. – С. 1063-1073.
5. *Лебедев Б.К., Лебедев О.Б., Лебедев В.Б.* Гибридизация роевого интеллекта и генетической эволюции на примере размещения // Программные продукты, системы и алгоритмы. – 2017. – № 4.
6. *Clerc M.* Particle Swarm Optimization. ISTE, London, UK, 2006.
7. *Kennedy J., Eberhart R.C.* Particle swarm optimization // In Proceedings of IEEE International Conference on Neural Networks. – 1995. – P. 1942-1948.
8. *Blum C., Roli A.* Metaheuristics in combinatorial optimization: overview and conceptual comparison // ACM computing surveys. – 2003. – No. 35. – P. 268-308.
9. *Лебедев Б.К., Лебедев О.Б.* Гибридный биоинспирированный алгоритм решения задачи символьной регрессии // Известия ЮФУ. Технические науки. – 2015. – № 6 (167). – С. 28-41.
10. *Емельянов В.В., Курейчик В.М., Курейчик В.В.* Теория и практика эволюционного моделирования. – М.: Физматлит, 2003. – 412 с.
11. *Лебедев В.Б.* Построение кратчайших связывающих сетей на основе роевого интеллекта // Известия ЮФУ. Технические науки. – 2011. – № 7 (120). – С. 37-44.
12. *Лебедев Б.К., Лебедев О.Б., Лебедева Е.М.* Меметический алгоритм разбиения // Вестник Ростовского государственного университета путей сообщения. – 2017. – № 2 (62). – С. 136-145.
13. *Лебедев Б.К., Лебедев О.Б., Лебедева Е.М.* Разбиение на классы методом альтернативной коллективной адаптации статья // Известия ЮФУ. Технические науки. – 2016. – № 7 (180). – С. 89-101.
14. *Курейчик В.В., Полупанов А.А.* Обзор эволюционных методов оптимизации на основе роевого интеллекта // Известия ЮФУ. Технические науки. – 2011. – № 7 (120). – С. 7-12.
15. *Sha D.Y. and Cheng-Yu.* A hybrid particle swarm optimization for job shop scheduling problem // Computers & Industrial Engineering. – 2006. – P. 791-808.
16. *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.
17. *Лебедев Б.К., Лебедев О.Б., Лебедева Е.М.* Модернизированный муравьиный алгоритм синтеза идентифицированного дерева гильотинного разреза при планировании СБИС // Известия ЮФУ. Технические науки. – 2017. – № 7 (192). – С. 15-28.
18. *Гладков Л.А., Курейчик В.В., Курейчик В.М., Сороколетов П.В.* Биоинспирированные методы в оптимизации: монография. – М.: Физматлит, 2009. – 384 с.
19. *Лебедев Б.К., Лебедев О.Б., Лебедева Е.О.* Роевой алгоритм планирования работы многопроцессорных вычислительных систем // Инженерный вестник Дона. – 2017. – № 3.
20. *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 2003. – P. 88-94.

REFERENCES

1. *Karpenko A.P.* Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: ucheb. posobie [Modern algorithms of search optimization. Algorithms inspired by nature: a tutorial]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2014, 448 p.
2. *Wang X.* Hybrid nature-inspired computation method for optimization: Doctoral Dissertation. Helsinki University of Technology, TKK Dissertations, Espoo 2009, 161 p.
3. *Raidl G.R.* A Unified View on Hybrid Metaheuristics. In: Lecture Notes In Computer Science, 2006. Springer, Verlag, pp. 1-12.
4. *Lebedev B.K., Lebedev O.B.* Raspredelenie resursov na osnove gibridnykh modeley roevogo intellekta [Distribution of resources on the basis of hybrid models of swarm intelligence], *Nauchno-tekhnicheskiy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki* [Scientific and Technical Herald of Information Technologies, Mechanics and Optics], 2017, Vol. 17, No. 6, pp. 1063-1073.
5. *Lebedev B.K., Lebedev O.B., Lebedev V.B.* Gibridizatsiya roevogo intellekta i geneticheskoy evolyutsii na primere razmeshcheniya [Hybridization of swarm intelligence and genetic evolution using the example of placement], *Programmnyye produkty, sistemy i algoritmy* [Software products, systems and algorithms], 2017, No. 4.
6. *Clerc M.* Particle Swarm Optimization. ISTE, London, UK, 2006.
7. *Kennedy J., Eberhart R.C.* Particle swarm optimization, *In Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.
8. *Blum C., Roli A.* Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM computing surveys*, 2003, No. 35, pp. 268-308.
9. *Lebedev B.K., Lebedev O.B.* Gibridnyy bioinspirovannyy algoritm resheniya zadachi simvol'noy regressii [Hybrid bioinspiral algorithm for solving the problem of symbolic regression], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 6 (167), pp. 28-41.
10. *Emel'yanov V.V., Kureychik V.M., Kureychik V.V.* Teoriya i praktika evolyutsionnogo modelirovaniya [Theory and practice of evolutionary modeling]. Moscow: Fizmatlit, 2003, 412 p.
11. *Lebedev V.B.* Postroenie krachayshikh svyazyvayushchikh setey na osnove roevogo intellekta [Construction of the shortest connecting networks on the basis of swarm intelligence], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2011, No. 7 (120), pp. 37-44.
12. *Lebedev B.K., Lebedev O.B., Lebedeva E.M.* Memeticheskyy algoritm razbieniya [The Memetic Algorithm of Partitioning], *Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya* [Bulletin of the Rostov State Transport University], 2017, No. (62), pp. 136-145.
13. *Lebedev B.K., Lebedev O.B., Lebedeva E.M.* Razbienie na klassy metodom al'ternativnoy kollektivnoy adaptatsii stat'ya [Splitting into classes by the method of alternative collective adaptation article], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2016, No. 7 (180), pp. 89-101.
14. *Kureychik, V.V., Polupanov A.A.* Obzor evolyutsionnykh metodov optimizatsii na osnove roevogo intellekta [Overview of evolutionary optimization techniques based on swarm intelligence], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2011, No. 7 (120), pp. 7-12.
15. *Sha D.Y. and Cheng-Yu.* A hybrid particle swarm optimization for job shop scheduling problem, *Computers & Industrial Engineering*, 2006, pp. 791-808.
16. *Lebedev B.K., Lebedev O.B., Lebedeva E.M.* Raspredelenie resursov na osnove gibridnykh modeley roevogo intellekta [Distribution of resources on the basis of hybrid models of swarm intelligence], *Nauchno-tekhnicheskiy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki* [Scientific and Technical Herald of Information Technologies, Mechanics and Optics], 2017, Vol. 17, No. 6, pp. 1063-1073.

17. *Lebedev B.K., Lebedev O.B. Lebedeva E.M.* Modernizirovannyi murav'inyi algoritm sinteza identifikirovannogo dereva gil'otinnogo razreza pri planirovani SBIS [A modernized ant algorithm for the synthesis of the identified guillotine tree tree in the course of VLSI planning], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2017, No. 7 (192), pp. 15-28.
18. *Gladkov L.A., Kureychik V.V., Kureychik V.M., Sorokoletov P.V.* Bioinspirirovannye metody v optimizatsii: monografiya [Bioinspired methods in optimization: monograph]. Moscow: Fizmatlit, 2009, 384 p.
19. *Lebedev B.K., Lebedev O.B. Lebedeva E.O.* Roeffy algoritm planirovaniya raboty mnogoprotseessornykh vychislitel'nykh sistem [Swarm algorithm for planning the work of multiprocessor computer systems], *Inzhenernyi vestnik Dona* [Engineer's Herald of the Don], 2017, No. 3.
20. *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 2003, pp. 88-94.

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

Лебедев Борис Константинович – Южный федеральный университет; e-mail: lebedev.b.k@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 89282897933; кафедра систем автоматизированного проектирования; профессор.

Лебедев Олег Борисович – e-mail: lebedev.ob@mail.ru; тел.: 89085135512; кафедра систем автоматизированного проектирования; доцент.

Лебедева Екатерина Олеговна – e-mail: lebedev.ob@mail.ru; тел.: 89289591426; кафедра систем автоматизированного проектирования; магистрант.

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

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

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

УДК 681.3.06

DOI 10.23683/2311-3103-2018-4-136-145

Д.Ю. Запорожец, Д.В. Заруба

ПАРАЛЛЕЛЬНЫЙ ПОПУЛЯЦИОННЫЙ АЛГОРИТМ*

Предложена архитектура параллельного поиска. Выделены три уровня распараллеливания: микроуровень – распараллеливание задач на уровне выполнения операторов генерации новых альтернативных решений; макроуровень – распараллеливание поиска на уровне генераций; метауровень – распараллеливание процесса поиска на уровне изолированного множества альтернативных решений. В качестве примера, в данной работе предложена реализация параллельного генетического алгоритма с распараллеливанием на микроуровне. Предложена обобщенная архитектура и описаны основные шаги нового параллельного генетического алгоритма с распараллеливанием на микроуровне. Основной идеей данного подхода является инициализация N потоков, в каждом из которых выполняется оператор кроссинговера и, с заданной долей вероятности, операторы мутации и инверсии. Для всех новых решений производится расчет целевой функции в том потоке, в котором данное решение было получено. Алгоритм ожидает окончания всех потоков. Далее производится

* Работа выполнена при финансовой поддержке гранта Президента Российской Федерации для государственной поддержки молодых российских ученых – кандидатов наук № МК-92.2017.8.