

Раздел I. Алгоритмы обработки информации

УДК 004.896

DOI 10.18522/2311-3103-2025-1-6-22

Б.К. Лебедев, О.Б. Лебедев, М.А. Ганжур, М.И. Бесхмельнов

БИОИНСПИРИРОВАННЫЙ ПОИСК В ПОЛНОМ ГРАФЕ СОВЕРШЕННОГО ПАРСОЧЕТАНИЯ МАКСИМАЛЬНОЙ МОЩНОСТИ

Разработана реконфигурируемая архитектура гибридной многоагентной системы поиска решений, базирующиеся на парадигмах роевых алгоритмов. Реконфигурируемая архитектура путем настройки позволяет реализовать следующие методы гибридизации: высокоуровневую и низкоуровневую гибридизацию вложением, типа препроцессор/постпроцессор, ко-алгоритмическую на базе одного или нескольких типов алгоритмов. Предложена методология синтеза совершенного паросочетания минимального веса в полном графе, основанная на базовых принципах гибридизации поисковых, эволюционных процедур. В работе агентами роя являются трансформирующиеся хромосомы, являющиеся генотипами решения. В качестве кода решения используется упорядоченный список множества вершин графа. Разработана структура упорядоченного кода паросочетания, главное достоинство которого заключается в том, что одному решению (паросочетанию) соответствует один код и наоборот. Определены свойства упорядоченного кода и разработаны алгоритмы кодирования и декодирования. Работа гибридной системы начинается с генерации роем пчел случайным образом произвольного множества отличающихся друг от друга решений в виде исходного множества хромосом. Ключевой операцией пчелиного алгоритма является исследование перспективных решений и их окрестностей в пространстве поиска. Разработан метод формирования окрестностей решений с регулируемой степенью подобия и близости между ними. На последующих этапах работы многоагентной системы выполняется поиск решений процедурами, построенными на основе гибридизации роевого и муравьиного алгоритмов. Отличительной особенностью гибридизации является сохранение автономии гибридизируемых алгоритмов. Отметим, что для представления решений в алгоритмах используется единая структура данных, что упрощает стыковку разработанных процедур. Предлагается подход к построению модифицированной парадигмы роя трансформирующихся хромосом. Поиск решений выполняется в аффинном пространстве. В процессе поиска осуществляется перманентные трансформации (переход) хромосом в состоянии с лучшим значением целевой функции решения (градиентная стратегия). Процесс поиска решений итерационный. На каждой итерации осуществляется трансформация (переход) хромосом в состояние с лучшими значениями целевой функции решения. Целью трансформации хромосомы, тяготеющей к лучшей хромосоме, в новое состояние является минимизация степени различия, путем изменения взаимного расположения элементов в упорядоченном списке, что соответствует увеличению веса аффинной связи. Обновленные после трансформации хромосомы являются, в свою очередь, базовыми точками в последующих трансформациях. В результате экспериментов было установлено, что показатели качества разработанных алгоритмов имеют более высокие значения чем в работах, представленных в литературе.

Поисковая оптимизация; методология; декомпозиция; роевой интеллект; муравьиная и пчелиная колонии; трансформирующиеся хромосомы; адаптивное поведение; гибридизация.

B.K. Lebedev, O.B. Lebedev, M.A. Ganzhur, M.I. Beskhmel'nov

BIOINSPIRED SEARCH IN THE COMPLETE GRAPH OF A PERFECT MATCH OF MAXIMUM POWER

A reconfigurable architecture of a hybrid multi-agent decision-making system based on swarm algorithm paradigms has been developed. The reconfigurable architecture allows implementing the following hybridization methods by tuning: high-level and low-level hybridization by nesting, preprocessor/postprocessor type, co-algorithmic based on one or several types of algorithms. A methodology for

synthesizing a perfect matching of minimum weight in a complete graph based on the basic principles of hybridization of search. evolutionary procedures has been proposed. In this paper, the swarm agents are transforming chromosomes, which are the genotypes of the solution. An ordered list of the set of graph vertices is used as the solution code. A structure of an ordered matching code has been developed, the main advantage of which is that one solution (matching) corresponds to one code and vice versa. The properties of the ordered code have been determined and encoding and decoding algorithms have been developed. The hybrid system operation starts with the random generation by a swarm of bees of an arbitrary set of solutions differing from each other in the form of an initial set of chromosomes. The key operation of the bee algorithm is the study of promising solutions and their neighborhoods in the search space. A method for forming neighborhoods of solutions with an adjustable degree of similarity and closeness between them has been developed. At subsequent stages of the multi-agent system operation, solutions are searched for by procedures built on the basis of hybridization of the swarm and ant algorithms. A distinctive feature of hybridization is the preservation of the autonomy of the hybridized algorithms. Note that a single data structure is used to represent solutions in the algorithms, which simplifies the docking of the developed procedures. An approach to constructing a modified paradigm of a swarm of transforming chromosomes is proposed. The search for solutions is performed in an affine space. In the process of searching, permanent transformations (transitions) of chromosomes into states with the best value of the objective function of the solution (gradient strategy) are carried out. The process of finding solutions is iterative. At each iteration, the chromosomes are transformed (transitioned) into states with better values of the objective function of the solution. The purpose of transforming a chromosome that tends to be the best chromosome into a new state is to minimize the degree of difference by changing the mutual arrangement of elements in an ordered list, which corresponds to an increase in the weight of the affine connection. The chromosomes updated after the transformation are, in turn, the base points in subsequent transformations. As a result of the experiments, it was found that the quality indicators of the developed algorithms have higher values than in the works presented in the literature.

Search engine optimization, methodology; decomposition; swarm intelligence; ant and bee colonies; transforming chromosomes; adaptive behavior; hybridization.

Введение. Среди комбинаторно-логических задач на графах важное место занимает проблема определения паросочетаний. Паросочетанием графа $G=(X,U)$ называется подмножество таких рёбер $Q \subset U$, что любые два ребра $u_i, u_j \in Q$ не имеют общих вершин, т.е. не смежны [1]. Паросочетание максимальной мощности определяется как паросочетание, включающее максимальное число рёбер [2]. Алгоритмы решения данной задачи применяются при проектировании инженерных сетей, коммуникаций, построения систем поддержки принятия решений в неопределённых условиях, проектировании СБИС и т.п. Задачи такого типа относятся к переборным задачам с экспоненциальной временной сложностью. В этой связи необходимо разрабатывать различные эвристики для построения алгоритмов с полиномиальной временной сложностью [3]. Группу «жадных» (градиентных) алгоритмов, составляют алгоритмы, действующие по принципу «максимальный выигрыш на каждом шаге» [4]. В группу наиболее эффективных и широко распространённых алгоритмов для построения максимального паросочетания входят: алгоритм Куна [5], алгоритм Хопкрофта-Карпа [3], алгоритм Радо-Эдмонса [6], алгоритм Форда-Фалкерсона [7], построенный по схеме поиска в глубину. Существуют алгоритмы определения паросочетаний в графе, основанные на использовании потоков в сетях [8], эволюционного моделирования [9], генетического поиска [10] и других эвристик, которые обеспечивают приемлемые результаты при решении задач малой и средней сложности. Часто эта процедура используется в итерационных адаптивных структурах [11]. Это предъявляет повышенные требования к качеству и времени решения задачи нахождения максимального паросочетания. Возникшие потребности в решении задач большой и очень большой размерности являются побудительным мотивом исследований и разработок новых эффективных алгоритмов. Анализ литературы показывает, что наиболее успешными в этих условиях являются математические методы, в которых заложены принципы природных механизмов принятия решений [12].

К таким методам можно отнести, прежде всего, методы адаптивного поведения [13], метод эволюционного моделирования [14], гибридные модели [15, 16], алгоритмы роевого интеллекта [17] и муравьиные алгоритмы [18]. Анализ методов решения сложных

прикладных задач показывает, что применение любого одного алгоритма оптимизации (как классического, так и популяционного) далеко не всегда приводит к успеху [19]. Для повышения эффективности поведенческих методов глобальной оптимизации в настоящее время используют два основных подхода – гибридизация и метаоптимизация. Интеграция метаэвристик популяционных алгоритмов обеспечивает более широкий обзор пространства поиска и более высокую вероятность локализации глобального экстремума задачи. Разработка общей структуры гибридных алгоритмов требует решение следующих задач: выбор гибридируемых алгоритмов; определение общей структуры поискового процесса; вычленение функций в каждом гибридируемом алгоритме, которые должны выполняться в составе гибридного поискового алгоритма; разработка методов трансформации данных при сращивании гибридируемых алгоритмов. Исследования ряда авторов показали, что использование гибридизации повышает эффективность поисковых алгоритмов оптимизации в среднем на 3-5% [12]. Связующим звеном гибридизации является структура данных, описывающая интерпретации решений проблемно-ориентированных задач. Разработка принципов синтеза пространства поиска решений, позволяет строить эффективные поисковые процедуры. В работе излагается методика решения задачи нахождения максимального паросочетания в графе, основанная на моделировании адаптивного поведения биологических систем [18].

Постановка задачи нахождения паросочетания в графе. Пусть задан полный граф $G=(X,U)$, где $X=\{x_i|i=1,2,\dots,n\}$ – множество вершин, а $U=\{u_k|k=1,2,\dots,n_u\}$ – множество ребер $u_k=(x_i, x_j)$. Каждое ребро $u_k=(x_i, x_j)$ связывает пару вершин.

Для каждой вершины x_i задается вес ψ_i . Вес вершин задается множеством $\Psi=\{\psi_i|i=1,2,\dots,n\}$, а вес ребер – множеством $\Phi_i=\{\phi_k|k=1,2,\dots,m\}$.

Вес ϕ_k ребра $u_k=(x_i, x_j)$ графа G , связывающего вершины x_i и x_j , определяется как сумма весов вершин x_i и x_j : $\phi_k=(\psi_i + \psi_j)$.

В качестве пространства поиска решений служит полный граф $G=(X,U)$.

Паросочетание Q в G – это множество попарно несмежных ребер, т.е. ребер, не имеющих общих вершин.

Основное свойство паросочетания Q в графе $G=(X,U)$ заключается в том, что любые два ребра $u_i \in Q$ и $u_j \in Q$ не имеют общих вершин, т.е. не смежны. Совершенным паросочетанием (или 1-фактором) называется паросочетание, в котором участвуют все вершины графа. То есть любая вершина графа инцидентна ровно одному ребру [3].

Обозначим как F суммарный вес ребер, входящих в паросочетание Q графа G .

Критерий оптимизации – суммарная стоимость ребер, входящих в совершенное паросочетание Q графа G . Цель оптимизации минимизация критерия:

$$F = \sum_k \phi_k (k|u_k \in Q) \rightarrow \min.$$

Методика и структура представления паросочетания в виде упорядоченной последовательности. Во многих приложениях используется кодированное представление решения: в терминах генетического алгоритма в виде хромосом, которые являются кодами решений. В качестве простейшей структуры данных, несущей информацию о решении, используется упорядоченный список (УС), задающий порядок декодирования. Решение представляется в виде кода, являющегося генотипом решения. Переход от кода к решению производится с помощью декодера. Декодер – оператор, позволяющий перейти от кода решения задачи (генотипа) к фенотипу. Фактически код решения (в том числе список) является интерпретацией решения в конкретной предметной области [7].

Пусть на полном графе $G=(X,U)$, $|X|=n$ с помощью какого-либо алгоритма построено решение – совершенное паросочетание, которое представляется в виде упорядоченного множества ребер $Q=\langle u_t|t=1,\dots,n_q \rangle$, $u_t=(x_i, x_j)$. Если n – четное число, то мощность совершенного паросочетания полного графа равна $n/2$. Пусть $Q=\langle u_1, u_2, u_3, u_4 \rangle$, $u_1=\langle x_4, x_2 \rangle$, $u_2=\langle x_7, x_1 \rangle$, $u_3=\langle x_6, x_5 \rangle$, $u_4=\langle x_3, x_8 \rangle$. Фактически вектор Q является чередующейся цепью в графе G – простой путь в котором ребра поочередно принадлежат/не принадлежат паросочетанию, а начальная и конечная вершины не принадлежат паросочетанию.

Вектор Q после замены u_i на (x_i, x_j) преобразуется в вектор $R = \langle r_l | l=1, \dots, n_q \rangle$ в котором элементы расположены по парам таким образом, что элементы пары $u_k = (x_i, x_j)$ расположены в позициях $P = \langle p_i | i=1, 2, \dots, n \rangle$ списка R : x_i в позиции $p_i = (2k-1)$ и x_j в позиции $p_j = 2k$. p_i – это позиция вершины x_i в векторе R .

Вектор $R = \langle x_4, x_2, x_7, x_1, x_6, x_5, x_3, x_8 \rangle$ – код решения задачи паросочетания. $P = \langle p_i | i=1, 2, \dots, n \rangle$; $p_1=4$; $p_2=2$; $p_3=7$; $p_4=1$; $p_5=6$; $p_6=5$; $p_7=3$; $p_8=8$.

В работе в качестве кода решения R используется упорядоченный список множества вершин графа G , расположенных в позициях $P = \langle p_i | i=1, 2, \dots, n \rangle$ вектора R , соответствующий чередующейся цепи.

Кодом решения является соответствие между множеством вершин $X = \langle x_i | i=1, 2, \dots, n \rangle = \langle x_1, x_2, x_3, x_4, \dots, x_n \rangle$ и множеством позиций $P = \langle p_i | i=1, 2, \dots, n \rangle$ вектора R . p_i – позиция вектора R , в которой расположена вершина x_i . Декодирование паросочетания производится путем последовательного просмотра кода решения (вектора R).

Отметим, что полученный в результате декодирования порядок, в котором сформировано множество ребер, входящих в состав паросочетания, а также порядок, в котором пара вершин (x_i, x_j) , инцидентных ребру u_k , расположена в векторе R , не изменяет сущности решения. Число паросочетаний в графе известно как индекс Хосойи [1]. Вычисление этого числа является NP – задачей [1].

Недостаток этого подхода заключается в том, что перестановка в векторе R любой пары вершин, инцидентных ребру, и перестановка последовательности ребер в списке паросочетания Q не изменяет решения – паросочетания, что приводит к увеличению пространства поиска и к повышенным затратам при поиске решения.

Пусть имеется некоторое решение задачи построения совершенного паросочетания Q в полном графе $G = (X, U)$, $|X| = n$. $|Q| = n/2$. Поскольку порядок сформированных $(n/2)$ ребер, и порядок, в котором вершины размещены в парах, не имеет значения, то одному коду решения будет соответствовать число списков P , равное $(n_b/2)! \cdot 2 \cdot n_b/2 = (n_b/2)! \cdot n_b$. В связи с этим при поиске решения актуальна проблема синтеза и просмотра только одного списка, соответствующего одному конкретному решению (одному коду).

Например: решение $u_1 = \langle x_7, x_1 \rangle$, $u_2 = \langle x_3, x_8 \rangle$, $u_3 = \langle x_4, x_2 \rangle$, $u_4 = \langle x_6, x_5 \rangle$, полученное в результате декодирования чередующейся цепи (списка) $R_2 = \langle x_7, x_1, x_3, x_8, x_4, x_2, x_6, x_5 \rangle$, и решение $u_1 = \langle x_6, x_5 \rangle$, $u_2 = \langle x_3, x_8 \rangle$, $u_3 = \langle x_7, x_1 \rangle$, $u_4 = \langle x_4, x_2 \rangle$, полученное в результате декодирования $R_3 = \langle x_6, x_5, x_3, x_8, x_7, x_1, x_4, x_2 \rangle$, эквивалентны решению, полученному при декодировании списка R_1 . В приведенном примере одному решению соответствует 3 списка.

Рассмотрим на примерах переход от неупорядоченного списка к упорядоченному путем двух трансформаций.

1. $R_1 = \langle x_4, x_2, x_7, x_1, x_6, x_5, x_3, x_8 \rangle$.
 $u_1 = \langle x_4, x_2 \rangle$, $u_2 = \langle x_7, x_1 \rangle$, $u_3 = \langle x_6, x_5 \rangle$, $u_4 = \langle x_3, x_8 \rangle$.
 $u_1 = \langle x_2, x_4 \rangle$, $u_2 = \langle x_1, x_7 \rangle$, $u_3 = \langle x_5, x_6 \rangle$, $u_4 = \langle x_3, x_8 \rangle$.
 $u_1 = \langle x_1, x_7 \rangle$, $u_2 = \langle x_2, x_4 \rangle$, $u_3 = \langle x_3, x_8 \rangle$, $u_4 = \langle x_5, x_6 \rangle$.
 $R_1^* = \langle x_1, x_7, x_2, x_4, x_3, x_8, x_5, x_6 \rangle$.
2. $R_2 = \langle x_7, x_1, x_3, x_8, x_4, x_2, x_6, x_5 \rangle$.
 $u_1 = \langle x_7, x_1 \rangle$, $u_2 = \langle x_3, x_8 \rangle$, $u_3 = \langle x_4, x_2 \rangle$, $u_4 = \langle x_6, x_5 \rangle$.
 $u_1 = \langle x_1, x_7 \rangle$, $u_2 = \langle x_3, x_8 \rangle$, $u_3 = \langle x_2, x_4 \rangle$, $u_4 = \langle x_5, x_6 \rangle$.
 $u_1 = \langle x_1, x_7 \rangle$, $u_2 = \langle x_2, x_4 \rangle$, $u_3 = \langle x_3, x_8 \rangle$, $u_4 = \langle x_5, x_6 \rangle$.
 $R_2^* = \langle x_1, x_7, x_2, x_4, x_3, x_8, x_5, x_6 \rangle$.
3. $R_3 = \langle x_6, x_5, x_3, x_8, x_7, x_1, x_4, x_2 \rangle$.
 $u_1 = \langle x_6, x_5 \rangle$, $u_2 = \langle x_3, x_8 \rangle$, $u_3 = \langle x_7, x_1 \rangle$, $u_4 = \langle x_4, x_2 \rangle$.
 $u_1 = \langle x_5, x_6 \rangle$, $u_2 = \langle x_3, x_8 \rangle$, $u_3 = \langle x_1, x_7 \rangle$, $u_4 = \langle x_2, x_4 \rangle$.
 $u_1 = \langle x_1, x_7 \rangle$, $u_2 = \langle x_2, x_4 \rangle$, $u_3 = \langle x_3, x_8 \rangle$, $u_4 = \langle x_5, x_6 \rangle$.
 $R_3^* = \langle x_1, x_7, x_2, x_4, x_3, x_8, x_5, x_6 \rangle$.

После первой трансформации упорядочиваются элементы в каждой паре: значение первого элемента пары больше значения второго элемента пары.

После второй трансформации последовательность пар упорядочивается по возрастанию значений первых элементов пар.

Как видно из примеров три неупорядоченных списка $R_1 = \langle x_4, x_2, x_7, x_1, x_6, x_5, x_3, x_8 \rangle$, $R_2 = \langle x_7, x_1, x_3, x_8, x_4, x_2, x_6, x_5 \rangle$, $R_3 = \langle x_6, x_5, x_3, x_8, x_7, x_1, x_4, x_2 \rangle$ эквивалентны одному упорядоченному списку $R^* = \langle x_1, x_7, x_2, x_4, x_3, x_8, x_5, x_6 \rangle$, являющемуся кодом.

И наоборот коду $R^* = \langle x_1, x_7, x_2, x_4, x_3, x_8, x_5, x_6 \rangle$ соответствует решение:

$$R = \langle x_1, x_7, x_2, x_4, x_3, x_8, x_5, x_6 \rangle. u_1 = \langle x_1, x_7 \rangle, u_2 = \langle x_2, x_4 \rangle, u_3 = \langle x_3, x_8 \rangle, u_4 = \langle x_5, x_6 \rangle.$$

В работе предложена структура упорядоченного кода (УК) паросочетания главное достоинство которого заключается в том, что одному решению (паросочетанию) соответствует один код и наоборот.

В качестве генотипа решения (паросочетания) служит упорядоченный список R в котором расположены вершины x_i исходного множества $X = \langle x_i | i = 1, 2, \dots, n \rangle = \langle x_1, x_2, x_3, x_4, \dots, x_n \rangle$.

Упорядоченный список R обладает следующими свойствами:

- ◆ элементы списка R – вершины графа G располагающиеся в позициях списка R от 1 до n , где n четное число, равно числу вершин полного графа $G = (X, U)$;
- ◆ список R формируется в виде упорядоченной последовательности $n/2$ пар вершин (x_i, x_j) таких, что $i < j$. Каждая пара (x_i, x_j) соответствует ребру u_k , входящему в паросочетание;
- ◆ вершины пары $u_k = (x_i, x_j)$ расположены в соседних позициях списка R : x_i в позиции $p_i = (2k-1)$ и x_j в позиции $p_j = 2k$. $p_j = p_i + 1$;
- ◆ для любых двух пар $u_k = (x_i, x_j)$ и $u_l = (x_m, x_n)$ последовательности R , если $l > k$, то номер позиции p_m , в которой находится x_m , больше номера позиции p_i , в которой находится x_i , $p_m > p_i$;
- ◆ первым элементом списка R всегда является x_1 ;
- ◆ в составе любой пары (x_i, x_j) элемент x_n не может быть первым.

Процедура формирования кода произвольного паросочетания в виде упорядоченного списка вершин. Построение упорядоченного списка, обладающего выше перечисленными свойствами, выполняется по шагам на базе опорного упорядоченного вектора $V = \langle x_1, x_2, x_3, \dots, x_n \rangle$ вершин графа $G = (X, U)$, $|X| = n$ с динамически изменяемым составом. Начальный состав вектора V включает все вершины от x_1 до x_n . Отметим, что, если число вершин n четное, то число ребер, входящих в совершенное паросочетание полного графа равно $n/2$. Если число вершин n нечетное, то в состав графа $G = (X, U)$ включается вершина с нулевой локальной степенью.

На каждом шаге t процесса формирования кода в обновленном составе вектора $V(t)$ выбирается очередная пара $u_t = (x_i, x_j)$ вершин для включения в упорядоченную последовательность (УП) R . На шаге t процесс выбора очередной пары ребра $u_t \in U$ включает две стадии. На первой стадии выбирается первый элемент очередной пары, на второй стадии выбирается второй элемент пары. После выбора двух элементов очередной пары они удаляются из опорного упорядоченного вектора $V(t)$.

Выбор ребра $u_t \in U$ включает две стадии. На первой стадии шага t в качестве первого элемента формируемой пары $u_t = (x_i, x_j)$ выбирается первый в векторе $V(t)$ элемент x_i , который помещается в позицию $p_i = (2t-1)$ вектора R , и удаляется из вектора $V(t)$. На второй стадии шага t в качестве второго элемента формируемой пары вероятностным способом в векторе $V(t)$ выбирается элемент $x_j \in V(t)$, который помещается в позицию $i = 2t$ вектора R и удаляется из вектора $V(t)$.

Пример. Пусть $n=10$, опорный вектор $V = \langle x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \rangle$. Число ребер совершенного паросочетания в полном графе равно $n/2=5$. t – номер шага, S_1 – первая стадия, S_2 – вторая стадия.

$$t=1.$$

S_1 . Выбор 1 элемента $A1 = x_1$ в $V(t)$.

Включение x_1 в $R(t) = \langle x_1 \rangle$. Удаление x_1 из $V(t)$.

$$V(t) = \langle x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \rangle.$$

S_2 . Случайный выбор 2-го элемента $A2 = x_5$ в $V(t)$.

Включение x_5 в список $R(t) = \langle x_1, x_5 \rangle$. Удаление x_5 из $V(t)$. $t=t+1$

$$V(t) = \langle x_2, x_3, x_4, x_6, x_7, x_8, x_9, x_{10} \rangle.$$

$t=2$.

S_1 . Выбор 1 элемента $A1 = x_2$ в $V(t)$.

Включение x_2 в $R(t) = \langle x_1, x_5, x_2 \rangle$. Удаление x_2 из $V(t)$.

$V(t) = \langle x_3, x_4, x_6, x_7, x_8, x_9, x_{10} \rangle$.

S_2 . Случайный выбор 2 элемента $A2 = x_3$ в $V(t)$.

Включение x_3 в $R(2) = \langle x_1, x_5, x_2, x_3 \rangle$. Удаление x_3 из $V(t)$. $t=t+1$.

$V(t) = \langle x_4, x_6, x_7, x_8, x_9, x_{10} \rangle$.

$t=3$.

S_1 . Выбор 1 элемента $A1 = x_4$ в $V(t)$.

Включение x_4 в $R(t) = \langle x_1, x_5, x_2, x_3, x_4 \rangle$. Удаление x_4 из $V(t)$.

$V(t) = \langle x_6, x_7, x_8, x_9, x_{10} \rangle$.

S_2 . Случайный выбор элемента $A2 = x_{10}$ в $V(t)$.

Включение x_{10} в $R(t) = \langle x_1, x_5, x_2, x_3, x_4, x_{10} \rangle$. Удаление x_{10} из $V(t)$.

$V(t) = \langle x_6, x_7, x_8, x_9 \rangle$.

$t=4$.

S_1 . Выбор 1 элемента $A1 = x_6$ в $V(4)$.

Включение x_6 в $R(t) = \langle x_1, x_5, x_2, x_3, x_4, x_{10}, x_6 \rangle$. Удаление x_6 из $V(t)$.

$V(t) = \langle x_7, x_8, x_9 \rangle$.

S_2 . Случайный выбор 2 элемента $A2 = x_9$ в $V(t)$.

Включение x_9 в $R(t) = \langle x_1, x_5, x_2, x_3, x_4, x_{10}, x_6, x_9 \rangle$. Удаление x_9 из $V(t)$.

$V(t) = \langle x_7, x_8 \rangle$.

$t=5$.

S_1 . Выбор 1 элемента $A1 = x_7$ в $V(t)$.

Включение x_7 в $R(t) = \langle x_1, x_5, x_2, x_3, x_4, x_{10}, x_6, x_9, x_7 \rangle$. Удаление x_7 из $V(t)$.

$V(t) = \langle x_8 \rangle$.

S_2 . Включение x_8 в $R(t) = \langle x_1, x_5, x_2, x_3, x_4, x_{10}, x_6, x_9, x_7, x_8 \rangle$. Удаление x_8 из $V(t)$.

$V(t) = \langle \rangle$.

Построенный код имеет вид в $R = \langle x_1, x_5, x_2, x_3, x_4, x_{10}, x_6, x_9, x_7, x_8 \rangle$.

Коду R соответствует паросочетание

$Q = \langle u_1, u_2, u_3, u_4, u_5 \rangle, u_1 = \langle x_1, x_5 \rangle, u_2 = \langle x_2, x_3 \rangle, u_3 = \langle x_4, x_{10} \rangle, u_4 = \langle x_6, x_9 \rangle, u_5 = \langle x_7, x_8 \rangle$.

В работе рассматриваются реконфигурируемая архитектура гибридной многоагентной системы поиска решений, базирующаяся на парадигмах роевых алгоритмов. Реконфигурируемая архитектура путем настройки позволяет реализовать следующие методы гибридизации: высокоуровневую и низкоуровневую гибридизацию вложением, типа препроцессор/постпроцессор, ко-алгоритмическую на базе одного или нескольких типов алгоритмов.

Гибридная ко-эволюционная многоагентная система поисковой оптимизации на базе метаэвристик роевых алгоритмов. Анализ методов решения сложных прикладных задач показывает, что применение любого одного алгоритма оптимизации (как классического, так и популяционного) далеко не всегда приводит к успеху [9, 10, 20]. Проблема, с которой сталкиваются разработчики эволюционных алгоритмов, заключается в снижении разнообразия популяции и сужение пространства поиска, что уменьшает вероятность нахождения эффективных решений.

Одним из широко распространенных подходов является гибридизация поисковых алгоритмов. Заложенные в каждой модели метаэвристики при их интеграции усиливают возможности и эффективность алгоритмов. Для поддержки разнообразия популяции решений, расширения области поиска решений разработана архитектура многоагентной биоинспирированной системы в основу которой положен иерархический мультипопуляционный подход.

Разработанный алгоритм решения задачи построения пасочетания, использует архитектуру бионического поиска, и состоит из объединенных процедур роя пчел (РП), роя хромосом (РХ), муравьиного алгоритма (МА), моделирования отжига (МО), что позволяет осуществлять выход из «локальных ям» и увеличивает сходимость алгоритма.

Прежде всего, решающее значение имеет интеграция этапов проектирования в связанный единой методологией маршрут, построенный на единой платформе.

Предложена методология синтеза поисковых алгоритмов, основанных на базовых принципах эволюционных алгоритмов.

Разработка общей структуры гибридных алгоритмов требует решение следующих задач: выбор гибридируемых алгоритмов; определение общей структуры поискового процесса; вычленение функций в каждом гибридируемом алгоритме, которые должны выполняться в составе гибридного поискового алгоритма; разработка методов трансформации данных при сращивании гибридируемых алгоритмов.

Процесс построения совершенного паросочетания выполняется на следующих этапах:

1. Генерация исходного роя хромосом $H = \{H_i | i = 1, 2, \dots, n\}$.
2. Модификация роя хромосом $H = \{H_i | i = 1, 2, \dots, n\}$ базовыми процедурами пчелиного и роевого алгоритмов.
3. Построения совершенного паросочетания гибридизацией роевого и муравьиного алгоритмов.

На рис. 1 представлена структура гибридной многоагентной системы построения совершенного паросочетания, базирующаяся на гибридизации механизмов муравьиной колонии, роевого и пчелиного алгоритмов. Отличительной особенностью гибридизации является сохранение автономии гибридируемых алгоритмов. Связующим звеном гибридизации является структура данных, описывающая интерпретации решений.

В качестве кода решения используется упорядоченный список R множества вершин графа G , расположенных в позициях $P = \langle p_i | i = 1, 2, \dots, n \rangle$ вектора R , принципы построения которого описаны в выше рассмотренных разделах.

Кодом решения является соответствие между множеством вершин $X = \langle x_i | i = 1, 2, \dots, n \rangle = \langle x_1, x_2, x_3, x_4, \dots, x_n \rangle$ и множеством позиций $P = \langle p_i | i = 1, 2, \dots, n \rangle$ вектора R . p_i – позиция в R , в которой размещена вершина x_i .

Формирование паросочетания производится путем последовательного просмотра кода решения (вектора R).

В алгоритмах, входящих в состав многоагентной системы, упорядоченный список R представляется в виде хромосомы $H_i = \{h_{ij} | j = 1, 2, \dots, n\}$.

Процедуры первого и второго этапов системы базируются на механизмах адаптивного поведения пчелиной колонии. Работа гибридной системы начинается с генерации роем пчел случайным образом произвольного множества отличающихся друг от друга решений в виде исходного множества хромосом $H^0 = \{H_i | i = 1, 2, \dots, n\}$.

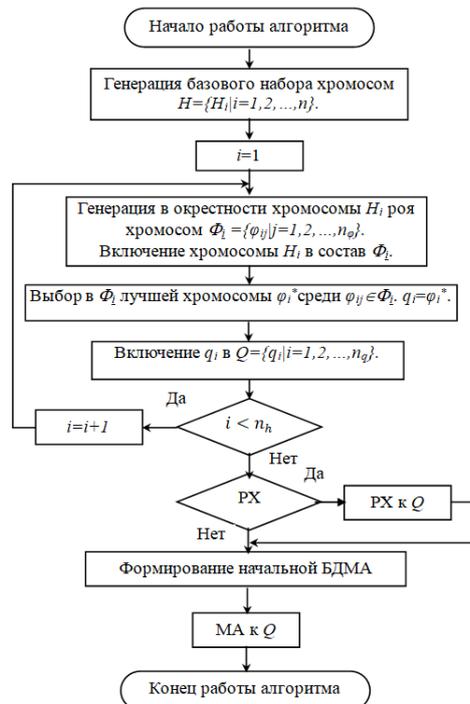


Рис. 1. Структура многоагентной системы построения паросочетания

Хромосома $H_i(t) = \{g_{il} | l=1, 2, \dots, n_i\}$ представляет собой совокупность n генов g_{il} . Значение гена g_{kil} равно значению соответствующего элемента списка R .

Для каждого решения h_i вычисляется значение целевой функции F_i . В множестве H^0 выбираются n_θ лучших решений, которые включаются в множество базовых решений (позиций) H .

При втором подходе реализуется вероятностный выбор H . Вероятность $p(H_i)$ выбора агентом фуражиром базовой хромосомы $H_i \in H^0$ пропорциональна значению целевой функции F_i в этой позиции и определяется как:

$$p(H_i) = F_i / \sum_i (F_i).$$

Далее в окрестности каждой хромосомы $H_i \in H$ формируется рой хромосом:

$$\Phi_i = \{\varphi_{ij} | j=1, 2, \dots, n_\theta\}.$$

Ключевой операцией пчелиного алгоритма является исследование перспективных решений H_i и их окрестностей $\Phi_i = \{\varphi_{ij} | j=1, 2, \dots, n_\theta\}$ в пространстве поиска. Остановимся на понятии окрестности. Смысл, изначально вкладываемый в понятие окрестности, заключается в том, что решения, лежащие в окрестности некоторого базового решения, обладают высокой степенью подобия и, как правило, незначительно отличаются друг от друга.

В работе предлагается метод формирования окрестностей Φ_i решений с регулируемой степенью подобия и близости между ними. Основное требование, предъявляемое к вновь сгенерированному решению φ_{ij} , лежащего в θ – окрестности Φ_i базового решения h_i , заключается в том, что оно должно быть представлено в виде кода, структура которого описана выше и отвечать свойствам упорядоченного списка R . Для удовлетворения требования разработаны механизмы специальной процедуры синтеза решений φ_{ij} в окрестности Φ_i базового решения H_i .

Формирование нового решения φ_{ij} , лежащего в θ – окрестности базового решения h_i производится путём θ выборочных (случайных) парных перестановок элементов в базовом решении h_i , не нарушающих структуры и свойств упорядоченного списка R .

Возможны два типа перестановок в упорядоченном списке R , удовлетворяющих вышеперечисленным условиям базового решения и сохраняющих свойства упорядоченного списка.

Будем считать, что решение $\varphi_{ij} \in \Phi_i$ лежит в θ – окрестности решения H_i , если φ_{ij} получено путём θ случайных парных перестановок соседних элементов в упорядоченном списке, соответствующем H_i .

В каждое Φ_i дополнительно включается H_i . В составе упорядоченного списка R_v , соответствующего решению H_v , произвольно выбираются две пары элементов: $\langle x_i, x_j \rangle$ и $\langle x_k, x_l \rangle$, соответствующих некоторой паре ребер. В соответствии со свойствами упорядоченной последовательности R_v : $i < j$, $k < l$, $k > i$.

1. Если $(j > k)$, то x_j можно поменять местами с x_l .

Например: в списке $R_v = \langle x_1, x_5, x_2, x_3, x_4, x_{10}, x_6, x_9, x_7, x_8 \rangle$ выбраны две пары: $u_1 = \langle x_1, x_5 \rangle$, $u_2 = \langle x_2, x_3 \rangle$. x_5 и x_3 отвечают условию 1. После перестановки в R_v пары примут вид: $u_1 = \langle x_1, x_3 \rangle$, $u_2 = \langle x_2, x_5 \rangle$.

Новые пары вершин отвечают свойствам упорядоченной последовательности.

Новый список примет вид $R_{v1} = \langle x_1, x_3, x_2, x_5, x_4, x_{10}, x_6, x_9, x_7, x_8 \rangle$.

2. Если $(k > i) \& (l > j)$, то x_j можно поменять местами с x_k .

Например: $u_1 = \langle x_2, x_3 \rangle$, $u_2 = \langle x_4, x_{10} \rangle$. x_4 и x_{10} отвечают условию 2.

После перестановки $u_1 = \langle x_2, x_4 \rangle$, $u_2 = \langle x_3, x_{10} \rangle$.

Новый список примет вид $R_{v1} = \langle x_1, x_5, x_2, x_4, x_3, x_{10}, x_6, x_9, x_7, x_8 \rangle$.

На последующих этапах работы многоагентной системы выполняется поиск решений процедурами, построенными на основе гибридизации роевого и муравьиного алгоритмов. Отличительной особенностью гибридизации является сохранение автономии гибридизируемых алгоритмов. Отметим, что для представления решений в алгоритмах используется единая структура данных, что упрощает стыковку разработанных процедур.

Реконфигурируемая архитектура путем настройки позволяет реализовать следующие маршруты оптимизации.

1 вариант (рис. 1). Генерация исходного роя хромосом $H = \{H_i | i = 1, 2, \dots, n\}$.

В окрестности каждой H_i формируется множество решений $\Phi_i = \{\varphi_{ij} | j = 1, 2, \dots, n_\varphi\}$;

В каждом множестве Φ_i выбирается лучшая хромосома φ_i^* с лучшей оценкой F_i .

На базе лучших решений φ_i^* формируется рой хромосом $Q = \{q_i | i = 1, 2, \dots, n_q\}$. q_i соответствует φ_i^* .

Сформированный рой $Q = \{q_i | i = 1, 2, \dots, n_q\}$ в соответствии с метаэвристикой роя агентов эволюционирует в рой Q^o . Далее выполняется МА.

2 вариант (рис. 2). Генерация исходного роя хромосом $H = \{H_i | i = 1, 2, \dots, n\}$.

В окрестности каждой H_i формируется множество решений $\Phi_i = \{\varphi_{ij} | j = 1, 2, \dots, n_\varphi\}$.

Каждый рой Φ_i эволюционирует в соответствии с метаэвристикой роя агентов в рой Φ_i^o .

В каждом множестве Φ_i^o выбирается лучшая хромосома φ_i^* с лучшей оценкой F_i .

На базе лучших решений φ_i^* формируется рой хромосом $Q = \{q_i | i = 1, 2, \dots, n_q\}$. q_i соответствует φ_i^* .

Рой $Q = \{q_i | i = 1, 2, \dots, n_q\}$ в соответствии с метаэвристикой роя агентов эволюционирует в рой Q^o .

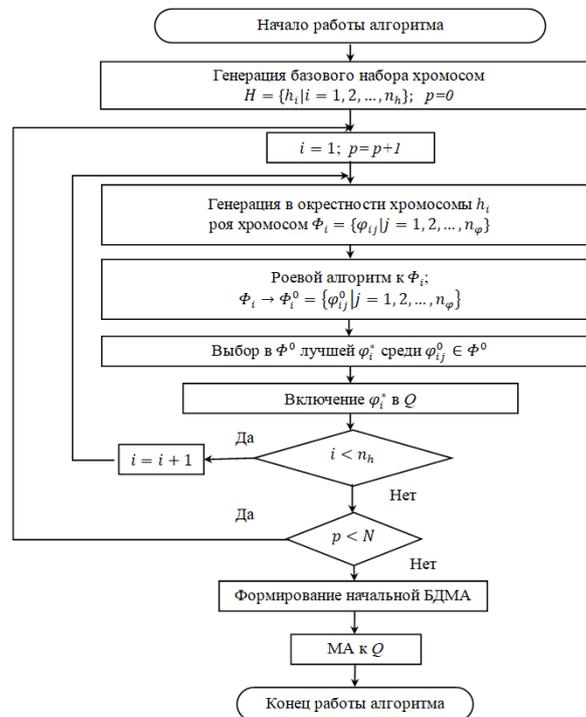


Рис. 2. Структура системы построения паросочетания (2-й вариант)

Модифицированная парадигм роя трансформирующихся хромосом (РТХ).

В работе предлагается подход к построению модифицированной парадигмы роя трансформирующихся хромосом (РТХ) $H = \{H_i | i = 1, 2, \dots, n\}$, $H_i \in H$, $H_i = \{h_{il} | l = 1, 2, \dots, n_l\}$ с целочисленными значениями генов h_{il} схожий с эволюционными алгоритмами. Этот метод моделирует роевое и стайное поведение животных [1]. В отличие от канонических популяционных методов РТХ работает с одной статической популяцией, члены которой постепенно с появлением информации о пространстве поиска улучшаются. Данный метод представляет собой вид направленной мутации (directed mutation). Решения в виде хромосом в РТХ мутируют в направлении наилучших найденных решений. Хромосомы в РТХ никогда не умирают (т.к. нет селекции).

Хромосома H_i является генотипом некоторого решения (фенотипа Φ_i). Оценка целевой функции фенотипа Φ_i является оценкой целевой функции генотипа H_i . Состояние хромосомы H_i полностью определяет состояние, в котором находится фенотип Φ_i .

Для организации поисковой процедуры необходимо представление объекта оптимизации, наличие операторов, преобразующих одно решение в другое (т.е. осуществляющих переход в пространстве решений от одного решения к другому), а также наличие стратегии поиска в пространстве решений. Какие изменения и на сколько – определяет стратегия поиска. В качестве объекта оптимизации рассматривается хромосома.

Разработана архитектура многоагентной системы, выполняющая поиск решения в аффинном пространстве [16–19], используя в качестве модели роевого интеллекта рой трансформирующихся хромосом.

В качестве аффинно-релаксационной модели (АРМ) роя хромосом используется граф G_w , вершины которого соответствуют хромосомам роя, а ребра соответствуют аффинным связям между хромосомами в аффинном пространстве.

В каноническом методе роя частиц [1, 2] используются геометрические меры близости решений. Решение, соответствующее частице, полностью определяется геометрическими параметрами позиции, в которой размещена частица. Изменение решения реализуется путем перехода частицы в новую позицию. Средство изменения решения представлено в виде универсального аналитического выражения синтеза геометрических параметров новой позиции (координаты в пространстве поиска) в предположении (с надеждой), что решение в данной позиции будет лучше (что не гарантировано).

В методе роя трансформирующиеся хромосом агентами роя являются генотипы решений [4]. В качестве меры близости (сходства) двух хромосом используется аффинность – мера эквивалентности, родственности (сходства, близости) двух хромосом. Вес ребра в графе G_w , связывающего две вершины графа G_w , соответствующие двум хромосомам H_i и H_j , равен аффинности, количественно описывающей степень родственности между H_i и H_j . Эквивалентом оценки, степени родственности (аффинность) двух хромосом H_i и H_j служит параметр степень различия двух хромосом H_i и H_j . В качестве оценки аффинной связи используется параметр степень различия двух хромосом H_i и H_j . Максимизация аффинности двух хромосом H_i и H_j соответствует минимизации степени различия.

В качестве средства изменения решения (трансформации) служит оператор трансформации (направленной мутации), суть которого заключается в изменении генотипа путем изменения целочисленных значений генов h_{il} в хромосоме $H_i(t) = \{h_{il}(t) | l=1, 2, \dots, n\}$. Целью перехода хромосомы в новое состояние является увеличению веса аффинной связи между хромосомами, что равносильно сокращению степени различия [5].

Введём на множестве трансформирующихся хромосом $H_i(t)$ отношение предпочтения \succ . Пусть имеются две хромосомы H_i и H_j . Хромосома $H_i(t)$ считается предпочтительней (лучшей), т.е. $H_i(t) \succ H_j(t)$, если выполняется приведённое ниже выражение: $f_i(t) > f_j(t)$ – где $f_i(t)$ и $f_j(t)$ – значение целевой функции H_i и H_j , $H(t) = \{H_i(t) | i=1, 2, \dots, n\}$ – рой хромосом, $H_i \in H$, $H_i = \{h_{il} | l=1, 2, \dots, n_i\}$.

$f_i(t)$ – значение целевой функции хромосомы $H_i(t)$;

T – число выполненных итераций;

$H^*_i(t)$ – хромосома H_i в лучшем состоянии, среди состояний, в которых она находилась с начала первой итерации до t -й итерации; $H^*_i(t) = \{h^*_{il}(t) | l=1, 2, \dots, n_i\}$;

$$(H^*_i \succ H_i(t)) \rightarrow f^*_i(t) > f_i(t);$$

$f^*_i(t)$ – значение целевой функции хромосомы в позиции $H^*_i(t)$;

$$f^*_i(t) = \min_{t \in [0; T]} f_i(t). \quad (1)$$

$H^*(t)$ – хромосома роя в лучшем состоянии среди всех состояний всех хромосом роя, в которых они находилась с начала первой итерации до t -й итерации; $H^*(t) = \{h^*_i(t) | i=1, 2, \dots, n\}$; $f^*(t)$ – значение целевой функции хромосомы $H^*(t)$.

$$f^*(t) = \min_{i \in [1:N]} f_i^*(t). \quad (2)$$

Алгоритм оптимизации роём трансформирующихся хромосом (РТХО) имеет вид:

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

t – номер итерации, T – число итераций.

В процессе поисковой процедуры происходит релаксационное сжатие аффинного графа.

Процесс поиска решений итерационный. В процессе поиска осуществляется перманентные трансформации (переход) хромосом в состояния с лучшим значением целевой функции решения (градиентная стратегия).

На каждой итерации t «лучшие», с точки зрения целевой функции хромосомы, являются аттракторами – «центром притяжения». Векторы трансформации всех хромосом в аффинном пространстве устремляются к этим аттракторам (центрам). Каждая хромосома $H_i(t)$ на шаге t переходит в аффинном пространстве с помощью релаксационной процедуры трансформации в новое состояние $H_i(t+1)$, при котором степень различия между хромосомой $H_i(t)$ и «центром притяжения» уменьшается.

Переход в новые состояния выполняется путем изменения значений генов и структуры хромосом. Целью трансформации хромосомы $H_i(t)$, тяготеющей к лучшей хромосоме $H^*(t)$, в новое состояние $H_i(t+1)$ является минимизация степени различия R_i между $H_i(t)$ и $H^*(t)$, что соответствует увеличению веса аффинной связи.

Трансформация, т.е. переход хромосомы $H_i(t)$ в новое состояние $H_i(t+1)$ осуществляется с учетом «степени различия» с лучшей хромосомой $H^*(t)$, либо с интегральной оценкой «степени различия» от группы соседних хромосом и с учетом вероятности перехода в новое состояние. На каждой итерации t осуществляется трансформация (переход) хромосом в состояния с лучшими значениями целевой функции решения (градиентная стратегия).

Если вес аффинной связи между хромосомами рассматривать как расстояние, то в процессе поисковой процедуры происходит релаксационное сжатие аффинного графа.

В качестве средства трансформации – перехода хромосомы $H_i(t)$ в хромосому $H_i(t+1)$ служит оператор направленной мутации (ОНМ), суть которого заключается в изменении целочисленных значений генов в хромосоме $H_i(t)$. ОНМ применяется к заданному числу генов с несовпадающими значениями.

Решением задачи является, построенная в процессе трансформации роя хромосом в аффинном пространстве хромосома, с лучшим значением целевой функции.

Для учета одновременного тяготения хромосомы $H_i(t)$ к хромосомам роя $H^*(t)$ и $H^*_i(t)$ на первом этапе формируется виртуальный центр притяжения роя $H^v(t+1)$. Формирование виртуального центра $H^v(t+1)$ осуществляется путем применения процедуры виртуальной трансформации к хромосоме $H_i(t)$, тяготеющей к хромосоме $H^*_i(t)$, в виртуальную хромосому $H^v(t+1)$.

После определения центра притяжения $H^v(t+1)$ хромосома $H_i(t)$ с помощью процедуры трансформации в направлении виртуальной позиции $H^v(t+1)$ трансформируется в хромосому $H_i(t+1)$.

После трансформации хромосомы $H_i(t)$ в новую $H_i(t+1)$ виртуальная хромосома $H^v(t+1)$ исключается.

Локальная цель трансформации хромосомы – достижение ею состояния с наилучшим значением целевой функции. Глобальная цель роя позиций – формирование оптимального решения задачи.

Трансформация $H_i(t)$ в $H_i(t+1)$ выполняется следующим образом. Последовательно просматриваются (начиная с $l=1$) локусы хромосом $H_i(t)$ и $H^*(t)$ и сравниваются соответствующие им гены, расположенные в одном локусе l .

Если на шаге l значения генов равны, т.е. $h_{il}(t) = h_{il}^*(t)$, то $h_{il}(t+1) = h_{il}^*(t)$.

Если же $h_{il}(t) \neq h_{il}^*(t)$, то в $H_i(t)$ отыскивается ген $h_{ik}(t)$, значение которого равно $h_{il}^*(t)$. Гены $h_{il}(t)$ и $h_{ik}(t)$ в хромосоме $H_i(t)$ меняются местами.

Рассмотрим процедуру *трансформации*, которая реализуется при помощи операции направленной мутации (ОНМ), разработанной авторами, и заключается в изменении взаимного расположения элементов в упорядоченном списке.

Пусть хромосомы $H_i(t) = \{h_{il} | l=1, 2, \dots, n_l\}$ и $H_z(t) = \{h_{zl} | l=1, 2, \dots, n_l\}$ имеют вид:

$$H_z(t) = \langle x_1, x_5, x_2, x_3, x_4, x_{10}, x_6, x_9, x_7, x_8 \rangle. \quad H_i(t) = \langle x_1, x_3, x_2, x_4, x_5, x_9, x_6, x_8, x_7, x_{10} \rangle.$$

Преобразуем вышеприведенные списки вершин в списки индексов вершин для удобства вычислений.

$$H_z^*(t) = \langle 15 \ 23 \ 410 \ 69 \ 78 \rangle. \quad H_i^*(t) = \langle 13 \ 24 \ 59 \ 68 \ 710 \rangle.$$

В качестве степени различия между двумя хромосомами $H_i(t)$ и $H^*(t)$ используется величина R_i^* – пропорциональная числу γ_i генов в трансформируемой хромосоме $H_i(t)$, значения которых не совпадают со значениями соответствующих генов в лучшей хромосоме $H^*(t)$.

$$R_i^* = \varepsilon \cdot \gamma_i. \quad \gamma_i = \sum_l \delta_{il}$$

где $\delta_{il}=0$, если в l -ом локусе хромосом $H_i(t)$ и $H^*(t)$ значения генов $h_{il} = h_{il}^*(t)$; $\delta_{il}=1$, если $h_{il} \neq h_{il}^*(t)$.

Целью трансформации хромосомы $H_i(t)$ является уменьшение степени различия R_i^* между хромосомами $H_i(t)$ и $H^*(t)$. Уменьшение степени различия R_i^* реализуется посредством парных перестановок генов в хромосоме $H_i(t)$.

Пусть в локусе l значения генов различны, $h_{il} \neq h_{il}^*$, $h_{il} \in H_i$, $h_{zl} \in H^*$. В хромосоме H_i отыскивается k -ый локус, в котором $h_{ik} = h_{il}^*$. Далее выполняется парная перестановка генов h_{il} и h_{ik} , размещенных в l -ом и k -ом локусах хромосомы H_i .

Трансформация хромосомы H_i выполняется следующим образом.

Последовательно просматриваются (начиная с первого) локусы хромосом H_i и H^* . Если $h_{il} \neq h_{zl}$, то в хромосоме H_i отыскивается локус k , в котором $h_{ik} = h_{il}$, выполняется парная перестановка генов, размещенных в l -ом и k -ом локусах хромосомы $H_i(t)$, т.е. $h_{il} = h_{ik}$, а $h_{ik} = h_{il}$.

Пример последовательной трансформации роя хромосом. Результат представлен в табл. 1.

Дано: H_i, H^* .

$l=1, h_{11} = h_{11}^*$.

$l=2, h_{12} \neq h_{12}^*, h_{12}=3, h_{12}^*=5$. Находим в $H_i(2)$ ген $h_{15}=5$. Обмен в $H_i(2)$ генами (h_{12} и h_{15}).

$l=3, h_{13} = h_{13}^*$.

$l=4, h_{14} \neq h_{14}^*, h_{14}=4, h_{14}^*=3$. Находим в $H_i(4)$ ген $h_{15}=3$. Обмен в $H_i(4)$ генами (h_{14} и h_{15}).

$l=5, h_{15} = h_{15}^*$.

$l=6, h_{16} \neq h_{16}^*, h_{16}=9, h_{16}^*=10$. Находим в $H_i(6)$ ген $h_{110}=9$. Обмен в $H_i(6)$ генами (h_{16} и h_{110}).

$l=7, h_{17} = h_{17}^*$.

$l=8, h_{18} \neq h_{18}^*, h_{18}=8, h_{18}^*=9$. Находим в $H_i(8)$ ген $h_{110}=9$. Обмен в $H_i(8)$ генами (h_{18} и h_{110}).

$l=9, h_{19} = h_{19}^*$.

$l=10, h_{110} = h_{110}^*$.

В приведенном примере выполнена полная трансформация хромосомы H_i в результате которой H_i стала эквивалентна H^* .

В работе для уменьшения степени различия R_i^* между хромосомами H_i и H^* задается число парных перестановок по формуле: $\lambda = k\gamma_i$. При этом λ позиций, подвергающихся трансформациям, выбираются случайным образом.

Таблица 1

Результат последовательной трансформации роя хромосом

| 1 | H_i | H^* | h_{il} | h^*_l | δ_{il} | H^*_i |
|----|------------------------------|-----------------------|----------|---------|---------------|------------------------------|
| 1 | 1324 59 68 710 | 1523 410 69 78. | 1 | 1 | 0 | 1324 59 68 710 |
| 2 | 1324 59 68 710 | 1523 410 69 78 | 3 | 5 | 1 | 1524 39 68 710 |
| 3 | 1524 39 68 710 | 1523 410 69 78 | 2 | 2 | 0 | 1524 39 68 710 |
| 4 | 1524 39 68 710 | 1523 410 69 78 | 4 | 3 | 1 | 1523 49 68 710 |
| 5 | 1523 49 68 710 | 1523 410 69 78 | 4 | 4 | 0 | 1523 410 69 78 |
| 6 | 1523 49 68 710 | 1523 410 69 78 | 9 | 10 | 1 | 1523 410 68 79 |
| 7 | 1523 410 68 79 | 1523 410 69 78 | 6 | 6 | 0 | 1523 410 68 79 |
| 8 | 1523 410 68 79 | 1523 410 69 78 | 8 | 9 | 1 | 1523 410 69 78 |
| 9 | 1523 410 69 78 | 1523 410 69 78 | 7 | 7 | 0 | 1523 410 69 78 |
| 10 | 15 23 410 69 78 | 15 23 410 69 78 | 8 | 8 | 0 | 15 23 410 69 78 |

Построенный на последней итерации рой хромосом используется муравьиным алгоритмом в качестве исходной информации.

Отметим, что для представления решений в алгоритмах используется единая структура данных в виде упорядоченного списка R . В вариантах гибридизации с использованием муравьиного алгоритма (МА) формируется начальная база данных БДМА. В соответствии с методологией МА формируется граф поиска решений G . Каждое решение $q_i \in Q^0$ начальной популяции Q^0 трансформируется в маршрут $m_i(1)$ на графе G . На первой итерации МА на каждом маршруте $m_i(1)$ графе G откладывается феромон в количестве пропорциональном оценке решения $q_i \in Q^0$. На последующих итерациях маршруты $m_i(t)$ формируются в соответствии с используемыми метаэвристиками МА. Ядром канонической структуры МА является конструктивный алгоритм построения маршрута на графе [18]. Отличительной чертой модифицированного МА, является процедура построения упорядоченного маршрута, соответствующего коду решения (списку R), базирующаяся на рассмотренных выше правилах.

Результаты эксперимента. Гибридная многоагентная система построения паросочетания реализована в виде комплекса программ, базирующихся на парадигмах роевых алгоритмов. Реконфигурируемая архитектура путем настройки позволяет реализовать следующие схемы гибридизации:

1. (ПА+РХ) – пчелиный алгоритм, алгоритм роя хромосом.
2. (ПА+МА) – пчелиный алгоритм, муравьиный алгоритм.
3. ((ПА+РХ)+МА) – пчелиный алгоритм, алгоритм роя хромосом, муравьиный алгоритм.
4. ПА+РХ+МА – пчелиный алгоритм, алгоритм роя хромосом, муравьиный алгоритм.

Тестирования вышеуказанных алгоритмов проводились на тестовых задачах.

Для оценки эффективности разработанных методик и программного обеспечения авторами разработаны контрольные примеры построения паросочетаний, для которых заранее известно оптимальное значение F суммарной стоимости ребер, входящих в совершенное паросочетание Q полного графа G .

Обозначим как:

nG – число вершин полного графа G .

nU – число ребер полного графа G .

nQ – число ребер, входящих в состав совершенного паросочетания полного графа G .

Множество U ребер полного графа G разбиваем на два подмножества $U=U_1 \cup U_2$. В состав U_1 входит nQ ребер полного графа G , а в состав U_2 входит остаток $(nU-nQ)$ ребер полного графа G . Ребрам множества U_1 , случайным образом, присваиваются значения диапозоне от $\partial 1$ до $\partial 2$, а ребрам множества U_2 , случайным образом, присваиваются значения, лежащие в диапозоне от $\partial 3$ до $\partial 4$, причем $\partial 3 > \partial 2$.

Поскольку вес любого ребра из U_1 меньше веса любого ребра из U_2 , то суммарный вес ω_1 ребер множества U_1 будет минимальным: $\omega_1 = \sum_i \varphi_i$, где φ_i вес ребра, входящего в U_1 .

С помощью разработанного программного средства было сгенерировано 5 примеров C1-C5.

C1 – ($nG=10, nU=45, nQ=5$); C2 – ($nG=20, nU=190, nQ=10$); C3 – ($nG=30, nU=435, nQ=15$); C4 – ($nG=40, nU=780, nQ=20$); C5 – ($nG=50, nU=1225, nQ=25$).

Каждый пример запускался 10 раз. В табл. 2 для каждого примера приводится среднее значение отклонения оценки решения от оптимального в %, а также среднее значение отклонения оценки решения для каждой схемы гибридизации.

Таблица 2

Результаты проведенных экспериментов

| Класс задач | C1 | C2 | C3 | C4 | C5 | Среднее значение | Количество лучших решений | |
|-------------|------|-----|-----|-----|-----|------------------|---------------------------|------|
| | nG | 10 | 20 | 30 | 40 | | | 50 |
| | nU | 45 | 190 | 435 | 780 | | | 1225 |
| nQ | 5 | 10 | 15 | 20 | 25 | | | |
| (ПА+PX) | 1,5 | 1,9 | 2 | 2,1 | 2,2 | 1,94 | 1 | |
| (ПА+МА) | 1,4 | 1,8 | 2 | 1,9 | 1,8 | 1,78 | 2 | |
| (ПА+PX)+МА) | 1,2 | 1,2 | 1,3 | 1,7 | 1,6 | 1,14 | 4 | |
| (ПА+PX+МА) | 1,3 | 1,4 | 1,4 | 1,8 | 1,7 | 1,52 | 2 | |

Диапазон веса ребер: $\partial Q=<2-4>$; $\partial U=<6-8>$

Процедура в роевом алгоритме в составе гибрида (ПА+PX) сходится быстрее чем процедура МА в составе гибрида (ПА+МА). Процедура в МА в составе гибрида (ПА+PX)+МА) сходится быстрее чем процедура МА в составе гибрида (ПА+МА).

Лучшие результаты показала программа (ПА+PX)+МА). Первая часть является гибридизацией алгоритма пчелиной колонии и роевого алгоритма. Вторая часть является гибридизацией первой части и алгоритма муравьиной колонии, рис. 3.

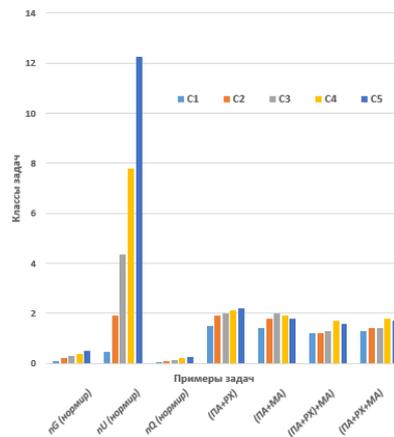


Рис. 3. Графическая иллюстрация результатов работы

В результаты экспериментов было установлено, что показатели качества разработанных алгоритмов имеют более высокие значения чем в работах [6–10]. По сравнению с существующими алгоритмами достигнуто улучшение результатов на 5-8%.

Временная сложность алгоритма (BCA), полученная экспериментальным путем, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет ($BCA \approx O(n^2)$).

Заключение. В статье представлена разработанная реконфигурируемая архитектура гибридной многоагентной системы поиска решений, базирующаяся на парадигмах роевых алгоритмов. Реконфигурируемая архитектура путем настройки позволяет реализовать следующие методы гибридизации: высокоуровневую и низкоуровневую гибридизацию.

цию вложением, типа препроцессор/постпроцессор, ко-алгоритмическую на базе одного или нескольких типов алгоритмов. Предложена методология синтеза совершенного паросочетания минимального веса в полном графе, основанная на базовых принципах гибридизации поисковых эволюционных процедур.

В работе представлен разработанный новый подход для решения оптимизационных задач на графах построенный на основе интеграции, комбинирования и ко-эволюции роевых методов, отличающийся тем, что заданное число роевых субалгоритмов, используют различные стратегии поиска, функционируют параллельно-последовательно, реализуют более широкий обзор пространства решений, что позволяет обеспечить более высокую вероятность локализации глобального экстремума задачи.

Эксперименты предложенных в работе алгоритмов проводились на тестовых задачах.

Для оценки эффективности разработанных методик и программного обеспечения авторами разработаны контрольные примеры построения паросочетаний, для которых заранее известно оптимальное значение суммарной стоимости ребер, входящих в совершенное паросочетание полного графа.

Первая часть разработанной программы является гибридизацией алгоритма пчелиной колонии и роевого алгоритма. Вторая часть является гибридизацией первой части и алгоритма муравьиной колонии. В результате экспериментов было установлено, что показатели качества разработанных алгоритмов имеют более высокие значения чем в существующих алгоритмах, при этом достигнуто улучшение результатов на 5-8%.

Предложенный новый подход к использованию роевых алгоритмов повысит эффективность, расширит сферы их применения, снизит комбинаторную сложность задачи, позволит организовывать эффективные структуры поиска, формировать компактные интерпретации решений, отражающие коллективную эволюционную память.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Андерсон Д.* Дискретная математика и комбинаторика. – М.: Вильямс, 2003.
2. *Кормен К., Лейзерсон Ч., Ривест Р.* Алгоритмы построение и анализ. – М.: МЦМНО, 2000.
3. *Асанов М., Баранский В., Расин В.* Дискретная математика: Графы, матроиды, алгоритмы. – СПб.: Изд-во «Лань», 2010.
4. *Blum C., Roli A.* Metaheuristics in combinatorial optimization: overview and conceptual comparison // ACM computing surveys. – 2003. – No. 35. – P. 268-308.
5. MAXimal. Алгоритм Куна нахождения наибольшего паросочетания.
6. *Bast H., Mehlhorn K., Schafer G. et al.* Matching Algorithms Are Fast in Sparse Random Graphs // Theory Comput Syst. – 2006. – P. 3-14.
7. *Ding J., Ma Z., Wu Y. et al.* Efficient random graph matching via degree profiles // Probab. Theory Relat. Fields. – 2021. – P. 29-115.
8. *Blum N.* A new approach to maximal matchings in general graphs // International Colloquium on Automata, Languages, and Programming. – 2017. – P. 586-597.
9. *Лебедев Б.К., Лебедев О.Б.* Эволюционный алгоритм нахождения максимального паросочетания // 3-й Международный НТС «Интегрированные модели и мягкие вычисления в искусственном интеллекте». – М.: Изд-во Физматлит, 2005. – С. 274-280.
10. *Курейчик В.В., Курейчик В.М.* Генетический алгоритм определения паросочетаний графа // Тр. 10 международной конференции «Knowledge-dialogue-solution». – 2003. – С. 246-251.
11. *Карпенко А.П., Кузьмина И.А.* Исследование эффективности структурно-параметрического синтеза одного класса популяционных алгоритмов глобальной оптимизации // Математические методы в технологиях и технике. – 2024. – № 4. – С. 82-89.
12. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2021. – 448 с.
13. *Пантелеев А.В., Метлицкая Д.В., Алешина Е.А.* Методы глобальной оптимизации: Метаэвристические стратегии и алгоритмы. – М.: Вузовская книга, 2013. – 44 с.
14. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Решение задачи покрытия на основе эволюционного моделирования // Известия РАН. Теория и системы управления. – 2009. – С. 101-117.
15. *Лебедев Б.К., Лебедев В.Б., Лебедев О.Б.* Гибридизация роевого интеллекта и генетической эволюции на примере размещения // Программные продукты, системы и алгоритмы. – 2017. – № 4.
16. *Лебедев Б.К., Лебедев О.Б.* Распределение ресурсов на основе гибридных моделей роевого интеллекта // Научно-технический вестник информационных технологий, механики и оптики. – 2017. – № 6. – С. 1063-1073.

17. *Карпенко А.П.* Типовые структуры популяционных алгоритмов глобальной оптимизации // Информационные и математические технологии в науке и управлении. – 2022. – № 1 (25). – С. 48-57.
18. *Лебедев О.Б.* Модели адаптивного поведения муравьиной колонии в задачах проектирования. – Таганрог: Изд-во ЮФУ, 2013. – 199 с.
19. *Raidl G.R.* A Unified view on hybrid Metaheuristics // Lecture Notes In Computer Science. – Springer, 2006. – P. 1-12.
20. *Cong J., Romesis M., Xie M.* Optimality, scalability and stability study of partitioning and placement algorithms // Proc. of the International Symposium on Physical Design. – 2003. – P. 88-94.

REFERENCES

1. *Anderson D.* Diskretnaya matematika i kombinatorika [Discrete Mathematics and combinatorics]. Moscow: Vil'yams, 2003.
2. *Kormen K., Leyzerson Ch., Rivest R.* Algoritmy postroenie i analiz [Algorithms for construction and analysis]. Moscow: MTSMNO, 2000.
3. *Asanov M., Baranskiy V., Rasin V.* Diskretnaya matematika: Grafy, matroidy, algoritmy [Discrete mathematics: Graphs, matroids, algorithms]. Saint Petersburg: Izd-vo «Lan'», 2010.
4. *Blum C., Roli A.* Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM computing surveys*, 2003, No. 35, pp. 268-308.
5. MAXimal. Algoritm Kuna nakhozheniya naibol'shego parosochetaniya [MAXimal. An algorithmic Kuna designed for the largest number of users.
6. *Bast H., Mehlhorn K., Schafer G. et al.* Matching Algorithms Are Fast in Sparse Random Graphs, *Theory Comput Syst.*, 2006, pp. 3-14.
7. *Ding J., Ma Z., Wu Y. et al.* Efficient random graph matching via degree profiles, *Probab. Theory Relat. Fields.*, 2021, pp. 29-115.
8. *Blum N.* A new approach to maximal matchings in general graphs, *International Colloquium on Automata, Languages, and Programming*, 2017, pp. 586-597.
9. *Lebedev B.K., Lebedev O.B.* Evolyutsionnyy algoritm nakhozheniya maksimal'nogo parosochetaniya [An evolutionary algorithm for finding the maximum match], 3-y *Mezhdunarodnyy NTS «Integrirrovannye modeli i myagkie vychisleniya v iskusstvennom intellekte»* [3rd International Scientific and Technical Conference «Integrated Models and Soft Computing in Artificial Intelligence». Moscow: Izd-vo Fizmatlit, 2005, pp. 274-280.
10. *Kureychik V.V., Kureychik V.M.* Geneticheskyy algoritm opredeleniya parosochetaniy grafa [A genetic algorithm for determining graph combinations], *Tr. 10 mezhdunarodnoy konferentsii «Knowledge-dialogue-solution»* [Proceedings of the 10th International Conference «Knowledge-dialogue-solution»], 2003, pp. 246-251.
11. *Karpenko A.P., Kuz'mina I.A.* Issledovanie effektivnosti strukturno-parametricheskogo sinteza odnogo klassa populyatsionnykh algoritmov global'noy optimizatsii [Study of the efficiency of structural-parametric synthesis of one class of population algorithms for global optimization], *Matematicheskie metody v tekhnologiyakh i tekhnike* [Mathematical Methods in Technology and Engineering], 2024, No. 4, pp. 82-89.
12. *Karpenko A.P.* Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature: a textbook]. Moscow: Izd-vo MGTU im. N.E. Bauman, 2021, 448 p.
13. *Panteleev A.V., Metlitskaya D.V., Aleshina E.A.* Metody global'noy optimizatsii: Metaevristicheskie strategii i algoritmy [Methods of global optimization: Metaheuristic strategies and algorithms]. Moscow: Vuzovskaya kniga, 2013, 44 p.
14. *Kureychik V.M., Lebedev B.K., Lebedev O.B.* Reshenie zadachi pokrytiya na osnove evolyutsionnogo modelirovaniya [Solving the problem of coverage based on evolutionary modeling], *Izvestiya RAN. Teoriya i sistemy upravleniya* [News of the Russian Academy of Sciences. Theory and control systems], 2009, pp. 101-117.
15. *Lebedev B.K., Lebedev V.B., Lebedev O.B.* Gibridizatsiya roevogo intellekta i geneticheskoy evolyutsii na primere razmeshcheniya [Hybridization of swarm intelligence and genetic evolution by the example of placement], *Programmnyye produkty, sistemy i algoritmy* [Software products, systems and Algorithms], 2017, No. 4.
16. *Lebedev B.K., Lebedev O.B.* Raspredelenie resursov na osnove gibridnykh modeley roevogo intellekta [Resource allocation based on hybrid models of swarm intelligence], *Nauchno-tekhnicheskyy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki* [Scientific and Technical Bulletin of Information Technologies, Mechanics and Optics], 2017, No. 6, pp. 1063-1073.
17. *Karpenko A.P.* Tipovyye struktury populyatsionnykh algoritmov global'noy optimizatsii [Typical structures of population algorithms for global optimization], *Informatsionnyye i matematicheskie tekhnologii v nauke i upravlenii* [Information and Mathematical Technologies in Science and Management], 2022, No. 1 (25), pp. 48-57.

18. *Lebedev O.B.* Modeli adaptivnogo povedeniya murav'inoi kolonii v zadachakh proektirovaniya [Models of adaptive behavior of an ant colony in design tasks]. Taganrog: Izd-vo YuFU, 2013, 199 p.
19. *Raidl G.R.* A Unified view on hybrid Metaheuristics, *Lecture Notes In Computer Science*. Springer, 2006, pp. 1-12.
20. *Cong J., Romesis M., Xie M.* Optimality, scalability and stability study of partitioning and placement algorithms, *Proc. of the International Symposium on Physical Design*, 2003, pp. 88-94.

Лебедев Борис Константинович – Южный федеральный университет; e-mail: lebedev.b.k@gmail.com; г. Таганрог, Россия; тел.: 89282897933; кафедра систем автоматизированного проектирования им. В.М. Курейчика; д.т.н.; профессор.

Лебедев Олег Борисович – МИРЭА – Российский технологический университет; e-mail: lebedev.ob@mail.ru; г. Москва, Россия; тел.: 89085135512; кафедра информатики; д.т.н.; профессор.

Ганзжур Марина Александровна – Донской государственный технический университет; e-mail: mganzhur@yandex.ru; г. Ростов-на-Дону, Россия; тел.: 89081819111; кафедра вычислительных систем и информационной безопасности; старший преподаватель.

Бесхмельнов Максим Игоревич – МИРЭА – Российский технологический университет; e-mail: m_beskhmelnov@mail.ru; г. Москва, Россия; тел.: 89252622177; кафедра информатики; преподаватель.

Lebedev Boris Konstantinovich – Southern Federal University; e-mail: lebedev.b.k@gmail.com; Taganrog, Russia; phone: +79282897933; the Department of Computer Aided Design named after V.M. Kureichik; dr. of eng. sc.; professor.

Lebedev Oleg Borisovich – MIREA – Russian University of Technology; e-mail: lebedev.ob@mail.ru; Moscow, Russia; phone: +79085135512; the Department of Computer Science; dr. of eng. sc.; professor.

Ganzhur Marina Aleksandrovna – Don State Technical University; e-mail: mganzhur@yandex.ru; Rostov-on-Don, Russia; phone: +79081819111; the Department of Computing Systems and Information; security senior lecturer.

Beskhmelnov Maxim Igorevich – MIREA – Russian University of Technology; e-mail: m_beskhmelnov@mail.ru; Moscow, Russia; phone: +79252622177; the Department of Computer Science; teacher.

УДК 519.612

DOI 10.18522/2311-3103-2025-1-22-37

Д.А. Береза, Л.А. Гладков, Н.В. Гладкова

РАЗРАБОТКА АГЕНТНО-ОРИЕНТИРОВАННОГО АЛГОРИТМА РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ БОЛЬШОЙ РАЗМЕРНОСТИ

Решение систем линейных алгебраических уравнений (СЛАУ) является одной из важнейших фундаментальных задач при разработке нового поколения систем проектирования в различных областях науки и техники. Актуальность данного исследования обусловлена ростом объема данных и усложнением задач. Традиционные методы решения СЛАУ, такие как метод Гаусса, метод прогонки, итерационные методы (метод Якоби, метод Зейделя и т.д.), хорошо зарекомендовали себя при работе с относительно небольшими системами. Однако при решении СЛАУ большой размерности данные методы недостаточно эффективны из-за высоких вычислительных затрат и требований к памяти. Одним из перспективных подходов к решению задач высокой сложности является использование агентно-ориентированных систем. Агентно-ориентированные системы предлагают новый способ организации вычислительных процессов, основанный на взаимодействии независимых агентов, каждый из которых выполняет определенную часть задачи. Такой подход позволяет более гибко распределять вычислительные ресурсы и эффективно решать сложные задачи в условиях больших данных. Представлена методика решения уравнений, описывающих математическую модель схемы, с учетом оптимизации соотношения между точностью расчетов и временем их выполнения. В данной работе предлагается агентно-ориентированный алгоритм для решения систем линейных алгебраических уравнений большой размерности. В ходе разработки данного алгоритма был проведен анализ существующих методов и алгоритмов решения СЛАУ, выявлены их преимущества и не-