

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

МУЛЬТИ-МЕТЕТИЧЕСКИЙ МУРАВЬИНЫЙ АЛГОРИТМ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ*

Рассматривается задача конструирования мульти-меметического муравьиного алгоритма (МА), включающая разработку 5 мемов и выбор целесообразной стратегии использования того или иного мема из роя доступных мемов. Работа МА дискретной оптимизации иллюстрируется на примере задачи разбиения графа на подграфы, широко применяющейся при решении таких задач, как кластеризация, раскраска, выделения клик, паросочетания, разбиения СБИС и т.д. Задан граф $G(X, U)$, где X – множество вершин, $|X| = n$, U – множество ребер. Необходимо разбить множество X на подмножества X_1 и X_2 , $X_1 \cup X_2 = X$, $X_1 \cap X_2 = \emptyset$, $X_i \neq \emptyset$. Для поиска решения задачи используется полный граф поиска решений (ГПР) $R(X, E)$. В отличие от канонической парадигмы МА агентом на ГПР $R(X, E)$ строится не маршрут, а формируется подграф. На первом этапе каждой итерации МА конструктивным алгоритмом (мемом) каждый из агентов формирует множество X_{1k} . Во всех мемах формирование множества X_{1k} осуществляется последовательно (пошагово). На каждом шаге t агент применяет вероятностное правило выбора следующей вершины для включения ее в формируемое множество $X_{1k}(t)$. В первом меме поиск решений осуществляется на ГПР $R(X, E)$, а феромон откладывается на множестве ребер E . Основное отличие второго мема от первого заключается в том, что из процесса поиска агентом решения исключается полный граф поиска $R(X, E)$. Феромон откладывается только на вершинах графа $G(X, U)$. Это приводит к резкому сокращению числа феромонных точек и, следовательно, памяти, так как число ребер графа $R(X, E)$ значительно больше числа вершин. Особенность третьего мема M_3 заключается в том, что на вершине x_i графа $G(X, U)$ формируются два показателя ϕ_{1k} и ϕ_{2k} : ϕ_{1k} – суммарный уровень феромона, отложенного на x_i только в тех случаях, когда x_i была в составе $X_{1k}(t)$; ϕ_{2k} – суммарный уровень феромона, отложенного на x_i только в тех случаях, когда x_i не входила в состав $X_{1k}(t)$. Показатель ϕ_{1k} характеризует для вершины x_i предпочтительность узла $X_{1k}(t)$. Четвертый мем отличается от третьего тем, что агентом формируются параллельно два множества $X_{1k}(t)$ и $X_{2k}(t)$. В пятом меме для уменьшения общей трудоемкости поисковой процедуры используется подход, основанный на декомпозиции структуры данных в процессах формирования множества вершин $X_{1k} \subset X$ и отложения феромона. В модифицированном подходе отложение феромона осуществляется на графе $R(X, E)$, а формирование подмножества вершин $X_{1k} \subset X$ осуществляется на ориентированном графе $V(X, V)$. Предложены два подхода к построению адаптивного мульти-меметического алгоритма, использующего рой мемов. Мульти-меметическая гибридизация приводит к алгоритму, обладающему свойствами самоадаптации, поскольку в процессе поиска мемы конкурируют между собой, и победивший на данном этапе поиска мем используется на последующих итерациях. За счет повышения эффективности поиска, гибридный алгоритм дает возможность уменьшить число агентов в системе и повысить тем самым его быстродействие.

Муравьиный алгоритм; мем; меметический алгоритм; мульти-меметическая гибридизация; адаптация; рой частиц; оптимизация.

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

MULTI-MEMETIC ANT DISCRETE OPTIMIZATION ALGORITHM

The work deals with the task of constructing a multi-meme ant algorithm (AA), which includes the development of 5 memes and the choice of an expedient strategy for using one meme or another from a swarm of available memes. The work of discrete optimization AA is illustrated by the example of the problem of splitting a graph into subgraphs, which is widely used in solving problems such as

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

clustering, coloring, selection of clicks, matching, partitioning of VLSI, etc. Given a graph $G(X, U)$, where X is the set of vertices, $|X|=n$, U is the set of edges. It is necessary to divide the set X into subsets X_1 and X_2 , $X_1 \cup X_2 = X$, $X_1 \cap X_2 = \emptyset$, $X_i \neq \emptyset$. To search for a solution to the problem, a complete solution search graph (SSG) $R(X, E)$ is used. In contrast to the canonical paradigm of the AA, the agent for the SSG $R(X, E)$ does not construct a route, but forms a subgraph. At the first stage of each iteration of the MA, the constructive algorithm (meme) of each agent forms the set X_{1k} . In all memes, the formation of the set X_{1k} is carried out sequentially (step by step). At each step t , the agent applies a probabilistic selection rule for the next vertex to include it in the generated set $X_{1k}(t)$. In the first meme, the search for solutions is carried out on the SSG $R(X, E)$, and the pheromone is deposited on the set of edges E . The main difference between the second meme and the first one is that the full search graph $R(X, E)$ is excluded from the search process by the decision agent. Pheromone is deposited only on the vertices of the graph $G(X, U)$. This leads to a sharp reduction in the number of pheromone points and, consequently, in memory, since the number of edges of the graph $R(X, E)$ is significantly greater than the number of vertices. The peculiarity of the third meme M_3 is that two indicators φ_{1k} and φ_{2k} are formed at the vertex x_i of the graph $G(X, U)$: φ_{1k} is the total level of the pheromone delayed by x_i , only when x_i was in $X_{1k}(t)$; φ_{2k} is the total level of pheromone deposited on x_i , only in cases when x_i was not included in $X_{1k}(t)$. The indicator φ_{1k} characterizes for the vertex x_i the preference of the node $X_{1k}(t)$. The fourth meme differs from the third one in that the agent generates two parallel sets $X_{1k}(t)$ and $X_{2k}(t)$ in parallel. In the fifth meme, an approach based on the decomposition of the data structure in the processes of forming the $X_{1k} \subset X$ vertex set and the pheromone deposit is used to reduce the overall laboriousness of the search procedure. In the modified approach, the pheromone is deposited on the graph $R(X, E)$, and the formation of a subset of the $X_{1k} \subset X$ vertices is carried out on the oriented graph $B(X, V)$. Two approaches to constructing an adaptive multimeme algorithm using a swarm of memes are proposed. Multimemeic hybridization leads to an algorithm with self-adaptive properties, since in the search process memes compete with each other, and the winner at this stage of the search is used in subsequent iterations. By increasing the search efficiency, the hybrid algorithm makes it possible to reduce the number of agents in the system and thereby increase its speed.

Ant algorithm; meme; memetic algorithm; multimeme hybridization; adaptation; particle swarm; optimization.

Введение. Одним из перспективных современных направлений в области эволюционных вычислений являются меметические алгоритмы (memetic algorithms) (МА) [1]. Они представляют собой популяционные метаэвристические методы поисковой оптимизации, основанные на нео-дарвиновском принципе эволюции и концепции мема (meme), предложенной Р. Докинзом в 1976 году [2].

Мем – единица передачи культурной информации, распространяемая от одного человека к другому, посредством имитации, научения и др. Термин меметических алгоритмов был впервые предложен Москато (Р. Moscato) в 1989 году [3] для обозначения класса стохастических методов глобального поиска. МА объединяют в себе сильные стороны методов локального поиска, ориентированных на конкретные практические задачи, и популяционных методов глобального поиска. На сегодняшний день методы данного класса успешно используются для решения задач комбинаторной, непрерывной, динамической и многокритериальной оптимизации [1–5].

В контексте МА, мем является реализацией какого-либо метода локальной оптимизации, уточняющий решение в процессе поиска. В ряде работ МА рассматривается как сочетание популяционного поиска глобального оптимума и процедур локального уточнения решений, которое дает синергетический эффект.

Меметические алгоритмы оказались успешными для решения задач оптимизации в различных сферах. Как показали исследования [6–9], выбор мемов оказывает очень большое влияние на производительность МА. Концепция меметических алгоритмов предоставляет широкие возможности для разработки различных модификаций этих алгоритмов, которые могут отличаться частотой выполнения локального поиска, условиями его окончания и так далее [5]. Практически значимые модификации меметических алгоритмов предполагают одновременное использование различных мемов. Такие алгоритмы называют мульти-меметическими.

На сегодняшний день существует большое число публикаций, показывающих различные варианты гибридизации метаэвристических методов с различными методами локального поиска [10]. Эти алгоритмы используют сложные эвристические методы локального поиска, ориентированные на конкретные задачи. Несмотря на их высокую эффективность, они имеют ограниченное применение, так как исследователи не обладают априорной информацией о том, какой мем является наиболее оптимальным для данной задачи. Для преодоления указанного недостатка были предложены, так называемые, мульти-меметические (multi-memes) адаптивные алгоритмы [2, 6, 7].

Использование наиболее эффективных наборов мемов позволяет отыскать оптимальное решение за меньшее число итераций по сравнению с менее эффективными наборами. Известной является стратегия случайного спуска (random descent choice). Здесь на первом шаге мем выбирается из роя M случайным образом. Выбранный мем используется до тех пор, пока он не перестанет обеспечивать локальные уточнения решений. Затем случайным образом выбирается другой мем и т.д.

Методы жадной стратегии применяют мемы к каждому агенту популяции, а затем выбирают мем, показавший наилучший результат. Недостатком жадных методов являются высокие затраты вычислительных ресурсов.

Интересным также является направление изучения механизма выбора мемов по нескольким различным критериям.

В работе рассматривается задача конструирования мульти-меметического муравьиного алгоритма, включающая разработку 5 мемов и выбор целесообразной стратегии использования того или иного мема из роя доступных мемов M .

В качестве критериев оценки эффективности используется среднее значение целевой функции по итогам мультистарта, а также число итераций, которое при заданном ограничении на число итераций локального поиска однозначно определяет максимальное число испытаний.

1. Постановка задачи. Работа муравьиного алгоритма дискретной оптимизации иллюстрируется на примере задачи разбиения графа на подграфы, широко применяющейся при решении таких задач, как кластеризация, раскраска, выделения клик, паросочетания, разбиения СБИС и т.д.

Пусть дан граф $G(X, U)$, где X – множество вершин, $|X|=n$, U – множество ребер. Необходимо разбить множество X на два непустых и непересекающихся подмножества X_1 и X_2 , $X_1 \cup X_2 = X$, $X_1 \cap X_2 = \emptyset$, $X_i \neq \emptyset$. На формируемые узлы (блоки, компоненты) накладываются ограничения: $|X_1|=n_1$, $|X_2|=n_2$, $n_1+n_2=n$. Критерий оптимизации число (суммарная стоимость) связей – F между X_1 и X_2 . Цель оптимизации минимизация критерия F .

Метаэвристика муравьиного алгоритма основывается на комбинации двух техник: общая схема строится на базовом методе, в которую включается встроенная процедура. Базовый метод заключается в реализации итерационной процедуры поиска лучшего решения, на основе механизмов адаптивного поведения муравьиной колонии. Встроенная процедура – это конструктивный алгоритм построения муравьем некоторой конкретной интерпретации решения [11]. Фактически встроенная процедура является мемом, а муравьиный алгоритм можно рассматривать, как гибридный алгоритм на основе интеграции базовой итерационной структуры с мемом – встроенной конструктивной процедурой локального поиска.

Для поиска решения задачи используется полный граф решений $R(X, E)$, где E – множество всех ребер полного графа, связывающих множество вершин X . На множестве ребер E будем откладывать феромон. Процесс поиска решений муравьиным алгоритмом – итерационный. Каноническая парадигма муравьиного алгоритма для поиска решения задачи использует полный граф решений $R(X, E)$, где E – множество всех ребер полного графа, связывающих множество вершин X .

На множестве ребер E откладывается феромон. Агентом на графе поиска решений $R(X, E)$ строится маршрут, являющийся интерпретацией решения. В работе в отличие от канонической парадигмы агентом на графе $R(X, E)$, строится не маршрут, а формируется подграф $R_I(X_I, E_I)$, $R_I \subset R$.

На начальном этапе на всех вершинах графа G откладывается одинаковое (небольшое) количество феромона Q/m , где $m=|E|$.

Каждая итерация t включает три этапа. На первом этапе каждой итерации выполняется рассматриваемый в качестве мема конструктивный алгоритм. На втором и третьем этапах выполняется базовая итерационная процедура.

В мульти-меметических алгоритмах используется несколько мемов для поиска оптимума. Решение о выборе мема для конкретного индивида в популяции очень часто принимается динамически. При этом методы локального поиска, ориентированные на различные задачи, конкурируют между собой. В результате эффективность работы МА в целом остается высокой несмотря на то, что предопределенная информация о решаемой задаче отсутствует.

В работе используется пять методов локальной оптимизации, лежащих в основе мемов. Для выбора того или иного мема на каждой итерации работы алгоритма использовалась гиперэвристика, когда для каждого решения, формируемого, агентом выбирают мем случайным образом из роя мемов и вероятность выбора каждого из мемов не меняется в процессе итераций.

Во всех мемах формирование множества X_{I_k} осуществляется последовательно (пошагово). На каждом шаге агент применяет вероятностное правило выбора следующей вершины для включения ее формируемое множество $X_{I_k}(t)$.

Рассмотрим первый мем M_1 .

На первом этапе каждой итерации муравьиного алгоритма конструктивным алгоритмом (мемом) каждый из агентов $z_k \in Z = \{z_k | k=1, 2, \dots, n\}$ формирует множество X_{I_k} где k – номер агента. Коллективом агентов формируется множество интерпретаций $W = \{W_k | k=1, 2, \dots, n\}$. $W_k = \{X_{I_k}, X_{2k}\}$. Поиск решений осуществляется на ГРП $R(X, E)$.

На втором этапе агенты на ребрах графа R откладывают феромон, на третьем этапе осуществляется испарение феромона. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромоны откладываются агентами на ребрах после полного формирования решений.

Локальной целью агента z_k при поиске решения является формирование множества X_{I_k} такого, чтобы на вершинах узла X_{I_k} , было отложено максимальное количество феромона. Это соответствует максимизации числа внутренних связей в узле X_{I_k} и минимизации числа внешних связей узла X_{I_k} с остальными узлами.

Агенты имеют собственную «память». Каждый агент хранит в памяти список пройденных им вершин графа. На каждом шаге t у агента z_k есть список вершин, уже включенных в формируемое множество – $X_{I_k}(t)$ и список оставшихся вершин $X_{ck}(t)$, $X_{I_k}(t) \cup X_{ck}(t) = X$. $X_{I_k}(0) = \emptyset$, $X_{2k}(0) = X$. В памяти хранится распределение феромона по вершинам и ребрам графа поиска решений $R(X, E)$, выполняющего функции хранилища коллективной эволюционной памяти.

На первом шаге априори в каждое формируемое множество $X_{I_k}(t)$, где $t=1$, включается вершина графа G , причем вершины графа G распределяются по узлам равномерно, то есть в каждом узле своя вершина, ($\forall i, j$) $[X_{I_i}(1) \cap X_{I_j}(1) = \emptyset]$. Такое распределение необходимо, чтобы все вершины графа G имели одинаковые шансы быть отправной точкой при формировании узла X_I . В модификациях алгоритма использовались также ln муравьев, причем каждая группа из l муравьев используют в качестве начального одно и то же $X_{I_i}(1)$. На конечном шаге $t=n_l$ агентом будет сформирован узел $X_{I_k}(n_l) = X_{I_k}$. $|X_I(n_l)| = n_l$.

На каждом шаге t агент просматривает все свободные на данном шаге вершины $X_{ck}(t)$. Для каждой вершины $x_i \in X_{ck}(t)$ рассчитываются два показателя (рис. 1):

φ_{ik} – суммарный уровень феромона на ребрах графа R , связывающих x_i с вершинами узла $X_{1k}(t)$;

s_{ik} – число связей на графе G между x_i и $X_{1k}(t)$.

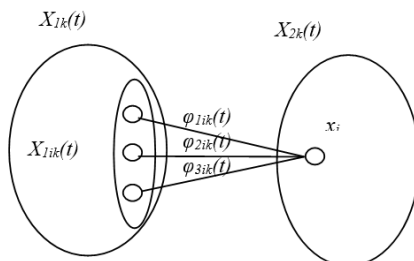


Рис. 1. Структура показателя φ_{ik}

Путем аддитивной свертки показателей φ_{ik} и s_{ik} определяется потенциальная стоимость Φ_{ik} связей $x_i \in X_{ck}(t)$ с $X_{1k}(t)$.

$$\Phi_{ik} = \alpha(\varphi_{ik}) + \beta(s_{ik}), \quad (1)$$

где α, β – управляющие параметры, которые подбираются экспериментально.

Показатель Φ_{ik} характеризует привлекательность (предпочтительность) включения x_i в $X_{1k}(t)$.

Вероятность P_{ik} включения вершины $x_i \in X_{ck}(t)$ в формируемый узел $X_{1k}(t)$ определяется следующим соотношением:

$$P_{ik} = \Phi_{ik} / \sum_i (\Phi_{ik}). \quad (2)$$

Агент z_k с вероятностью P_{ik} выбирает одну из вершин, которая включается в множество $X_{1k}(t)$ и исключается из множества $X_{ck}(t)$.

На втором этапе итерации, каждый агент z_k откладывает феромон на ребрах полного подграфа $R_{1k} \subset R$, построенного на вершинах множества X_{1k} в графе R . В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромон откладывается агентами на ребрах графе R после полного формирования решений.

Количество феромона $\tau_k(l)$, откладываемое агентом z_k на каждом ребре подграфа $R_{1k} \subset R$, построенного на l -ой итерации, определяется следующим образом:

$$\tau_k(l) = Q / D_k(l), \quad (3)$$

где l – номер итерации, Q – общее количество феромона, откладываемое k -ым муравьем на ребрах подграфа $R_{1k} \subset R$, $D_k(l)$ – число связей на графе G между множествами X_{1k} и X_{2k} , сформированными k -ым муравьем на l -ой итерации (другими словами, целевая функция для данного решения).

Основное отличие второго мема от первого заключается в том, что из процесса поиска агентом решения исключается полный граф поиска $R(X, E)$. Феромон, в соответствии с формулой (3), откладывается только на вершинах графа $G(X, U)$. Это приводит к резкому сокращению числа феромонных точек и, следовательно, памяти, так как число ребер графа $R(X, E)$ значительно больше числа вершин.

Потенциальная стоимость Φ_{ik} связей $x_i \in X_{ck}(t)$ с $X_{1k}(t)$ определяется как:

$$\Phi_{ik} = \alpha(\varphi_{ik}) + \beta(s_{ik}), \quad (4)$$

где φ_{ik} – суммарный уровень феромона на вершине x_i графа $G(X, U)$; s_{ik} – число связей на графе G между x_i и $X_{1k}(t)$, связывающих x_i с вершинами узла $X_{1k}(t)$.

Вероятность P_{ik} включения вершины $x_i \in X_{ck}(t)$ в формируемый узел $X_{Ik}(t)$ определяется соотношением (2), а отложение феромона соотношением (3).

Особенность третьего мема M_3 заключается в том, что на вершине x_i графа $G(X, U)$ формируются два показателя φ_{i1k} и φ_{i2k} :

φ_{i1k} – суммарный уровень феромона, отложенного на x_i только в тех случаях, когда x_i была в составе $X_{Ik}(t)$;

φ_{i2k} – суммарный уровень феромона, отложенного на x_i только в тех случаях, когда x_i не входила в состав $X_{Ik}(t)$.

Показатель φ_{1i} характеризует для вершины x_i предпочтительность узла $X_{Ik}(t)$.

Показатель φ_{2i} характеризует для вершины x_i предпочтительность узла $X_{2k}(t)$.

Процесс отложения феромона агентом выполняется следующим образом. Пусть агентом z_k сформировано множество X_{Ik} . Значения множества показателей φ_{1i} , соответствующих вершинам, вошедшим в состав X_{Ik} , а также значения множества показателей φ_{2i} , соответствующим вершинам, не вошедшим в состав X_{Ik} , увеличивается на величину, определяемую выражением (3).

Введем показатель $\theta_i = \varphi_{1i} - \varphi_{2i}$. Среди показателей θ_i с отрицательным значением определяется максимальное отрицательное значение δ : $(\forall i/\theta_i < 0)[\delta \geq |\theta_i|]$. Для ликвидации отрицательных значений показателей θ_i всем показателям присваивается новое значение $\theta_i + \delta$.

Потенциальная стоимость Φ_i связей $x_i \in X_{ck}(t)$ с $X_{Ik}(t)$ определяется как:

$$\Phi_{ik} = \alpha(\theta_i) + \beta(s_{ik}). \quad (5)$$

Вероятность P_{ik} включения вершины $x_i \in X_{ck}(t)$ в формируемый узел $X_{Ik}(t)$ определяется соотношением (3), а отложение феромона соотношением (4).

Четвертый мем отличается от третьего тем, что агентом формируются параллельно два множества $X_{Ik}(t)$ и $X_{2k}(t)$. Пусть $X_{ck}(t)$ множество еще не назначенных вершин. На шаге t для каждой вершины $x_i \in X_{ck}(t)$ определяется потенциальная стоимость Φ_{i1k} связей $x_i \in X_{ck}(t)$ с $X_{Ik}(t)$ и Φ_{i2k} – с $X_{2k}(t)$ определяется как:

$$\Phi_{i1k} = \alpha(\varphi_{i1k}) + \beta(s_{i1k}); \quad \Phi_{i2k} = \alpha(\varphi_{i2k}) + \beta(s_{i2k}). \quad (6)$$

Вероятность P_{i1k} включения вершины $x_i \in X_{ck}(t)$ в формируемый узел $X_{Ik}(t)$ определяется следующим соотношением:

$$P_{i1k} = \Phi_{i1k} / (\Phi_{i1k} + \Phi_{i2k}),$$

в формируемый узел $X_{Ik}(t)$ (7)

$$P_{i2k} = \Phi_{i2k} / (\Phi_{i1k} + \Phi_{i2k}).$$

В пятом меме для уменьшения общей трудоемкости поисковой процедуры используется подход, основанный на декомпозиции структуры данных в процессах формирования множества вершин $X_{Ik} \subset X$ и отложения феромона [12]. В рассмотренном выше меме M_1 формирование подмножества вершин $X_{Ik} \subset X$ и отложение феромона осуществляется на одном графе $R(X, E)$. В модифицированном подходе отложение феромона осуществляется на графе $R(X, E)$, а формирование подмножества вершин $X_{Ik} \subset X$ будем осуществлять в ориентированном графе $B(X, V)$, где X – множество вершин, $|X| = n$. Множество ориентированных ребер V формируется следующим образом. Из каждой вершины $x_i \in X$ выходит $(n-i)$ ребер, которые входят в вершины с номерами от $i+1$ до n .

Расположим (упорядочим) вершины графа $B(X, V)$ в линейку, так чтобы все ребра были направлены слева направо (рис. 2). Введем стартовую вершину St , связанную со всеми вершинами графа $B(X, V)$. Вершина St используется для начального размещения агентов и не входит в состав формируемого подмножества. Назовем граф $B(X, V)$ с расположенными в линейку (упорядоченными) вершинами шаблоном графа $B(X, V)$.

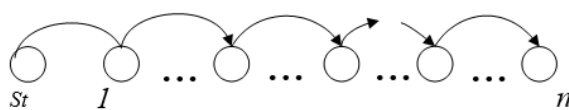


Рис. 2. Маршрут в графе $B(X, V)$

Сформировать подмножество X_t , содержащее n_t вершин, это значит построить на шаблоне графа $B(X, V)$ ориентированный маршрут M_{tk} , содержащий n_t вершин (рис. 2). Каждый из агентов формирует свое решение путем перемещения от одной вершины к другой на шаблоне графа поиска решений $B(X, V)$. На множестве ребер E будем откладывать феромон. Количество феромона на ребре характеризует преимущество выбора данного ребра по сравнению с другими при перемещении [12].

Процесс построения ориентированного маршрута M_{tk} пошаговый, начиная от вершины St поэтому, вначале, все агенты помещаются в вершину St . На каждом шаге t агент применяет вероятностное правило выбора очередной вершины на шаблоне графа $B(X, V)$ для включения ее формируемый маршрут M_{tk} . Выбор очередной вершины производится с учетом ограничения, накладываемого на список кандидатов, обеспечивающего возможность построения ориентированного маршрута, содержащего заданное число вершин. На первом шаге в маршрут M_{tk} может быть включена одна из вершин шаблона графа $B(X, V)$ с номерами от 1 до $n - (n_t - 1)$. Последние $(n_t - 1)$ вершин в линейке шаблона графа $B(X, V)$ (рис. 3) резервируются для того, чтобы можно было построить ориентированный маршрут, содержащий n_t вершин.

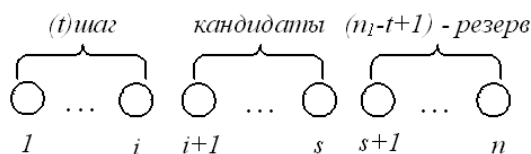


Рис. 3. Формирование маршрута

Пусть выполнено t шагов. Это значит, что в маршрут M_{tk} включено t вершин шаблона графа $B(X, V)$. Пусть последней вершиной маршрута, выбранной на шаге t , является вершина с номером i . Кандидатами на включение в маршрут на шаге $t+1$ являются вершины с номерами от $(i+1)$ до $s = n - (n_t - t + 1)$. Ограничения, накладываемые на список кандидатов (ОСК), резервируют $(n_t - t)$ последних в линейке вершин шаблона графа $B(X, V)$, начиная с номера $s+1$, для того, чтобы гарантировать возможность построения ориентированного маршрута, содержащего n_t вершин.

После построения маршрута, все вершины маршрута образуют множество X_1 , а все вершины, не входящие в маршрут, образуют множество X_2 .

При формировании подмножества X_1 , в графе $R(X, E)$ на первом шаге рассматривается n альтернатив, на втором шаге $(n-1)$ и т.д. На последнем шаге – $(n - (n_t - 1))$ альтернатив. Всего при формировании подмножества, содержащего n_t вершин, за n_t шагов всегда рассматривается:

$$n + (n-1) + (n-2) + \dots + (n - (n_t - 1)) = \left\{ \frac{n + (n - (n_t - 1))}{2} \right\} \cdot n_t = \left(\frac{2n - n_t + 1}{2} \right) \cdot n_t$$

альтернатив.

При построении маршрута в ориентированном графе $B(X, V)$ на первом шаге всегда рассматривается $(n - (n_t - 1)) = (n - n_t + 1)$ альтернатив. За $t-1$ предыдущих шагов в худшем случае могли быть выбранными $t-1$ вершин, расположенных в шаблоне гра-

фа $B(X, V)$ в позициях с l по $t-l$. На последующем шаге t эти вершины в количестве $t-l$ уже не могут быть кандидатами на включении в маршрут M_{lk} . Кроме того, из числа кандидатов шаге t для резервирования исключаются последних вершин в линейке вершин шаблона графа $B(X, V)$. Таким образом, на шаге t могут рассматриваться $[(n-(t-l)-(n_l-t))=(n-n_l+1)]$ альтернатив. Всего при формировании подмножества, содержащего n_l вершин, за n_l шагов всегда рассматривается $[(n-n_l+1)+(n-n_l+1)]/2 \cdot n_l = [(2n-2n_l+2)]/2 \cdot n_l$. Из сравнения формул видно, что при втором подходе даже в худшем случае трудоемкость поисковой процедуры значительно меньше.

При втором подходе отложение феромона на ребрах графа $R(X, E)$ осуществляется дважды. Вначале, также, как и при первом подходе, каждый муравей откладывает феромон на ребрах полного подграфа $R_{lk} \subset R$, построенного на вершинах множества X_{lk} , а затем откладывается дополнительное количество феромона на всех ребрах построенного муравьем ориентированного маршрута, включающего вершины множества X_{lk} . Количество добавляемого феромона рассчитывается в соответствии с оценкой маршрута.

2. Мульти-меметическая гибридизация. Результаты ряда исследований [1–9] свидетельствуют о том, что задачу выбора лучшего мема следует рассматривать как многокритериальную задачу. Основной задачей конструирования таких алгоритмов является выбор целесообразной стратегии использования того или иного мема из роя доступных мемов. Задача относится к классу задач метаоптимизации [10, 14–19]. Указанную стратегию иногда называют гиперэвристикой (hyperheuristic). Наиболее известны три следующие категории гиперэвристик, используемых в гиперэвристических адаптивных мульти-меметических алгоритмах: случайные (random) гиперэвристики, жадные (greedy) гиперэвристики, гиперэвристики с функцией выбора (choice-function). В предлагаемом алгоритме использован один из вариантов жадной гиперэвристики. Мульти-меметическая гибридизация приводит к алгоритму, обладающему свойствами самоадаптации, поскольку в процессе поиска мемы конкурируют между собой и победивший на данном этапе поиска мем используется на последующих этапах.

В работе используется следующий адаптивный алгоритм. На каждой итерации l определяется набор мемов $M(l) = \{M_k | k=1, 2, \dots, v\}$, который будет использоваться на этой итерации. Популяция агентов Z разбивается на v подпопуляций Z_k равной мощности, число которых равно числу мемов, используемых агентами на итерации l . Число агентов в подпопуляции Z_k пропорционально среднему значению целевой функции $D_k(l-1)$ решений, полученных агентами с использованием мема M_k по итогам работы на предыдущих итерациях. Далее на итерации l сначала выполняется процедура, соответствующая первому этапу муравьиного алгоритма. Каждая подпопуляция агентов Z_k формирует множество решений с использованием мема M_k . Рассчитываются значения целевой функции для всех решений. После этого выполняется второй этап МА. Все агенты всех подпопуляций откладывают феромон на графах $R(X, E)$, $G(X, U)$, в соответствии оценками решений, построенных на итерации l . На третьем этапе МА выполняется испарение феромона на графах $R(X, E)$, $G(X, U)$. Далее по итогам работы на итерации l для каждой подпопуляции Z_k подсчитывается среднее значение целевой функции $D_k(l)$ решений, полученных агентами с использованием мема M_k . Определяется подпопуляция Z_k с худшим значением $D_k(l)$. Из набора $M(l)$ удаляется мем M_k : $M(l+1) = M(l) \setminus M_k$. Популяция агентов разбивается на $v-l$ подпопуляций Z_k равной мощности. На следующей итерации агенты работают с набором мемов $M(l+1)$. Поскольку число итераций муравьиного алгоритма, как правило, значительно больше числа мемов, то с одним набором мемов $M(l)$ выполняется несколько итераций МА. Другими словами, модификация набора мемов выполняется после выполнения порции итераций. При этом подсчитывается среднее значение целевой функции $D_k(l)$ решений, полученных на всех итерациях агентами подпопуляции Z_k с использованием мема M_k .

Второй подход предусматривает перераспределения *количественного* состава подпопуляций Z_k после каждого выполнения порции итераций. Подсчитывается среднее значение целевой функции D_k решений, полученных на всех итерациях агентами подпопуляции Z_k с использованием мема M_k . Определяется минимальная оценка D_{min} . ($\forall k$) [$D_{min} \leq D_k$]. Для каждой подпопуляции Z_k определяется оценка $\delta_k = D_k - D_{min}$. Количественный состав N_k подпопуляции Z_k пропорциональный оценке δ_k определяется по формуле $N_k \approx (\delta_k / \sum_k(\delta_k)) \cdot |Z|$. Результаты статистического анализа показывают, что предложенный адаптивный мульти-меметический алгоритм, использующий рой мемов дает наилучший результат за тоже число итераций, что и одномемные алгоритмы. При этом результаты статистических тестов показывают, что использование первого подхода имеет повышенную сходимость алгоритма в области некоторого локального минимума.

3. Экспериментальные исследования. Для проведения экспериментов была использована процедура синтеза контрольных примеров с известным оптимумом F_{opt} по аналогии с известным методом BEKU (Partitioning Examples with Tight Upper Bound of Optimal Solution) [13,20].

Эксперименты проводились на десяти сгенерированных графах. Исследованию подвергались примеры, содержащие до 200 вершин ($n_1=20, n_2=40, n_3=60, n_4=80, n_5=100, n_6=120, n_7=140, n_8=160, n_9=180, n_{10}=200$). Число ребер m в графе G в среднем равно $n^2/4$. Вес вершин принимался равным нулю, а вес всех ребер – единице. При этом графы «разбивались» на два подграфа с равным количеством вершин в каждом подграфе.

В рамках работы проведено сравнительное исследование эффективности работы мульти-меметического алгоритма в зависимости от используемых алгоритмов локальной оптимизации и их числа. Поскольку эффективность алгоритма существенно зависит от случайных значений свободных параметров, во всех исследованиях каждый вычислительный эксперимент проводился по методу мульти-старта 100 раз; максимальное число итераций локального поиска при каждом запуске мема равно 100. Число агентов равно 100. В качестве критериев оценки эффективности используется среднее значение целевой функции по итогам мульти-старта, а также число итераций, которое при заданном ограничении на число итераций локального поиска однозначно определяет максимальное число испытаний. Результаты испытаний приведены на рис. 4, представлены средние значения отклонения целевых функций от оптимального значения F_{opt} в %.

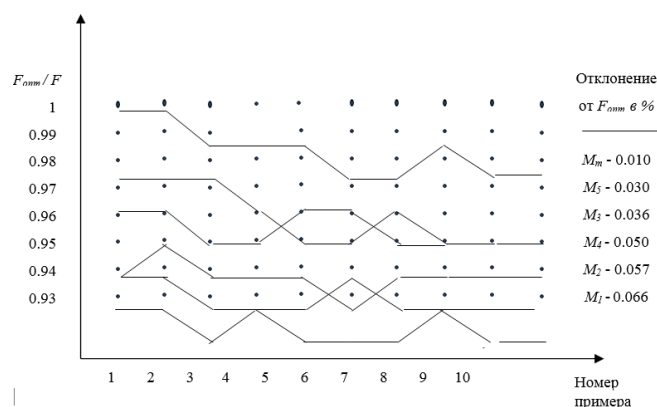


Рис. 4. Результаты экспериментов

В начале испытания проводились для каждого мема в отдельности. Все мемы обеспечили высокие результаты. Наилучшие результаты были получены при использовании мемов M_3 и M_4 . Алгоритм с мемом M_1 завершал работу через небольшое число итераций, что свидетельствует о преждевременной сходимости метода в области некоторого локального минимума. Использование всех мемов наиболее эффективно для примеров малой и средней размерности.

Затем испытания проводились с использованием для решения всех мемов в случайной последовательности. Исследования показали, что результирующая оценка и суммарное время решения задачи мало зависят от последовательности. Однако значительно вырастает общее время решения задачи.

Результаты статистического анализа показывают, что предложенный адаптивный мульти-меметический алгоритм, использующий рой мемов $M_m = \{M_1, M_2, M_3, M_4, M_5\}$ дает наилучший результат за тоже число итераций, что и одномемные алгоритмы. При этом результаты статистических тестов показывают, что использование первого варианта с мемом M_1 имеет повышенную сходимость алгоритма в области некоторого локального минимума.

Заключение. В работе приведены модификации стандартной встроенной конструктивной процедуры муравьиного алгоритма. Используется двойное и двухуровневое отложение феромона, декомпозиция структуры данных, используемой в процессах построения маршрута (формирование подграфа) и для отложения феромона. В отличие от канонической парадигмы муравьиного алгоритма муравьем на графе поиска решений $R(X,E)$ строится не маршрут, а формируется подграф. Рассмотрена концепция представления МА как гибридизация мема с базовой итерационной структурой. Разработан рой мемов в количестве 5 штук. Исследованы МА построенные на каждом меме. Анализ среднего числа итераций показывает, что использование мемов использующих отложение феромона на вершинах графа позволяет отыскать оптимальное решение за меньшее число итераций. Выполнен широкий вычислительный эксперимент по исследованию эффективности различных наборов мемов в зависимости от их вида и количества. Предложены два подхода к построению адаптивного мульти-меметического алгоритма, использующего рой мемов. Мульти-меметическая гибридизация приводит к алгоритму, обладающему свойствами самоадаптации, поскольку в процессе поиска мемы конкурируют между собой и победивший на данном этапе поиска мем используется на последующих итерациях. Результаты исследования демонстрируют, что мульти-меметический метод оказался наиболее эффективным для выбранных примеров. Представленные результаты исследования позволяют сделать вывод, что предложенная гибридизация муравьиного алгоритма позволяет не только увеличить вероятность локализации глобального экстремума целевой функции, но и повысить точность решения. Кроме того, за счет повышения эффективности поиска, гибридный алгоритм дает возможность уменьшить число агентов в системе и повысить тем самым его быстродействие.

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. Cotta C., Moscato P.B. Handbook of memetic algorithms. – Springer, 2012. – 368 p.
3. Moscato P., Corne D., Glover F., Dorigo M. Memetic algorithms: A short introduction // In book: New Ideas in Optimization. – McGraw-Hill, 1999. – P. 219-234.
4. Krasnogor N. Studies on the theory and design space of memetic algorithms. Doct. diss. – Bristol: Univ. of the West of England, 2002. – 412 p.
5. Neri F., Cotta C. Memetic algorithms and memetic computing optimization: A literature review // Swarm and Evolutionary Computation. – 2012. – P. 1-14.

6. *Karpenko A.P., Sakharov M.K.* Мультиимеявая глобальная оптимизация на основе алгоритма эволюции разума // Информационные технологии. – 2014. – № 7. – С. 23-30.
7. *Karpenko A.P., Chernobrivchenko K.A.* Мультиимеявая модификация гибридного муравьиного алгоритма непрерывной оптимизации HCIAC // Электронный научно-технический журнал. Эл. №ФС7748211, 2012.
8. *Lebedev B.K., Lebedev O.B.* Меметический алгоритм разбиения // Вестник Ростовского государственного университета путей сообщения. – 2017. – № 2 (62). – С. 136-145.
9. *Lebedev B.K., Lebedev O.B.* Гибридный биоинспирированный алгоритм на основе интеграции метода ветвей и границ и метода муравьиной колонии // Вестник Ростовского государственного университета путей сообщения. – 2018. – № 2 (70). – С. 77-88.
10. *Yew-Soon Ong, Meng-Hiot Lim, Ning Zhu, Kok-Wai Wong.* Classification of adaptive memetic algorithms: A comparative study // IEEE Trans. on Systems, Man, and Cybernetics. – 2006. – Vol. 36, Issue 1. – P. 141-152.
11. *Dorigo M. and Stützle T.* Ant Colony Optimization. – MIT Press, Cambridge, MA, 2004. – 345 p.
12. *Lebedev B.K., Lebedev O.B., Lebedeva E.M.* Распределение ресурсов на основе гибридных моделей роевого интеллекта // Научно-технический вестник информационных технологий, механики и оптики. – 2017. – Т. 17, № 6. – С. 1063-1073.
13. *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, 2003. – P. 88-94.
14. *Воробьева Е.Ю., Карпенко А.П., Селиверстов Е.Ю.* Ко-гибридизация алгоритмов роя частиц // Наука и образование. МГТУ им. Н.Э. Баумана. Электронный журнал. – 2012. – № 4.
15. *Lebedev B.K., Lebedev O.B., Lebedev V.B.* Гибридизация роевого интеллекта и генетической эволюции на примере размещения // Программные продукты, системы и алгоритмы. – DOI: 10.15827/2311-6749.25.280. – <http://swsys-web.ru/hybridization-of-swarm-intelligence-and-genetic-evolution.html>.
16. *Cheng Yu., Sha D.Y.* A hybrid particle swarm optimization for job shop scheduling problem // Computers & Industrial Engineering. – 2006. – P. 791-808.
17. *Агасиев Т.А., Карпенко А.П.* Современные техники глобальной оптимизации // Информационные технологии. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2018. – № 6. – С. 370-386.
18. *Clerc M.* Particle Swarm Optimization. – ISTE, London, UK, 2006. – 248 p.
19. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Разбиение на основе моделирования адаптивного поведения биологических систем // Нейрокомпьютеры: разработка, применение. – 2010. – № 2. – С. 28-33.
20. *Ong Y.S., Keane A.J.* Meta-Lamarckian learning in memetic algorithms // IEEE Transactions on Evolutionary Computation. – 2004. – Vol. 8 (2). – P. 99-110.

REFERENCES

1. *Karpenko A.P.* Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Nature-inspired algorithms: textbook]. Moscow: Izd-vo MGTU im. N.E. Bauman, 2014, 448 p.
2. *Cotta C., Moscato P.B.* Handbook of memetic algorithms. Springer, 2012, 368 p.
3. *Moscato P., Corne D., Glover F., Dorigo M.* Memetic algorithms: A short introduction, *In book: New Ideas in Optimization.* – McGraw-Hill, 1999, pp. 219-234.
4. *Krasnogor N.* Studies on the theory and design space of memetic algorithms. Doct. diss. Bristol: Univ. of the West of England, 2002, 412 p.
5. *Neri F., Cotta C.* Memetic algorithms and memetic computing optimization: A literature review, *Swarm and Evolutionary Computation*, 2012, pp. 1-14.
6. *Karpenko A.P., Sakharov M.K.* Multimeevaya global'naya optimizatsiya na osnove algoritma evolyutsii razuma [Multimeevoy global optimization based on the algorithm of evolution of the mind], *Informatsionnye tekhnologii* [Information technologies], 2014, No. 7, pp. 23-30.
7. *Karpenko A.P., Chernobrivchenko K.A.* Multimeevaya modifikatsiya gibridnogo murav'inogo algoritma nepreryvnoy optimizatsii HCIAC [Multimeic modification of the hybrid ant algorithm of continuous optimization HCIAC], *Elektronnyy nauchno-tekhnicheskiy zhurnal* [Electronic Scientific and Technical Journal]. El. No.FS7748211, 2012.
8. *Lebedev B.K., Lebedev O.B.* Memeticheskiy algoritm razbieniya [Memetic algorithm of splitting], *Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya* [Bulletin of the Rostov State University of Communications], 2017, No. 2 (62), pp. 136-145.

9. *Lebedev B.K., Lebedev O.B.* Gibridnyy bioinspirirovannyy algoritm na osnove integratsii metoda vetvey i granits i metoda murav'inoy kolonii [Hybrid bioinspired algorithm based on the integration of the branch and bound method and the ant colony method], *Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya* [Bulletin of the Rostov State University of Communications], 2018, No. 2 (70), pp. 77-88.
10. *Yew-Soon Ong, Meng-Hiot Lim, Ning Zhu, Kok-Wai Wong.* Classification of adaptive memetic algorithms: A comparative study, *IEEE Trans. on Systems, Man, and Cybernetics*, 2006, Vol. 36, Issue 1, pp. 141-152.
11. *Dorigo M. and Stützle T.* Ant Colony Optimization. MIT Press, Cambridge, MA, 2004, 345 p.
12. *Lebedev B.K., Lebedev O.B., Lebedeva E.M.* Raspredelenie resursov na osnove gibridnykh modeley roevogo intellekta [Resource allocation based on hybrid models of swarm intelligence], *Nauchno-tekhnicheskiiy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics], 2017, Vol. 17, No. 6, pp. 1063-1073.
13. *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, 2003*, pp. 88-94.
14. *Vorob'eva E.Yu., Karpenko A.P., Seliverstov E.Yu.* Ko-gibridizatsiya algoritmov roya chastits [Co-hybridization of particle swarm algorithms], *Nauka i obrazovanie. MGTU im. N.E. Baumana. Elektronnyy zhurnal* [Science and Education. MGTU them. N.E. Bauman. Electronic journal], 2012, No. 4.
15. *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 on the example of placement], *Programmnye produkty, sistemy i algoritmy* [Software Products, Systems and Algorithms]. Available at: <http://swwsweb.ru/hybridization-of-swarm-intelligence-and-genetic-evolution.html>. DOI: 10.15827/2311-6749.25.280.
16. *Cheng Yu., Sha D.Y.* A hybrid particle swarm optimization for job shop scheduling problem, *Computers & Industrial Engineering*, 2006, pp. 791-808.
17. *Agasiev T.A., Karpenko A.P.* Sovremennyye tekhniki global'noy optimizatsii [Modern technology of global optimization], *Informatsionnye tekhnologii* [Information technologies]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2018, No. 6, pp. 370-386.
18. *Clerc M.* Particle Swarm Optimization. ISTE, London, UK, 2006, 248 p.
19. *Kureychik V.M., Lebedev B.K., Lebedev O.B.* Razbienie na osnove modelirovaniya adaptivnogo povedeniya biologicheskikh sistem [Splitting based on modeling adaptive behavior of biological systems], *Neyrokomp'yutery: razrabotka, primeneniye* [Neurocomputers: development, application], 2010, No. 2, pp. 28-33.
20. *Ong Y.S., Keane A.J.* Meta-Lamarckian learning in memetic algorithms, *IEEE Transactions on Evolutionary Computation*, 2004, Vol. 8 (2), pp. 99-110.

Статью рекомендовал к опубликованию д.т.н., профессор А.В. Чернов.

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: lbedevakate@mail.ru; тел.: 89289591426; кафедра систем автоматизированного проектирования; аспирант.

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: lbedevakate@mail.ru; phone: +79289591426; the department of computer aided design; graduate student.