

# Оптимизация, системный анализ и исследование операций

© 2023 г. **И.Н. СИНИЦЫН**, д-р техн. наук (sinitsin@dol.ru),  
**Ю.П. ТИТОВ**, канд. техн. наук (kalengul@mail.ru)  
(Московский авиационный институт  
(национальный исследовательский университет);  
Федеральный исследовательский центр  
“Информатика и управление” РАН, Москва)

## УПРАВЛЕНИЕ НАБОРАМИ ЗНАЧЕНИЙ ПАРАМЕТРОВ СИСТЕМЫ МЕТОДОМ МУРАВЬИНЫХ КОЛОНИЙ

Рассматриваются модификация и применение метода муравьиных колоний для задачи направленного перебора значений параметров системы при выполнении расчетных многократных вычислений. Взаимодействие с пользователем позволяет останавливать процесс полного перебора наборов значений параметров, а применение модификации метода муравьиных колоний позволит рассмотреть рациональные наборы на ранних итерациях. Если пользователь не завершает работу алгоритма, то предложенные модификации позволяют перебрать все решения методом муравьиных колоний. Для модификации метода муравьиных колоний предложены новая вероятностная формула и различные алгоритмы метода муравьиных колоний, позволяющие для каждого агента находить новый набор значений параметров. Оптимальным алгоритмом, по результатам исследований, является применение повторного бесконечного циклического поиска нового решения. Данная модификация позволяет рассмотреть все решения и при этом найти все оптимальные решения среди первых 5% рассмотренных решений.

*Ключевые слова:* метод муравьиных колоний, параметрический граф, изменение порядка следования, вычислительный кластер, оптимизация гиперпараметров.

**DOI:** 10.31857/S000523102308010X, **EDN:** HDNFSR

### 1. Введение

В настоящее время в связи с развитием вычислительных кластеров множество расчетных и оптимизационных задач переносится с человека-эксперта на вычислительные машины. Для таких систем возникают задачи нахождения рациональных значений параметров для решения расчетной задачи, называемые оптимизацией гиперпараметров [1]. Среди алгоритмов оптимизации

гиперпараметров выделяют Байесовский оптимизатор, позволяющий на основе гипотез и апостериорной информации находить закономерности влияния значений отдельных параметров (и различных комбинаций) на критерии эффективности [2]. Для многокритериальных и многоэкстремальных задач часто необходимо рассмотрение всех наборов значений параметров, обычно методом полного перебора. В работе рассматривается возможность применения метода муравьиных колоний для решения задачи направленного перебора наборов гиперпараметров перед отправкой на вычислительную систему. За счет взаимодействия с пользователем возможна остановка работы метода до рассмотрения всех наборов значений параметров при нахождении удовлетворяющего пользователя набора. Направленный перебор методом муравьиных колоний позволит рассмотреть рациональные наборы значений параметров как можно раньше, но в случае неудовлетворенности пользователя результатами перебрать все наборы значений.

Метод муравьиных колоний первоначально разрабатывался для решения задачи коммивояжера [3, 4]. Современные исследования позволяют применять метод муравьиных колоний для поиска непрерывной оптимизации. Поиск методами CASO (ContinuousAntColonyOptimization — непрерывная оптимизация колонией муравьев) [5], ACOR (Ant Colony Optimization for continuous domain) [6] и CIAC (ContinuousInteractingAntColony — непрерывно взаимодействующая колония муравьев) [7, 8] не предполагает использования графа и активно исследовался, в том числе и российскими исследователями [9, 10]. Исследования описывают возможность применения метода муравьиных колоний для решения задач на графах: задачи о назначении с нечетким временем выполнения [11, 12], задачи нахождения оптимальных маршрутов для группы коммивояжеров [13, 14], задачи поддержки процессов поставок запасных частей [15]. В мировом сообществе активно исследуются параметрические задачи, связанные с поиском оптимального набора параметров, классификации, выявления зависимостей и т.д. [16–19]. Для подобных задач происходит создание специальной структуры графа. Представленные методы и модификации метода муравьиных колоний предназначены для поиска приближенных, рациональных решений. Обычно все агенты (муравьи) должны сойтись к одному решению. Поиск новых решений осуществляется путем мультистарта [9].

Для направленного перебора значений параметров необходимо не сойтись к одному решению (набору значений параметров), а последовательно рассматривать новые решения до остановки работы метода пользователем или рассмотрения всех возможных решений. В работе предложены модификации алгоритма для рассмотрения всех решений, а не сходимости к одному. Предложенный подход позволяет решать задачи с векторным критерием оптимальности и многомодальными целевыми функциями без повторного запуска алгоритма. При этом сохраняется свойство алгоритмов оптимизации: наискорейшее нахождение оптимальных решений.

## 2. Модели и методы

В основе метода муравьиных колоний лежит вероятностный поиск дуги в графе в соответствии с формулой

$$(1) \quad P_{ij,k}(t) = \frac{\tau_{ij}^{\alpha} \cdot \mu_{ij,t}^{\beta}}{\sum_{z \in J_{i,k}} \left( \tau_{iz}^{\alpha} \cdot \mu_{iz,t}^{\beta} \right)}.$$

С помощью (1) определяется вероятность перехода агента из текущей вершины  $i$  в вершины из множества  $J_{i,k}$  на итерации  $t$ . По результатам вычислений определяется вероятность перехода в вершину  $j$  для  $k$ -го агента. В (1) учитывается информация о длине дуги  $\tau_{ij}^{\alpha}$  (удаленность) и некотором весе  $\mu_{ij,t}^{\beta}$  (феромоне). Значение  $\tau_{ij}^{\alpha}$  является фиксированным и не зависит от номера итерации  $t$ . Количество весов  $\mu_{ij,t}^{\beta}$  изменяется между итерациями, обновляя состояние графа, внешнюю среду для движения агентов.

Для определения значений параметров системы (решения) предлагается представить наборы значений в виде параметрического графа. Каждое конкретное значение одного параметра представляет вершину графа. Все значения одного параметра объединяются в слои. Из каждой вершины одного слоя имеется дуга в каждую вершину следующего слоя. Слои вершин расположены в определенном порядке, что позволяет уменьшить количество дуг в графе. Пример параметрического графа приведен на рис. 1. Подобные графы встречались в [13, 15, 18–21].

Веса (феромон) в подобном графе заносятся не на дуги, а на вершины. В результате дуги являются фиктивными, и подобный параметрический граф может быть представлен в виде множества слоев (параметров) и множества вершин (значений параметра) [22].

Для параметрического графа в вероятностной формуле (1) значение  $\tau_{ij}^{\alpha}$  может быть задано только на основе априорной информации от эксперта, но в общем виде данный параметр не может быть определен. При использовании в вероятностной формуле одного сомножителя увеличивается стагнация

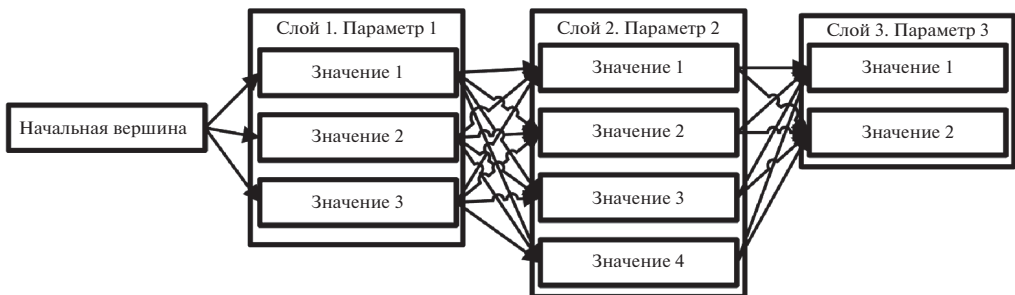


Рис. 1. Схема структуры параметрического графа.

процесса поиска решения. При этом алгоритм сходится к первому хорошему решению и не продолжает поиск оптимального.

Для решения проблемы стагнации можно производить «сброс» параметрического графа, перевод состояния вершин графа в начальное состояние или модифицировать вероятностную формулу

$$(2) \quad P_{ij,k}(t) = \frac{k1 \cdot \mu_{norm\ ij,t}^\alpha + k2 \cdot \left(\frac{1}{kol(t)_{ij,t}}\right)^\beta + k3 \cdot \left(\frac{kol(t)_{ij,t}}{MaxKol_{j,t}}\right)^\gamma}{\sum_{z \in J_{i,k}} \left(k1 \cdot \mu_{norm\ iz,t}^\alpha + k2 \cdot \left(\frac{1}{kol(t)_{iz,t}}\right)^\beta + k3 \cdot \left(\frac{kol(t)_{iz,t}}{MaxKol_{z,t}}\right)^\gamma\right)}.$$

Формула (2) представляет собой линейную свертку (а не мультипликативную, как в (1)) трех слагаемых и весовых коэффициентов. Первое слагаемое  $\mu_{norm\ ij,t}^\alpha$  определяется количеством весов у  $i$ -й вершины  $j$ -го параметра (слоя параметрического графа) на итерации  $t$ . Для применения данного параметра во взвешенной сумме необходимо использовать нормализованное значение. Второе слагаемое  $\left(\frac{1}{kol(t)_{ij,t}}\right)^\beta$  определяется количеством посещений агентами  $i$ -й вершины  $j$ -го параметра за время работы алгоритма. Данное слагаемое увеличивает вероятность посетить вершину, которая редко присутствует в решениях и позволяет избегать стагнации на ранних итерациях метода муравьиных колоний. В третьем слагаемом  $\left(\frac{kol(t)_{ij,t}}{MaxKol_{j,t}}\right)^\gamma$  учитывается максимальное количество возможных посещений вершины — значения  $MaxKol_{j,t}$  для параметра  $j$  на итерации  $t$ . Так как в параметрическом графе на каждом слое необходимо выбрать одну вершину (одно значение параметра), то можно вычислить общее количество решений, наборов значений параметров. Общее количество решений может быть вычислено как произведение количества вершин в каждом слое. Максимальное количество решений, в которых может содержаться конкретная вершина параметрического графа, вычисляется как отношение общего количества решений и количества вершин в данном слое, т.е. для каждой вершины слоя  $j$  значение  $MaxKol_{j,t}$  будет одинаковым. Третье слагаемое на поздних итерациях позволяет увеличить вероятность выбора вершины, для которой рассмотрено большинство вариантов решений. Если для вершины рассмотрены все возможные решения, то из вероятностного поиска данную вершину можно исключить. Аддитивная свертка позволяет компенсировать значения слагаемых. Вершины с большим количеством весов и частыми посещениями (часто рассматриваемые вершины) могут быть компенсированы вершинами с небольшим количеством посещений (редкие вершины) или вершинами, для которых рассмотрены практически все решения.

Другой особенностью модификации алгоритма является необходимость взаимодействия с внешней вычислительной системой. Для таких модификаций необходимо хранить состояние системы, вычисленное для конкретного решения. Если метод муравьиных колоний повторно находит решение, то данное решение повторно не отправляется на вычислитель, а значение целевой функции берется из Хэш-таблицы [15, 21].

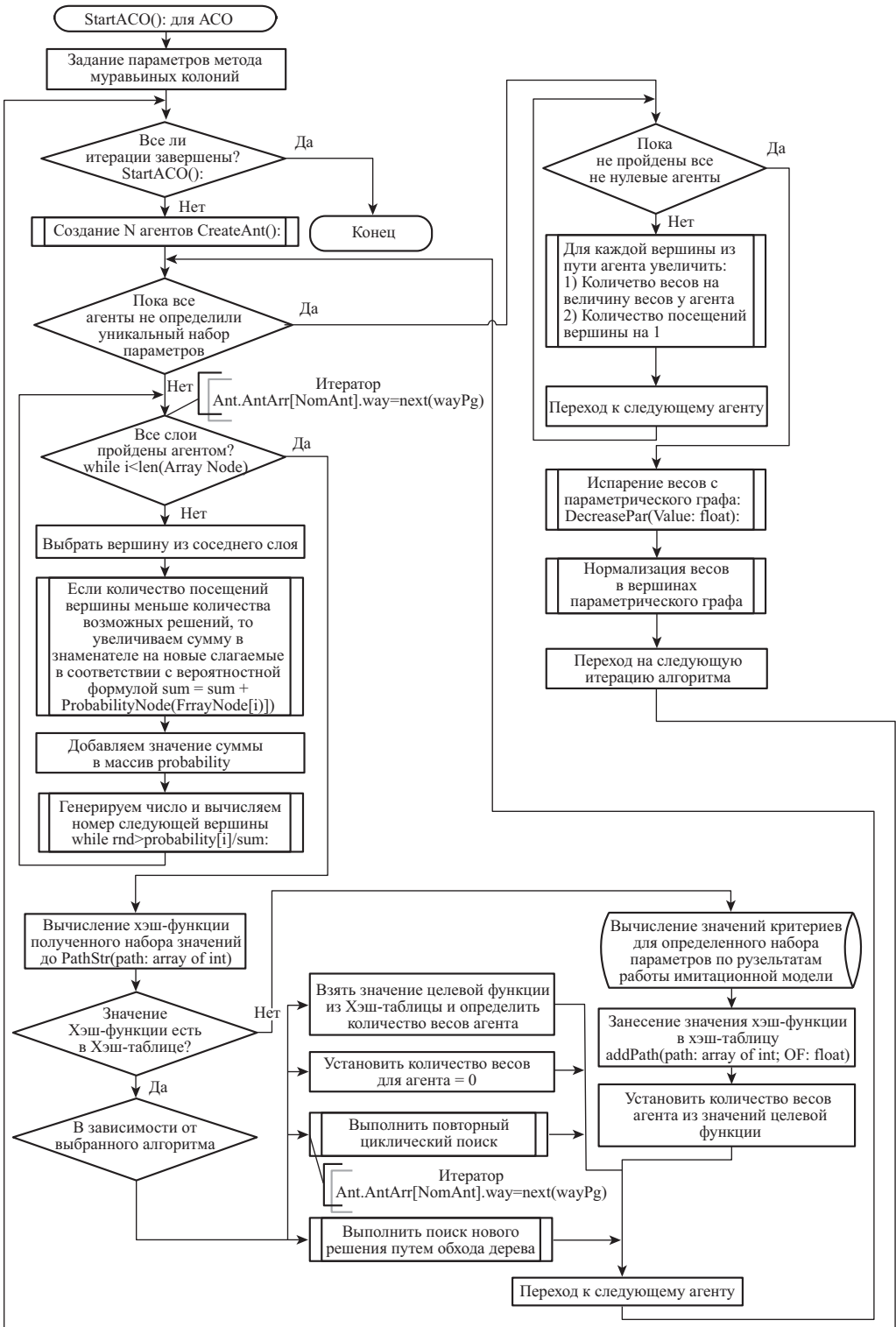


Рис. 2. Схема алгоритма модифицированного метода муравьиных колоний.

Для рассматриваемого алгоритма важен поиск нового решения на каждой итерации и поэтому, если решение уже имеется в Хэш-таблице, возможны различные действия:

1. Используя значения целевой функции из Хэш-таблицы, занести веса, как в оригинальном алгоритме.
2. Игнорировать агента. Агент не заносит веса на параметрический граф.
3. Повторный поиск нового, еще не рассмотренного на вычислителе, решения методом муравьиных колоний с ограничением на количество итераций. Если за установленное количество итераций не найдено новое решение, то агент игнорируется.
4. Повторный поиск нового, еще не рассмотренного на вычислителе, решения методом муравьиных колоний неограниченным количеством итераций. Ограничение в пункте № 3 может решить проблему стагнации.
5. Повторный поиск нового решения другим алгоритмом. Рассматривалась возможность обхода параметрического графа как дерева.

Схема алгоритма предлагаемой модификации метода муравьиных колоний приведена на рис. 2.

### 3. Эксперимент

Задачами направленного перебора значений параметров являются:

- перебор всех, без исключения, наборов значений параметров;
- наискорейшее получение оптимального набора значений параметра.

Можно заметить, что обе задачи противоречат друг другу, так как среди всех наборов значений параметров всегда будет наилучший. Но так как система взаимодействует с вычислителем и пользователем в режиме реального времени, то имеется возможность остановки программы пользователем, если найдено удовлетворяющее его решение. Также следует отметить возможность использования векторного критерия, переводящего задачу в область поддержки принятия решений и многокритериальной оптимизации [20].

Эффективность модификаций метода муравьиных колоний определяется двумя основными оценками:

- возможность алгоритма рассмотреть все решения. Определяется значением критерия: «оценка вероятности найти новое решение агентом». Данная оценка вычисляется путем отношения количества найденных решений за время работы алгоритма на общее количество рассмотренных решений;
- скорость нахождения оптимального решения. Определяется значением критерия: «оценка математического ожидания номера итерации, на котором данное решение было найдено».

Проведение экспериментов осуществлялось на программном обеспечении АСО Cluster, написанном на языке Python. В качестве тестовых данных рассматривалось большинство из бенчмарков, взятых из [22, 23], и тестовые графы большой размерности [21]. Структура графа большой размерности приведена на рис. 3. Представленный граф большой размерности содержит бо-

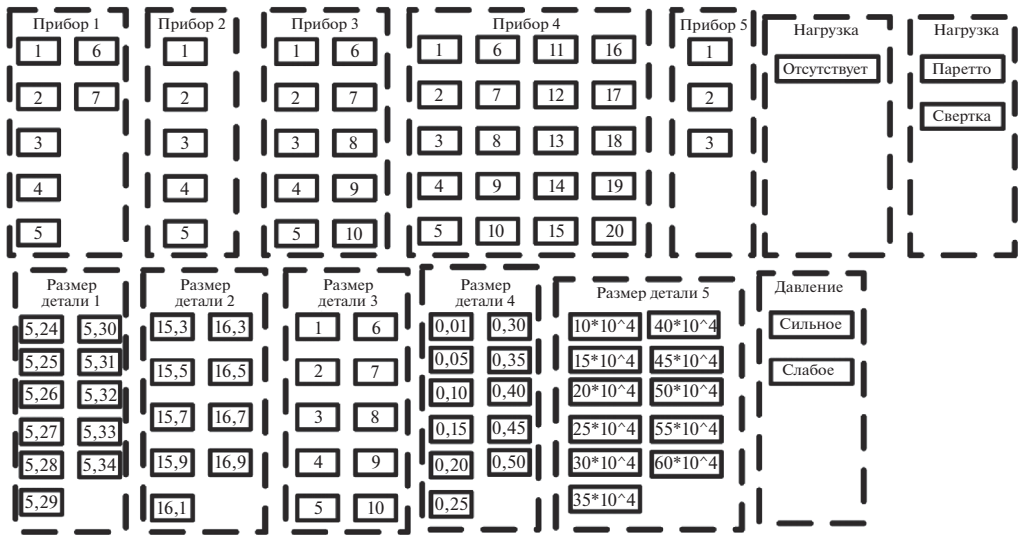


Рис. 3. Схема параметрического графа большой размерности.

более  $10^9$  решений по 13 параметрам, заданных в дискретных и качественных шкалах. При 25 агентах на одной итерации и 20 000 итерациях максимально может быть рассмотрено только 0,05% решений от общего числа.

Для анализа работы алгоритма при поиске последних (еще не рассмотренных) решений в результатах приведены исследования функции «Camom table function»:

$$(3) \quad f(x) = - \frac{\left( \cos(x_1) \cos(x_2) e^{|100 - (x_1^2 + x_2^2)^{0,5}|} \right)^2}{30}.$$

Параметрический граф для функции (3) содержит два слоя, для параметров  $x_1$  и  $x_2$ . В каждом слое параметрического графа имеется по 201 вершине, определяющих конкретные значения параметра в диапазоне  $[-10, 10]$  с точностью 0,1. Для рассмотрения всех решений 25 агентами необходимо 1936 итераций метода муравьиных колоний при условии, что каждый агент найдет новое решение.

#### 4. Результат

В работе проведено исследование следующих модификаций алгоритма муравьиных колоний:

– ACOCC (ACO Cluster Classic) — Классический метод муравьиных колоний. В (2) коэффициенты  $k_2$  и  $k_3$  равны 0. В данном случае формула аддитивной свертки 2 становится мультипликативной сверткой формулы (1) с параметром  $\tau_{ij}^\alpha = 1$ . Если найдено уже рассмотренное алгоритмом решение, то значение целевой функции определяется из Хэш-таблицы.

**Таблица 1.** Оценка математического ожидания времени (в сек.) поиска решения одним агентом

Кол-во итер.	АСОСС	АСОСН	АСОСНИ	АСОССу3	АСОССуI	АСОСТ
2500	$1,404 \cdot 10^{-4}$	$1,547 \cdot 10^{-4}$	$1,562 \cdot 10^{-4}$	$1,627 \cdot 10^{-4}$	$1,619 \cdot 10^{-4}$	$1,887 \cdot 10^{-4}$
5000	$1,381 \cdot 10^{-4}$	$1,517 \cdot 10^{-4}$	$1,560 \cdot 10^{-4}$	$1,648 \cdot 10^{-4}$	$1,636 \cdot 10^{-4}$	$2,745 \cdot 10^{-4}$
7500	$1,388 \cdot 10^{-4}$	$1,505 \cdot 10^{-4}$	$1,567 \cdot 10^{-4}$	$1,665 \cdot 10^{-4}$	$1,647 \cdot 10^{-4}$	$4,158 \cdot 10^{-4}$
10 000	$1,391 \cdot 10^{-4}$	$1,501 \cdot 10^{-4}$	$1,578 \cdot 10^{-4}$	$1,690 \cdot 10^{-4}$	$1,654 \cdot 10^{-4}$	$5,645 \cdot 10^{-4}$
12 500	$1,370 \cdot 10^{-4}$	$1,547 \cdot 10^{-4}$	$1,562 \cdot 10^{-4}$	$1,706 \cdot 10^{-4}$	$1,657 \cdot 10^{-4}$	$6,980 \cdot 10^{-4}$
15 000	$1,364 \cdot 10^{-4}$	$1,526 \cdot 10^{-4}$	$1,569 \cdot 10^{-4}$	$1,700 \cdot 10^{-4}$	$1,650 \cdot 10^{-4}$	$8,593 \cdot 10^{-4}$
17 500	$1,328 \cdot 10^{-4}$	$1,472 \cdot 10^{-4}$	$1,585 \cdot 10^{-4}$	$1,695 \cdot 10^{-4}$	$1,655 \cdot 10^{-4}$	$9,776 \cdot 10^{-4}$
20 000	$1,325 \cdot 10^{-4}$	$1,469 \cdot 10^{-4}$	$1,582 \cdot 10^{-4}$	$1,741 \cdot 10^{-4}$	$1,688 \cdot 10^{-4}$	$10,549 \cdot 10^{-4}$

**Таблица 2.** Оценка вероятности найти новое решение одним агентом на итерации метода муравьиных колоний

Кол-во итер.	АСОСС	АСОСН	АСОСНИ	АСОССу3	АСОССуI	АСОСТ
2500	0,248	0,618	0,966	1,000	1,000	1,000
5000	0,122	0,381	0,951	1,000	1,000	1,000
7500	0,082	0,282	0,942	1,000	1,000	1,000
10 000	0,063	0,223	0,935	1,000	1,000	1,000
12 500	0,049	0,184	0,929	1,000	1,000	1,000
15 000	0,041	0,158	0,925	1,000	1,000	1,000
17 500	0,035	0,136	0,921	1,000	1,000	1,000
20 000	0,030	0,125	0,918	1,000	1,000	1,000

– АСОСН (ACO Cluster New) – Аналогичный классическому методу муравьиных колоний, но в (2) коэффициенты  $k_2$  и  $k_3$  равны 1. Вероятность перехода определяется аддитивной сверткой.

– АСОСНИ (ACO Cluster New Ignor) – В (2) коэффициенты  $k_2$  и  $k_3$  равны 1. Если найдено решение, уже записанное в Хэш-таблицу, то данный агент игнорируется, не заносит веса на вершины параметрического графа.

– АСОССу3 (ACO Cluster Cycle3) – В (2) коэффициенты  $k_2$  и  $k_3$  равны 1. Если найдено решение, уже записанное в Хэш-таблицу, то производится повторный поиск решения методом муравьиных колоний. Поиск нового решения производится циклически. Установлено ограничение на количество итераций повторного поиска, равное 3. Если за установленное количество итераций не найдено новое решение, то агент игнорируется.

– АСОССуI (ACO Cluster Cycle Infinity) – В (2) коэффициенты  $k_2$  и  $k_3$  равны 1. Если найдено решение, уже записанное в Хэш-таблицу, то производится повторный поиск решения методом муравьиных колоний. Поиск нового решения производится циклически без ограничения на количество итераций.

– АСОСТ (ACO Cluster Tree) – В (2) коэффициенты  $k_2$  и  $k_3$  равны 1. Если найдено решение, уже записанное в Хэш-таблицу, то производится по-



**Таблица 3.** Оценка вероятности найти новое решение одним агентом на итерации метода муравьиных колоний

Кол-во итер.	АСОСС	АСОСН	АСОСНИ	АСОССу3	АСОССуI	АСОСТ
2500	0,03	0,31	0,21	0,27	0,24	0,13
5000	0,03	0,49	0,26	0,26	0,33	0,25
7500	0,03	0,50	0,35	0,30	0,32	0,24
10 000	0,02	0,60	0,22	0,32	0,31	0,33
12 500	0,03	0,78	0,32	0,37	0,42	0,27
15 000	0,02	0,71	0,31	0,44	0,37	0,31
17 500	0,03	0,74	0,28	0,39	0,44	0,26
20 000	0,03	0,72	0,41	0,46	0,43	0,39

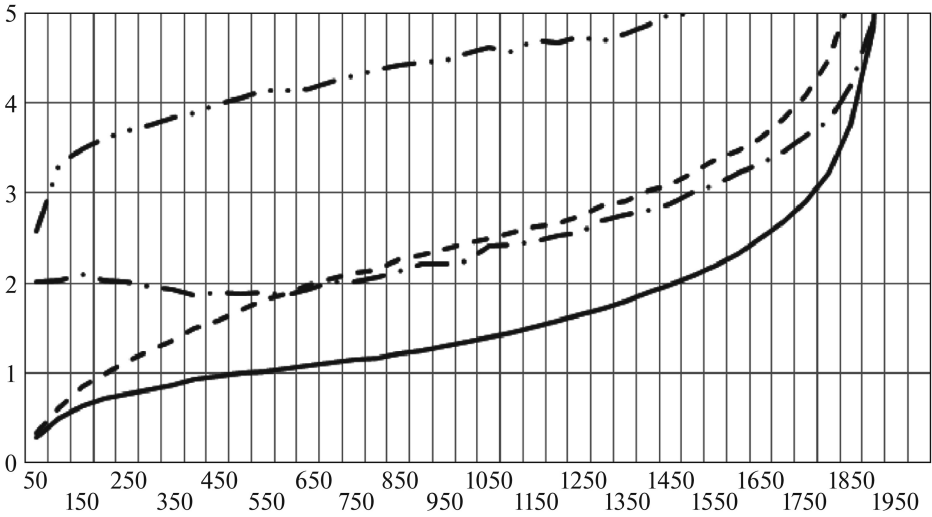
**Таблица 4.** Оценка математического ожидания номера решения (в %), на котором были найдены оптимальные значения параметров

Кол-во итер.	АСОСС	АСОСН	АСОСНИ	АСОССу3	АСОССуI	АСОСТ
2500	$0,801 \cdot 10^{-6}$	$2,956 \cdot 10^{-6}$	$3,404 \cdot 10^{-6}$	$2,970 \cdot 10^{-6}$	$3,299 \cdot 10^{-6}$	$2,863 \cdot 10^{-6}$
5000	$1,122 \cdot 10^{-6}$	$3,191 \cdot 10^{-6}$	$4,501 \cdot 10^{-6}$	$4,415 \cdot 10^{-6}$	$5,648 \cdot 10^{-6}$	$5,030 \cdot 10^{-6}$
7500	$1,014 \cdot 10^{-6}$	$3,378 \cdot 10^{-6}$	$5,701 \cdot 10^{-6}$	$6,137 \cdot 10^{-6}$	$5,830 \cdot 10^{-6}$	$6,838 \cdot 10^{-6}$
10 000	$1,142 \cdot 10^{-6}$	$3,667 \cdot 10^{-6}$	$5,693 \cdot 10^{-6}$	$5,337 \cdot 10^{-6}$	$5,910 \cdot 10^{-6}$	$7,704 \cdot 10^{-6}$
12 500	$0,829 \cdot 10^{-6}$	$3,770 \cdot 10^{-6}$	$6,779 \cdot 10^{-6}$	$6,320 \cdot 10^{-6}$	$8,108 \cdot 10^{-6}$	$5,269 \cdot 10^{-6}$
15 000	$0,740 \cdot 10^{-6}$	$3,741 \cdot 10^{-6}$	$8,240 \cdot 10^{-6}$	$10,174 \cdot 10^{-6}$	$8,794 \cdot 10^{-6}$	$7,834 \cdot 10^{-6}$
17 500	$1,393 \cdot 10^{-6}$	$3,845 \cdot 10^{-6}$	$8,175 \cdot 10^{-6}$	$9,221 \cdot 10^{-6}$	$11,678 \cdot 10^{-6}$	$15,820 \cdot 10^{-6}$
20 000	$1,119 \cdot 10^{-6}$	$3,936 \cdot 10^{-6}$	$9,864 \cdot 10^{-6}$	$11,344 \cdot 10^{-6}$	$10,513 \cdot 10^{-6}$	$23,592 \cdot 10^{-6}$

вторный поиск нового решения другим алгоритмом. Рассматривается обход параметрического графа как дерева.

Оценки результатов работы модификаций для параметрического графа большой размерности приведены в табл. 1–4. При применении алгоритма АСОСН количество решений, необходимых для поиска наилучшего набора значения параметров, минимально, и слабо возрастает с увеличением количества итераций (3-й столбец табл. 4). Быстрее находит решение только алгоритм АСОСС, но вероятность найти оптимальное решение алгоритмом АСОСС менее 0,1 (2-й столбец табл. 3). Алгоритм АСОСС стагнирует на ближайшем рациональном решении. Алгоритмы, использующие повторный поиск для нулевых агентов (АСОССу3, АСОССуI и АСОСТ), для каждого агента находят новое решение (столбцы 5–7 табл. 2). Для 25 агентов за  $2 \cdot 10^4$  итераций представленными алгоритмами будет рассмотрено  $5 \cdot 10^5$  наборов значений параметров. Время, затраченное на поиск агентом решения, для большинства алгоритмов не зависит от количества итераций алгоритма (табл. 1). В результате возможно прогнозировать необходимое количество итераций для поиска всех решений. Исключением является алгоритм АСОСТ, который при увеличении количества итераций требует больше времени на поиск пути одним агентом. По возрастанию времени поиска агентом решения можно про-

Оценка математического ожидания количества дополнительных итераций на одного агента



Оценка математического ожидания номера решения (в %), на котором был найден оптимальный набор параметров

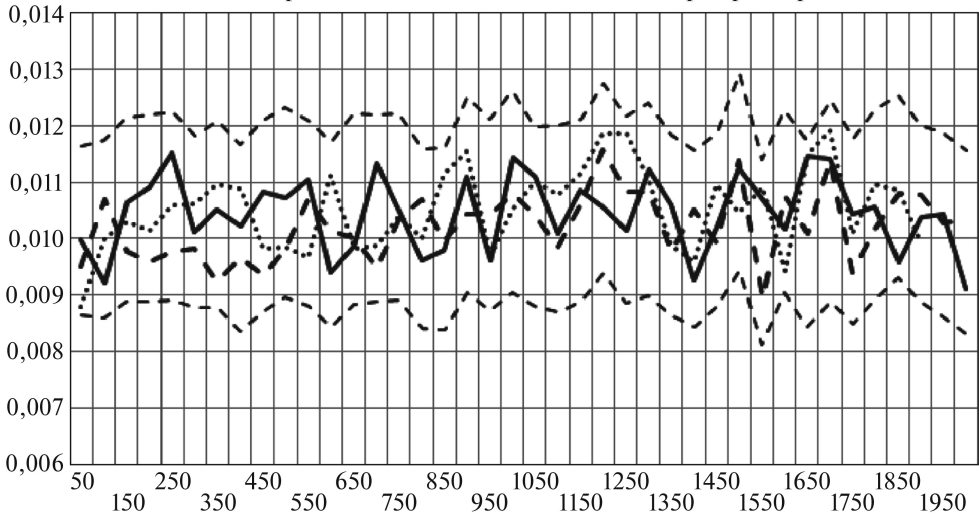


Рис. 4. Зависимость эффективности работы алгоритма от количества итераций метода муравьиных колоний для параметрического графа «Carrom table function».

ранжировать алгоритмы АСОСС, АСОСН, АСОСНИ, АСОССуЗ и АСОССуІ. Время поиска агентом решения в алгоритмах АСОССуЗ и АСОССуІ близкое, так как количество дополнительных итераций чаще всего меньше ограничения в 3 итерации.

Следует отметить, что циклический поиск фактически увеличивает количество итераций, так как работает по правилам метода муравьиных колоний. Если среди десяти агентов на итерации один не нашел нового решения и ему

**Таблица 5.** Оценка математического ожидания количества дополнительных итераций модификации АСОССуI

№ итер.	1900	1910	1920	1930	1940	1950
$k_2=0; k_3=0;$	8,379	8,712	9,582	10,487	23,004	25,153
$k_2=0; k_3=0; ignor;$	8,399	9,028	9,880	10,719	19,658	18,996
$k_2=0; k_3=1;$	5,035	5,248	5,735	6,536	13,408	12,861
$k_2=0; k_3=1; ignor;$	5,054	5,291	5,711	6,558	10,658	10,299
$k_2=1; k_3=0;$	7,049	7,780	8,990	11,493	44,962	42,247
$k_2=1; k_3=0; ignor;$	7,113	7,744	8,698	10,263	13,658	13,533
$k_2=1; k_3=1;$	4,864	5,312	5,990	7,383	19,585	19,828
$k_2=1; k_3=1; ignor;$	4,874	5,314	5,977	7,268	11,172	11,292

потребовалось десять дополнительных итераций, то фактически было выполнено двадцать итераций. В отличие от двадцати итераций оригинального алгоритма для данных итераций не происходит обновления весов на параметрическом графе и возможен контроль необходимого количества итераций.

Так как модификации АСОССу3 и АСОССуI показали хорошую сходимость к оптимальному решению и каждый агент в данном алгоритме находит новое решение (табл. 2), то исследуем возможность нахождения всех решений в параметрическом графе. Для исследования воспользуемся графом малой размерности (40 401 решение) с целевой функцией (3) «Carrom table function». Результаты применения модификации АСОССуI приведены на рис. 4. В отличие от АСОССуI модификацией АСОСNI за 2000 итераций рассмотрено только 35% решений. При исследовании модификации АСОССуI изменялись значения коэффициентов  $k_2$  и  $k_3$  в (2).

Длинные штрихпунктирные линии с точками на рис. 4 определяют модификацию, у которой  $k_2 = 0$  (при  $k_3 = 0$  — двойная точка, при  $k_3 = 1$  — одна точка). При  $k_2 = 0$  оптимальное решение с вероятностью 1 определяется только после 1000 итераций (более 60% рассмотренных решений). Более эффективными являются модификации, в которых  $k_2 = 1$  (при  $k_3 = 0$  — штриховая линия, при  $k_3 = 1$  — сплошная линия). При любом количестве итераций оптимальное решение находится после рассмотрения от 0,009% ( $40\,401 \cdot 0,009 = 363$ ) решений до 0,012% (484 решений) — доверительный интервал с доверительной вероятностью 0,99 (нижний график рис. 4). Для нахождения 0,99% оптимального решения требуется в 2 раза меньше рассмотренных решений, наборов значений параметров. Для модификации АСОСNI (линия из точек) требуется аналогичное количество рассмотренных решений для нахождения оптимального.

На верхнем графике рис. 4 отображено количество требуемых дополнительных итераций на одного агента. В среднем агенту требуется менее 5 дополнительных итераций при рассмотрении большинства возможных решений. В результате устанавливать ограничение, как в модификации АСОССу3, неэффективно. Минимальное количество дополнительных итераций, необхо-

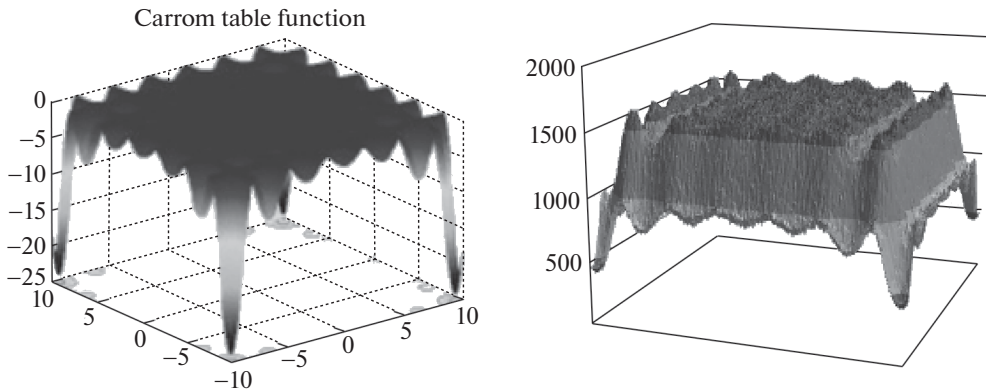


Рис. 5. График функции и оценки номера решения для «Carrom table function».

димых для поиска нового, еще не рассмотренного, набора значений параметров, гарантирует минимальные задержки по времени работы алгоритма.

Неэффективное поведение модификации происходит на последних 36-ти итерациях, на которых необходимо найти последние 900 оставшихся не рассмотренными решений (табл. 5). Четные строки таблицы с пометкой *ignor* — это результаты работы модификации АСОССуІ с условием игнорирования вершин, для которых рассмотрены все решения. Модификация не рассматривает вершины, у которых  $kol(t)_{ij,t} = MaxKol_{j,t}$  в вероятностной формуле (2).

Результаты, представленные в табл. 5, доказывают эффективность применения третьего слагаемого суммы в формуле (2). При  $k3 = 1$  количество необходимых дополнительных итераций существенно меньше, чем при  $k3 = 0$ . Меньшее количество дополнительных итераций соответствует меньшему времени поиска оставшихся не рассмотренными наборов значений параметров. Высокую эффективность показывает игнорирование вершин, для которых рассмотрены все возможные решения (строки с надписью *ignor*).

Оба тестовых параметрических графа имеют несколько оптимальных (равных по значению целевой функции) значений параметров. На рис. 5 приведены график функции «Carrom table function» и график оценки номера решения, на котором рассматривались конкретные значения параметров  $x_1$  и  $x_2$  при 3000 прогонах модификации АСОССуІ метода муравьиных колоний. При множестве прогонов алгоритм находит все 4 оптимума на самых ранних итерациях. График оценки номера итерации визуально повторяет график функции, т.е. оптимальные и рациональные решения, предложенные модификацией метода муравьиных колоний, статистически определяются на начальных операциях.

## 5. Заключение

В работе рассматривалась задача направленного перебора наборов значений параметров. Для решения задачи используется параметрический граф, в

котором набор параметров представлен в виде множества вершин, объединенных в группы — слои. В данном графе отсутствуют дуги, и необходимо определиться с порядком следования слоев, что может приводить к расхождению в эффективности предложенных модификаций. Но достоинством данного вида параметрического графа является простота его создания для пользователя. Для метода муравьиных колоний рассматривалась задача перебора всех значений параметров системы. Оптимизационные свойства метода позволяют рассмотреть оптимальный набор значений параметров как можно раньше. Предложены модификации, позволяющие упростить и контролировать поиск новых наборов значений параметров методом муравьиных колоний. Предполагается использование интерфейса для взаимодействия с пользователем с целью обеспечения возможности остановки работы метода при нахождении удовлетворяющего пользователя набора значений параметров системы. Если такой набор не найден, то метод муравьиных колоний рассматривает все комбинации дискретных значений параметров. Данный подход также позволит решать многоэкстремальные и многокритериальные задачи.

Рассмотренный метод муравьиных колоний является достаточно мощным инструментом решения поставленной задачи, благодаря вероятностной формуле выбора следующей вершины. Предложена простая модификация вероятностной формулы, позволяющая существенно улучшить показатели эффективности работы алгоритма. Данная модификация определяет вероятность выбора следующей вершины как аддитивную свертку трех компонентов: весов в вершине, количества посещений вершины агентами и количества оставшихся решений, содержащих данную вершину.

В работе предложены модификации, позволяющие выбирать различные алгоритмы поведения при получении агентом уже рассмотренного решения. Проверка найденного решения осуществляется с помощью Хэш-таблицы. Повторный циклический поиск показал хорошую сходимость к оптимальному решению и наилучшую работу при поиске последних наборов значений параметров на различных параметрических графах. На графе большой размерности повторный циклический поиск показывает результаты хуже, чем работа оригинального алгоритма с новой вероятностной формулой.

Рассмотрение всех возможных комбинаций значений параметров системы является специфической задачей, так как перебрать все варианты можно полным перебором и в результате найти оптимальное решение. Большинство алгоритмов перебора вариантов системы нацелены на поиск рационального решения путем сходимости к нему, и “плохие” решения подобные алгоритмы не рассматривают. Несмотря на вероятностную сходимость метода муравьиных колоний к одному решению, в работе доказана возможность алгоритмом рассмотреть все решения, даже все неоптимальные, при любом распределении вероятностной формулы. Для многомодальных функций можно отказаться от процедуры мультистарта, вносящей дополнительную стохастичность в работу алгоритма.

К недостаткам метода муравьиных колоний относят наличие большого количества свободных параметров [9]. Параметры, относящиеся к “классическому” методу муравьиных колоний (количество агентов на итерации, коэффициент испарения весов и коэффициент добавления), подробно рассмотрены в [21]. При исследовании асинхронного поведения метода муравьиных колоний при взаимодействии с асинхронным вычислительным кластером предполагается устанавливать количество агентов равным количеству потоков, осуществляющих вычисления на кластере. Отсутствие сходимости алгоритма к одному решению не требует установки граничных условий и параметров мультистарта. Но наличие коэффициентов в новой вероятностной формуле требует дополнительных исследований. Исследованы различные дискретные значения весовых и степенных коэффициентов в формуле (2). Оптимальными являются значения, равные 1. При этом следует отметить, что ситуация, когда весовые коэффициенты принимают вещественные значения и в сумме дают 1, требует дальнейших исследований. Также дальнейших исследований требует динамическое изменение значений коэффициентов, так как коэффициент  $k1$  эффективен на ранних итерациях для быстрого поиска оптимального набора значений параметров, а коэффициент  $k3$  — для поиска оставшихся решений на последних итерациях.

Дальнейшее развитие предполагается в следующих направлениях.

1. Предложенный альтернативный алгоритм поиска нового решения путем обхода дерева показал низкую эффективность и требует доработки.
2. Не рассматривалась ценность отдельных слоев-параметров параметрического графа. Данное исследование позволит выделить наиболее и наименее значимые параметры технической системы.
3. Вероятностная формула выбора агентом следующей вершины является очень мощным инструментом, и возможна ее модификация с учетом ценности отдельных слоев.
4. Необходимо дальнейшее исследование структуры параметрического графа, разделение слоя на подслои, перестановки слоев и способов формирования графа.
5. Применение предложенного метода для решения задач с векторным критерием. Исследования модификаций для быстрого рассмотрения на кластере всех решений из множества Парето.
6. При работе метода муравьиных колоний совместно с асинхронным вычислителем. Предполагается получение значений целевой функции асинхронно по отношению к работе метода муравьиных колоний. Для такой модификации предлагается каждого агента просчитывать в своем потоке, и в результате рассмотреть асинхронную работу метода муравьиных колоний. Добавление и испарение весов с параметрического графа может осуществляться в виде отдельных потоков.

## СПИСОК ЛИТЕРАТУРЫ

1. *Feurer M., Hutter F., Vanschoren J.* Hyperparameter Optimization // The Springer Series on Challenges in Machine Learning. Springer, Cham. 2019. [https://doi.org/10.1007/978-3-030-05318-5\\_1](https://doi.org/10.1007/978-3-030-05318-5_1)
2. *Koehrsen W.* A conceptual explanation of bayesian hyperparameter optimization for machine learning. 2018. (Открытый доступ 18.01.2023: <https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f>)
3. *Colomi A., Dorigo M., Maniezzo V.* Distributed Optimization by Ant Colonies // Proc. First Eur. Conf. on Artific. Life, Paris, France, Elsevier Publishing. 1992. P. 134–142.
4. *Dorigo M., Stützle T.* Ant Colony Optimization // MIT Press. 2004. P. 321.
5. *Socha K., Dorigo M.* Ant colony optimization for continuous domains // Eur. J. Oper. Res., 2008, V. 185. Issue 3. pp. 1155–1173. <https://doi.org/10.1016/j.ejor.2006.06.046>
6. *Mohamad M., Tokhi M., Omar O.M.* Continuous Ant Colony Optimization for Active Vibration Control of Flexible Beam Structures // IEEE International Conf. on Mechatronics (ICM). Apr., 2011. P. 803–808.
7. *Карпенко А.П., Чернобривченко К.А.* Эффективность оптимизации методом непрерывно взаимодействующей колонии муравьев (СИАС) // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. 2011. № 2. <https://doi.org/10.7463/0211.0165551>
8. *Карпенко А.П., Чернобривченко К.А.* Мультимемевая модификация гибридного муравьиного алгоритма непрерывной оптимизации НСИАС. // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. 2012. № 9. <https://doi.org/10.7463/0912.0470529>
9. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой // М.: Изд-во МГТУ им. Баумана. 2-е изд. 2017. 446 с.
10. *Саймон Д.* Алгоритмы эволюционной оптимизации : практическое руководство. М.: ДМК Пресс. 2020. 1002 с.
11. *Sudakov V.A., Titov Y.P.* Modified Method of Ant Colonies Application in Search for Rational Assignment of Employees to Tasks // Proceedings of 4th Computational Methods in Systems and Software 2020. Vol. 2, Vsetin: Springer Nature, 2020. P. 342–348. DOI 10.1007/978-3-030-63319-6\_30
12. *Хажулин Г.Ф., Титов Ю.П.* Система поддержки решений поставок запасных частей летательных аппаратов военного назначения // Изв. Самарского научн. центра РАН. 2014. Т. 16. № 1–5. С. 1619–1623.
13. *Синицын И.Н., Титов Ю.П.* Развитие стохастических алгоритмов муравьиной организации // Бионика – 60 лет. Итоги и перспективы. Сборник статей Первой Междунар. науч.-практ. конф. Под редакцией А.П. Карпенко. 17–19 декабря 2021 г., г. Москва. Под ред. 2022. С. 210–220. [https://doi.org/10.53677/9785919160496\\_210\\_220](https://doi.org/10.53677/9785919160496_210_220)
14. *Титов Ю.П.* Модификации метода муравьиных колоний для решения задач разработки авиационных маршрутов // АиТ. 2015. № 3. С. 108–124. *Titov Y.P.* Modifications of the ant colony method for aviation routing // Autom. Remote Control. 2015. V. 76. no 3. P. 458–471. <https://doi.org/10.1134/S0005117915030091>



15. *Судаков В.А., Батъковский А.М., Тумов Ю.П.* Алгоритмы ускорения работы модификации метода муравьиных колоний для поиска рационального назначения сотрудников на задачи с нечетким временем выполнения // *Современные информационные технологии и ИТ-образование*. 2020. Т. 16. № 2. С. 338–350. <https://doi.org/10.25559/SITITO.16.202002.338-350>
16. *Parpinelli R., Lopes H., Freitas A.* Data mining with an ant colony optimization algorithm // *IEEE Trans. Evol. Comput.* 2002. V. 6. No. 4 P. 321–332.
17. *Junior I.C.* Data mining with ant colony algorithms // *ICIC. LNCS*. 2013. V. 7996. P. 30–38.
18. *Martens D., De Backer M., Haesen R., Vanthienen J.* Classification with ant colony optimization // *IEEE Trans. Evol. Comput.* 2007. V. 11. No. 5. P. 651–665.
19. *Pasia J.M., Hartl R.F., Doerner K.F.* Solving a Bi-objective Flowshop Scheduling Problem by Pareto-Ant Colony Optimization // *ANTS 2006*. P. 294–305.
20. *Тумов Ю.П.* Опыт моделирования планирования поставок с применением модификаций метода муравьиных колоний в системах высокой доступности // *Системы высокой доступности*. 2018. Т. 14. № 1. С. 27–42.
21. *Синицын И.Н., Тумов Ю.П.* Оптимизация порядка следования гиперпараметров вычислительного кластера методом муравьиных колоний // *Системы высокой доступности*. 2022. Т. 18. № 3. С. 23–37. <https://doi.org/10.18127/j20729472-202203-02>
22. *Mishra Sudhanshu K.* Some New Test Functions for Global Optimization and Performance of Repulsive Particle Swarm Method // *University Library of Munich, Germany, MPRA Paper*. 2006. <https://doi.org/10.2139/ssrn.926132>
23. *Layeb Abdesslem.* New hard benchmark functions for global optimization. 2022. <https://doi.org/10.48550/arXiv.2202.04606>

*Статья представлена к публикации членом редколлегии О.П. Кузнецовым.*

Поступила в редакцию 23.01.2023

После доработки 21.03.2023

Принята к публикации 09.06.2023