ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2025, № 1, С. 73–89

СИСТЕМНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

УДК 519.856

МОДИФИКАЦИИ МЕТОДА МУРАВЬИНЫХ КОЛОНИЙ ДЛЯ ПЕРЕБОРА ВАРИАНТОВ РЕШЕНИЯ ДИСКРЕТНОЗНАЧНОЙ ПАРАМЕТРИЧЕСКОЙ ЗАДАЧИ

© 2025 г. Е. А. Давыдкина[a], *, В. А. Нестеров[a], **, В. А. Судаков[a, b], *, К. И. Сыпало[c], ****, Ю. П. Титов[a], *****

[a]МАИ (Национальный исследовательский университет), Москва, Россия
[b]ИПМ им. М.В. Келдыша РАН, Москва, Россия
[c]Центральный аэрогидродинамический институт им. проф. Н.Е. Жуковского, Жуковский, Россия
**e-mail: davydkinaelena@yandex.ru, **e-mail: nesterov46@inbox.ru, ***e-mail: sudakov@ws-dss.com, ****e-mail: ksypalo@tsagi.ru, *****e-mail: kalengul@mail.ru*

Поступила в редакцию 27.10.2023 г.
После доработки 13.10.2024 г.
Принята к публикации 13.01.2025 г.

Рассматривается задача поиска рациональных решений многоэкстремальной параметрической задачи методом муравьиных колоний. Рациональные решения – это решения, близкие по значению целевой функции к оптимальному, но не обязательно являющиеся таковыми. Для решения многоэкстремальной задачи предложена модификация метода муравьиных колоний, которая не сходится к одному решению, а продолжает его поиск. Отсутствие сходимости к одному решению преодолевает проблему стагнации предложенного в работе алгоритма метода муравьиных колоний. Найденные решения сохраняются в хэш-таблице, что позволяет избежать повторного вычисления значения целевой функции для уже вычисленного решения на вычислителе и осуществить поиск нового решения для каждого агента. Описана новая формула определения вероятности перехода муравья (агента) к новому параметру. Задачей данной формулы является решение проблемы стагнации алгоритма на ранних итерациях путем увеличения вероятности посещения агентом еще нерассмотренных компонентов решений. Алгоритм данной модификации метода муравьиных колоний позволяет решать дискретные параметрические задачи, определять рациональные значения параметров из дискретного множества. Изучается зависимость эффективности работы метода от параметров предложенной модификации метода муравьиных колоний. Исследование на тестовых задачах и задачах большой размерности показало зависимость от параметров аддитивной свертки и коэффициента испарения, отвечающего за уменьшение весов, которые получены на прошлых итерациях.

Ключевые слова: параметрическая задача, метод муравьиных колоний, целевая функция, гиперпараметры, хэш-таблица

DOI: 10.31857/S0002338825010067 **EDN:** AGZEUB

MODIFICATIONS OF THE ANT COLONIES METHOD FOR SOLVING A DISCRETE-VALUED PARAMETRIC PROBLEM

E. A. Davydkina[a], *, V. A. Nesterov[a], **, V. A. Sudakov[a, b], *, K. I. Sypalo[c], ****, Yu. P. Titov[a], *****

[a]MAI (national research university), Moscow, Russia
[b]Keldysh Institute of Applied Mathematics of RAS, Moscow, Russia
cCentral Central Aerohydrodynamic Institute, Zhukovsky, Russia
**e-mail: davydkinaelena@yandex.ru, **e-mail: nesterov46@inbox.ru, ***e-mail: sudakov@ws-dss.com, ****e-mail: ksypalo@tsagi.ru, *****e-mail: kalengul@mail.ru*

The problem of finding rational solutions to a multi-extremal parametric problem using the ant colony method is considered. Rational solutions are solutions that are close to the optimal one in terms of the value of the objective function, but are not necessarily such. To solve for a multi-extreme problem, a modification

of the ant colony method is proposed, which does not converge to a single solution, but continues to search for a solution. The ant colony method underlying the modification allows one to consider all rational solutions at the earliest iterations. The lack of convergence to a single solution solves the problem of stagnation of the proposed algorithm of the ant colony method. The solutions considered are stored in a hash table, which allows avoiding recalculation of the value of the objective function for a given solution on the computer and searching for a new solution for each agent. A new formula for determining the probability of the ant's (agent's) transition to a new parameter. The purpose of this formula is to solve the problem of algorithm stagnation in early iterations by increasing the probability of the agent visiting components of solutions that have not yet been considered. The algorithm of this modification of the ant colony method allows solving discrete parametric problems, determining rational values of parameters from discrete set. The paper investigates the dependence of the efficiency of the method on the parameters of the proposed modification of the ant colony method. The study on test problems and large-scale problems showed the dependence on the parameters of the additive convolution and the evaporation coefficient, which is responsible for reducing the weights obtained in previous iterations.

Keywords: parametric problem, ant colony method, objective function, hyperparameters, hash table

Введение. В настоящее время с развитием вычислительных кластеров и систем возможно перенести множество расчетных и оптимизационных задач с человека-эксперта на вычислительные машины. Программное обеспечение, такое как Ansys и CATIA и др., обеспечивает автоматизацию расчетов, которые ранее выполнялись вручную инженерами. Дальнейшее развитие систем позволяет решать больший спектр задач и проводить параллельные вычисления. Обычно оптимизационную задачу по определению оптимальных параметров системы выполняет инженер, используя итеративный процесс: задание параметров системы, расчеты на вычислительном устройстве, получение значений критериев от программы. Если критерии не удовлетворяют требованиям, то производится эмпирический анализ зависимостей и повторные расчеты. Данный циклический процесс происходит до тех пор, пока значения критериев не удовлетворят инженера.

Дальнейшее развитие предполагает автоматизацию процессов оптимизации различными методами. С использованием параллельных вычислений возможна автоматизация задач оптимизации системы с помощью вычислительных кластеров или суперкомпьютеров. В этом случае задача оптимизации перекладывается на программное обеспечение, которое перебирает варианты параметров системы методом полного перебора. Последовательность взаимодействия с такой системой представляет собой задание диапазонов значений для каждого параметра, формирование множества наборов значений параметров, отправку наборов на вычислительные кластеры для определения значений критериев и выдачу результатов разработчику. Оптимизация может проводиться с применением методов многокритериального анализа и систем поддержки принятия решений.

Оптимизация системы с использованием параллельных вычислений может быть автоматизирована при помощи вычислительных кластеров и суперкомпьютеров. Для этого задача оптимизации перекладывается на программное обеспечение, которое перебирает варианты параметров системы методом грубой силы. При этом порядок наборов значений параметров неважен, а оптимизация проводится после рассмотрения всех наборов. Подобные задачи нахождения рациональных значений параметров для решения расчетной задачи называются оптимизацией гиперпараметров [1, 2]. Одним из алгоритмов оптимизации гиперпараметров является Байесовский оптимизатор, который позволяет находить закономерности влияния значений отдельных параметров на критерии эффективности. IBM (international business machines) разработала систему BOA (bayesian optimization accelerator), использующую искусственный интеллект на базе Байесовской оптимизации для быстрого нахождения рациональных решений [3–5]. BOA работает как прослойка между вычислительной программой и хранилищем входных параметров. В случае неверной работы BOA результат будет таким же, как и при применении метода грубой силы. Для решения задачи переупорядочивания наборов гиперпараметров перед отправкой на вычислительную систему может быть использован метод муравьиных колоний.

В работе приводятся модификации метаэвристического метода оптимизации – метода муравьиных колоний, позволяющие решать задачу поиска рациональных значений параметров, т.е. осуществлять поиск решений, приемлемых для лица, принимающего решения (ЛПР). Ал-

горитмы оптимизации дают возможность рассмотреть не все наборы значений параметров, а путем небольших вычислений найти близкие к оптимальным значения параметров (если говорить про метаэвристические алгоритмы, например метод муравьиных колоний, то рациональных значений параметров). При наличии нескольких значений критериев многоэкстремальных и сложных овражных функций методы оптимизации могут находить не глобальное оптимальное решение, а локальный экстремум. Для решения данной проблемы применяют мультистарт – многократный повторный запуск алгоритма из различных начальных состояний [6].

Недостатком полного перебора является необходимость проанализировать все решения, в том числе дающие наихудшее значение целевой функции, физически не реализуемые, нелогичные. Модификации алгоритмов полного перебора стараются выделить точки, которые надо рассмотреть раньше других. Например, атака по словарю (метод подбора пароля с помощью часто используемых слов) проводит полный перебор всех паролей, но сначала изучает все комбинации паролей из словаря. Предлагается применить динамическое переупорядочивание приведенных решений в процессе их перебора.

Данный алгоритм требует взаимодействия с пользователем, инженером, решающим вычислительную задачу, который фактически выступает ЛПР. Если пользователь задаст область допустимых значений критерия (область решений, которая устраивает ЛПР), то возможна остановка алгоритма полного перебора в случае, если найдено решение, попадающее в данную область. Переупорядочивание решений предлагается осуществлять с помощью метода муравьиных колоний.

Метод муравьиных колоний ACO (ant colony optimization), предложенный M. Dorigo, решал задачу коммивояжера и позже был расширен на большой класс оптимизационных задач [7, 8]. Элитный, ранжированный и Min-Max алгоритмы описаны давно и применяются во множестве задач [8]. В работе описывается задача параметрической оптимизации: поиск оптимальных значений параметров с последующим продолжением поиска рациональных решений. Задача поиска оптимальных значений параметров методом муравьиных колоний рассматривалась в публикациях многих авторов: комбинаторная оптимизация [9, 10], решения задачи составления расписаний [11], составления заявки на поставки [12], задачи о назначении [13], классификации [14] и других параметрических задач [15, 16]. Отдельно следует отметить оптимизацию гиперпараметров для нейронных сетей [17], управления дискретными детерминированными системами [18, 19], непрерывную [6, 19–23] и многоэкстремальную модификации метода муравьиных колоний [24, 25]. Также широко применяются различные гибридные методы, в составе которых имеется метод муравьиных колоний или роевой вероятностный выбор решения [26].

В статье представлена задача, когда метод муравьиных колоний осуществляет поиск рациональных решений параметрической задачи, но не сходится к ним, а продолжает поиск новых решений. Данный подход основан на работе метода муравьиных колоний совместно с вычислительно сложными аналитическими и имитационными моделями, в которых вычисляются значения функций. Важно, чтобы каждое решение рассматривалось один раз. В результате требуется хранение уже вычисленных решений. Необходима модификация метода муравьиных колоний, позволяющая не сходиться к одному решению, как в оригинальном алгоритме, а продолжить поиск новых решений [15]. Для выборки решений с помощью метода муравьиных колоний пользователь (инженер) задает границы изменения значений и шаг изменения значения параметров системы. Возможно и непосредственное задание качественных значений параметров. На основе информации от пользователя создается структура данных, в которой содержатся параметры и их дискретные значения для каждого параметра системы [15, 16]. Метод муравьиных колоний определяет конкретные значения для каждого параметра (решение), которое отправляется на вычислительный кластер для определения значения критерия. Полученные значения используются методом муравьиных колоний для решения задачи оптимизации. Вычислительный кластер и его целевая функция для исследуемой системы являются черным ящиком (black box). Вычисление может быть ресурсоемко, требовать существенного времени. Поэтому для повышения производительности требуется хранить значения критерия, которые уже были вычислены для заданных значений параметров, в хэш-таблице для минимизации запросов к вычислительному кластеру. Перед отправкой на кластер решение ищется

в хэш-таблице, из-за этого не требуется сходимость всех муравьев к одному решению, так как возможен поиск оптимума (путем сравнения значений критерия) в конечном множестве рассмотренных методом муравьиных колоний решений [15]. В статье описывается подход, позволяющий для каждого агента-муравья находить новое, еще не рассмотренное на кластере решение [27].

1. Постановка задачи. Пусть имеется дискретное множество параметров $P=\{p_1, p_2 \dots p_i, \dots p_n\}$. У каждого i-го параметра существует множество допустимых значений $V_i=\{v_{1,i}, v_{2,i}, \dots v_{j,i}, \dots v_{m_i,i},\}$, $i=\overline{1,n}$. Количество допустимых значений определяется величиной $|V_i|=m_i>0$, зависящей от номера параметра. Так для различных параметров может быть найдено различное количество возможных значений. В результате работы алгоритма оптимизации определяется вектор значений параметров, называемый далее решением: $X_k=(x_{1,k}, x_{2,k} \dots x_{i,k}, \dots x_{n,k})$, $|X_k|=n$, где $\forall i \exists j\colon (x_{i,k}=v_{j,i}) \wedge (v_{j,i} \in V_i)$. Здесь k – номер агента (муравья), который получает решение независимо от других агентов. Всего на итерации метода муравьиных колоний решение задают одновременно K агентов. Таким образом возможно параллельное получение каждого решения X_k. Найденный вектор значений параметров отправляется на вычислитель, который возвращает значение критерия, целевой функции $f(X_k)$. Решения рассматриваются в дискретном пространстве значений параметров, и требований непрерывности и дифференцируемости на $f(X_k)$ не накладывается.

Требуется найти множество решений $Y=\{X_1^*, X_2^*, \dots X_z^*\}$, таких, что $f(X_i^*) \in \psi, \forall i,$ где ψ – область удовлетворительных значений целевой функции. Число решений z не задается ЛПР, достаточно одного решения ($z \geq 1$). Их может быть больше одного в случае, если решение ищут несколько агентов $k>1$. Агент, нашедший удовлетворительное решение, останавливает поиск, т.е. $1 \leq z \leq K$. Действительно для задачи минимизации достаточно указать только правую границу B для приемлемой области $f(X_i^*) \leq B$, а $\psi=(-\infty; B]$. Таким образом определяется область приемлемых для ЛПР значений критериев. Данная задача формально относится к классу задач удовлетворения ограничений (constraint satisfaction problems (CSP) [28].

2. Метод решения. В методе муравьиных колоний каждое значение параметра определяется множеством: P, V, τ, η, где P – множество параметров; V – множество допустимых значений; τ – вес (количество феромона) для значения параметра; η – количество решений, в которых было использовано данное значение параметра. В оригинальном методе муравьиных колоний применялась формула вероятностного перехода (2.1) из вершины i в вершину j, в которой применялась информация о длине дуги μ и феромоне τ [7, 8]. В данной задаче вершина графа – это допустимое значение параметра $v_{j,i}$, а дуги связывают все значения параметров с соседними номерами i и $i+1$ для $i=\overline{1,n-1}$. Параметры алгоритма α, β описывают соотношение длины дуги и количества феромона. Так как оригинальный алгоритм должен вычислять путь коммивояжера в графе, то посещенные вершины недоступны для повторного посещения, множество $J_{i,k}$ – множество доступных вершин для посещения из вершины i для агента k. Ниже $P_{i,j,k}(t)$ определяет вероятность перехода из вершины i в вершину j для агента k на итерации t:

$$P_{i,j,k}(t)=\frac{\tau_{i,j}^{\alpha}(t)\mu_{i,j}^{\beta}(t)}{\sum\limits_{j \in J_{i,k}} \tau_{i,l}^{\alpha}(t)\mu_{i,l}^{\beta}(t)} \quad j \in J_{i,k}, \quad P_{i,j,k}(t)=0, j \notin J_{i,k}. \tag{2.1}$$

Для решения параметрических или других задач, в которых в графе отсутствует априорная информация о эффективности перемещения агента в вершину, формула (2.1) изменяется и в ней остается только один нормализованный сомножитель:

$$P_{i,j,k}(t)=\frac{\tau_{i,j}^{\alpha}(t)}{\sum\limits_{j \in J_{i,k}} \tau_{i,l}^{\alpha}(t)} \quad j \in J_{i,k},$$

$$P_{i,j,k}(t)=0, j \notin J_{i,k}.$$

В таком случае, если в формуле остаются только веса вершин, алгоритм начинает стагнировать, сходиться к первому хорошему решению [6, 29]. Это происходит из-за того, что на начальных итерациях нет никакой информации о графе и все вершины имеют одинаковую вероятность выбора. После нескольких первых итераций вершины рассмотренных решений будут иметь большую вероятность выбора, чем другие вершины.

Предложим новую формулу для расчета вероятности выбора значения j для параметра i:

$$P_{i,j,k}(t) = \frac{z_{i,j}}{\sum\limits_{l=1}^{m_i}[z_{i,l}]}, \quad j \notin \overline{1,m_i}, \tag{2.2}$$

$$z_{i,j} = \lambda_1 \tau_{norm,i,j}^{\alpha}(t) + \lambda_2 \left(\frac{1}{\eta_{i,j}(t)}\right)^{\beta} + \lambda_3 \left(\frac{\eta_{i,j}(t)}{\eta_{\max,i}}\right)^{\gamma}, \quad P_{i,j}(t) = 0, j \notin \overline{1,m_i},$$

где i – номер параметра $p_i \in P$; j – номер значения параметра $v_{j,i} \in V_i$; t – номер итерации; $\lambda_1,\lambda_2,\lambda_3 \in \mathbb{R}$ – коэффициенты аддитивной свертки, $\lambda_1,\lambda_2,\lambda_3 \in [0,\infty)$, $\alpha,\beta,\gamma \in \mathbb{R}$ – степени слагаемых аддитивной свертки; $\tau_{norm,i,j}(t) = \tau_{i,j}(t)/(\tau_{i,1}(t) + \tau_{i,2}(t) + \ldots + \tau_{i,m_i}(t))$ – нормированный вес для значения с номером j параметра с номером i для итерации t; $h_{i,j}(t)$ – количество применений в рассмотренных решениях для значения с номером j параметра с номером i для итерации t; $\eta_{\max,i}$ – максимально возможное количество применений значений для параметра с номером i.

Недостатки линейной свертки, взаимокомпенсация слагаемых, для вероятностной формулы являются достоинством [29]. По результатам взаимокомпенсации на разных итерациях алгоритма разные слагаемые в большей степени влияют на вероятность выбора значения параметра, что позволяет оставаться алгоритму эффективным на любых итерациях. Первое слагаемое $\tau_{norm,i,j}^{\alpha}(t)$ зависит только от нормированного количества весов у параметра с номером i, его значения с номером j. Второе слагаемое $(1/\eta_{i,j}(t))^{\beta}$ зависит от количества посещений значения параметра агентом, т.е. сколько раз значение параметра рассматривалось в решениях. Данное слагаемое позволяет увеличить вероятность выбора значений параметра с наименьшим количество раз посещения. Добавление данного параметра дает возможность избежать стагнации алгоритма. После нескольких итераций алгоритма влияние данного параметра уменьшается из-за возросшего значения $\eta_{i,j}(t)$. Третье слагаемое, наоборот, увеличивается, когда количество решений с определенным значением параметра $\eta_{i,j}(t)$ приближается к максимальному количеству $h_{\max,i}$. Значение $h_{\max,i}$ для дискретной задачи можно точно вычислить по формуле:

$$\eta_{\max,i} = \prod_{s=1}^{n} m_s \,/\, m_i .$$

Данное слагаемое помогает найти последние решения и практически не влияет на работу алгоритма на ранних итерациях.

На каждой итерации метода муравьиных колоний поколение агентов размерностью K находит решения $X_k, k \in K$, которые отправляются на вычислитель для получения значения критерия $f(X_k)$. После получения решений для всех агентов (X_k) из одного поколения определяется новое состояние:

1. Увеличивается номер итерации: $t' = t + 1$.

2. Определяются добавочные веса $\Delta\tau_{i,j,k}(t)$, добавляемые для каждого значения с номером j параметра с номером i агентом k для итерации t (2.3); $Q \in \mathbb{R}$ – параметр метода муравьиных колоний, определяющий количество весов, добавляемых на единицу значения целевой функции:

$$\Delta\tau_{i,j,k}(t) = Q/f(X_k) \text{ для } x_{i,k} = v_{j,i}, \text{ если } f(X_k) \to \min, \tag{2.3}$$

$$\Delta\tau_{i,j,k}(t) = Qf(X_k) \text{ для } x_{i,k} = v_{j,i}, \text{если } f(X_k) \to \max.$$

3. Задается количество весов для новой итерации:

$$\tau_{i,j}(t') = \rho\tau_{i,j}(t) + (1-\rho)\sum_{k=1}^{K}\Delta\tau_{i,j,k}(t), \tag{2.4}$$

где $\rho \in [0, 1]$ – коэффициент испарения – параметр, определяющий скорость "устаревания" информации о предыдущем состоянии.

В (2.4) добавлен коэффициент $(1 - \rho)$ для добавленного значения весов. Коэффициент добавлен для аналогичности формуле экспоненциального скользящего среднего. В формуле "прогнозным" значением является предыдущее значение, а "установившимся" – количество добавляемого феромона.

Последующие агенты будут определять решения для итерации t'. Алгоритм продолжает итеративно находить решения, пока не будет остановлен. Критерием остановки работы алгоритма служит достижение заданного удовлетворительного значения целевой функции $f(X_k) \in \psi$.

В оригинальном методе муравьиных колоний в качестве целевой функции используется информация о длине пройденного пути агентом. Данная величина, как и количество феромона, является положительной и в (2.1) вычисление вероятности не приводит к проблемам. Но в задачах с неизвестными значениями целевой функции $f(X_k)$ возможно получение для определенных решений X_k отрицательных значений целевой функции, которая может привести к отрицательным значениям феромона. Для решения проблемы можно применить сдвиг значений целевой функции, но для данного подхода требуется заранее знать $\min_{X_k} f(X_k)$. В работе [30] предложен подход, который позволяет осуществлять сдвиг целевой функции итерационно при получении отрицательных значений. Можно пересчитать феромон во всех вершинах, что хоть и требует дополнительной памяти для хранения количества агентов, посетивших вершину на каждой итерации, но позволяет не заботиться о значениях целевой функции:

$$\tau_{i,j}(t') = \tau'_{i,j}(t) + \sum_{s=1}^{t}\rho(1-\rho)^s\, Q^s fv'(t) k_{i,j,t-s}, \tag{2.5}$$

где $\tau'_{i,j}(t)$ – значение феромона в вершине для значения с номером j параметра с номером i для итерации t по (2.4); $fv'(t+1)$ – сдвиг целевой функции на итерации $t+1$, $fv'(t+1) = \max_k |f(X_k) + fv'(t)|$, если $f(X_k) + fv'(t) < 0; fv'(t+1) = fv'(t)$, если $f(X_k) + fv'(t) \geq 0$; $fv'(t)$ – величина сдвига целевой функции на предыдущих итерациях, начальное значение $fv'(t) = 0$; $k_{i,j,t-s}$ – количество муравьев-агентов в вершине для значения с номером j, параметра с номером i на итерации $t-s$.

Хранение вычисленных значений целевой функции $f(X_k)$ для набора значений параметров X_k осуществляется в хэш-таблице. Хэш-таблица – это структура данных, которая применяет хэш-функцию для быстрого поиска, вставки и удаления элементов. Хэш-функция принимает значение параметров X_k и преобразует их в уникальное значение фиксированной длины – хэш, в работе используется алгоритм SHA-256. Если значение хэш отсутствует в таблице, то в нее заносятся значения X_k и значение целевой функции $f(X_k)$, иначе возникает коллизия.

Хэш-таблица используется при вычислении значений целевой функции, для разгрузки вычислительного кластера. После нахождения решения X_k оно проверяется на присутствие в хэш-таблице. Если решение найдено, то назовем агента "нулевым" и для него возможно выполнение различных действий: значение критерия $f(X_k)$ определяется из хэш-таблицы (стандартный алгоритм с промежуточным хранением решений); агент не будет добавлять веса для значений параметров $\Delta\tau_{i,j,k}(t) = 0$; найти новое решение X_k. Среди алгоритмов поиска нового решения рассматривался дальнейший циклический поиск решения, которого нет в хэш-таблице, модифицированным методом муравьиных колоний. По результатам исследований наиболее эффективным является модификация метода муравьиных колоний с циклическим

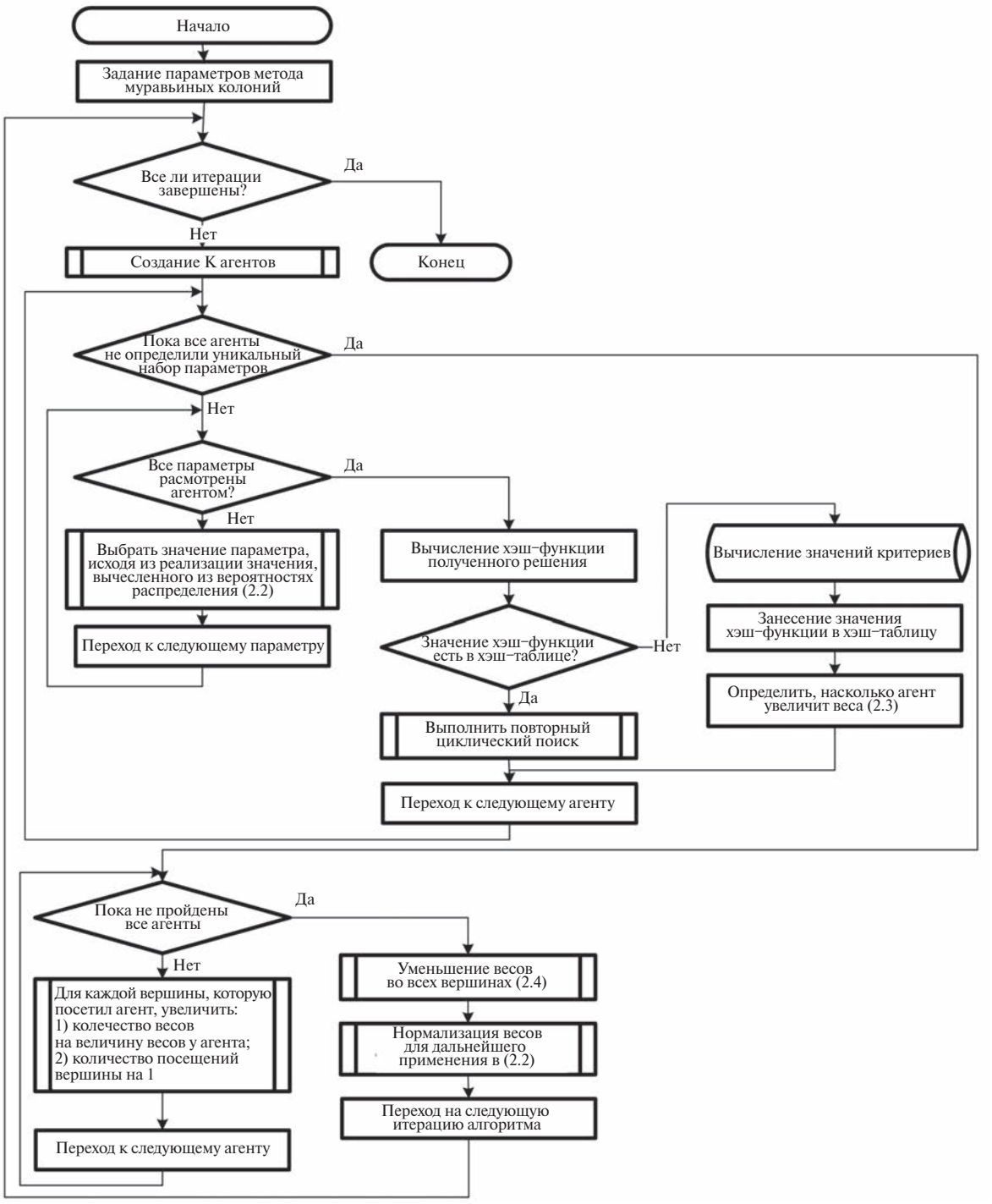


Рис 1. Алгоритм модифицированного метода муравьиных колоний (ACOCCyI).

поиском нового решения для нулевого агента (ACO cluster cycle infinity – ACOCCyI) [31]. Отличие такого подхода от обычной работы метода муравьиных колоний в том, что поиск нового решения происходит без изменения состояния. При условии, что все агенты найдут новое решение, можно вычислить необходимое количество итераций алгоритма и определить критерий остановки работы алгоритма – требуемое количество итераций. Если полный поиск всех решений не требуется, то можно остановиться после выполнения определенного процента итераций, например 5–10%. Алгоритм работы модифицированного метода муравьиных колоний приведен на рис. 1.

При решении поставленной задачи критерием остановки алгоритма является нахождение решения, удовлетворяющего условию $f(X_i^*) \in \psi$. Дополнительным техническим критерием остановки алгоритма выступает превышение допустимого времени работы алгоритма. В этом случае нельзя говорить о том, что задача строго решена в поставленной постановке, но тем не менее найденное алгоритмом лучшее решение может быть полезно для ЛПР, если оно близко к области удовлетворительных значений целевой функции.

3. Исследование эффективности метода. Исходными задачами исследования являлись различные параметрические задачи и тесты. Рассматривались функция Розенброка, синусоидальная функция Швефеля, функции Шаффера, Растригина и др. [32–34]. Результаты тестирования будут показаны на функции "Carrom table function":

$$f(x) = -\left(\cos x_1 \cos x_2 e^{\left|1-\frac{\sqrt{x_1^2+x_2^2}}{\pi}\right|}\right)^2 \Big/ 30; \quad x_1, x_2 \in [-10,10]. \tag{3.1}$$

В данном случае рассматривается задача минимизации. Так как представленные тесты – непрерывные функции, то для решения дискретной параметрической задачи необходимо дискретизировать параметры x_1 и x_2. Значения x_1 и x_2 определялись на интервалах $[-10,10]$ с точностью 10^{-3}. Таким образом, для каждого параметра было определено 20 001 значение от 10 до -10 с шагом 0.001. При решении задачи минимизации целевой функции "Carrom table function" (3.1) имеет четыре оптимальные точки ($(x_1; x_2)$: (9.646; 9.646), (9.646; –9.646), (–9.646; 9.646), (–9.646; –9.646)) с равными значениями целевой функции $f(x) = -24.15681551650653$ (рис. 2). Рассматривались также аналогичные многоэкстремальные функции с большим количеством оптимальных решений.

В данном примере область удовлетворительных значений целевой функции $\psi = (-\infty; -24.15681]$. Другими словами, правая граница интервала – это значение целевой функции, которое немного больше, чем оптимальное значение целевой функции. Таким образом, считается, что мы допускаем нахождение не строгого оптимального решения, а решения, которое может быть немного хуже оптимального. Для работы с отрицательными значениями целевой функции использовался метод динамического сдвига (2.5) [30]. Тестовый пример с заведомо известным оптимумом позволил лучше оценить эффективность предложенного метода.

Для изучения эффективности работы метода создано программное обеспечение на языке программирования Python [35]. Так как метод муравьиных колоний осуществляет вероятностный выбор значения параметра, то необходимо оценить статистические результаты: оценки вероятности и математического ожидания с вычислением доверительных интервалов. В рамках исследования вычисляются следующие оценки и их доверительные интервалы:

- оценка вероятности найти все попадающее в область ψ значение целевой функции за ограниченное число итераций;атематическое ожидание порядкового номера решения, на котором было определено попадающее в область ψ значение целевой функции;
- математическое ожидание количества дополнительных итераций метода муравьиных колоний;
- математическое ожидание времени поиска агентом одного нового решения.

Результаты исследования модификаций метода муравьиных колоний и оценки других характеристик работы алгоритма имеются в репозитории [35]. https://github.com/kalengul/ACO_Cluster

На рис. 2 приведены графики функций "Carrom table function" (сверху), "Мульти-функции" (посередине) и функции "Шеффеля" (снизу) и оценка (по результатам проведения 3000 прогонов) математического ожидания порядкового номера решения, на котором было рассмотрены данные значения параметров x_1 и x_2. По результатам можно сделать вывод, что предложенные модификации метода муравьиных колоний позволяют осуществить процедуру полного перебора всех комбинаций значений параметров за прогнозируемое число итераций, так как на правых графиках (рис. 2) представлены все точки без пропуска значений. Такой подход гарантирует нахождение оптимального значения даже сложной многоэкстремальной функции без применения процедуры мультистарта. При этом оценка математического ожидания номера итерации повторяет очертания графика функции. В результате перебора зна-

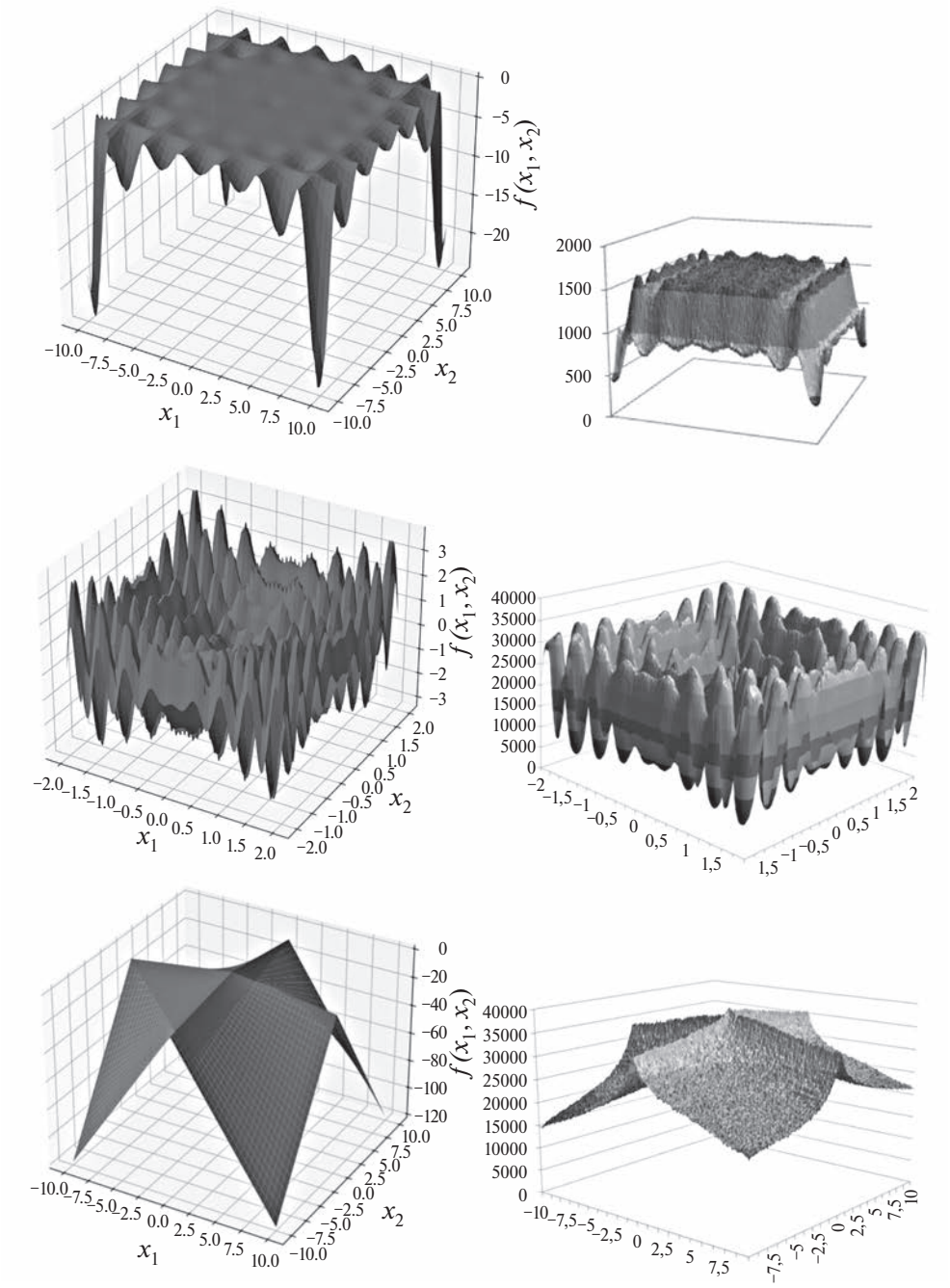


Рис 2. График функции “Carrom table function” (сверху), “Мультифункции” (посередине) и функции “Шеффеля” (снизу).

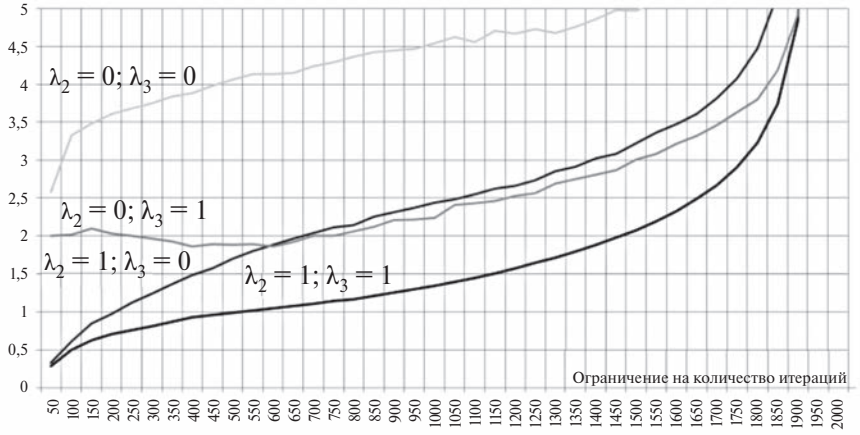
чений параметров статистически оптимальные наборы значений параметров определяются на ранних итерациях. После нахождения первого оптимального решения на ранних итерациях алгоритм не прекращает свою работу и позволяет определить все оптимальные решения, например для “Carrom table function” алгоритм рассматривает все четыре решения, попадающие в область ψ, менее чем за 500 итераций.

Одной из проблем применения метода муравьиных колоний является большое количество и сложность определения параметров алгоритма [6, 8]. Предложенная вероятностная формула (2.2) добавляет к стандартной формуле еще четыре параметра. Поэтому исследуем поведение алгоритма при различных значениях параметров для выработки рекомендаций.

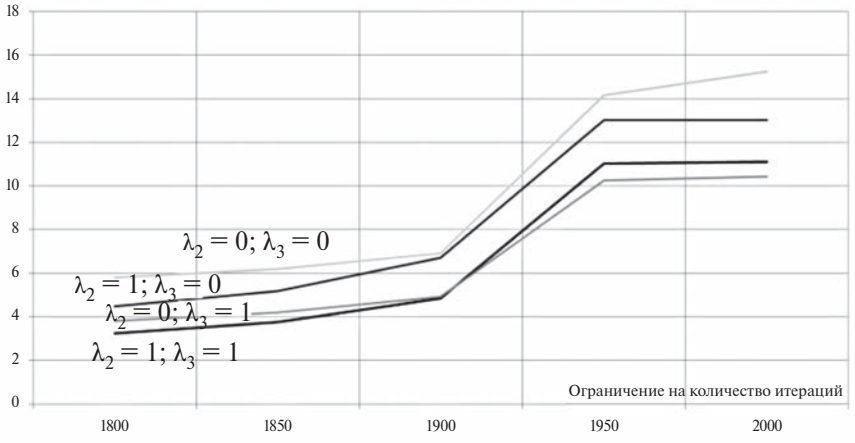
Изучим эффективность применения всех слагаемых вероятностной формулы (рис. 3) на “Carrom table function” (3.1) (далее – задача малой размерности) с дискретизацией до 10^{-1} (201

значение каждого из параметров x_1 и x_2) с 25 агентами на итерации. Так как каждый агент в модификации ACOCCyI находит новое решение, то за 1936 итераций будут рассмотрены все 40 401 решение [31]. Задача большой размерности представлена тестовой задачей, состоящей из 13 параметров и более 10^7 решений и некоторой авторской одноэкстремальной целевой функцией (подробнее см. [35]). При исследовании зависимости и варьировании параметров метода муравьиных колоний большинство из них оставались неизменными и модифицировался только один параметр метода. Поэтому определим “начальные, стандартные” значения параметров метода муравьиных колоний:

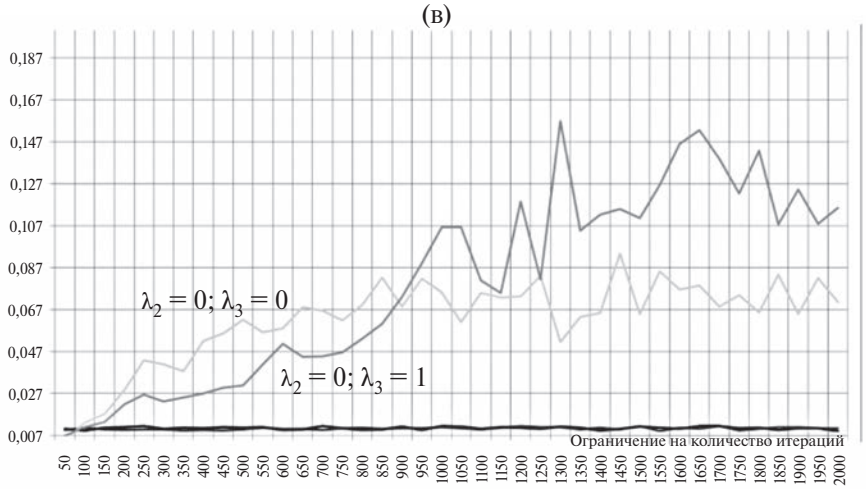
(а)



(б)



(в)



(г)

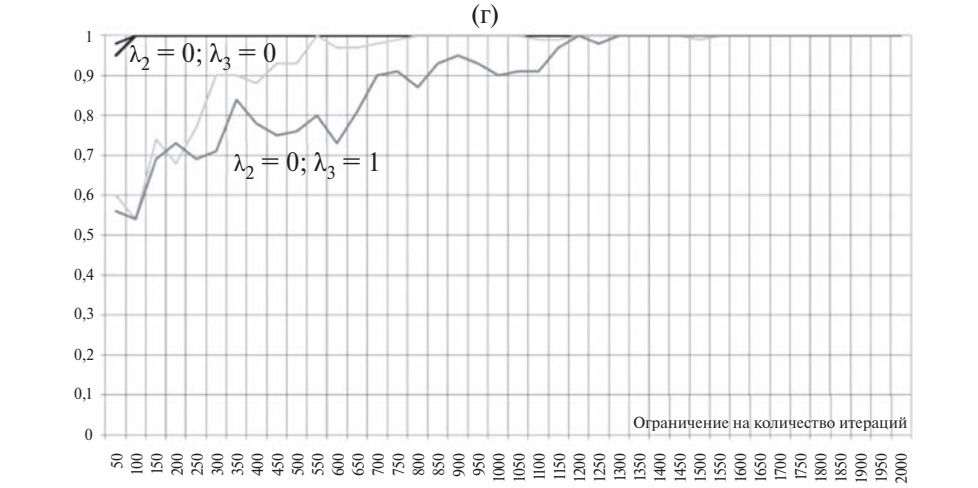


Рис 3. Оценки статистических параметров при изменении параметров $\lambda_1, \lambda_2, \lambda_3$ и ограничения на максимальное количество итераций: а, б – оценка математического ожидания количества дополнительных итераций на одного агента, в – оценка математического ожидания номера решения, на которой были найдены оптимальные значения параметров, г – оценка вероятности найти оптимальное решение.

для формулы вычисления вероятности выбора вершины (2.2):

$$\lambda_1 = 1,\ \lambda_2 = 1,\ \lambda_3 = 1, \alpha = 1, \beta = 1, \gamma = 1;$$

количество агентов на итерации $N = 25$;
коэффициент испарения $\rho = 0.9$;
коэффициент $Q = 5$.

Проведем исследование важности слагаемых в формуле вероятностного выбора значения параметра (2.2) на задаче малой размерности. Первое слагаемое обязательно присутствует в вероятностной формуле, так как это слагаемое является основным для работы метода муравьиных колоний: $\lambda_3 = 1$. Из графиков (рис. 3, светло-серые и темно-серые линии) видно, что без второго слагаемого в формуле (2.2) метод муравьиных колоний стагнирует и плохо находит решения (попадающие в область ψ значения целевой функции). Оценка вероятности определить решение равна 1 только после рассмотрения 72% решений (1400 итераций из 1936). А третье слагаемое в формуле (2.2) позволяет на последних итерациях быстрее найти последние неэффективные решения, оценка математического ожидания количества дополнительных итераций на одного агента составляет около 11 итераций при $\lambda_3 = 1$ и 13–15 итераций при $\lambda_3 = 0$. Также следует отметить, что резкое возрастание числа дополнительных итераций одного агента происходит только при поиске последней 1000 решений.

Из-за нормировки количества весов для значений параметра $\tau_{norm,i,j}(t)$ (2.2) параметр метода муравьиных колоний – Q (2.3) не влияет на эффективность работы алгоритма. Подставив в формулу $\tau_{norm,i,j}(t')$ значение $\tau_{i,j}(t')$ из (2.4), получим:

$$\tau_{norm,i,j}(t') = \frac{\rho\tau_{i,j}(t) + \sum_{k=1}^{K}(1-\rho)\Delta\tau_{i,j,k}(t)}{\sum_{l=1}^{m_i}\rho\tau_{i,l}(t) + \sum_{k=1}^{K}(1-\rho)\Delta\tau_{i,j,k}(t)}. \tag{3.2}$$

Выразив $\Delta\tau_{i,j,k}(t)$ из (2.3) и подставив в (3.2), найдем:

$$\tau_{norm,i,j}(t') = \frac{\rho\tau_{i,j}(t) + \sum_{k=1}^{K}(1-\rho)\dfrac{Q}{f(X_k)}}{\sum_{l=1}^{m_i}\rho\tau_{i,l}(t) + \sum_{k=1}^{K}(1-\rho)\dfrac{Q}{f(X_k)}},$$

если $f(X_k) \to \min$.

Параметр Q является постоянным для метода муравьиных колоний, не зависит от номера агента и может быть вынесен из под знака суммы:

$$\tau_{norm,i,j}(t') = \frac{\rho\tau_{i,j}(t) + Q(1-\rho)\sum_{k=1}^{K}\frac{1}{f(X_k)}}{\sum_{l=1}^{m_i}\rho\tau_{i,l}(t) + Q(1-\rho)\sum_{k=1}^{K}\frac{1}{f(X_k)}}, \text{ если } f(X_k) \to \min. \quad (3.3)$$

Подставив в (3.3) вместо $\tau_{i,j}(t)$ значение, полученное на предыдущей итерации t' из (2.4), получим:

$$\tau_{norm,i,j}(t') = \frac{\rho\left(\rho\tau_{i,j}(t') + Q(1-\rho)\sum_{k=1}^{K}\frac{1}{f(X_k)'}\right) + Q(1-\rho)\sum_{k=1}^{K}\frac{1}{f(X_k)}}{\sum_{l=1}^{m_i}\rho\left(\rho\tau_{i,l}(t') + Q(1-\rho)\sum_{k=1}^{K}\frac{1}{f(X_k)'}\right) + Q(1-\rho)\sum_{k=1}^{K}\frac{1}{f(X_k)}}, \quad (3.4)$$

где $f(X_k)'$ определяет значение целевой функции для агентов на предыдущей итерации. Единственное значение $\tau_{i,j}(t)$, которое не может быть так декомпозировано, – начальное количество весов, которое для всех значений параметров одинаковое. Поделив все значения в формуле (3.4) на Q, запишем:

$$\tau_{norm,i,j}(t') = \left(\rho\left(\frac{\rho\tau_{i,j}(t')}{Q} + (1-\rho)\sum_{k=1}^{K}\frac{1}{f(X_k)'}\right) + (1-\rho)\sum_{k=1}^{K}\frac{1}{f(X_k)}\right)\Big/Q. \quad (3.5)$$

Формулу (3.5) можно сократить на значение Q (постоянен для любого $l \in m_i$), оставив только постоянную величину $\rho\ \tau_{i,j}(t')/Q$, при условии, что $\tau_{i,j}(t')$ – начальный вес у каждого значения каждого параметра.

На рис. 4 показаны результаты исследования влияния различных значений параметра Q в зависимости от ограничения на количество проведенных итераций. Следует отметить статистическую неразличимость оценок, так как все они находятся в пределах доверительного интервала.

Исследования коэффициента испарения ρ (2.4) показали, что для задачи малой размерности только малое значение коэффициента $\rho \le 0.01$ существенно влияет на статистические характеристики. При изучении задачи большой размерности подтвердилось предположение, что большие значения параметра испарения улучшают работу алгоритма. Следует отметить, что при небольшом количестве итераций (первые столбцы нижнего графика рис. 5) значение $\rho = 0.95$ наиболее эффективно. Рекомендуется выбирать $0.9 \le \rho < 1$.

При исследовании количества агентов K на итерации (рис. 6), в случае фиксированного количества итераций метода муравьиных колоний, можно сделать вывод, что чем больше агентов, тем лучше, так как агенты рассмотрят большее число решений, а следовательно, и с большей вероятностью обнаружат решение, попадающее в область ψ значения целевой функции. На рис. 6 показаны результаты исследований при ограничении на количество рассмотренных решений (количество итераций до остановки алгоритма зависит от числа агентов на итерации), что позволяет определить эффективное количество агентов на итерации.

По результатам исследований выявлено, что при достаточно большом значении количества агентов на итерации $K > 35$ дальнейшее увеличение статистически не влияет на эффективность работы алгоритма, оценку вероятности найти решение, попадающее в область ψ значения целевой функции (максимизируемый критерий эффективности, верхний график рис. 6) и количества дополнительных итераций (минимизируемый критерий эффективности, нижний график рис.6). При исследовании “Carrom table function” статистически различима эффективность работы алгоритма только при пяти агентах на итерации.

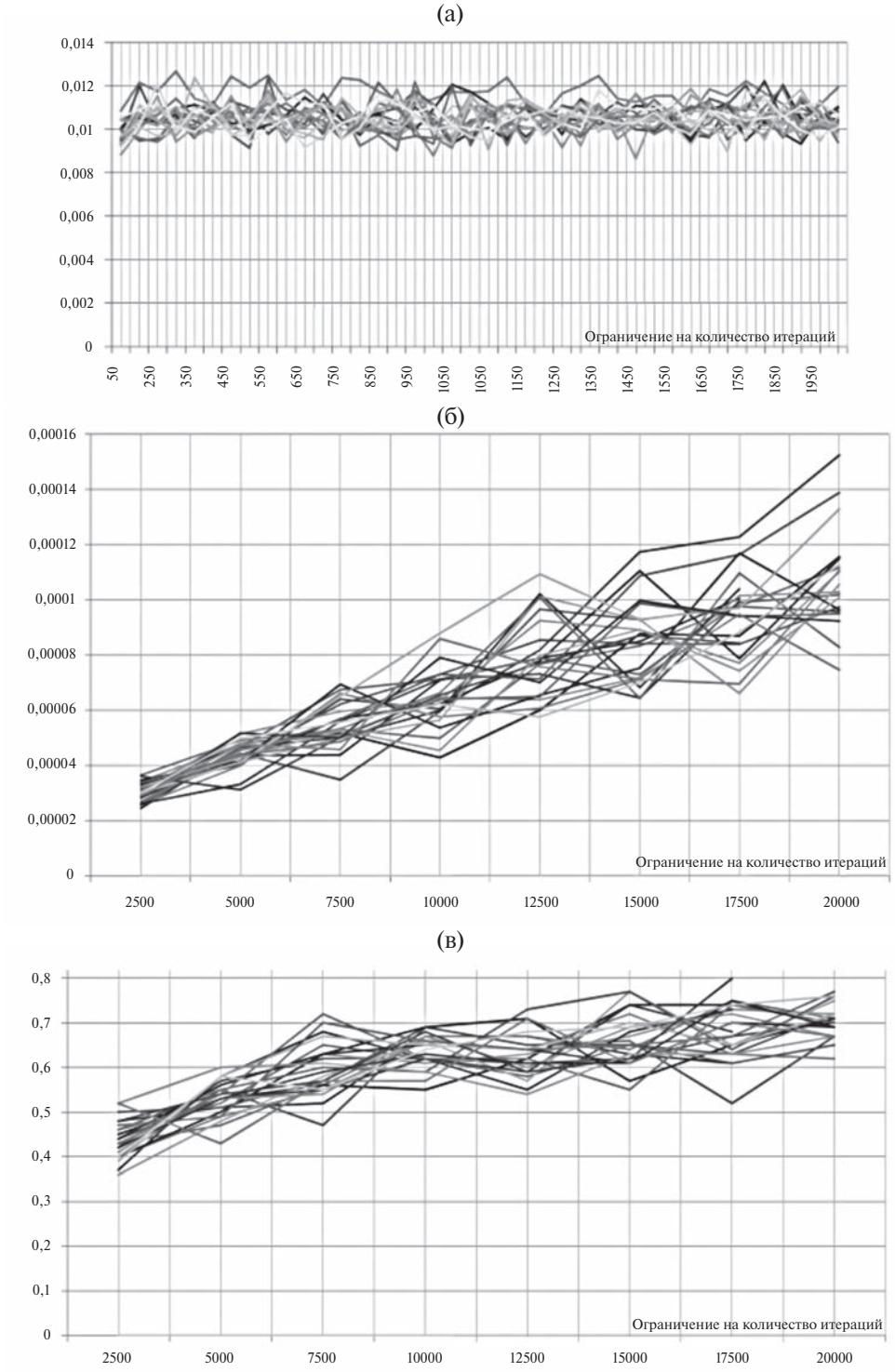


Рис. 4. Оценки статистических критериев эффективности работы алгоритма при изменении значений параметра Q и ограничения на количества итераций: а – оценка математического ожидания номера решения, на которой были найдены оптимальные значения параметров для графа малой размерности (бенчмарк); б – оценка вероятности найти оптимальное решение для графа большой размерности; в – оценка математического ожидания номера решения, на которой были найдены оптимальные значения параметров для графа большой размерности.

Наиболее сложными для исследования являются коэффициенты и значения параметров степеней в вероятностной формуле (2.2). На рис. 7 приведены результаты исследования взаимовлияния коэффициентов $\lambda_1, \lambda_2, \lambda_3$ и степеней α, β, γ. Больше результатов исследований можно найти в работе [35].

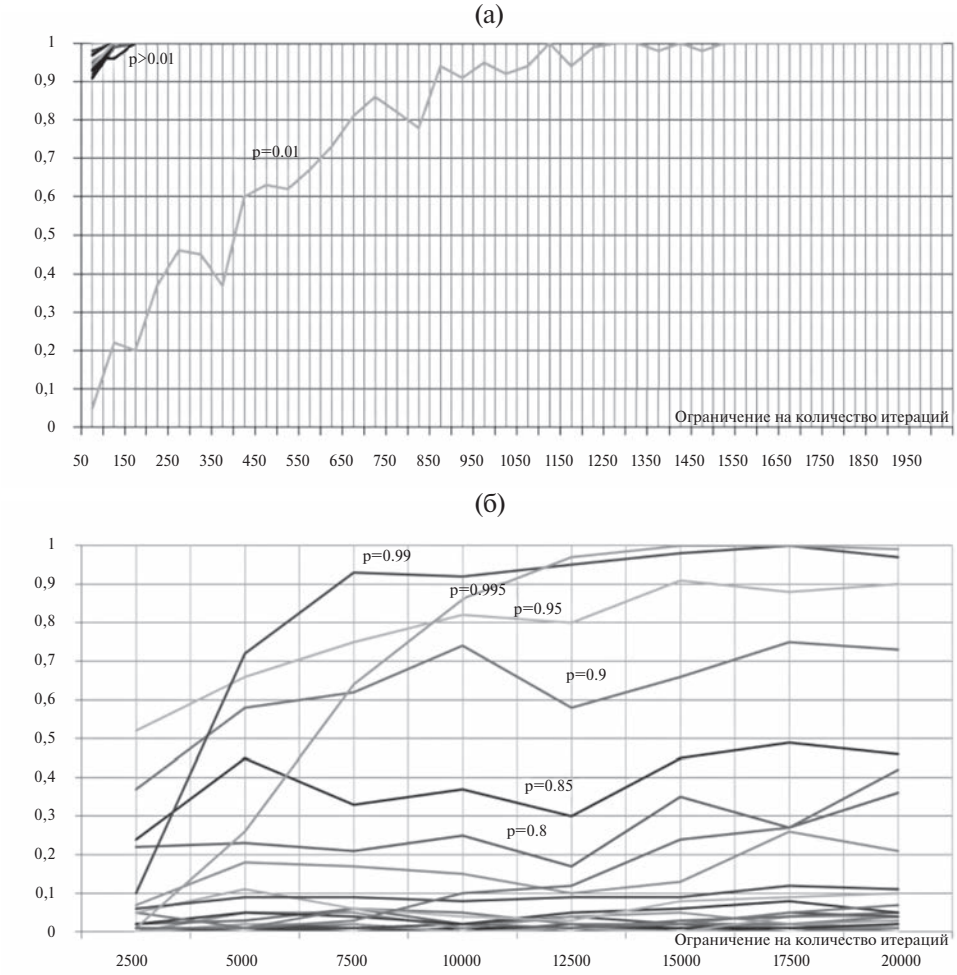
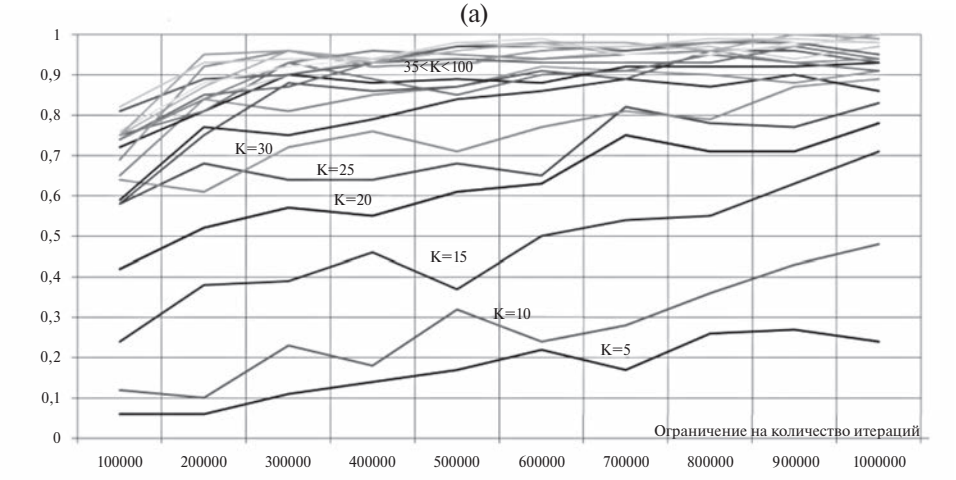


Рис. 5. Оценки статистических критериев эффективности работы алгоритма при изменении значений параметра ρ и ограничения на количества итераций: а – оценка вероятности найти оптимальное решение для графа малой размерности (бенчмарк), б – оценка вероятности найти оптимальное решение для графа большой размерности.

Так как все слагаемые в формуле (2.2) меньше 1, то увеличение степени приводит к уменьшению значения слагаемого, а коэффициент увеличивает значение. Верхний график на рис. 7 показывает влияние коэффициента на оценку вероятности найти решение, попадающее в область ψ значения целевой функции, в задаче большой размерности, для удобства часть линий скрыта. Наиболее эффективными являются значения $\alpha=1$, $\beta=1$ или $\beta=2$, $\gamma=0.5$. Таким образом, самый высокий вес будет у третьего слагаемого. Далее по эффективности идут группы,



(б)

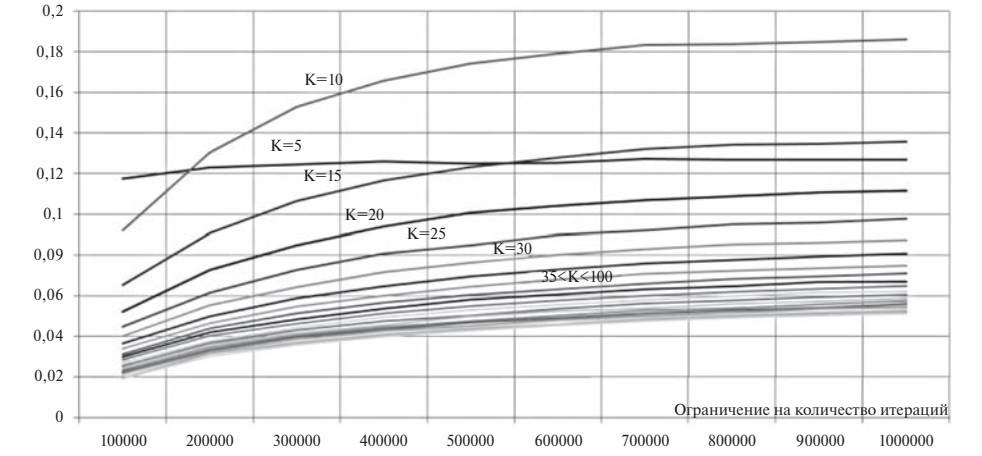
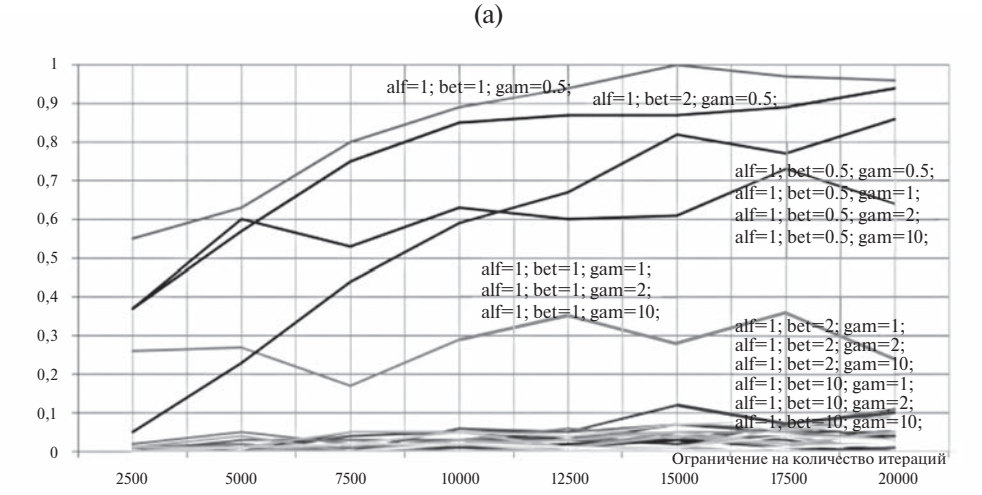


Рис. 6. Оценки статистических критериев эффективности работы алгоритма при изменении количества агентов на итерации K и ограничения на количества итераций: а – оценка вероятности найти оптимальное решение для графа большой размерности, б – оценка математического ожидания количества дополнительных итераций на одного агента для графа большой размерности.

(а)



(б)

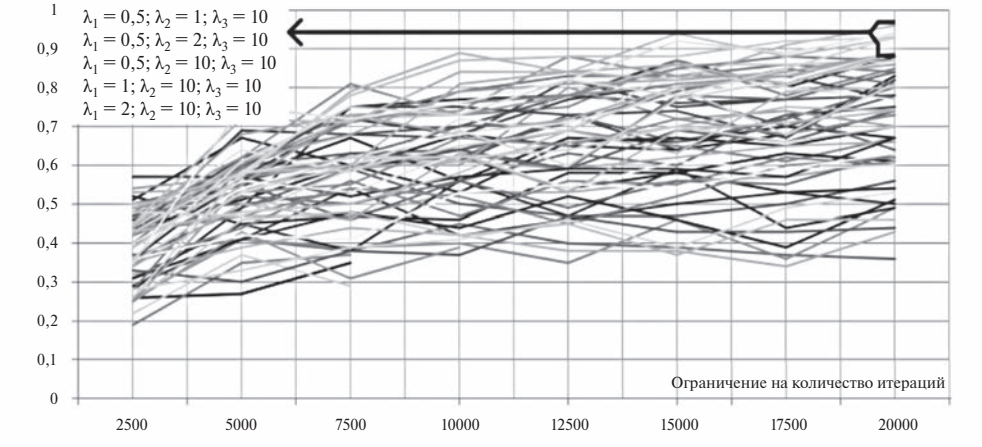


Рис. 7. Оценки статистических критериев эффективности работы алгоритма при изменении степенней α, β, γ и коэффициентов $\lambda_1, \lambda_2, \lambda_3$: а – влияние параметров степени α, β, γ на оценку вероятности найти оптимальное решение для графа большой размерности, б – влияние коэффициентов $\lambda_1, \lambda_2, \lambda_3$ на оценку вероятности найти оптимальное решение для графа большой размерности.

у которых $\alpha = 1$, $\beta = 0.5$ и любое значение γ, потом $\alpha = 1$, $\beta = 1$ и другие значения γ и $\alpha = 1$, $\beta = 2$ или $\beta = 10$ и $\gamma > 1$. В случае, когда $\alpha \neq 1$ эффективность алгоритма снижается. Рекомендуется не использовать данный параметр, а значение степени принимать равным 1 для α, β, γ. Такой выбор позволяет получить приемлемые характеристики эффективности работы алгоритма и при этом настройку алгоритма осуществлять с помощью коэффициентов $\lambda_1, \lambda_2, \lambda_3$.

Из рис. 7 (нижний график) видно, что влияние значений коэффициентов $\lambda_1, \lambda_2, \lambda_3$ разнообразно и зависит от относительных значений. Наилучшие показатели эффективности (оценка вероятности нахождения оптимального решения) связаны с соотношениями $\lambda_1 < \lambda_2 < \lambda_3$, причем разница может отличаться на порядок. Такое соотношение позволяет рассмотреть больше различных решений за счет высоких значений у коэффициентов λ_2 и λ_3, а далее учитывать распределение феромона.

Заключение. Предложена модификация метода муравьиных колоний для осуществления перебора значений параметров в задачах поиска решений приемлемых для ЛПР. Метод показал свою эффективность на задачах большой и малой размерности. Исследованы зависимости эффективности работы от заданных параметров метода. Созданные алгоритмы и программные модули позволяют находить приемлемые решения при анализе и проектировании сложных систем. Они могут быть полезны при решении задач мультидисциплинарной оптимизации в авиационной и ракетно-космической отраслях, когда для расчета значения целевой функции требуется провести вычислительные эксперименты с целым рядом сложных, в том числе имитационных моделей, включающих механизмы автоматического управления. Метод не гарантирует нахождения строго оптимального решения, но он позволяет определить удовлетворительное значение целевой функции за приемлемое время.

СПИСОК ЛИТЕРАТУРЫ

1. *Bergstra J.S., Bardenet R., Bengio Y., Kégl B.* Algorithms for Hyper-parameter Optimization // Adv. Neural Inform. Proc. Systems. 2011. V. 24. P. 2546–2554.
2. *Akiba T., Shotaro S., Toshihiko Y., Takeru O., Masanori K.* Optuna: A Next-generation Hyperparameter Optimization Framework // 25th ACM SIGKDD Intern. Conf. on Knowledge Discovery & Data Mining. N.Y., USA, 2019. P. 2623–2631;
https://doi.org/10.48550/arXiv.1907.10902
3. *Koehrsen W.* A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning. 2018 (открытый доступ 23.10.2022).
https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f)
4. *Dewancker I., McCourt M., Scott C.* Bayesian Optimization Primer (открытый доступ 23. 23.10.2022).
https://static.sigopt.com/b/20a144d208ef255d3b981ce419667ec25d8412e2/static/pdf/SigOpt_Bayesian_Optimization_Primer.pdf
5. IBM Bayesian Optimization Accelerator 1.1 Helps Identify Optimal Product Designs Faster with Breakthrough Performance for Scientific Discovery and High-performance Computing Simulation (открытый доступ 23.10.2022).
https://www.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/6/877/ENUSZP20-0186/index.html&request_locale=en
6. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. 2-е изд. М.: Изд-во МГТУ им. Баумана, 2017. 446 с.
7. *Colorni A., Dorigo M., Maniezzo V.* Distributed Optimization by Ant Colonies // Proc. First Eur. Conf. on Artific. Life / Eds F. Varela, P. Bourgine. Paris, France: Elsevier Publishing. 1992. P. 134–142.
8. *Dorigo M., Stützle T.* Ant Colony Optimization. Cambridge, Massachusetts: MIT Press, 2004. 321 p.
9. *Юхименко Б.И., Титов Н.А., Ушаков В.О.* Разработка и исследование алгоритмов муравьиной колонии для решения некоторых задач комбинаторной оптимизации // Актуальные научные исследования в современном мире. 2020. № 11-2 (67). С. 101–115.
10. *Семенкина О.Е., Семенкин Е.С.* О сравнении эффективности муравьиного и генетического алгоритмов при решении задач комбинаторной оптимизации // Актуальные проблемы авиации и космонавтики. 2011. Т. 1. № 7. С. 338–339.
11. *Semenkina O.E., Popov E.* Adaptive Ant Colony Optimization Algorithm for Hierarchical Scheduling Problem. Proc. Intern. Conf. on Information Technologies, Constantine and Elena. Varna, Bulgaria, 2019. P. 8860897;
https://doi.org/10.1109/InfoTech.2019.8860897
12. *Хахулин Г.Ф. Титов Ю.П.* Система поддержки решений поставок запасных частей летательных аппаратов военного назначения // Изв. Самарского научного центра Российской академии наук. 2014. Т. 16. № 1–5. С. 1619–1623.
13. *Судаков В.А., Батьковский А.М., Титов 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

14. *Martens D., De Backer M., Haesen R., Vanthienen J., Snoeck M., Baesens B.* Classification with Ant Colony Optimization // IEEE Trans. Evol. Comput. 2007. V. 11. № 5. P. 651–665.
15. *Синицын И.Н., Титов 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
16. *Судаков В.А., Титов Ю.П., Сивакова Т.В., Иванова П.М.* Применение метода муравьиных колоний для поиска рациональных значений параметров технической системы: Препринт № 38. М.: ИПМ, 2023. С. 1–15; https://doi.org/10.20948/prepr-2023-38
17. *Krzysztof S. Christian B.* An Ant Colony Optimization Algorithm for Continuous Optimization: Application to Feed-Forward Neural Network Training // Neural Comput. Appl. 2007. V. 3. № 16. P. 235–247; https://doi.org/10.1007/s00521-007-0084-z
18. *Пантелеев А.В., Алёшина Е.А.* Применение непрерывной модификации метода муравьиных колоний к задаче поиска оптимального управления дискретными детерминированными системами // Научный вестник МГТУ ГА. 2013. № 8 (194).
19. *Пантелеев А.В., Пановский В.Н.* Применение гибридного меметического алгоритма в задачах оптимального управления нелинейными стохастическими системами с неполной обратной связью // Научный вестник МГТУ ГА. 2018. Т. 21, № 2. С. 59–70; https://doi.org/10.26467/2079-0619-2018-21-2-59-70
20. *Саймон Д.* Алгоритмы эволюционной оптимизации / Пер. с англ. А.В. Логунова. М.: ДМК Пресс, 2020. 1002 с.
21. *Socha K., Dorigo M.* Ant Colony Optimization for Continuous Domains // Europ. J. of Oper. Res. 2008. V. 185. № 3. P. 1155–1173; https://doi.org/10.1016/j.ejor.2006.06.046
22. *Mohamad M., Tokhi M., Omar O.M.* Continuous Ant Colony Optimization for Active Vibration Control of Flexible Beam Structures // IEEE Intern. Conf. on Mechatronics (ICM). Istanbul, Turkey, 2011. P. 803–808.
23. *Карпенко А.П., Чернобривченко К.А.* Эффективность оптимизации методом непрерывно взаимодействующей колонии муравьев (CIAC) // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. 2011. № 2; https://doi.org/10.7463/0211.0165551
24. *Abdelbar A.M., Salama K.M., Falcón-Cardona J.G., Coello C.A.C.* An Adaptive Recombination-Based Extension of the iMOACOR Algorithm // IEEE Sympos. Series on Computational Intelligence (SSCI). Bangalore, India. 2018. P. 735–742; https://doi.org/10.1109/SSCI.2018.8628657
25. *Abdelbar A.M., Humphries T., Falcón-Cardona J.G., Coello C.A.* An Extension of the iMOACO Algorithm Based on Layer-Set Selection // Swarm Intelligence. ANTS 2022. Lecture Notes in Computer Science. 2022. V. 13491; https://doi.org/10.1007/978-3-031-20176-9_22
26. *Карпенко А.П., Чернобривченко К.А.* Мультимемеевая модификация гибридного муравьиного алгоритма непрерывной оптимизации HCIAC // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. 2012. № 9; https://doi.org/10.7463/0912.0470529
27. *Синицын И.Н., Титов Ю.П.* Исследование возможности получения всех решений методом муравьиных колоний для задачи // Системы высокой доступности. 2023. Т. 20. № 2. С. 55–69; https://doi.org/10.31857/S000523102308010X
28. *Russell S.J., Norvig P.* Artificial Intelligence: A Modern Approach: Pearson Series In Artificial Intelligence. Fourth Edition. Hoboken: Pearson, 2021. 1245 с.
29. *Синицын И.Н., Титов Ю.П.* Управление наборами значений параметров системы методом муравьиных колоний // АиТ. 2023. № 8. С. 153–168; https://doi.org/10.31857/S000523102308010X
30. *Синицын И.Н., Титов Ю.П.* Оптимизация параметрических задач с отрицательным значением целевой функции методом муравьиных колоний // Системы высокой доступности. 2024. Т. 20. № 1. С. 30–37; https://doi.org/10.18127/j20729472-202401-03
31. *Синицын И.Н., Титов Ю.П.* Исследование алгоритмов циклического поиска дополнительных решений при оптимизации порядка следования гиперпараметров методом муравьиных колоний // Системы высокой доступности. 2023. Т. 19. № 1. С. 59–73; https://doi.org/10.18127/j20729472-202301-05
32. *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
33. *Abdesslem L.* New Hard Benchmark Functions for Global Optimization. N.Y.: Cornell Tech, 2022; https://doi.org/10.48550/arXiv.2202.04606
34. Википедия. Тестовые функции для оптимизации (открытый доступ 23.10.2022) https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B5_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8_%D0%B4%D0%BB%D1%8F_%D0%BE%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8
35. Реализация модификаций алгоритма ACO (открытый доступ 23.10.2022). https://github.com/kalengul/ACO_Cluster/tree/master/Rezult