

УДК 519.6

DOI: [10.26102/2310-6018/2021.33.2.031](https://doi.org/10.26102/2310-6018/2021.33.2.031)

Исследование эффективности алгоритма глобальной оптимизации, вдохновленного некоторыми аспектами поведения тараканов

Д.С. Дубровкин, А.П. Карпенко, Н.В. Пивоварова

*Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет),
Москва, Российская федерация*

Резюме: Рассматриваем оптимизационный алгоритм нашествия тараканов (Roach Infestation Optimization, RIO), который относится к классу популяционных алгоритмов, вдохновленных живой природой. Алгоритм RIO предложен в 2008 г. и его можно считать глубокой модификацией широко известного и одного из наиболее эффективных оптимизационных алгоритмов роя частиц (Particle swarm optimization, PSO). Интерес к алгоритму RIO обусловлен тем, что в силу высокой эффективности алгоритма PSO для широкого круга задач глобальной оптимизации, особый интерес представляет исследование модификации этого алгоритма, которую представляет алгоритм RIO. Цель работы состоит в программной реализации и исследовании эффективности алгоритма RIO для известных сложных мультимодальных тестовых функций Ратригина и Экли. Особенностью исследования является поиск глобального экстремума (минимума) указанных функций в широкой области пространства поиска, в которой число локальных минимумов этих функций чрезвычайно велико. Представляем постановку рассматриваемой задачи глобальной оптимизации. Приводим описание алгоритма RIO, отличительным признаком которого является использование не оригинальных обозначений авторов этого алгоритма, но унифицированных обозначений, используемых нами при рассмотрении других популяционных алгоритмов. Описываем программное обеспечение, реализующее алгоритм, и организацию вычислительных экспериментов по исследованию его эффективности. Представляем результаты исследований, показывающие высокую перспективность алгоритма RIO для решения задач глобальной оптимизации.

Ключевые слова: глобальная безусловная оптимизация, популяционный алгоритм, алгоритм оптимизации роем частиц, функция Ратригина, функция Экли.

Для цитирования: Дубровкин Д.С., Карпенко А.П., Пивоварова Н.В. Исследование эффективности алгоритма глобальной оптимизации, вдохновленного некоторыми аспектами поведения тараканов. *Моделирование, оптимизация и информационные технологии*. 2021;9(2). Доступно по: <https://moitvvt.ru/ru/journal/pdf?id=984> DOI: 10.26102/2310-6018/2021.33.2.031

Effectiveness research of a global optimization algorithm inspired by some aspects of the cockroach behavior

D.S. Dubrovkin, A.P. Karpenko, N.V. Pivovarova

Bauman Moscow State Technical University (National research university of technology), Moscow, Russian Federation

Abstract: Consider the Roach Infestation Optimization (RIO) algorithm, which belongs to the class of population-based algorithms inspired by wildlife. The RIO algorithm was proposed in 2008 and can be considered a profound modification of the well-known and one of the most effective particle swarm optimization (PSO) algorithms. The interest in the RIO algorithm is, due to the high efficiency of the PSO algorithm for a wide range of global optimization problems, the study of the modification of this

algorithm, which is represented by the RIO algorithm, is of particular interest. The purpose of this paper is to implement software and study the efficiency of the RIO algorithm for the well-known complex multimodal test of Ratrigin and Ackley functions. A feature of the study is the search for a global extremum (minimum) of these functions in a broad region of the search space, in which the number of local minima of these functions is critically large. We present the global optimization problem definition, as well as a description of the RIO algorithm. Its distinctive feature is the use not of the original designations of the authors of this scheme but of the unified terms that we use when considering other population algorithms. We describe the software that implements the algorithm and the organization of computational experiments to study its effectiveness. Finally, the article presents the research results showing the high prospects of the RIO algorithm for solving global optimization problems.

Keywords: global unconstrained optimization, population-based algorithm, particle swarm optimization algorithm, Ratrigin function, Ackley function.

For citation: Dubrovkin D.S., Karpenko A.P., Pivovarova N.V. Effectiveness research of a global optimization algorithm inspired by some aspects of cockroach behaviour. *Modeling, Optimization and Information Technology*. 2021;9(2). Available from: <https://moitvvt.ru/ru/journal/pdf?id=984> DOI: 10.26102/2310-6018/2021.33.2.031 (In Russ).

Введение

В науке и технике часто возникают задачи глобальной параметрической оптимизации, отличительной особенностью которых является наличие, часто, большого или очень большого числа локально-оптимальных решений. В силу высокой практической значимости этих задач разработано большое число алгоритмы их решения. Вслед за [1], разделяем алгоритмы глобальной оптимизации на классические и не классические (популяционные). Популяционные алгоритмы (П-алгоритмы) часто обеспечивают более высокую эффективность (например, вероятность локализации глобального экстремума целевой функции) при решении сложных многомерных мультимодальных задач глобальной оптимизации.

Известно несколько классификаций П-алгоритмов глобальной оптимизации. Обычно выделяют следующие классы этих алгоритмов: эволюционные алгоритмы, включая генетические; П-алгоритмы, вдохновленные живой природой; алгоритмы, вдохновленные неживой природой; алгоритмы, инспирированные человеческим обществом; прочие алгоритмы [1].

Рассматриваемый в данной работе оптимизационный алгоритм нашествия тараканов (Roach Infestation Optimization, RIO) [2] относится к классу алгоритмов, вдохновленных живой природой. Этому классу принадлежит большое число «кластеров» П-алгоритмов: бактериальные алгоритмы; алгоритмы, вдохновленные насекомыми, млекопитающими, птицами, земноводными, рыбами, растениями и т. д. [3]. Алгоритм RIO входит в кластер П-алгоритмов, инспирированных поведением в природе насекомых. Кроме алгоритма RIO этому кластеру принадлежат муравьиные, пчелиные, светлячковые и некоторые другие алгоритмы [3].

Алгоритм RIO разработан на основе наблюдений за поведением тараканов [4-8] и его можно считать глубокой модификацией широко известного и одного из наиболее эффективных оптимизационных алгоритмов роя частиц (Particle swarm optimization, PSO) [9, 10].

RIO моделирует три следующих аспекта поведения тараканов.

Find Darkness. Тараканы ищут самое «темное» место в пространство поиска $\mathfrak{R}^{|X|}$.

Полагаем, что степень темноты в точке $X \in \mathfrak{R}^{|X|}$ прямо пропорциональна значению фитнес-функция в этой точке $\varphi(X)$.

Find Friends. Тараканы вероятно ищут соседних тараканов (тараканов своей группы). В качестве метрики близости используется евклидова метрика пространства $\mathbb{R}^{|X|}$.

Find Food. Тараканы периодически становятся голодными и в поисках пищи покидают достигнутые темные места и свои группы.

В силу высокой эффективности алгоритма PSO для широкого круга задач глобальной оптимизации, особый интерес представляет исследование данной модификации этого алгоритма. Цель работы состоит в программной реализации и исследовании эффективности алгоритма RIO для известных сложных мультимодальных тестовых функций Ратригина и Экли. Особенностью исследования является поиск глобального экстремума (минимума) указанных функций в широкой области пространства поиска, в которой число локальных минимумов этих функций чрезвычайно велико.

Постановка задачи и основные обозначения

Рассматриваем детерминированную задачу глобальной безусловной минимизации

$$\min_{X \in R^{|X|}} f(X) = f(X^*) = f^*, \quad (1)$$

где $|X|$ – размерность вектора варьируемых параметров $X = (x_k, k \in [1:|X|])$; $f(X)$ – целевая функция со значениями в пространстве R^1 ; X^* , f^* – искомые оптимальные решение и значение целевой функции соответственно. Полагаем, что поиск решения задачи (1) (то есть инициализация популяции) начинается в $|X|$ -мерном параллелепипеде $\Pi = \{X \mid X^- \leq X \leq X^+\} \subset \mathbb{R}^{|X|}$, где $X^-, X^+ \in \mathbb{R}^{|X|}$ и неравенства понимаются покомпонентно. Полагаем, что фитнес-функция $\varphi(X)$ П-алгоритма также подлежит минимизации. Найденные П-алгоритмом приближенные значения величины f^* и компонентов вектора X^* обозначаем \tilde{f}^* , \tilde{X}^* соответственно.

Отметим, что задача глобальной условной минимизации может быть сведена к аналогичной задаче безусловной оптимизации методом штрафных функций [1].

Примем еще следующие обозначения: $|S|$ – число особей (тараканов) в популяции; $\Delta X^-, \Delta X^+$ – $|X|$ -мерные векторы, определяющие параллелепипед $\Pi_\Delta = \{\Delta X \mid \Delta X^- \leq \Delta X \leq \Delta X^+\} \subset \mathbb{R}^{|X|}$ допустимых «скоростей» особей; c_0, c_{max} – вещественные константы, задающие веса составляющих «скорости»; $\hat{h} \in Z$ – целочисленный порог уровня «голода» тараканов; \hat{t} – максимально допустимое число итераций алгоритма; $\Delta h \in Z$ – целочисленное приращение уровня «голода»; ξ^1, ξ^2, ξ^3 – вероятности, определяющие число «соседей» каждого из тараканов (см. ниже).

Схема алгоритма

1) Инициализация

1.1) Инициализация пользователем свободных параметров алгоритма $|S|, |X|, X^-, X^+, \Delta X^-, \Delta X^+, c_0, c_{max}, \hat{h}, \hat{t}, \Delta h, \xi^1, \xi^2, \xi^3$, рекомендуемые значения которых имеют следующие значения: $|S| = 50$; $c_0 = 0,7$; $c_{max} = 1,43$; $\hat{h} = 100$; $\hat{t} = 1000$; $\xi^1 = 0,49$; $\xi^2 = 0,63$; $\xi^3 = 0,65$.

1.2) Инициализация начальных положений $X_i(0)$ и «скоростей» $\Delta X_i(0)$ особей популяции S . Равномерно случайно распределяем особей $s_i, i \in [1:|S|]$ в области Π , а их скорости - в области $\Pi_\Delta = \{\Delta X \mid \Delta X^- \leq \Delta X \leq \Delta X^+\}$.

1.3) Инициализация начального уровня голода тараканов $h_i(0)$, $i \in [1:|S|]$. Полагаем величины $h_i(0)$ равномерно случайно распределенными в интервале $[0: (\hat{h} - 1)]$.

2) Эволюция популяции (схема одной итерации алгоритма)

2.1 Вычисляем значения элементов $(|S| \times |S|)$ -матрицы $D = (d_{i,j}; i, j \in [1:|S|])$, имеющих смысл текущих евклидовых расстояний $\|\cdot\|_E$ между особями популяции: $d_{i,j} = \|X_i - X_j\|_E$.

2.2) С использованием матрицы D вычисляем текущее среднее расстояние между двумя особями популяции

$$d_g = \text{median} \{d_{i,j}; 1 \leq i < j \leq |S|\}.$$

2.3) Для каждой из особей $s_i, i \in [1:|S|]$ выполняем следующие действия.

а) Среди текущих следов особи s_i выбираем наилучшее положение $X_i(\tau_i)$ такое, что

$$\min_{\tau \in [0:\tau_i]} \varphi(X_i(\tau)) = \varphi(X_i^{hbest}).$$

б) Определяем множество текущих соседей $NB_i \subset S$ особи s_i (группу таракана s_i) в пространстве $\mathfrak{R}^{|X|}$ по формуле

$$NB_i = \{s_{i_j} \mid \|X_i - X_{i_j}\|_E < d_g; i_j \in [1:|S|], i_j \neq i\}. \quad (2)$$

в) Среди следов текущих соседей NB_i особи s_i с вероятностью ξ_i выбираем наилучшее положение $X_{i_g}^{NBhbest}$:

$$\min_{s_{i_j} \in NB_i} \min_{\tau \in [0:\tau_i]} \varphi(X_{i_j}(\tau)) = \varphi(X_{i_g}^{NBhbest}).$$

Здесь вероятность ξ_i определяет формула

$$\xi_i = \begin{cases} \xi^1, & |NB_i| = 1, \\ \xi^2, & |NB_i| = 2, \\ \xi^3, & |NB_i| \geq 3. \end{cases} \quad (3)$$

г) Если текущий уровень голода таракана s_i не превышает заданный уровень \hat{h} , то есть если $h_i \leq \hat{h}$, то применяем рандомизирующий трех шаговый оператор перемещения особи s_i :

$$X'_i = X_i + \Delta X_i = X_i(t+1); \quad (4)$$

$$\Delta X_i = c_0 \Delta X_i(t-1) + U_{|X|}(0; c_{\max}) \otimes (X_i^{hbest} - X_i) + U_{|X|}(0; c_{\max}) \otimes (X_{i_g}^{NBhbest} - X_i). \quad (5)$$

д) Если уровень голода того же таракана превышает \hat{h} , то есть если $h_i > \hat{h}$, то выполняем следующие действия:

- перемещаем особь s_i в случайную точку $X_i(t+1)$, созданную в параллелепипеде Π по правилам инициализации начальной популяции;

- полагаем $h_i = 0$.

2.4) Повышаем уровень голода таракана s_i по правилу $h'_i = h_i + \Delta h$.

3) *Завершение эволюционного процесса.*

Условием окончания поиска является достижение числа итераций эволюционного процесса, равного \hat{t} .

Отметим, что поскольку в определении множество соседей NB_i особи s_i используется ограничение $i_j \neq i$, в это множество не входит особь s_i (см. (2)). Отсюда следует, что наилучшее положение $X_i^{NB_{best}}$ не может совпадать с наилучшим положением $X_i^{h_{best}}$.

Поведения *Find Darkness*, *Find Friends* моделируют формула (4) и первые два слагаемых формулы (5) или третье слагаемое формулы (5) соответственно. Правило поведения *Find Food* формализует п. 2.3д схемы алгоритма. Таракан s_i перемещается в пространстве поиска по правилам *Find Darkness*, *Find Friends* если его текущий уровень голода h_i не превышает заданный уровень \hat{h} . В противном случае этот таракан перемещается по правилам *Find Food*.

Программное обеспечение и организация вычислительных экспериментов

Алгоритм нашествия тараканов реализован на языке программирования Python и IDE PyCharm. Каждый из тараканов является объектом класса *Cockroach*, а рой – объектом класса *Swarm*. Используются указанные выше рекомендованные значения свободных параметров алгоритма.

Для тестирования алгоритма и разработанного программного обеспечения использована сферическая функция

$$f(X) = \sum_{k=1}^{|X|} x_k^2,$$

минимум которой достигается в точке $X = 0_{|X|}$ и равен, очевидно, нулю.

Исследование эффективности алгоритма RIO выполнено на двух тестовых функциях с известными точными положениями X^* и значениями f^* глобального минимума [1]:

- функции Растригина

$$f(X) = 10 * |X| + \sum_{k=1}^{|X|} [x_k^2 - 10 \cos(2\pi x_k)];$$

- функция Экли

$$f(X) = 20 + e - 20 \exp \left(-0,2 \sqrt{\frac{1}{|X|} \sum_{k=1}^{|X|} x_k^2} \right) - \exp \left(\frac{1}{|X|} \sum_{k=1}^{|X|} [\cos(2\pi x_k)] \right).$$

Во всех случаях инициализация начальной популяции осуществлена в кубе, так что $abs(x_k^-) = x_k^+$, $k \in [1:|X|]$ (см. ниже).

Принято, что фитнес-функция совпадает с целевой функцией, то есть что имеет место равенство $\varphi(X) = f(X)$. Критерием окончания поиска является стагнация вычислительного процесса: если на протяжении δt итераций значение фитнес-функции

улучшается менее чем на $\delta f = 10^{-6}$, то поиск прекращается. Используем значение величины δt , равное 150.

Вычислительные эксперименты выполнены для размерностей пространства поиска $|X|$, равных 2, 4, 8, 16, 32 и 64. Для каждого из экспериментов применен метод мультистарта с числом запусков алгоритма, равным $n = 100$.

Рассматриваем следующие индикаторы эффективности алгоритма:

1) среднее число итераций алгоритма \bar{t} ;
 2) оценка математического ожидания $\bar{\varepsilon}_f$ достигнутой абсолютной точности в пространстве значений целевой функции - среднее по мультистарту значение величины $abs(f^* - \tilde{f}^*)$;

3) аналогичная оценка математического ожидания $\bar{\varepsilon}_X$ достигнутой значения абсолютной точности в пространстве поиска – оценка среднего по мультистарту значения величины $\|X^* - \tilde{X}^*\|_E$;

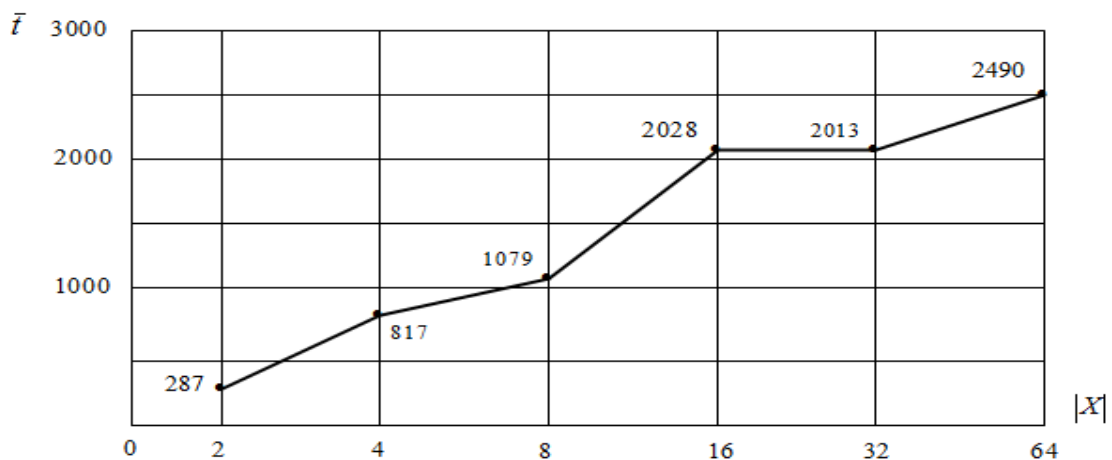
4) оценка вероятности локализации глобального экстремума $\tilde{p} = \frac{n^+}{n}$, где n^+ – число успешных стартов алгоритма, то есть стартов, в которых был выполнен критерий локализации глобального экстремума

$$\|\tilde{X}^* - X^*\| \leq \lambda l \sqrt{|X|}.$$

Здесь $\lambda = 0,01$ - заданная константа; l – длина ребра начального куба Π с центром в точке $X = 0_{|X|}$. Для всех рассматриваемых размерностей вектора варьируемых параметров $|X|$ эта длина принята равной $l=200$, то есть компоненты x_k^-, x_k^+ векторов X^-, X^+ во всех случаях выбирались так, чтобы это равенство было справедливым.

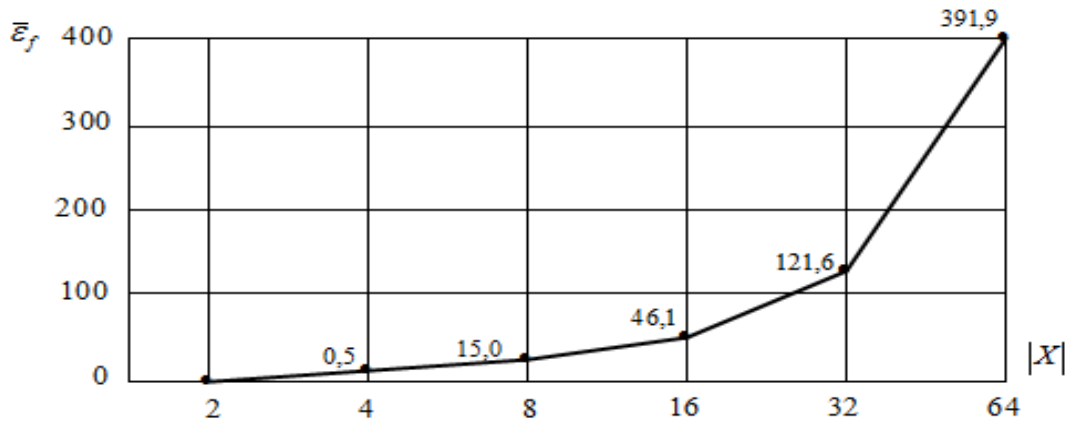
Результаты вычислительных экспериментов и их обсуждение

На Рисунке 1 представлены некоторые результаты исследования эффективности для функции Растригина.

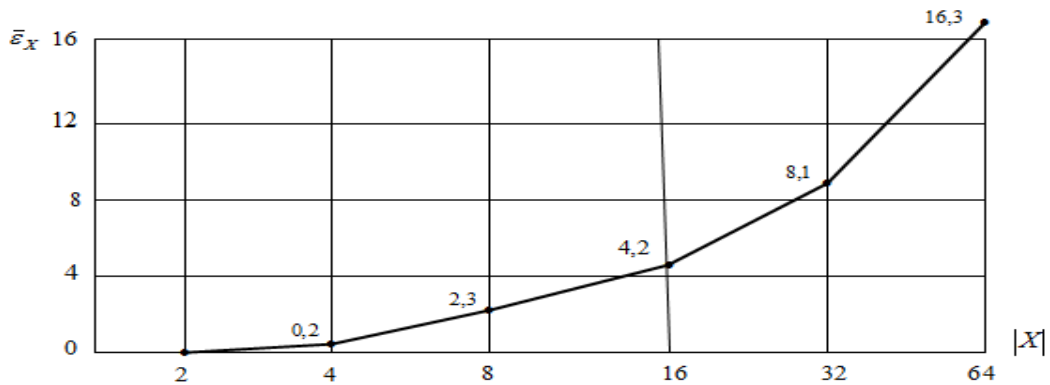


а) Число итераций \bar{t} в функции размерности пространства поиска $|X|$

a) The number of iterations \bar{t} in the function of the dimension of the search space $|X|$

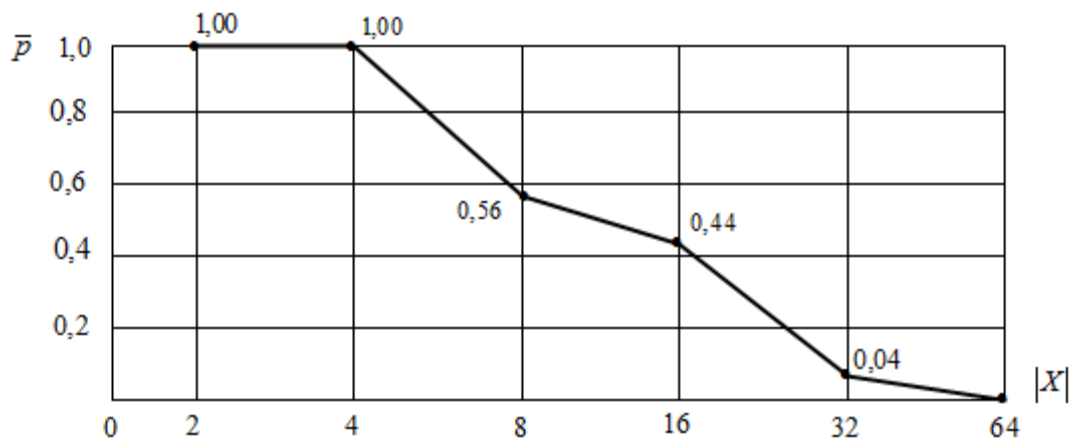


б) Средняя достигнутая точность по значениям целевой функции в зависимости от $|X|$
 b) The average achieved accuracy for the values of the objective function, depending on $|X|$



в) Средняя достигнутая точность положения результирующей точки в пространстве поиска в функции $|X|$

с) The average achieved accuracy of the position of the resulting point in the search space in the function $|X|$



г) Оценка вероятности \bar{p} локализации глобального минимума в зависимости от $|X|$

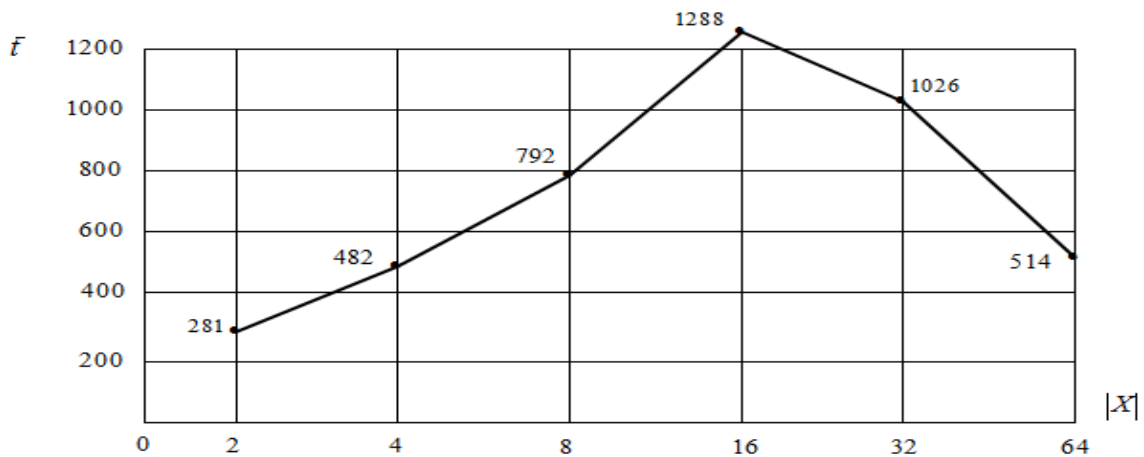
д) Estimation of the probability of localization of the global minimum depending on $|X|$

Рисунок 1 – Результаты вычислительных экспериментов: функция Растригина

Figure 1 – Results of computational experiments: Rastrigin function

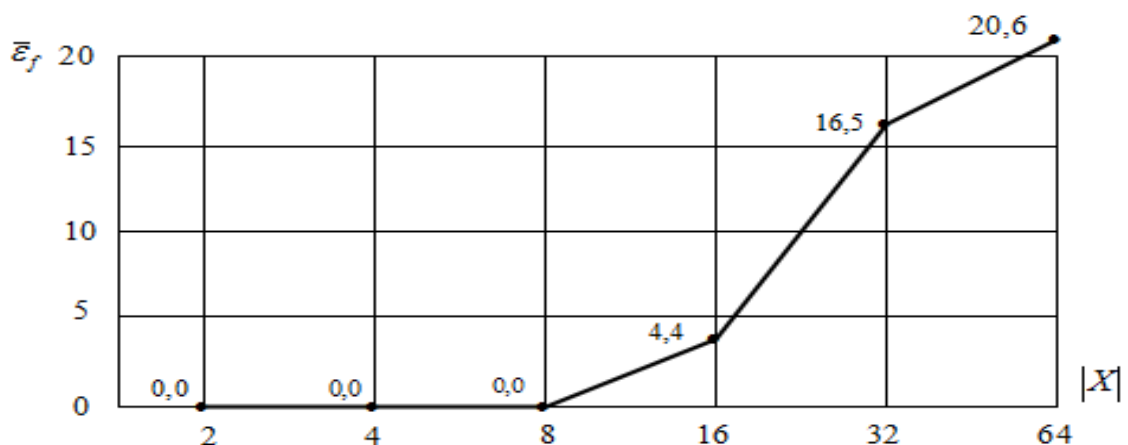
Прежде, чем анализировать представленные результаты, отметим, что из-за большого размера начального куба Π (длина ребра $l = 200$), число локальных минимумов функции Растргина (равно, как и функции Экли) в этой области чрезвычайно велико и быстро растет с ростом размерности $|X|$ пространства поиска [1]. Потому отыскание глобального минимума этих функций в указанной области представляет собой экстремально сложную задачу для любого из популяционных алгоритмов, особенно, при $|X| > 10$.

Результаты, представленные на Рисунках 1а – 1в, ожидаемо показывают, что с ростом размерности задачи $|X|$ быстро растет число итераций \bar{t} до стагнации вычислительного процесса, а также погрешности $\bar{\varepsilon}_f$, $\bar{\varepsilon}_x$. Интегрально, эффективность алгоритма и реализующей его программы характеризует Рисунок 1г, из которого следует, что до $|X| \approx 8$ обеспечивается приемлемая вероятность локализации глобального минимума функции Растргина. Напомним, что нами использовано число особей популяции, равное $|S| = 50$, и метод мултистарта с числом стартов $n = 100$. Повысить вероятность локализации глобального минимума, естественно, за счет увеличения вычислительных затрат на поиск, можно путем увеличения числа особей и/или числа стартов.



На Рисунке 2, аналогичном Рисунку 1, представлены результаты исследования эффективности для функции Экли.

а) $\bar{t}(|X|)$



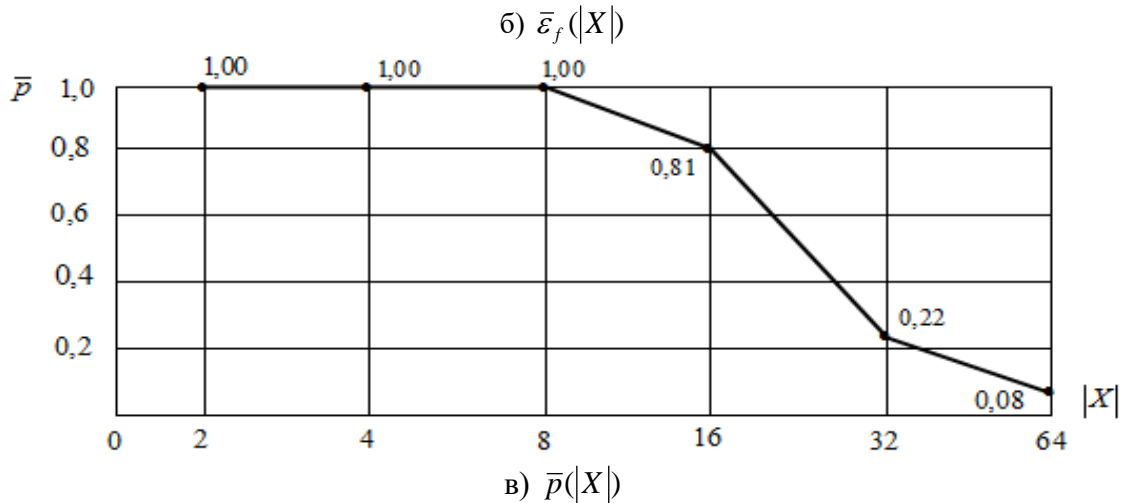


Рисунок 2 – Результаты вычислительных экспериментов: функция Экли
 Figure 2 – Results of computational experiments: Ackley function

В силу близости ландшафтов функций Растригина и Экли, качественно, представленные результаты для функции Экли близки результатам исследования для функции Растригина. Отметим неожиданное уменьшение числа итераций \bar{t} при размерности вектора варьируемых параметров $|X|$, превышающем 16. Причина этого эффекта не понятна и требует самостоятельного исследования. Оценка вероятности \bar{p} локализации глобального минимума в данном случае оказывается более высокой, чем в случае функции Растригина: сохраняется равной единице вплоть до $|X| = 8$, уменьшаясь до приемлемой величина при $|X| = 16$. Как и в предыдущем случае, повысить эффективность поиска можно путем увеличения числа особей и/или число стартов алгоритма.

Заключение

Легко видеть, что формулы (4), 5) близки по структуре к формулам, которые определяют эволюцию особей в алгоритме роя части PSO [1]. Основное отличие заключается в том, что при вычислении вектора скорости особи (см. (5)) используется не лучший результат всей популяции, как в алгоритме PSO, а лучший результат группы этой особи. Такое решение имеет свои позитивные и негативные стороны. Тот факт, что в алгоритме RIO не приходится отыскивать глобально лучшее решение, может позволить, например, эффективно распараллелить этот алгоритм для вычислительных систем с распределенной памятью. С другой стороны, отказ в алгоритме RIO от использования метрики близости в топологическом пространстве в пользу метрики пространства поиска $\mathcal{R}^{|X|}$ снижает диверсификационные свойства алгоритма. Можно предположить, что именно поэтому авторам алгоритма RIO пришлось ввести правило поведения *Find Food*, призванное предотвратить преждевременную стагнацию вычислительного процесса. Кроме того, использование метрики пространства $\mathcal{R}^{|X|}$ вынуждает на каждой итерации алгоритма RIO вычислять евклидовы расстояния между всеми парами особей популяции, что требует значительных вычислительных затрат и глобальной информации о всей популяции.

Для диверсификации поиска алгоритм RIO использует в механизме образования групп особей NB_i следующий прием. В качестве точки $X_{i_g}^{NB_{hbest}}$ выбирается не

обязательно самая лучшая точка среди следов всех соседей NB_i особи s_i , но одна из лучших таких точек. Выбор происходит с вероятностью ξ_i , зависящей от размера группы (см. (3)). В основе правила определения вероятности ξ_i лежат биологические предпосылки [4].

Поясним, наконец, оригинальный механизм диверсификации поиска, реализующий правило поведения особей *Find Food*. С каждым тараканом s_i алгоритм связывает его целочисленный уровень «голода» h_i , который на этапе инициализации полагаем распределенным равномерно случайно в заданном интервале. Значение величины h_i на каждом шаге эволюции популяции увеличиваем на фиксированную целочисленную величину Δh до выполнения условия $h_i > \hat{h}$, где \hat{h} - заданный уровень. При этом по правилам инициализации начальной популяции перемещаем таракана s_i в случайную точку области поиска и обнуляем его уровень голода. Такой механизм диверсификации поиска можно интерпретировать, как частичную «встряску» популяции [1].

Результаты представленного исследования позволяют утверждать, что оптимизационный алгоритм нашествия тараканов RIO перспективен для поиска экстремума сложных многоэкстремальных функций. Как почти всякий популяционный алгоритм, алгоритм RIO имеет значительное число свободных параметров, так что эффективность этого алгоритма применительно к той или иной задаче глобальной оптимизации может быть значительно повышена за счет его параметрической настройки [1]. В развитии работы авторы предполагают выполнить такую настройку алгоритма RIO, а также реализовать широкое сравнительное исследование этого алгоритма и алгоритма роя частиц PSO.

ЛИТЕРАТУРА

1. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы вдохновленные природой. Москва: Издательство МГТУ им. Н.Э. Баумана; 2014.
2. Havens T.C. et al. Roach infestation optimization. *In: Proceedings of the 2008 IEEE Swarm Intelligence Symposium*, St. Louis, MO, USA. 2008:21-23.
3. Bo Xing, Wen-Jing Gao. *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*. Springer International Publishing Switzerland; 2014.
4. Jeanson R. et al. Selforganized aggregation in cockroaches. *Animal Behaviour*. 2005; 69: 169–180.
5. Halloy J. et al. Social integration of robots into groups of cockroaches to control self-organized choices. *Science*. 2007;318(5853):1155-1158.
6. Ame J. et al. Collegial decision making based on social amplification leads to optimal group formation. *Proc. Natl. Acad. Sci.* 2006;103(15):5835-5840.
7. Garnier S. et al. Collective decision-making by a group of cockroach-like robots. *Proc. 2005 IEEE Swarm Intelligence Symposium (SIS 2005)*. - Pasadena, CA, USA. 2005;233-240.
8. Watanabe H., Mizunami M. Pavlov's cockroach: Classical conditioning of salivation in an insect. *PLoS ONE*. 2007;2(6):529.
9. Kennedy J., Eberhardt R. Particle swarm optimization. *Proceedings of the IEEE Int. Conf. on Neural Networks*, Piscataway, NJ. 1995;1942-1948.
10. Clerc M. Particle Swarm Optimization. Newport Beach, CA: ISTE USA, 2006.

REFERENCES

1. Karpenko A.P. *Sovremennye algoritmy poiskovoj optimizacii. Algoritmy vdohnovlennye prirodoj*. Moskva: Izdatel'stvo MGTU im. N.E. Baumana; 2014.
2. Havens T.C. et al. Roach infestation optimization. In: *Proceedings of the 2008 IEEE Swarm Intelligence Symposium*, St. Louis, MO, USA. 2008:21-23.
3. Bo Xing, Wen-Jing Gao. *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*. Springer International Publishing Switzerland; 2014.
4. Jeanson R. et al. Selforganized aggregation in cockroaches. *Animal Behaviour*. 2005;69:169-180.
5. Halloy J. et al. Social integration of robots into groups of cockroaches to control self-organized choices. *Science*. 2007;318(5853):1155-1158.
6. Ame J. et al. Collegial decision making based on social amplification leads to optimal group formation. *Proc. Natl. Acad. Sci.* 2006;103(15):5835-5840.
7. Garnier S. et al. Collective decision-making by a group of cockroach-like robots. *Proc. 2005 IEEE Swarm Intelligence Symposium (SIS 2005)*. Pasadena, CA, USA. 2005; 233-240.
8. Watanabe H., Mizunami M. Pavlov's cockroach: Classical conditioning of salivation in an insect. *PLoS ONE*. 2007;2(6):529.
9. Kennedy J., Eberhardt R. Particle swarm optimization. *Proceedings of the IEEE Int. Conf. on Neural Networks*, Piscataway, NJ. 1995;1942-1948.
10. Clerc M. Particle Swarm Optimization. Newport Beach, CA: ISTE USA, 2006.

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Дубровкин Дмитрий Станиславович, аспирант, Московский государственный технический университет имени Н.Э. Баумана, Москва, Российская Федерация. **Dmitry S. Dubrovkin**, PhD student, Bauman Moscow State Technical University, Moscow, Russian Federation.
e-mail: dmitry.dim-2011@yandex.ru

Карпенко Анатолий Павлович, д.ф.-м.н., профессор, зав. кафедрой, Московский государственный технический университет имени Н.Э. Баумана, Москва, Российская Федерация. **Anatoly P. Karpenko**, professor, head of department, Bauman Moscow State Technical University, Moscow, Russian Federation.
e-mail: apkarpenro@mail.ru
ORCID: [0000-0001-7904-5057](https://orcid.org/0000-0001-7904-5057)

Пивоварова Наталья Владимировна, к.т.н., доцент, Московский государственный технический университет имени Н.Э. Баумана, Москва, Российская Федерация. **Natalia V. Pivovarova**, assistant professor, Bauman Moscow State Technical University, Moscow, Russian Federation.
e-mail: pivovarova.natasha2013@yandex.ru