

DOI: 10.51790/2712-9942-2022-3-4-10

**ПРИМЕНИМОСТЬ АЛГОРИТМОВ РОЕВОГО ИНТЕЛЛЕКТА ДЛЯ РЕШЕНИЯ ЗАДАЧ
МИНИМИЗАЦИИ ФУНКЦИЙ РАЗНЫХ КЛАССОВ****В. А. Галкин^a, Т. В. Гавриленко^b, А. Д. Смородинов^c, О. П. Бобровская^d***Сургутский государственный университет, г. Сургут, Российская Федерация
Сургутский филиал Федерального государственного учреждения «Федеральный научный центр
Научно-исследовательский институт системных исследований Российской академии наук»,
г. Сургут, Российская Федерация*^a ORCID: <http://orcid.org/0000-0002-9721-4026>, val-gal@yandex.ru^b ORCID: <http://orcid.org/0000-0002-3243-2751>, taras.gavrilenko@gmail.com^c ORCID: <http://orcid.org/0000-0002-9324-1844>, sachenka_1998@mail.ru^d ORCID: <http://orcid.org/0000-0001-7045-9085>, o-bobrovskaya@mail.ru

Аннотация: в статье рассматриваются существующие роевые алгоритмы, приводятся их особенности. Подробно описывается алгоритм роя частиц, обеспечивающий оптимизацию функции. Проводятся эксперименты с различными параметрами роевого алгоритма (размера роя и времени жизни роя) для функций разных классов, как с одной точкой минимума, так и с несколькими. Делаются выводы о применимости алгоритма роя частиц для решения задач оптимизации.

Ключевые слова: роевой интеллект, роевой алгоритм, алгоритм роя частиц, минимизация функции, максимизация функции, поиск точек экстремума функций.

Благодарности: публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (выполнение фундаментальных научных исследований ГП 47) по теме № FNEF-2022-0007 «Развитие методов математического моделирования распределенных систем и соответствующих методов вычисления», рег. № 1021060909180-7-1.2.1.

Для цитирования: Галкин В. А., Гавриленко Т. В., Смородинов А. Д., Бобровская О. П. Применимость алгоритмов роевого интеллекта для решения задач минимизации функций разных классов. *Успехи кибернетики*. 2022;3(4):84–97. DOI: 10.51790/2712-9942-2022-3-4-10.

**SWARM INTELLIGENCE ALGORITHMS APPLICABILITY FOR MINIMIZING VARIOUS
FUNCTIONS****V. A. Galkin^a, T. V. Gavrilenko^b, A. D. Smorodinov^c, O. P. Bobrovskaya^d***Surgut State University, Surgut, Russian Federation
Surgut Branch of Federal State Institute “Scientific Research Institute for System Analysis of the Russian
Academy of Sciences”, Surgut, Russian Federation*^a ORCID: <http://orcid.org/0000-0002-9721-4026>, val-gal@yandex.ru^b ORCID: <http://orcid.org/0000-0002-3243-2751>, taras.gavrilenko@gmail.com^c ORCID: <http://orcid.org/0000-0002-9324-1844>, sachenka_1998@mail.ru^d ORCID: <http://orcid.org/0000-0001-7045-9085>, o-bobrovskaya@mail.ru

Abstract: this study considers the existing swarm intelligence algorithms and their features. The particle swarm algorithm used for function optimization is described in detail. We adjusted various swarm algorithm properties (swarm size and lifetime) for minimizing functions of different classes, both with one and multiple minimums. It was found that the particle swarm algorithm is applicable to solving optimization problems.

Keywords: swarm intelligence, swarm algorithm, particle swarm algorithm, function minimization, function maximization, function extremum seeking.

Acknowledgements: this study is a part of the government contract 47 GP with the Scientific Research Institute for System Analysis, Russian Academy of Sciences, project No. FNEF-2022-0007 Advancing Distribution System Simulation and Computation Methods, reg. № 1021060909180-7-1.2.1.

Cite this article: Galkin V. A., Gavrilenko T. V., Smorodinov A. D., Bobrovskaya O. P. Swarm Intelligence Algorithms Applicability for Minimizing Various Functions. *Russian Journal of Cybernetics*. 2022;3(4):84–97. DOI: 10.51790/2712-9942-2022-3-4-10.

Введение

Несколько десятилетий назад появилось новое направление в разработке искусственного интеллекта — роевой интеллект. **Роевой интеллект** (англ. Swarm intelligence) описывает коллективное поведение децентрализованной самоорганизующейся системы. Изучается в теории искусственного интеллекта как особый метод решения задач оптимизации. Впервые подобная идея была рассмотрена Станиславом Лемом в романе «Непобедимый» (1964) и эссе «Системы оружия двадцать первого века, или Эволюция вверх ногами» (1983). Термин был введен Херардо Бени и Ван Цзином в 1989 году [1] в контексте системы клеточных автоматов (роботов).

Роевой интеллект обычно создается как множество агентов, которые локально взаимодействуют между собой и окружающей средой. Поведение алгоритма строится на основе некоторых реальных объектов природы, чаще всего за основу берут биологическую систему. Каждый исследователь, работающий с роевыми алгоритмами, дает свое наименование элементам роя, например, частица, боид, квант — все это синонимы к словосочетанию «элемент роя». В некоторых случаях элемент роя могут называть муравьем или волком, если речь идет о муравьином алгоритме или алгоритме серых волков соответственно. В данной работе элемент роя будет именоваться частицей. Выбор этого синонима обоснован несколькими причинами: во-первых, понятия «боид» и «квант» введены зарубежными авторами и русскоязычным читателям, которые впервые знакомятся с данной темой, могут быть непонятны. Частица же достаточно распространенное и точное наименование элемента роя. Во-вторых, именно такое название используется в базовом роевом алгоритме (рое частиц).

Обычно у роя нет центра управления, который предписывал бы каждому элементу роя, что ему делать в данный момент. Каждая частица подчиняется достаточно простым правилам, но все равно локальные взаимодействия с внесением небольшой случайной составляющей приводят к возникновению группового поведения, которое неподконтрольно отдельным частицам роя. В настоящий момент не существует точного определения роевого интеллекта, есть только набор условий, при соответствии которым алгоритм можно назвать роевым [2]:

1. Роевой интеллект должен представлять собой многоагентную систему.
2. Система должна обладать самоорганизующимся поведением.
3. Поведение роя должно иметь признаки «разумного» поведения.

Первые два условия достаточно очевидны: система должна состоять из нескольких частиц, которые подчиняются некоторым правилам, формируемым создателем роя, третье же не так однозначно. Достаточно сложно определить, является ли поведение созданного роя «разумным». Единственное, что можно сказать определенно — поведение роя должно отличаться от алгоритма случайного перебора.

Роевые алгоритмы применяются для решения задач, которые обладают следующими особенностями [3]:

- 1) нелинейность;
- 2) недифференцируемость;
- 3) многоэкстремальность;
- 4) овражность;
- 5) отсутствие аналитического выражения;
- 6) высокая вычислительная сложность;
- 7) высокая размерность пространства поиска;
- 8) сложная топология области допустимых значений.

Рой можно определить как децентрализованную систему, которая состоит из простых и однообразных элементов, взаимодействующих между собой определенным образом. Примерами из реального биологического мира могут служить колонии муравьев, рой пчел, стаи птиц или косяки рыб. С концептуальной точки зрения такие алгоритмы строятся на эффекте аддитивности и синергии, который проявляется себя при объединении частиц в одну систему.

На данный момент известно не менее 22 различных алгоритмов роя:

1. Метод роя частиц.
2. Генетический алгоритм.
3. Муравьиный алгоритм.
4. Пчелиный алгоритм.
5. Искусственная иммунная система.

6. Алгоритм серых волков.
7. Алгоритм летучих мышей.
8. Алгоритм гравитационного поиска.
9. Алгоритм альтруизма.
10. Светляковый алгоритм.
11. Алгоритм капель воды.
12. Метод формирования реки.
13. Метод самоходных частиц.
14. Стохастический диффузионный поиск.
15. Многороевая оптимизация.
16. Алгоритм кукушки (кукушкин поиск).
17. Оптимизация передвижением бактерий.
18. Обезьяний алгоритм.
19. Поиск косяком рыб.
20. Тасующий алгоритм роя лягушек.
21. Электромагнитный поиск.
22. Гармонический поиск.

Это неполный перечень, который постоянно дополняется новыми алгоритмами или модификациями старых. Изначально авторы многих алгоритмов не относили их к роевым, раньше некоторые алгоритмы носили название поведенческий, интеллектуальный метаэвристический, вдохновленный (инспирированный) природой, многоагентный, популяционный. В настоящий момент все эти алгоритмы объединены под одним названием — роевой интеллект.

Далее в данной работе будет рассмотрен базовый алгоритм роя частиц и проведена серия численных экспериментов, в которых будет изучена применимость данного алгоритма для решения задач минимизации функций разных классов.

Метод роя частиц

Базовый алгоритм роя частиц — это метод численной оптимизации, для использования которого не требуется знать точного градиента оптимизируемой функции. Впервые он был показан Кеннеди, Эберхартом и Ши [4, 5, 6], и изначально предназначался для имитации социального поведения. В дальнейшем алгоритм был упрощен и было замечено, что он пригоден для решения задач оптимизации. Обширное исследование способов применения было сделано Поли [7, 8]. Метод роя частиц оптимизирует функцию, поддерживая популяцию возможных решений, называемых частицами и перемещая эти частицы в пространстве решений согласно простой формуле. Перемещения подчиняются принципу наилучшего найденного в этом пространстве положения, которое постоянно изменяется при нахождении частицами более выгодных положений [9]. Если говорить о данном методе с точки зрения математического аппарата, то алгоритм можно описать следующим образом:

Шаг 0:

1. Задается оптимизируемая функция — $F(X)$.
2. Выбирается тип задачи (максимум, минимум).
3. Определяется область решения — $G(X)$.
4. Выбирается критерий остановки.
5. Выбирается размер роя — n .
6. Указывается максимальная и минимальная скорость роя — (V_{min}, V_{max}) .
7. Указываются гиперпараметры:
 - 1) характеристика инерционного свойства частицы — Om ;
 - 2) коэффициент, определяющий степень учета индивидуального опыта, — A_1 ;
 - 3) коэффициент, определяющий степень учета группового опыта, — A_2 .

Шаг 1: Генерация начальных положений и скоростей частиц роя $j = 1$ по формулам:

$$X_{i1} = \text{rand}(G(X)), \quad i = 1, \dots, n, \quad \text{где } n \text{ — количество частиц в рое} \quad (1)$$

$$v_{i1} = \text{rand}(V_{min}, V_{max}), \quad i = 1, \dots, n \quad (2)$$

$X_{i1}^{best} = X_{ij}$ – сгенерированная точка считается «лучшей», в которой побывала частица.

Случайным образом выбирается лучшая точка роя $M = X_{i1}^{best}$, где i – произвольная частица на первой итерации.

Шаг 2: Перемещение частиц

$$V_{ij+1} = V_{ij} \cdot Om + A_1 \cdot (X_{ij}^{best} - X_{ij}) \cdot \text{rand}(0,1) + A_2 \cdot (M - X_{ij}) \cdot \text{rand}(0,1) \quad (3)$$

$$V_{ij+1} = \begin{cases} V_{ij+1}, & V_{min} \leq V_{ij+1} \leq V_{max} \\ V_{min}, & V_{ij+1} \leq V_{min} \\ V_{max}, & V_{ij+1} \geq V_{max}. \end{cases} \quad (4)$$

$$X_{ij+1} = \begin{cases} X_{ij} + V_{ij+1}, & G(X_{ij} + V_{ij+1}) = 1 \\ X_{ij}, & G(X_{ij} + V_{ij+1}) = 0 \end{cases} \quad (5)$$

Шаг 3: Определение лучшего положения частицы в рое

Определяется, является ли текущее положение частицы лучшим, чем предыдущее:

$$X_{ij}^{best} = X_{ij}, \quad F(X_{ij}^{best}) < F(X_{ij}) \quad (6)$$

Определяется, является ли лучшее положение какой-либо частицы роя лучшим во всем рое:

$$M = X_{ij}^{best}, \quad F(M) < F(X_{ij}^{best}). \quad (7)$$

Шаг 4: Проверка критериев остановки:

1. Время жизни роя закончилось (количество шагов либо время в секундах).
2. Лучшая частица роя перестала изменяться на протяжении M шагов.

Если критерий остановки не достигнут, то возвращаемся к шагу 2.

Далее опишем результаты численных экспериментов, которые показывают возможность применения данного алгоритма к решению задач минимизации и максимизации (на примере задачи минимизации).

Эксперимент 1

В данном эксперименте рассматривалась следующая функция:

$$F(x) = -e^{-0.1 \cdot x} \cdot \sin^2 x \quad (8)$$

$$F'(x) = -e^{-0.1x} (\sin 2x - 0.1 \sin^2 x) \quad (9)$$

при $F'(x) = 0$

$x = \pi k$ – точки максимума

$x = \text{arctg}(20) + \pi k$ – точки минимума

График функции на отрезке от -107 до 5 с шагом 0.01 представлен на рис. 1.

На данном промежутке 36 точек минимума, проверим, сможет ли рой найти минимальное значение функции. Для этого в рамках эксперимента было проведено несколько серий вычислений. Значения гиперпараметров для данного эксперимента следующие:

- минимальная и максимальная скорость движения частицы 0.05 ;
- коэффициент инерции 1.3 ;
- коэффициент учета индивидуального опыта 2 ;
- коэффициент учета группового опыта 3 .

Первая серия вычислений заключалась в следующем: 25 раз генерировали рой размером 100 , количество шагов роя (время жизни роя) равнялось 1000 . В табл. 1 приведены полученные результаты.

При такой конфигурации роя он может найти в 21 из 25 случаев минимальное значение функции.

Проведем следующую серию экспериментов и уменьшим размер роя в 10 раз, увеличив время жизни в 10 раз, что в теории должно компенсировать уменьшение размера роя, т. к. общее количество

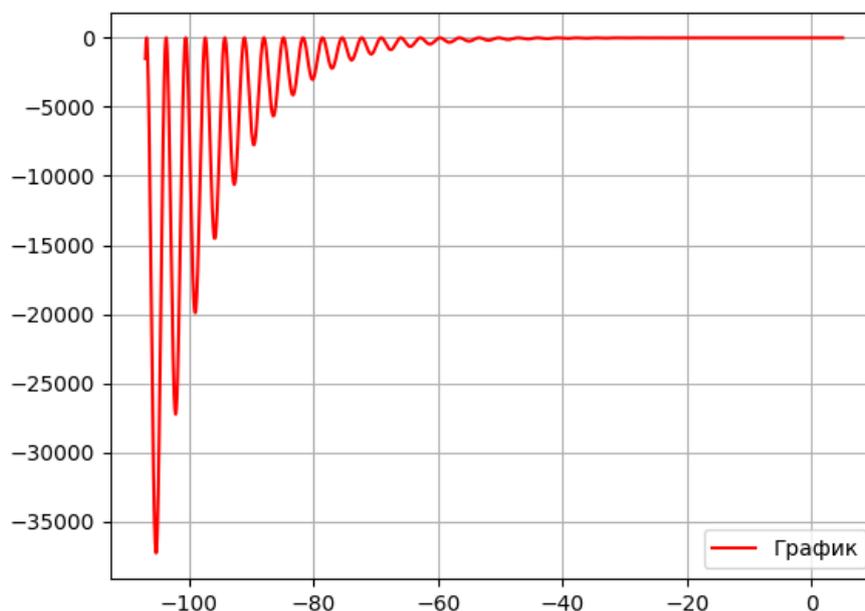


Рис. 1. График функции $F(x) = -e^{-0.1 \cdot x} \cdot \sin^2 x$ на промежутке от -107 до 5

Таблица 1

Результаты эксперимента при размере роя 100 и времени жизни 1000 шагов

Значение X	Значение функции F(X)	Значение X	Значение функции F(X)
-105.289	-37302.539	-105.263	-37269.007
-105.292	-37303.148	-105.295	-37303.058
-105.294	-37303.124	-105.295	-37303.081
-105.293	-37303.187	-105.287	-37302.011
-105.294	-37303.174	-105.292	-37303.145
-102.153	-27246.286	-105.293	-37303.198
-105.292	-37303.154	-105.293	-37303.196
-102.151	-27246.355	-105.294	-37303.164
-105.292	-37303.141	-102.142	-27243.831
-105.252	-37239.924	-105.293	-37303.195
-105.292	-37303.171	-102.148	-27246.122
-105.291	-37303.036	-102.151	-27246.353
-102.152	-27246.347		

Таблица 2

Результаты эксперимента при размере роя 10 и времени жизни 10000 шагов

Значение X	Значение функции F(X)	Значение X	Значение функции F(X)
-77.069	-2201.439	4.663	-0.625
-106.987	-1311.099	-36.182	-37.162
-106.998	-1490.949	-77.012	-2206.934
-45.600	-95.373	-17.360	-5.637
-106.999	-1493.358	-106.986	-1297.356
4.672	-0.625	-86.406	-5656.035
-95.881	-14533.024	-45.603	-95.374
-86.423	-5661.593	-106.999	-1505.936
-106.985	-1290.114	-89.587	-7754.534
-39.320	-50.881	-105.297	-37302.570
-66.081	-8.602	-106.998	-1476.886
-89.590	-7754.371	-106.996	-1453.808
-85.719	-3220.200		

шагов роя (каждая частица роя делает 1000 шагов, частиц 100, значит, шагов роя 100 000, в новом эксперименте каждая частица будет делать 10000 шагов, но частиц всего 10, общее количество шагов 100 000) остается неизменным. Результаты серии вычислений представлены в таблице 2.

В данном случае правильный ответ дал только 1 рой из 25, из чего можно сделать вывод, что компенсировать размер роя временем жизни нельзя.

Далее проведем эксперимент, в котором уменьшим относительно первой серии вычислений время жизни в 10 раз, при этом увеличив в 10 раз размер роя (табл. 3).

Таблица 3

Результаты эксперимента при размере роя 1000 и времени жизни 100 шагов

Значение X	Значение функции F(X)	Значение X	Значение функции F(X)
-105.292	-37303.187	-105.292	-37303.193
-105.283	-37299.738	-105.290	-37302.888
-105.294	-37303.172	-105.293	-37303.196
-105.292	-37303.172	-105.293	-37303.198
-105.292	-37303.191	-105.293	-37303.198
-105.295	-37302.957	-105.292	-37303.191
-105.294	-37303.159	-105.293	-37303.198
-105.292	-37303.191	-105.291	-37303.113
-105.292	-37303.151	-105.291	-37303.026
-105.294	-37303.173	-105.292	-37303.181
-105.292	-37303.189	-105.293	-37303.195
-105.293	-37303.197	-105.343	-37210.404
-105.289	-37302.749		

На рис. 2 представлен график, наглядно изображающий приведенные в таблицах 1–3 значения X и Y, на которых остановился роевой алгоритм.

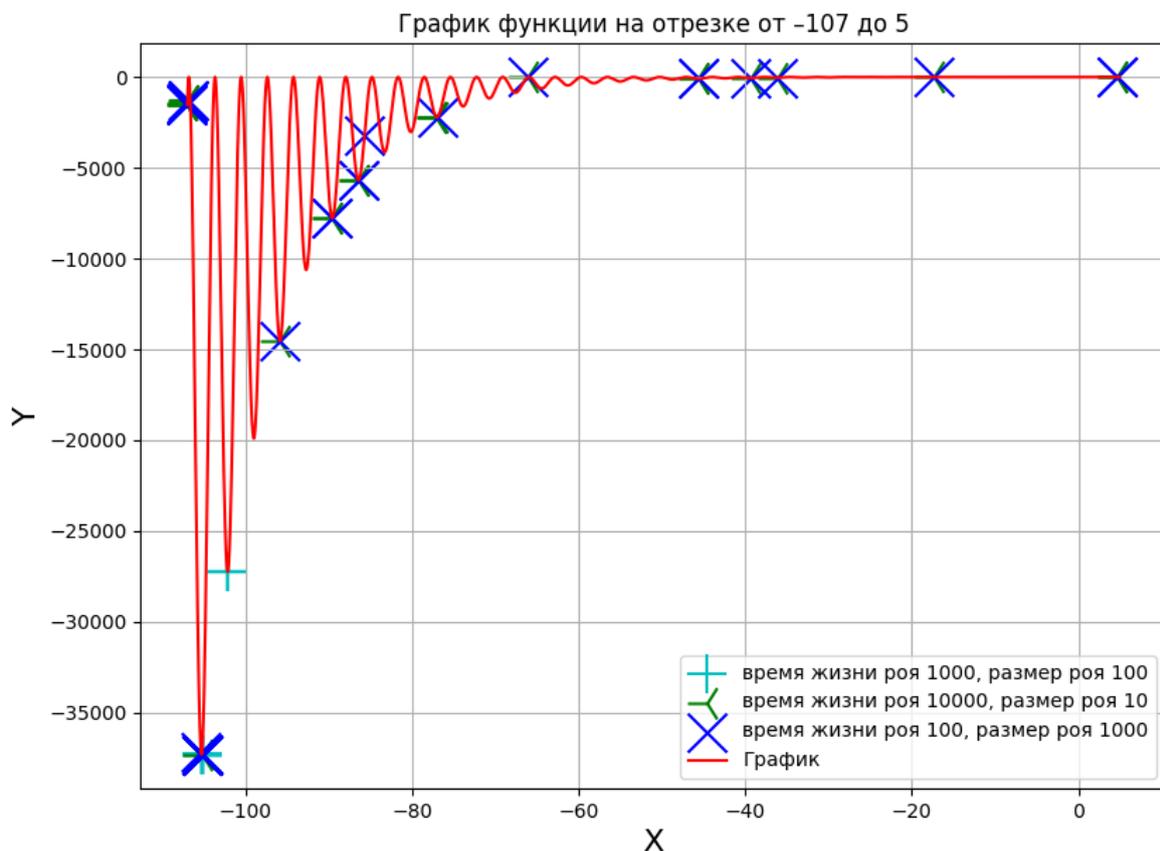


Рис. 2. Найденные роем значения минимума

Как видно из таблицы 4 и графика, почти каждый рой находит правильную точку. Но здесь необходимо учесть тот факт, что некоторые частицы роя могут изначально находиться в окрестности точки минимума.

Количество частиц роя, которые будут находиться в пределах от -106 до -104 при генерации в промежутке от -107 до 5 при равномерном распределении, представлено в таблице 4.

Таблица 4

Количество частиц в заданном промежутке в зависимости от размера роя

Размер роя	Количество частиц в заданном пределе	Размер роя	Количество частиц в заданном пределе	Размер роя	Количество частиц в заданном пределе
10	0	100	4	1000	17
10	0	100	1	1000	18
10	0	100	0	1000	16
10	2	100	1	1000	17
10	0	100	0	1000	24
10	0	100	2	1000	19
10	1	100	2	1000	23
10	0	100	3	1000	12

Как видно из данной таблицы, при размере роя в 1000 частиц как минимум 12 частиц находится в заведомо выигрышном положении, а следовательно, нельзя говорить о том, что при 1000 частицах правильную точку находит рой, т. к. это есть не что иное, как простой перебор точек.

Но при размере роя в 100 частиц можно с уверенностью сказать, что рой может решить задачу минимизации при наличии нескольких (или бесконечного числа) точек экстремума.

Эксперимент 2

Далее проведем серию вычислений на двумерных мультимодальных функциях:

$$f(x, y) = -20 \exp\left(-0.2\sqrt{0.5(x^2 + y^2)}\right) - \exp(0.5 \cos(2\pi x) + \cos(2\pi y)) + e + 20 - \text{функция Экли.} \quad (10)$$

На промежутке от -5 до 5 глобальный минимум $f(0,0) = 0$ (рис. 3).

На рис. 3 изображен график функции.

Для данных серий вычислений минимальную и максимальную скорость установили в значение 0.1, остальные коэффициенты оставили без изменений.

В первой серии вычислений размер роя равнялся 100 частицам, а время жизни роя составляло 1000 шагов. Ниже приведена таблица 5 результатов роя, по которым видно, что алгоритм роя частиц нашел глобальный минимум функции Экли.

Как видно из результатов проведенной серии экспериментов, почти все сгенерированные рои нашли точку минимума с точностью не ниже 10^{-2} . Но стоит учесть, что 100 частиц достаточно много и некоторые частицы роя могли изначально находиться в окрестности точки ноль, поэтому уменьшим количество частиц в рое до 15, оставив время жизни роя без изменения (табл. 6).

Как видно из проведенной серии вычислений, количество роев, которые нашли точку с точностью 10^{-2} , снизилось, но тем не менее шанс изначального попадания частицы роя в окрестность точки минимума все еще достаточно велик, поэтому уменьшим размер роя до 10. Результаты вычислений представлены в таблице 7 [9].

График найденных роем значений, приведенных в таблицах 5–7, представлен на рис. 4.

Как видно, уже при 10 частицах в рое количество роев, давших правильный ответ, равняется 3, среди которых лишь один рой, нашедший точку минимума с точностью 10^{-2} . При этом в данном варианте частица с очень низкой вероятностью изначально находится в окрестности точки $(0,0)$. Из этого можно сделать вывод, что количество частиц в рое является ключевым параметром, определяющим успешность работы алгоритма. Можно с уверенностью сказать, что использование роевого алгоритма для решения задачи минимизации функции такого класса возможно.

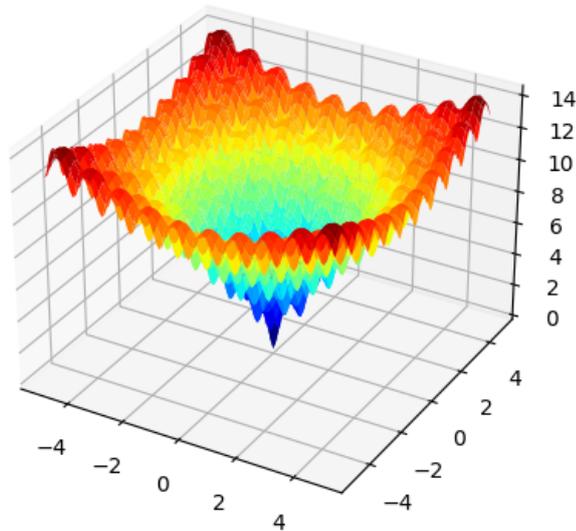


Рис. 3. График функции Экли на промежутке от -5 до 5

Таблица 5

Результаты эксперимента при размере роя 100 и времени жизни 1000 шагов

Значение X1	Значение X2	Значение функции F(X)	Значение X1	Значение X2	Значение функции F(X)
0.0023	0.004	0.014	-0.002	0.004	0.013
-0.007	0.021	0.078	0.011	0.003	0.037
0.003	-0.0001	0.008	-0.005	-0.006	0.023
0.002	-0.017	0.057	0.005	0.004	0.019
-0.002	0.002	0.009	0.001	-0.001	0.004
0.001	-0.001	0.004	-0.004	0.003	0.015
-0.002	-0.001	0.005	0.003	-0.008	0.027
0.002	0.001	0.005	-0.010	0.001	0.033
-0.002	0.005	0.017	.003	0.005	0.018
-0.007	-0.006	0.028	0.013	0.008	0.049
-0.001	0.003	0.010	-0.004	0.004	0.016
-0.001	0.005	0.014	0.001	0.004	0.011
0.001	0.006	0.017			

Эксперимент 3

Далее посмотрим, как себя поведет рой, если точек минимума будет несколько.

$$f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \quad \text{— функция Химмельброу.} \quad (11)$$

На промежутке от -5 до 5 глобальные минимумы (рис. 5):

$$f(3.0; 2.0) = 0$$

$$f(-2.805118; 3.131312) = 0$$

$$f(-3.779310; -3.283186) = 0$$

$$f(3.584428; -1.848126) = 0.$$

Как и в прошлом эксперименте, вычисления начнем с роя размером 100 частиц. Результаты вычислений представлены в таблице 8.

Таблица 6

Результаты эксперимента при размере роя 15 и времени жизни 1000 шагов

Значение X1	Значение X2	Значение функции F(X)	Значение X1	Значение X2	Значение функции F(X)
0.0001	-0.016	0.052	-0.006	-0.046	0.187
-0.043	0.025	0.207	0.007	-0.025	0.09
-0.003	0.013	0.042	-0.011	0.0111	0.051
0.010	-0.018	0.068	-0.002	0.00005	0.006
0.029	-0.074	0.388	-0.048	0.003	0.197
-0.003	0.131	0.771	-0.022	-0.002	0.077
0.011	-0.029	0.112	0.006	0.011	0.04
-0.080	0.097	0.739	-0.016	0.004	0.053
-0.0377	-0.045	0.255	-0.118	0.105	1.012
-0.001	0.029	0.103	0.0318	-0.022	0.149
-0.051	0.022	0.235	-0.020	0.012	0.079
0.0152	0.022	0.094	-0.0004	0.003	0.012
0.012	-0.003	0.04			

Таблица 7

Результаты эксперимента при размере роя 10 и времени жизни 1000 шагов

Значение X1	Значение X2	Значение функции F(X)	Значение X1	Значение X2	Значение функции F(X)
-0.023	-0.002	0.078	0.0860	-0.801	3.075
0.0196	4.910	10.225	4.923	0.007	10.184
-0.958	-1.979	5.403	4.993	-2.986	11.222
1.078	2.056	5.834	-0.050	0.022	0.236
1.005	4.925	10.32	-2.081	4.937	10.887
0.0182	-4.914	10.214	-1.027	-4.918	10.362
-4.967	4.922	12.743	0.906	-3.599	9.876
-0.008	-4.950	10.136	-4.923	0.007	10.183
-4.924	-1.028	10.348	0.139	-0.048	0.915
4.926	-0.0273	10.194	-4.91	-1.975	10.757
0.03	-4.947	10.162	-4.917	-2.035	10.782
0.0293	0.013	0.117	-4.945	-0.943	10.348
-0.069	1.954	5.009			

Как видно, рой справился с задачей, но, помня предыдущий эксперимент, проведем еще две серии вычислений с размером роя 15 и 10; результаты представлены в таблице 9.

На рис. 6 представлены приведенные в таблицах 8–9 значения, которые были найдены роем.

Размер роя был уменьшен до 15, а потом и до 10, как и при работе с предыдущей функцией. Но можно заметить, что с данной функцией рой справился лучше. Количество роев, которые дали правильный ответ с точностью 10^{-2} , в 10 раз больше. Данный факт можно объяснить тем, что шанс попасть в окрестность точки минимума выше в 4 раза. Поэтому возьмем другую функцию, в которой расширим область поиска в два раза по каждой координате.

Эксперимент 4

$$-\left| \sin x \cdot \cos y \cdot \exp \left(\left| 1 - \frac{\sqrt{x^2 + y^2}}{\pi} \right| \right) \right| \quad - \text{функция Хольдера.} \quad (12)$$

На промежутке от -10 до 10 глобальный минимум (рис. 7):

$$f(8.05502; 9.66459) = -19.2085$$

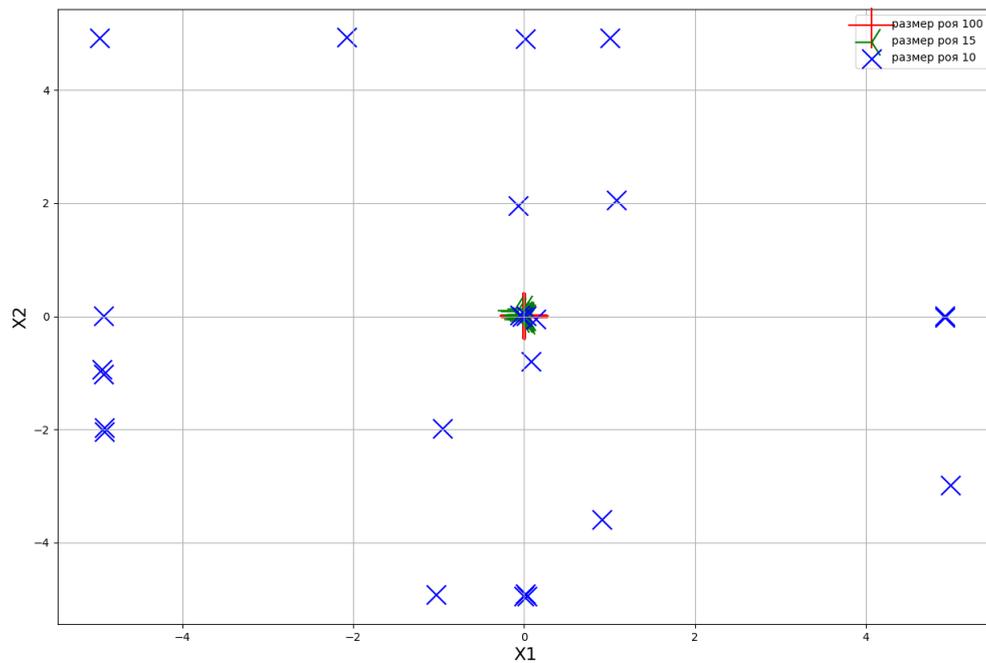


Рис. 4. Найденные роем значения минимума для функции Экли. Время жизни роя 1000

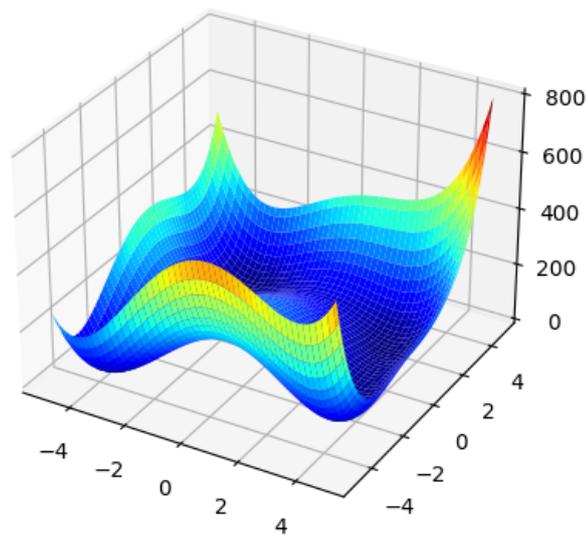


Рис. 5. Функция Химмельброу на промежутке от -5 до 5

$$f(-8.05502; 9.66459) = -19.2085$$

$$f(8.05502; -9.66459) = -19.2085$$

$$f(-8.05502; -9.66459) = -19.2085.$$

Значение гиперпараметров роевого алгоритма оставим неизменным по сравнению с предыдущим экспериментом. Оценим, как рой размером 100, 15, 10 справится с поиском точки минимума. В правом столбце в данном случае будет погрешность, вычисленная по формуле:

$$\Delta = |f(x) - f(x_r)|, \quad (13)$$

где $f(x)$ — минимальное значение функции Хольдера, а $f(x_r)$ — значение функции Хольдера в точке, которую указал рой.

Таблица 8

Результаты эксперимента при размере роя 100 и времени жизни 1000 шагов

Значение X1	Значение X2	Значение функции F(X)	Значение X1	Значение X2	Значение функции F(X)
2.992	2.014	0.004	2.998	2.002	0.0002
3.006	2.009	0.004	3.0	2.01	0.001
-2.794	3.105	0.031	3.004	1.997	0.0005
2.988	2.014	0.005	3.584	-1.856	0.0009
2.986	2.004	0.007	3.009	1.973	0.010
-2.799	3.124	0.003	2.996	2.018	0.005
2.993	2.010	0.002	3.018	1.973	0.015
3.014	1.974	0.012	-3.796	-3.278	0.017
3.584	-1.829	0.006	3.0	2.001	0.000019
-2.806	3.138	0.002	3.586	-1.846	0.0002
3.574	-1.841	0.005	2.995	2.003	0.0007
3.584	-1.842	0.0006	-2.776	3.099	0.0687
3.003	1.999	0.0002			

Таблица 9

Результаты экспериментов при размере роя 15 и 10 и времени жизни 1000 шагов

Значение X1	Значение X2	Значение функции F(X)	Значение X1	Значение X2	Значение функции F(X)
Размер роя 15			Размер роя 10		
3.485	-1.755	0.557	3.3256	1.630	3.890
3.572	-1.852	0.008	3.065	1.891	0.212
3.001	1.979	0.007	-2.833	3.143	0.032
3.016	1.935	0.059	-2.788	3.245	0.553
3.054	1.962	0.091	-3.209	3.381	8.668
3.521	-1.815	0.208	-3.088	2.142	30.678
3.536	-1.831	0.119	3.428	-1.717	1.327
-3.785	-3.319	0.052	3.009	2.003	0.004
-2.778	3.147	0.033	-2.793	3.132	0.005
-2.805	3.141	0.004	-3.767	-3.28	0.008
3.057	1.891	0.191	-2.81	3.152	0.018
2.96	2.088	0.124	3.566	-1.762	0.112
3.034	2.075	0.194	-2.821	3.106	0.033
3.263	1.316	4.95	3.477	1.227	9.453
3.027	2.025	0.050	3.594	-1.867	0.009
-2.789	3.123	0.011	-3.784	-3.177	0.497
-2.845	3.117	0.061	-2.751	3.103	0.123
-2.906	3.155	0.364	-4.285	-4.178	48.169
-2.775	3.136	0.03	3.672	-1.946	0.504
3.0102	1.974	0.01	-2.741	3.145	0.139
2.972	2.015	0.024	-2.37	3.791	27.536
3.597	-1.835	0.013	-2.825	3.149	0.0247
3.506	-1.74	0.416	-2.748	3.153	0.125
3.016	1.966	0.019	-2.833	3.133	0.026
-2.826	3.154	0.034	3.552	-1.815	0.061

Ниже представлена таблица 10 с результатами серии вычислений.

Значения X1 и X2 из таблицы 10 представлены на графике на рис. 8.

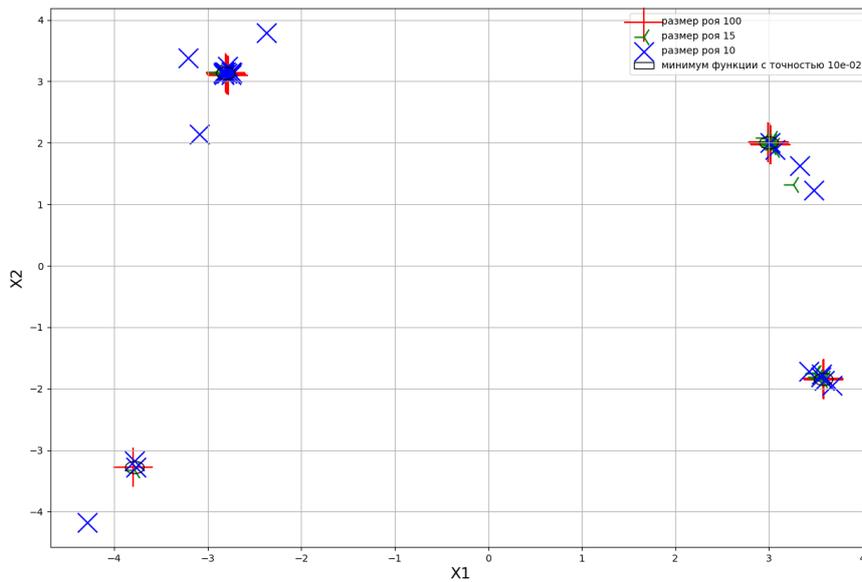


Рис. 6. Найденные роем значения минимума для функции Химмельброу. Время жизни роя 1000

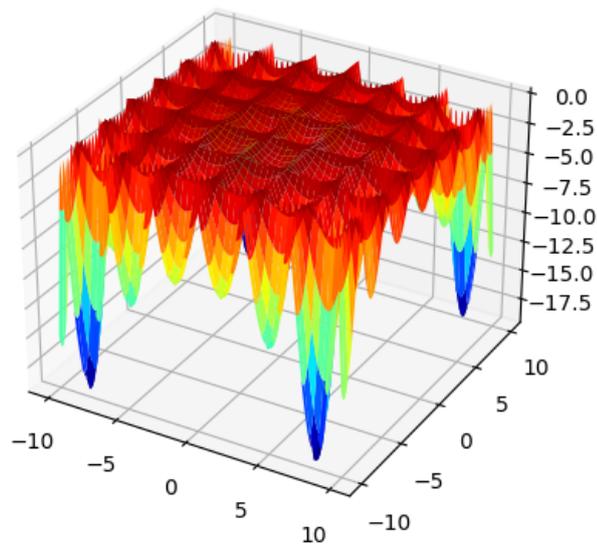


Рис. 7. График функции Хольдера на промежутке от -10 до 10

По графику можно отметить, что не нашедшие точку минимума функции рои остановились в локальных минимумах. Как видно из проведенной серии вычислений, при размере роя 10 алгоритм не смог найти точку минимума функции, но при размере 15 алгоритм справился с поставленной задачей. На основании этого можно сделать вывод, что данный алгоритм применим для решения задач такого класса.

Заключение

Роевые алгоритмы — относительно новое направление в решении задач оптимизации, и вопрос о возможности их применения, и доказательная база сходимости для большого спектра задач до сих пор остаются открытыми. Проведя серию вычислений и проанализировав результаты, можно сделать вывод, что роевые алгоритмы позволяют найти глобальный экстремум даже достаточно сложных функций, имеющих множество точек экстремума или имеющих значительное количество локальных экстремумов.

Таблица 10

Результаты экспериментов при размере роя 100, 15 и 10 и времени жизни 1000 шагов

Значение X1	Значение X2	Значение функции F(X)	Значение X1	Значение X2	Значение функции F(X)
Размер роя 100			Размер роя 15		
-8.051	9.661	-19.208	-8.196	3.293	-5.6995
-8.044	-9.639	-19.2006	-4.877	9.71	-11.0661
8.054	-9.655	-19.2075	4.903	9.668	-11.0498
-8.052	9.671	-19.2079	1.59	9.724	-8.09081
-8.044	-9.679	-19.2052	-8.090	-6.396	-9.4713
-8.067	-9.648	-19.2042	8.090	-6.45	-9.5001
8.06	-9.662	-19.2081	4.84	-9.693	-11.0681
8.042	9.664	-19.2068	8.072	9.753	-19.1255
8.063	9.656	-19.2072	7.99	-9.617	-19.1433
8.061	-9.682	-19.2051	-4.946	9.703	-11.0224
8.053	9.642	-19.2034	4.93	6.551	-4.7095
-8.061	-9.67	-19.2077	-8.048	-9.67	-19.2077
8.055	-9.672	-19.208	8.019	9.66	-19.1952
-8.058	9.685	-19.2042	8.073	9.658	-19.205
8.057	9.692	-19.20058	4.836	9.681	-11.0652
8.058	-9.657	-19.2078	-8.044	-9.638	-19.2005
8.050	-9.654	-19.2072	8.073	9.716	-19.1781
-4.841	-9.701	-11.0686	-8.241	3.321	-5.6695
-8.073	-9.649	-19.2029	-8.03	9.652	-19.2007
-8.044	-9.662	-19.2073	7.936	9.583	-19.0031
-8.092	9.635	-19.1864	4.942	3.297	-2.3451
8.071	9.681	-19.2034	-8.076	9.669	-19.2041
8.063	-9.676	-19.2066	-8.146	9.575	-19.0486
-8.055	9.68	-19.2063	-4.937	9.777	-10.9961
-4.837	-9.734	-11.0621	-8.167	-3.289	-5.7076
Размер роя 10					
9.981	-5.987	-7.547	8.079	6.416	-9.4835
9.984	9.761	-15.6797	1.538	-9.673	-8.0527
8.042	6.338	-9.3932	1.655	-9.764	-8.0852
4.961	3.344	-2.3455	-8.174	-6.526	-9.4641
-4.816	9.726	-11.0584	-9.985	-0.646	-3.7722
9.954	-9.934	-14.2653	8.210	6.492	-9.4393
-1.61	9.74	-8.0939	8.013	-6.593	-9.4079
1.606	9.73	-8.0939	-8.116	-6.436	-9.4939
-4.838	-9.695	-11.0678	2.021	7.303	-1.9344
-1.836	-0.107	-1.4527	4.98	0.024	-1.731
-9.966	9.517	-15.1618	9.997	-9.981	-15.1753
-5.7	-8.292	-2.11697	8.237	6.588	-9.3457
4.928	-6.573	-4.7064	14.502		

Однако особенностью применения такого алгоритма является необходимость правильного подбора размера роя, т. к. при слишком маленьком рое он не способен найти точку экстремума, а при слишком большом — алгоритм роя частиц превращается в простой подбор и поиск точки экстремума.

Кроме того, остается неисследованным вопрос о величине таких параметров алгоритма, как коэффициенты инерции, учета индивидуального опыта, учета группового опыта. Данные параметры были установлены экспериментальным путем и не имеют под собой математического обоснования. Как следствие, данная проблема является перспективной и требует детального исследования.

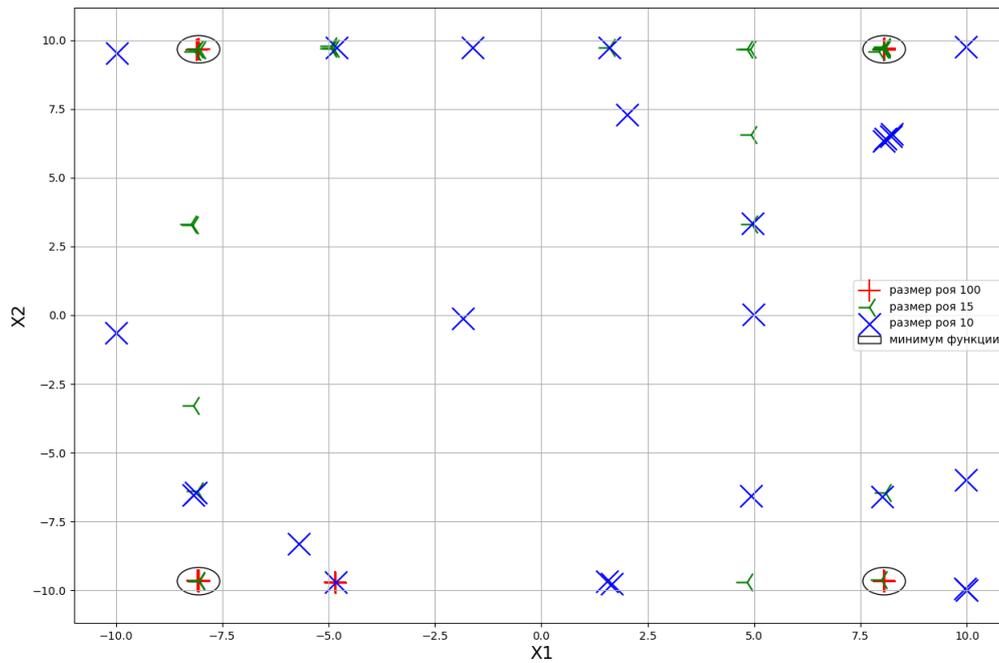


Рис. 8. Найденные роем значения минимума для функции Хольдера. Время жизни роя 1000

ЛИТЕРАТУРА

1. Beni G., Wang J. Swarm Intelligence in Cellular Robotic Systems. *Proceed. NATO Advanced Workshop on Robots and Biological Systems*. Tuscany, Italy, June 26–30 (1989). Режим доступа: <http://www-users.york.ac.uk/~jo115/paper-reviews/SwarmIntelligenceinCellularRoboticSystems.pdf>.
2. Зайцев А. А., Курейчик В. В., Полупанов А. А. Обзор эволюционных методов оптимизации на основе роевого интеллекта. *Известия ЮФУ. Технические науки*. 2010;12. Режим доступа: <https://cyberleninka.ru/article/n/obzor-evolyutsionnyh-metodov-optimizatsii-na-osnove-roevogo-intellekta>.
3. Карпенко А. П. Популяционные алгоритмы глобальной оптимизации. Обзор новых и малоизвестных алгоритмов. *Информационные технологии. Приложение*. 2012;7:1–32.
4. Водолазский И. А. и др. Роевой интеллект и его наиболее распространенные методы реализации. *Молодой ученый*. 2017;4(138):147–153.
5. Dazahra M. N., Elmariami F., Belfqih A., Boukherouaa J. Optimal Location of SVC Using Particle Swarm Optimization and Voltage Stability Indexes. *International Journal of Electrical and Computer Engineering*. 2016;6(6):2581–2588.
6. Eberhart R. C., Shi Y. Particle Swarm Optimization: Developments, Applications and Resources. *Congress on Evolutionary Computation*. Seoul, South Korea. 2001;81–86.
7. Poli R. An Analysis of Publications on Particle Swarm Optimisation Applications. *Technical Report CSM-469*. Department of Computer Science, Univ. of Essex, UK. 2007.
8. Poli R. Analysis of the Publications on the Applications of Particle Swarm Optimisation. *Journal of Artificial Evolution and Applications*. 2008;1–10. DOI: 10.1155/2008/685175.
9. Смородинов А. Д., Галкин В. А., Гавриленко Т. В. Применимость алгоритмов роевого интеллекта для решения задач минимизации функции «Экли». *Наука и инновации XXI века : Сборник статей по материалам VIII Всероссийской конференции молодых ученых*. В 4-х томах. Сургут, 23 декабря 2021 года. Сургут: Сургутский государственный университет; 2022. С. 23–27.