

ИССЛЕДОВАНИЕ ПОПУЛЯЦИОННОГО АЛГОРИТМА ПОИСКОВОЙ ОПТИМИЗАЦИИ, ВДОХНОВЛЕННОГО ПОВЕДЕНИЕМ ЛЕТУЧИХ МЫШЕЙ

А.А. Зуева

anna.926@outlook.com

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Для решения недифференцируемых мультимодальных и овражных задач минимизации с высокой размерностью пространства поиска наиболее эффективными являются метаэвристические алгоритмы. Представлено исследование одного из таких методов — алгоритма, вдохновленного поведением летучих мышей. Метод реализован с помощью объектно-ориентированного языка программирования C++ с выделением класса летучих мышей и контейнерного класса популяции. Анализ проведен на основе овражной и многоэкстремальной функций по данным о сходимости алгоритма и лучших найденных значениях при различных размерностях вектора варьируемых параметров. Выполнено сравнение исследуемого метода с модификациями алгоритма эволюционной стратегии по наименьшим найденным решениям и скоростям сходимости алгоритмов при разных размерностях. Сделаны выводы, что алгоритм летучих мышей высокоэффективен и на овражных, и на многоэкстремальных функциях различных размерностей.

Ключевые слова

Метаэвристический поиск, алгоритм летучих мышей, глобальная оптимизация, популяционный алгоритм, исследование эффективности, групповой интеллект, структурная оптимизация, минимизация

Поступила в редакцию 24.06.2022

© МГТУ им. Н.Э. Баумана, 2022

Введение. Традиционные подходы к решению задач минимизации бывают неэффективными при решении современных инженерных задач. Современные инженерные системы слишком сложны для применения традиционных методов оптимизации, которые по причине использования поисковых алгоритмов¹ на основе градиента позволяют находить локальный оптимум вместо глобального. Весомые преимущества по сравнению с традиционными подходами в различных аспектах демонстрируют новые методы поисковой оптимизации, которые также называют популяционными или метаэвристическими.

¹ Алгоритмы поисковой оптимизации — это алгоритмы для нахождения экстремума (минимума или максимума) целевой функции в некоторой области конечномерного векторного пространства.

Метаэвристические алгоритмы были разработаны для решения недифференцируемых мультимодальных¹ и овражных задач² с высокой размерностью пространства поиска. Они не зависят от информации о градиенте, могут работать и с дискретными, и с непрерывными переменными, выполняют глобальный поиск, что позволяет найти близкие к оптимальным решения сложных задач. Эти подходы основаны на методах без производных и вдохновлены природой [1].

Одним из таких методов является алгоритм, вдохновленный поведением летучих мышей и предложенный Янгом в 2010 г. [2]. Здесь для поиска используется эхолокационное поведение летучих мышей для поиска оптимального решения. Иначе говоря, алгоритм базируется на способности летучих мышей к эхолокации и их поведении во время ее использования [3].

Постановка задачи. Рассмотрим детерминированную непрерывную задачу глобальной безусловной минимизации:

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

где $f(X) \in R^1$ — скалярная целевая функция $|X|$ переменных (критерий оптимальности); $f(X^*) = f^*$ — искомый глобальный минимум целевой функции; $X = (x_1, x_2, \dots, x_{|X|})$ — $|X|$ -мерный вектор варьируемых параметров в пространстве $R^{|X|}$. Кроме того, задан гиперкуб $\Pi = \{X \mid x^- \leq x_i \leq x^+, i = 1, 2, \dots, |X|\}$ соответствующей области, в пределах которой будут инициированы особи начальной популяции.

Цель данной работы — исследование эффективности популяционного алгоритма поисковой оптимизации, вдохновленного поведением летучих мышей, для решения указанной задачи при различных целевых функциях.

Описание алгоритма. Большинство летучих мышей обладают способностью к эхолокации, которую они используют для ориентирования в пространстве и поиска добычи. Параметры лоцирующего звукового импульса у разных мышей зависят от их охотничьих стратегий.

Для измерения расстояния до добычи или препятствия при приеме сигнала мыши оперируют с различными громкостями и временем прихода этого сигнала. Для анализа пространства мыши используют эффект Доплера.

¹ Мультимодальная функция — экстремальная задача, имеющая несколько или неизвестное число локальных экстремумов.

² Овражная функция — функция, для которой незначительные изменения одних переменных ведут к быстрым изменениям значений функции, однако вдоль других направлений функция меняется медленно.

Чувствительность этого механизма такова, что мыши могут распознавать даже насекомых по частоте колебаний их крыльев.

Мыши применяют и другие органы чувств, однако алгоритм, вдохновленный поведением летучих мышей, моделирует именно эффект эхолокации. Алгоритм базируется на трех главных параметрах эхолокации: длительности, частоте и громкости. Большинство летучих мышей используют короткие частотно-модулированные сигналы. Ширина полосы сигнала варьируется в зависимости от вида летучей мыши и часто увеличивается с помощью использования большего количества гармоник. Длительность каждого из импульсов эхолокации составляет от 8 до 10 мс, а частота ультразвуковых колебаний в импульсе может меняться от 25 до 150 кГц. Длина волны в таком диапазоне частоты примерно совпадает с обычными размерами добычи летучих мышей и составляет 2...14 мм. Обычно мышь испускает 10...20 имп./с, при приближении к добыче это число может увеличиваться до 200 имп./с: чем ближе препятствие, тем короче импульсы. Громкость ультразвуковых колебаний зависит от цели испускания импульсов летучей мышью и варьируется от 50 до 120 дБ [4].

Алгоритм строится на некоторых приближениях поведения летучих мышей [5–8]. Летучие мыши, они же агенты популяции, перемещаются случайным образом со скоростью v_i и соответствуют положению X_i , частота импульсов — ω_i , длина волны λ_i и громкость a_i для поиска добычи (наименьшего значения целевой функции) постоянно меняются. Алгоритм построен на предположении, что громкость изменяется от наибольшего значения до наименьшего a_{\min} . Более высокие частоты имеют короткие длины волн и распространяются на меньшее расстояние. Для простоты можно считать, что $\omega \in [0; \omega_{\max}]$. Значение частоты импульсов находится в диапазоне $[0; 1]$, где 0 означает отсутствие импульсов вообще, а 1 — максимальную скорость излучения импульсов. Летучие мыши также могут самостоятельно регулировать скорость излучения импульсов $r \in [0; 1]$ в зависимости от близости к наилучшему решению.

Шаг 1. Инициализация популяции летучих мышей

Популяцию S инициализируем присвоением числа ее агентов $|S|$ и установкой положения X и скорости v каждого из них. Начальные значения компонент $x_i \in X, i = 1, 2, \dots, |X|$ полагаем равномерно распределенными в интервале $[x^-; x^+]$, заданном областью Π .

Шаг 2. Инициализация параметров эхолокации

Частота колебаний ω и громкость сигналов a варьируются в интервале $0 \dots 1, 0$, задаем их случайным образом для каждой мыши. Частоту испускаемых импульсов r выбираем случайно в промежутке $0,5 \dots 1, 0$:

$$\omega_{\min} \leq \omega_i \leq \omega_{\max};$$

$$a_{\min} \leq a_i \leq a_{\max};$$

$$\omega_{\min} = 0,0, \omega_{\max} = 1,0, a_{\min} = 0,0, a_{\max} = 1,0, r_{\min} = 0,5, r_{\max} = 1,0.$$

Шаг 3. Поиск лучшего решения

Из всех агентов популяции выбираем летучую мышь с наименьшим значением целевой функции.

Шаг 4. Миграция агентов

Выполняем перемещение всех агентов на один шаг в соответствии с используемой миграционной процедурой:

$$\omega'_i = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \cdot U(0; 1);$$

$$v'_i = v_i + \omega'_i (X_i - X^{**});$$

$$X'_i = X_i - v_i,$$

где $U(0; 1)$ — случайное число с равномерным распределением от 0 до 1 [4].

Шаг 5. Локальный поиск

Локальный поиск реализуем с вероятностью ξ_i^r в окрестности лучшего агента популяции:

$$X'_i = X_i + a'' \cdot U(-1; 1),$$

где a'' — среднее значение громкостей всех мышей, $U(-1; 1)$ — случайное число с равномерным распределением от -1 до 1 .

Шаг 6. Глобальный поиск

В окрестности текущего агента популяции случайным образом генерируем решение X . Если значение функции в новой точке меньше минимального, найденного за прошлые итерации, то с вероятностью ξ_i^r принимаем новое решение в качестве текущего.

Шаг 7. Поиск глобально лучшего решения

Реализуем поиск наименьшего значения среди всех агентов популяции.

Шаг 8. Выбор случайного решения в окрестности текущего лучшего и эволюция параметров

Случайным образом в окрестности агента популяции с наименьшим значением целевой функции генерируем новое решение

$$X'_i = X_i + U(-1; 1).$$

С вероятностью ξ_i^a оно принимается. Реализуется эволюция параметров a и r :

$$a'_i = 0,9a_i;$$

$$r'_i = r_0 (1 - e^{(1-0,9i)}).$$

Шаг 9. Окончание итераций

Выполняем проверку удовлетворения условиям окончания итераций: превышение предельного числа итераций, когда разница между лучшим значением на прошлой и текущей итерациях не превышает заданного значений. Если оно выполнено, в качестве решения принимается текущая точка X^{**} . В противном случае происходит переход к шагу 4.

Исследование эффективности алгоритма. Тестовые функции. Для исследования эффективности алгоритма были использованы овражная и многоэкстремальная функции.

- Функция Розенброка (овражная функция):

$$f(X) = \sum_{i=1}^{|X|-1} ((1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2) \rightarrow \min_x.$$

Функция имеет глобальный минимум в точке $X = 1_{|X|}$ с наименьшим значением $f^* = 0$ [3], $|X|$ — размерность вектора варьируемых параметров.

- Функция Экли (многоэкстремальная функция):

$$f(X) = -20 \exp\left(-0,2 \sqrt{\frac{1}{|X|} \sum_{i=1}^{|X|} x_i^2}\right) - \exp\left(\frac{1}{|X|} \sum_{i=1}^{|X|} \cos(2\pi x_i)\right) + 20 + e \rightarrow \min_x.$$

Функция имеет глобальный минимум в точке $X = 0_{|X|}$ с наименьшим значением $f^* = 0$ [3], $|X|$ — размерность вектора варьируемых параметров.

Вычислительный эксперимент. Программная реализация алгоритма выполнена на объектно-ориентированном языке C++ [9].

Число агентов популяции $|S| = 300$. В качестве условия окончания поиска использовано условие стагнации итерационного процесса, когда на протяжении не более, чем $t = 150$ итераций разница между лучшими значениями на прошлой и текущей итерациях $\Delta f \leq 10^{-7}$. Для области Π использованы значения $x^- = -3$ и $x^+ = 3$ по каждой компоненте вектора X . Размерность вектора варьируемых параметров $|X| \in \{2, 4, 8, 16, 32\}$.

Эффективность алгоритма исследована методом мультистарта для числа стартов $N = 100$ по следующим критериям:

\tilde{f}^* — лучшее найденное значение;

T — число вычислений целевой функции, которое потребовалось для достижения полученного значения;

t — число итераций на лучшем старте;

$\Omega_\epsilon = t/N$ оценка вероятности достижения требуемой точности $\epsilon(\tilde{f}^* < \epsilon)$ для $\epsilon = 0,1$ и $\epsilon = 0,01$.

Результаты эксперимента на функции Розенброка. На гистограмме найденных лучших значений функции Розенброка в зависимости от размерно-

сти вектора варьируемых параметров (рис. 1) видно, что результаты, которые находит алгоритм, ухудшаются с ростом размерности вектора варьируемых параметров, а при размерностях 2 и 4 алгоритм работает лучше всего: найденные лучшие значения максимально близки к наименьшему значению.

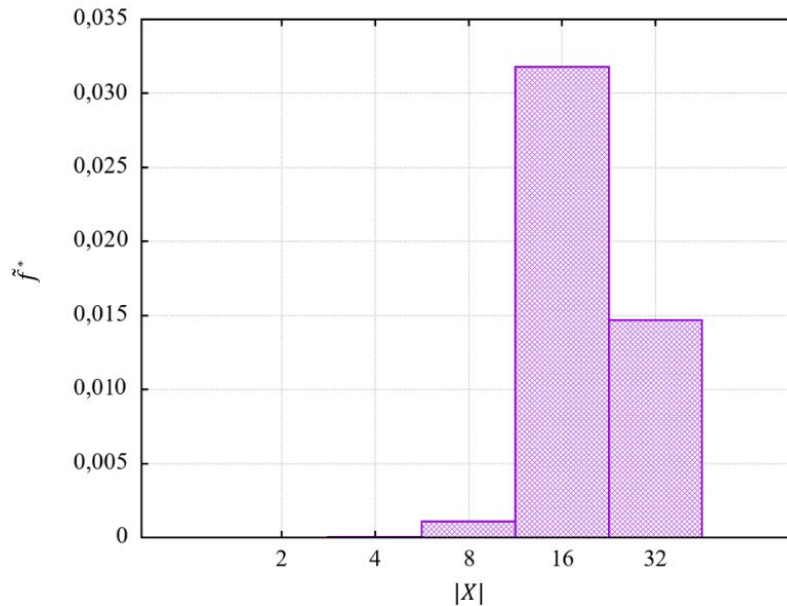


Рис. 1. Гистограмма найденных лучших значений для функции Розенброка в зависимости от размерности вектора варьируемых параметров

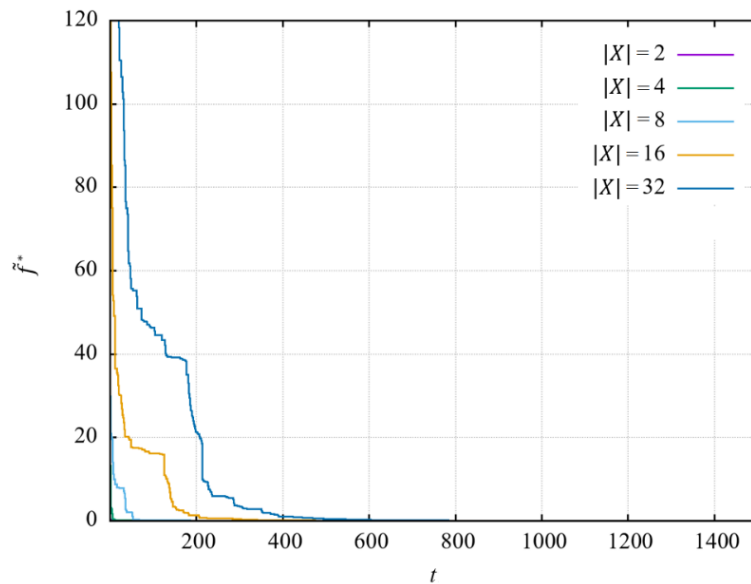


Рис. 2. График сходимости алгоритма для функции Розенброка при различных размерностях вектора варьируемых параметров

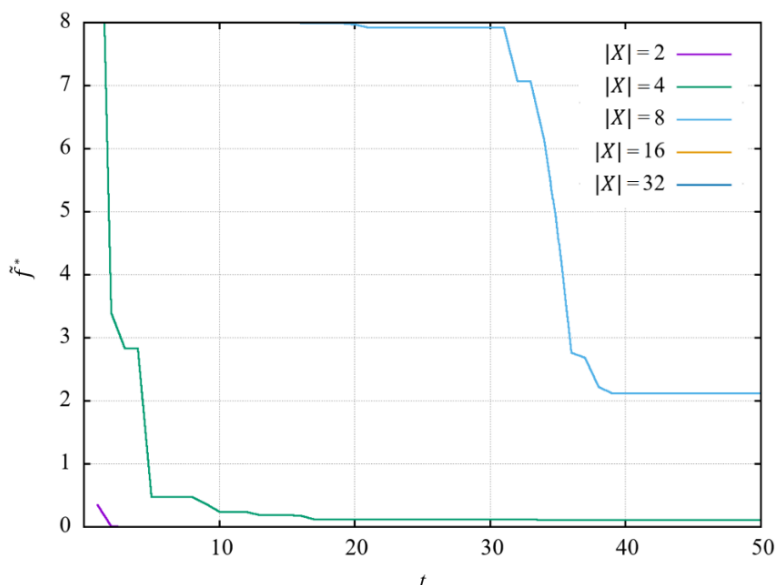


Рис. 3. Увеличенная область графика сходимости алгоритма для функции Розенброка при различных размерностях вектора варьируемых параметров

По графикам сходимости алгоритма для различных размерностей (рис. 2, 3) можно сделать вывод, что скорость сходимости алгоритма примерно обратно пропорциональна размерности вектора варьируемого параметра: для функции Розенброка она велика на начальных итерациях, однако с увеличением итераций скорость сходимости падает и каждое следующее найденное лучшее значение все меньше отличается от предыдущего. Наилучшая скорость сходимости алгоритма достигается при размерностях 2 и 4 вектора варьируемых параметров (рис. 3).

Результаты работы алгоритма для функции представлены в табл. 1. Заметно, что число необходимых итераций растет с увеличением размерности вектора $|X|$, но этот рост не является постоянным.

Таблица 1

Результаты эксперимента для функции Розенброка

$ X $	\tilde{f}^*	T	$\Omega_{0,1}$	$\Omega_{0,01}$
2	0,000	586 400 000	1,00	1,00
4	0,000	1 100 000 000	1,00	0,99
8	0,011	1 100 000 000	1,00	0,00
16	0,031	759 300 000	0,78	0,00
32	0,014	1 300 000 000	0,83	0,00

Результаты эксперимента на функции Экли. На гистограмме найденных лучших значений функции Экли в зависимости от размерности вектора варьируемых параметров (рис. 4) видно, что результаты, которые находит алгоритм, ухудшаются с ростом размерности вектора варьируемых параметров. Алгоритм демонстрирует близкую к экспоненциальной зависимость найденного лучшего значения функции от размерности вектора $|X|$. Заметно также, что рост найденного наименьшего значения в зависимости от размерности вектора варьируемых параметров медленнее, чем аналогичный рост для функции Розенброка.

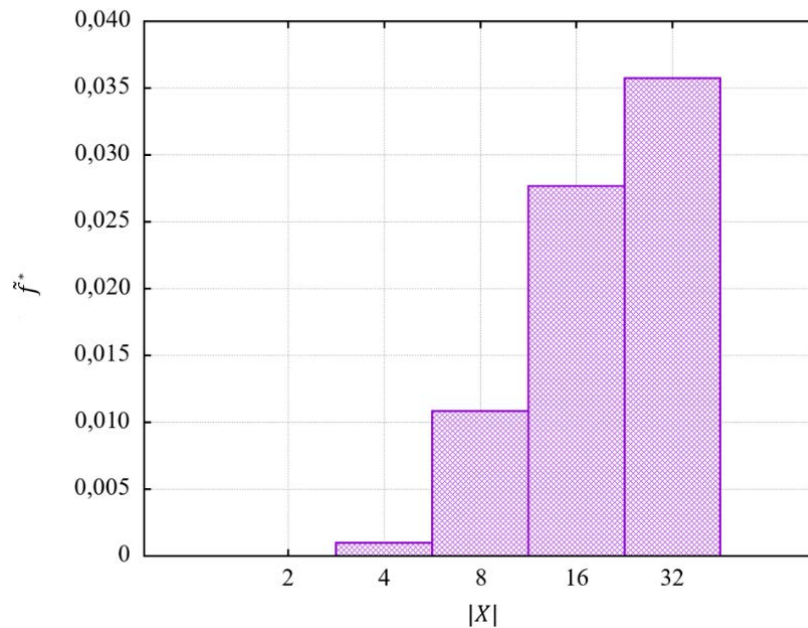


Рис. 4. Гистограмма найденных лучших значений для функции Экли в зависимости от размерности вектора варьируемых параметров

По графикам сходимости алгоритма для различных размерностей (рис. 5, 6) можно сделать вывод, что скорость сходимости алгоритма на функции Экли достаточно низкая и примерно обратно пропорциональна размерности вектора варьируемого параметра. Наилучшая скорость сходимости алгоритма достигается на начальных итерациях при размерностях 2 и 4 вектора варьируемых параметров (рис. 6), но начиная примерно с 15-й итерации для размерности 2 и 25-й итерации для размерности 4 скорость сходимости резко падает.

Результаты работе алгоритма для функции Экли представлены в табл. 2. Заметно, что число необходимых итераций растет с увеличением размерности вектора $|X|$, но этот рост значительно медленнее, чем для функции Розенброка, и не всегда является тенденцией. Оценка вероятности попадания найденных на 100 стартах лучших значений в заданную точность уменьшается при увеличении

размерности, однако при $\epsilon = 0,1$ на размерностях 16 и 32 это значение выше, чем при работе алгоритма на функции Розенброка.

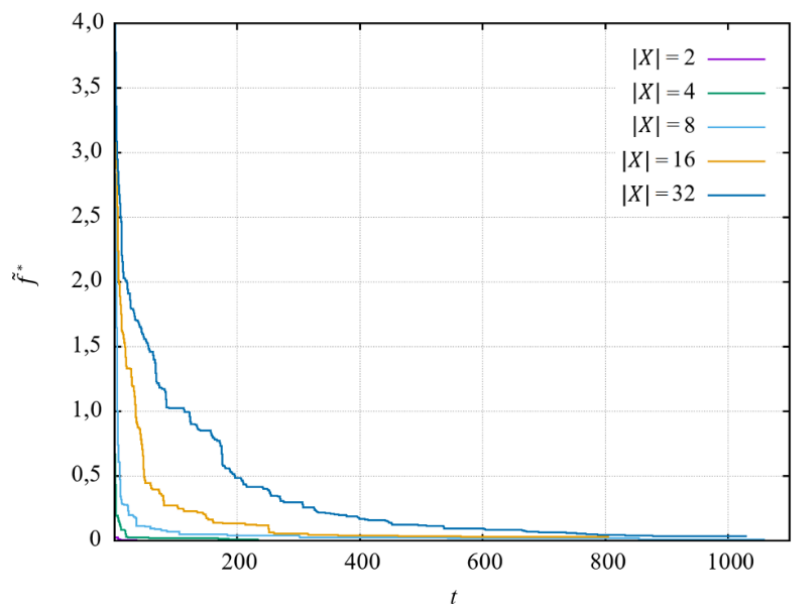


Рис. 5. График сходимости алгоритма на функции Экли для различных размерностей вектора варьируемых параметров

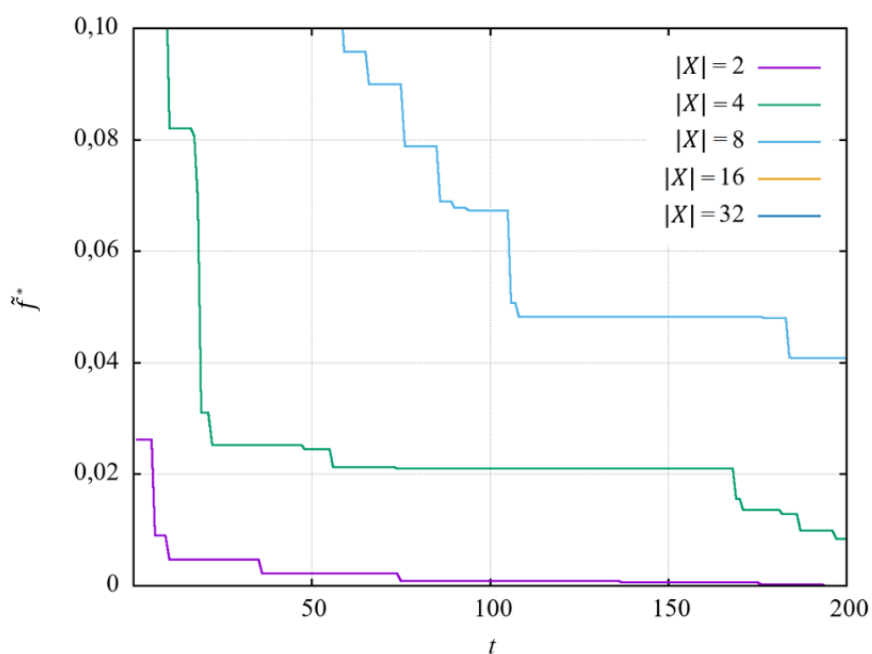


Рис. 6. Увеличенная область графика сходимости алгоритма на функции Экли для различных размерностей вектора варьируемых параметров

Результаты эксперимента на функции Экли

$ X $	\tilde{f}^*	T	$\Omega_{0,1}$	$\Omega_{0,01}$
2	0,000	189 000 000	1,00	1,00
4	0,000	479 000 000	1,00	0,86
8	0,010	983 000 000	1,00	0,00
16	0,027	747 000 000	0,99	0,00
32	0,035	956 000 000	0,97	0,00

Сравнение с различными модификациями алгоритма эволюционной стратегии. Для оценки эффективности алгоритма, вдохновленного поведением летучих мышей, были выбраны некоторые модификации алгоритма эволюционной стратегии. Наиболее универсальными по результатам исследования [10] являются $\alpha\beta$ -алгоритм без самоадаптации и $(\alpha + \beta)$ -алгоритм с самоадаптацией параметра ширины мутации, поэтому именно они будут использованы для сравнения.

Исходя из результатов, представленных в исследовании [10], на овражной функции алгоритм поисковой оптимизации, вдохновленный поведением летучих мышей, демонстрирует на порядок лучшие результаты, чем $(\alpha + \beta)$ -алгоритм с самоадаптацией, достигаемые за большее на 1–4 порядка число вычислений целевой функции.

На многоэкстремальной функции исследуемый алгоритм находит значения, меньшие на один-два порядка относительно решений, найденных $\alpha\beta$ -алгоритмом без самоадаптации и $(\alpha + \beta)$ -алгоритмом с самоадаптацией. Для достижения такого результата ему требуется на 3–4 порядка большее число вычислений целевой функции.

Заключение. Исследование эффективности алгоритма, вдохновленного поведением летучих мышей, проведено на двух функциях: овражной функции Розенброка и многоэкстремальной функции Экли. Результаты исследований показали, что алгоритм достаточно эффективен как на многоэкстремальных функциях, так и на овражных. Отметим, что хотя при поиске экстремума обеих функций найденное лучшее решение увеличивается с ростом размерности вектора варьируемых параметров, для функции Розенброка разница между найденными лучшими значениями на соседних размерностях выше, чем для функции Экли.

Скорость сходимости алгоритма на обеих функциях обратно пропорциональна размерности вектора варьируемых параметров. Но скорость сходимости алгоритма на начальных итерациях на функции Розенброка значительно выше, чем на последующих. На функции Экли алгоритм демонстрирует достаточно низкую скорость сходимости.

Можно сделать вывод, что алгоритм, вдохновленный поведением летучих мышей, одинаково высокоэффективен при использовании как овражной функции, так и многоэкстремальной: низкая скорость сходимости многоэкстремальной функции компенсируется высокой точностью найденных лучших значений даже при больших размерностях.

Литература

- [1] Lamberti L., Pappalettere C. Metaheuristic design optimization of skeletal structures: a review. *Comput. Technol. Rev.*, 2011, vol. 4, p. 1–32.
DOI: <http://dx.doi.org/10.4203/CTR.4.1>
- [2] Yang X.S. A new metaheuristic bat-inspired algorithm. In: *Nature inspired cooperative strategies for optimization*. Springer, 2010, pp. 65–74.
DOI: https://doi.org/10.1007/978-3-642-12538-6_6
- [3] Hasançebi O., Teke T., Pekcan O. A bat-inspired algorithm for structural optimization. *Comput. Struct.*, 2013, vol. 128, pp. 77–90.
DOI: <https://doi.org/10.1016/j.compstruc.2013.07.006>
- [4] Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. М., Изд-во МГТУ им. Н.Э. Баумана, 2014.
- [5] Öztürk M.A bat-inspired algorithm for prioritizing test cases. *Vietnam J. Comput. Sci.*, 2018, vol. 5, no. 1, pp. 45–57. DOI: <https://doi.org/10.1007/s40595-017-0100-x>
- [6] Tariq F., Alelyani S., Abbas G. et al. Solving renewables-integrated economic load dispatch problem by variant of metaheuristic bat-inspired algorithm. *Energies*, 2020, vol. 13, no. 23, art. 6225. DOI: <https://doi.org/10.3390/en13236225>
- [7] Akhtar S., Ahmad A.R., Abdel-Rahman E.M. A metaheuristic bat-inspired algorithm for full body human pose estimation. *2012 Ninth Conf. on Computer and Robot Vision*, 2012, pp. 369–375. DOI: <https://doi.org/10.1109/CRV.2012.55>
- [8] Menassel R., Gaba I., Titi K. Introducing BAT inspired algorithm to improve fractal image compression. *Int. J. Comput. Appl.*, 2020, vol. 42, no. 7, pp. 697–704.
DOI: <https://doi.org/10.1080/1206212X.2019.1638631>
- [9] Stroustrup B. The C++ programming language. Addison-Wesley, 1997.
- [10] Козов А.В. Сравнение эффективности некоторых модификаций алгоритма эволюционной стратегии. *Политехнический молодежный журнал*, 2018, № 5.
DOI: <http://dx.doi.org/10.18698/2541-8009-2018-5-309>

Зуева Анна Андреевна — студентка кафедры «Системы автоматизированного проектирования», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Козов Алексей Владимирович, старший преподаватель кафедры «Системы автоматизированного проектирования», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:

Зуева А.А. Исследование популяционного алгоритма поисковой оптимизации, вдохновленного поведением летучих мышей. *Политехнический молодежный журнал*, 2022, № 08(73). <http://dx.doi.org/10.18698/2541-8009-2022-08-816>

A POPULATION-BASED SEARCH ENGINE OPTIMIZATION ALGORITHM INSPIRED BY THE BEHAVIOR OF BATS

A.A. Zueva

anna.926@outlook.com

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

Metaheuristic algorithms are the most effective for solving undifferentiated multimodal and ravine minimization problems with high dimensional search space. The paper presents a study of one of such methods – an algorithm inspired by the behavior of bats. The method is implemented using the object-oriented C++ programming language, with a class of bats and a container class of the population. Analysis was performed based on ravine and multi-extremal functions using data on algorithm convergence and the best found values at different dimensions of the vector of varying parameters. A comparison is made between the method under study and modifications of the evolutionary strategy algorithm in terms of the smallest solutions found and the convergence rates of the algorithms at different dimensions. It is concluded that the bat algorithm is highly efficient on both ravine and multiextremal functions of different dimensions.

Keywords

Metaheuristic search, bat algorithm, global optimization, population algorithm, efficiency research, group intelligence, structural optimization, minimization

Received 24.06.2022

© Bauman Moscow State Technical University, 2022

References

- [1] Lamberti L., Pappalettere C. Metaheuristic design optimization of skeletal structures: a review. *Comput. Technol. Rev.*, 2011, vol. 4, p. 1–32. DOI: <http://dx.doi.org/10.4203/CTR.4.1>
- [2] Yang X.S. A new metaheuristic bat-inspired algorithm. In: *Nature inspired cooperative strategies for optimization*. Springer, 2010, pp. 65–74. DOI: https://doi.org/10.1007/978-3-642-12538-6_6
- [3] Hasançebi O., Teke T., Pekcan O. A bat-inspired algorithm for structural optimization. *Comput. Struct.*, 2013, vol. 128, pp. 77–90. DOI: <https://doi.org/10.1016/j.compstruc.2013.07.006>
- [4] Karpenko A.P. *Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy* [Modern search optimization algorithms. Algorithms inspired by nature]. Moscow, Bauman MSTU Publ., 2014. (in Russ.).
- [5] Öztürk M.A. bat-inspired algorithm for prioritizing test cases. *Vietnam J. Comput. Sci.*, 2018, vol. 5, no. 1, pp. 45–57. DOI: <https://doi.org/10.1007/s40595-017-0100-x>
- [6] Tariq F., Alelyani S., Abbas G. et al. Solving renewables-integrated economic load dispatch problem by variant of metaheuristic bat-inspired algorithm. *Energies*, 2020, vol. 13, no. 23, art. 6225. DOI: <https://doi.org/10.3390/en13236225>

- [7] Akhtar S., Ahmad A.R., Abdel-Rahman E.M. A metaheuristic bat-inspired algorithm for full body human pose estimation. *2012 Ninth Conf. on Computer and Robot Vision*, 2012, pp. 369–375. DOI: <https://doi.org/10.1109/CRV.2012.55>
- [8] Menassel R., Gaba I., Titi K. Introducing BAT inspired algorithm to improve fractal image compression. *Int. J. Comput. Appl.*, 2020, vol. 42, no. 7, pp. 697–704. DOI: <https://doi.org/10.1080/1206212X.2019.1638631>
- [9] Stroustrup B. *The C++ programming language*. Addison-Wesley, 1997.
- [10] Kozov A.V. Comparing the efficiency of some modifications of the evolutionary strategy algorithm. *Politekhnikheskiy molodezhnyy zhurnal* [Politechnical Student Journal], 2018, no. 5. DOI: <http://dx.doi.org/10.18698/2541-8009-2018-5-309> (in Russ.).

Zueva A.A. — student, Department of Computer-Aided Design Systems, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Kozov A.V., Senior Lecturer, Department of Computer-Aided Design Systems, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:

Zueva A.A. A population-based search engine optimization algorithm inspired by the behavior of bats. *Politekhnikheskiy molodezhnyy zhurnal* [Politechnical student journal], 2022, no. 08(73). <http://dx.doi.org/10.18698/2541-8009-2022-08-816.html> (in Russ.).