ORES

Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

УДК 681.5.015.23

ВЫБОР ОПТИМАЛЬНЫХ ПАРАМЕТРОВ АЛГОРИТМА ЛЕТУЧИХ МЫШЕЙ

Лагунова Алина Дмитриевна
аспирант
ФГБУН Институт автоматики и процессов управления Дальневосточного отделения Российской академии наук
Владивосток, Россия
690041, Владивосток, ул. Радио, 5
e-mail: schimka_06@mail.ru

Аннотация

В данной работе проведено исследование влияния количества итераций и размера популяции на среднюю погрешность в работе алгоритма летучих мышей (Bat Algorithm, BA). Были выведены формулы нахождения оптимальных значений популяции и итераций (в зависимости от размерности задачи) для унимодальных, овражных и мультимодальных функций, а также общие формулы, в случае если вид функции не известен заранее. Для исследования алгоритма был разработан программный модуль на языке Python.

Ключевые слова: алгоритм летучих мышей, оптимизация, эвристический алгоритм, роевая оптимизация, стохастическая оптимизация.

CHOOSING THE OPTIMAL PARAMETERS OF THE BAT ALGORITHM (BA)

Alina D. Lagunova
Institute of Automation and Control Processes Far Eastern Branch of Russian Academy of Sciences
Vladivostok, Russia
690041, Vladivostok, 5 Radio Street
e-mail: schimka_06@mail.ru

ABSTRACT

In this paper, we study the influence of the number of iterations and the size of the population on the average error in the Bat Algorithm (BA). Formulas for finding optimal population values and iterations (depending on the dimension of problem) for unimodal, gully, and multimodal functions were derived, as well as General formulas if the type of function is not known in advance. To study the algorithm, a software module in Python was developed.

Key words: bat algorithm (BA), optimization, heuristic algorithm, swarm optimization, stochastic optimization.

Введение

Методы оптимизации – это методы поиска экстремума функции при наличии ограничений или без ограничений. Данные методы имеют широкое применение в различных задачах проектирования и управления: организация процессов производства и перевозок, выбор наилучших технологических режимов, планирование крупномасштабного производства, выбор структуры технологических цепочек и элементов конструкций, материальное и техническое снабжение, повышение доходности, управление запасами и ресурсами и т.д. Сложность функций, описывающих задачу, и их нелинейный характер делают процесс поиска оптимального решения классическими методами в аналитической форме крайне затруднительным, а зачастую и вообще невозможным. Высокая вычислительная трудоемкость решения оптимизационных задач со стохастическим критерием вынуждает искать способы достаточно быстрого получения желаемых решений [1, 5, 9].

Эвристические алгоритмы, активно развивающиеся в последние годы, включают в себя практический метод, не являющийся гарантированно точным или оптимальным, но достаточный для решения задачи [3, 4, 6, 11, 20]. При решении задач с большим количеством переменных важную роль играет время вычислений. Эвристические алгоритмы позволяют сократить время вычислений в 100-1000 раз. Кроме того, эвристические алгоритмы позволяют найти решение даже в тех случаях, когда точное решение не может быть найдено или его поиск имеет большую вычислительную сложность.

Роевый интеллект, описывающий коллективное поведение децентрализованной самоорганизующейся системы, является наиболее известным представителем эвристических методов. На данный момент существует большое количество роевых алгоритмов, например: метод роя частиц, алгоритм стаи серых волков, алгоритм кукушки, муравьиный алгоритм, и пр. [6, 8, 10, 11, 13]. Одним из последних разработанных алгоритмов этого типа является алгоритм поиска летучих мышей. Алгоритм летучих мышей (Bat Algorithm, BA) – метаэвристический алгоритм, разработанный Янгом в 2010 году [19], имитирующий свойство эхолокации летучих мышей [15]. Являясь чуть более сложным, чем большинство других алгоритмов, показывает высокую эффективность, при меньших временных затратах [7,14, 19].

Цель данной работы – исследование влияния количества итераций и размера популяции на среднюю погрешность алгоритма BA, а также предложение указаний по подбору оптимальных параметров, позволяющих минимизировать среднюю погрешность и вычислительную сложность.

Алгоритм летучих мышей

В основу алгоритма BA положены следующие правила.

1. Благодаря эхолокации мыши определяют расстояние до добычи.

2. Передвижение мышей - хаотично. Текущее положение и скорость i-ой мыши b_i, где $i = 1, ..., N$, N – размерность популяции, равны $x_i = (x_1, ..., x_D)$ и $v_i = (v_1, ..., v_D)$ соответственно, D – размерность пространства поиска. Для поиска добычи мыши генерируют сигналы с частотой i_ω и громкость a_i. В процессе поиска мыши могут менять частоту, а также частоту повторения излучаемых импульсов $r \in [0;1]$;

3. Частота сигналов изменяется в диапазоне $[\omega^{min}, \omega^{max}]$, $\omega^{max} > \omega^{min} \geq 0$, а громкость сигналов в пределах [0;1].

На этапе инициализации алгоритма начальные значения частот ω_i^0, громкостей a_i^0 и частот повторения импульсов r_i^0, где i = 1, ..., N, полагаются равномерно распределенными в соответствующих интервалах $[\omega^{min}, \omega^{max}]$, $[a^{min}, a^{max}]$, $[0,1]$. Миграция летучей мыши b_i , i = 1, ..., N, осуществляется по формулам [2,19]:

$$x_i' = x_i + v_i',$$
$$v_i' = v_i + \omega_i * (x_i - x^{best}),$$
$$\omega_i = \omega^{min} + \left(\omega^{max} - \omega^{min}\right) * R(0;1),$$

где x_i', x_i – новые и старые координаты i-ой мыши, v_i', v_i – новое и старое значения скорости i-ой мыши, $R(0;1)$ – случайное число из интервала (0;1).

Случайный локальный поиск выполняется по следующей схеме:

1. Случайным образом варьируется текущее положение i-ой летучей мыши b_i:

$$x_i' = x_i + \bar{a} * R(-1,1),$$

где i = 1, ..., N, x_i', x_i – новые и старые координаты i-ой мыши, $\bar{a}$ – текущее среднее значение громкостей всех летучих мышей в популяции, такое что

$$\bar{a} = \frac{1}{N} * \sum_{i=1}^{N} a_i;$$

$R(-1,1)$ – величина, равномерно распределенная на интервале от -1 до 1;

2. Вычисляется значение целевой функции в новой точке: $f(x_i') = f_i'$. Если оно лучше предыдущего значения, то есть $f_i' < f_i$, то процедура локального поиска завершается, в противном случае фиксированное число раз осуществляется возврат к шагу 1.

Эволюция параметров a_i и r_i осуществляется по правилам:

$$a_i^{t+1} = b_a * a_i^t,$$
$$r_i^{t+1} = r_i^0 * (1 - \exp(-b_r * t)),$$

где i = 1, ..., N, a_i^{t+1}, a_i^t – громкости i-ой мыши на (t + 1)-ой и t-ой итерациях соответственно, r_i^0 – частота повторения импульсов i-ой мышью при инициализации, r_i^{t+1} – частота повторения импульсов i-ой мышью на (t + 1)-ой итерации, t – номер поколения (итерации), $b_a \in (0,1), b_r > 0$ – заданные константы (свободные параметры), значения которых равны 0.9 [2, 14, 19].

Выбор оптимальных параметров алгоритма

В ходе исследований, алгоритм ВА был реализован в классическом виде для решения задачи оптимизации. Также была реализована возможность задать такие параметры работы алгоритма, как: максимальное количество итераций, размер популяции и количество варьируемых параметров для нахождения оптимального значения целевой функции.

Для теста были выбраны три функции (рис.1) [12,16,17,18]:

- Функция Де-Джонга:

$$f(x) = \sum_{i=1}^{n} x_i^2 \ , \ x_i \in [-5.12; 5.12], \ \min \ = \ 0 \text{ при } x_i \ = \ 0$$

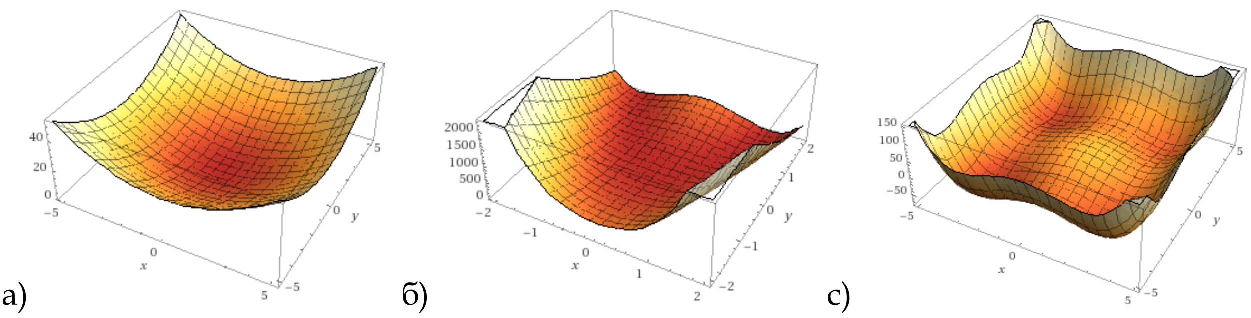
- Функция Розенброка:

$$f(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 \right] , \ x_i \in [-2{,}048; 2{,}048], \ \min = 0 \text{ при } x_i \ = \ 1$$

.

- Функция Стыбинского-Танга:

$$f(x) = \frac{1}{2} \sum_{i=1}^{n} x_i^4 - 16x_i^2 + 5x_i \ , \ x_i \in [-5; 5],$$
$$-39.16617n < min < -39.16616n, \text{ при } x_i = \text{-}2.903534$$

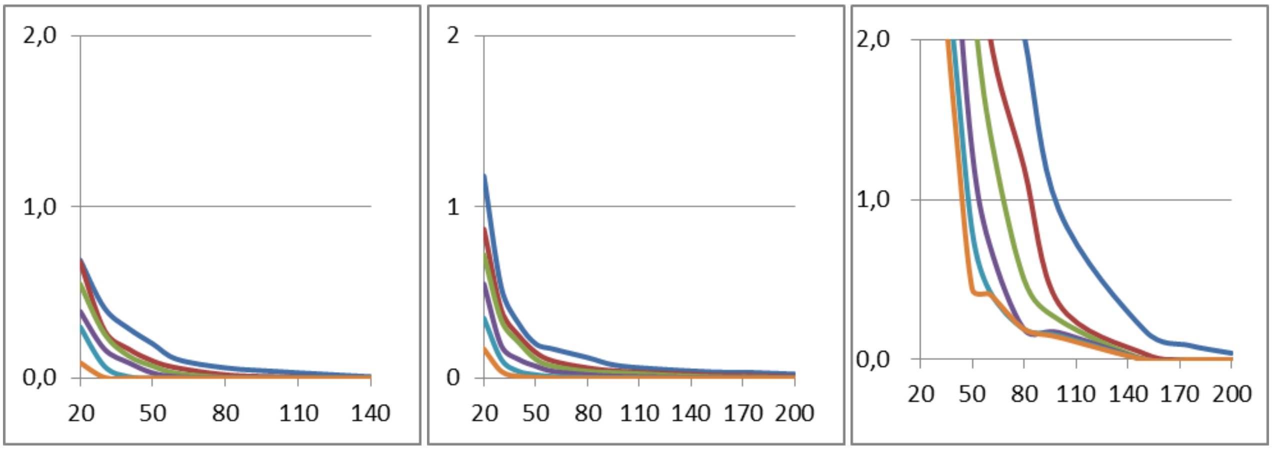


а) б) с)

Рисунок 1. *а) функция Де-Джонга, б) функция Розенброка, с) функция Стыбинского-Танга*

Для оценки эффективности алгоритма использовались такие параметры, как: количество итераций (*i*), размер популяции (*P*), погрешность (*S*), наименьшее время работы алгоритма в секундах (*time*) и количество переменных (*N*).

Алгоритм летучих мышей запускался 300 раз для каждого набора входных данных (*N*, *P*, *i*), после чего вычислялись средние значения *S* и *time* из всей серии тестов. Полученные графики зависимости средней ошибки от популяции и количества итераций представлены на Рисунках 2-5. Среднее время работы алгоритма для функций 8 переменных представлено в таблице 1.

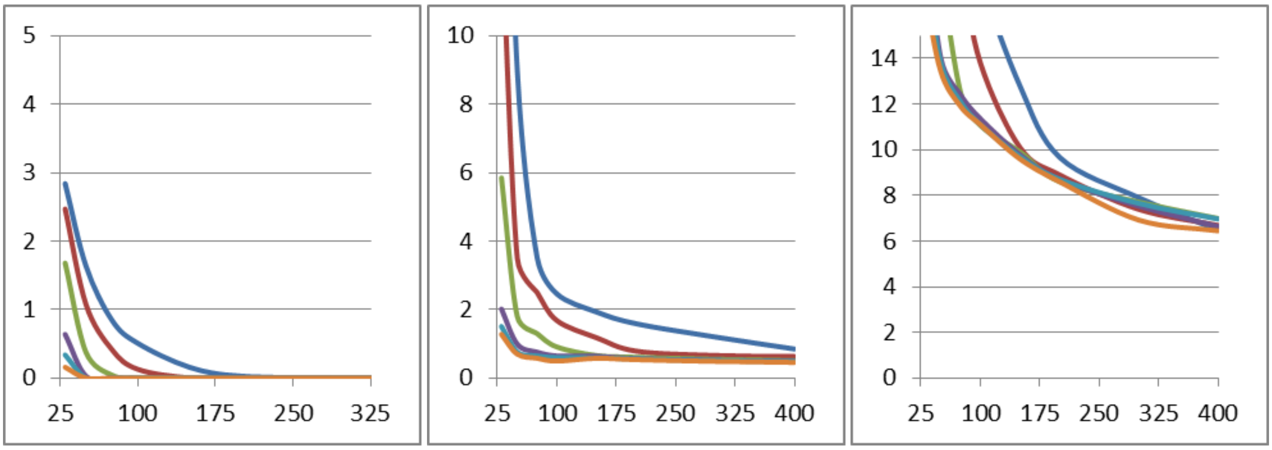


а) б) в)

***Рисунок 2.** Зависимость средней ошибки от популяции и количества итераций*

для функций 2-х переменных: а) Де-Джонг, б) Розенброк, в) Стыбинский-Танг

Итерации: 10 20 30 50 100 200

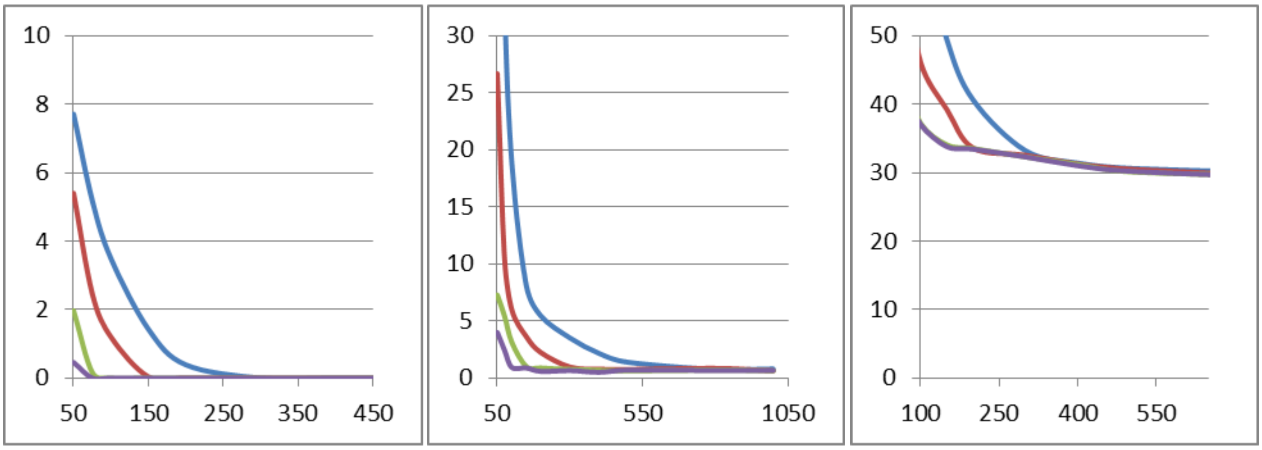


а) б) в)

***Рисунок 3.** Зависимость средней ошибки от популяции и количества итераций*

для функций 4-х переменных: а) Де-Джонг, б) Розенброк, в) Стыбинский-Танг

Итерации: 25 50 100 200 250 300

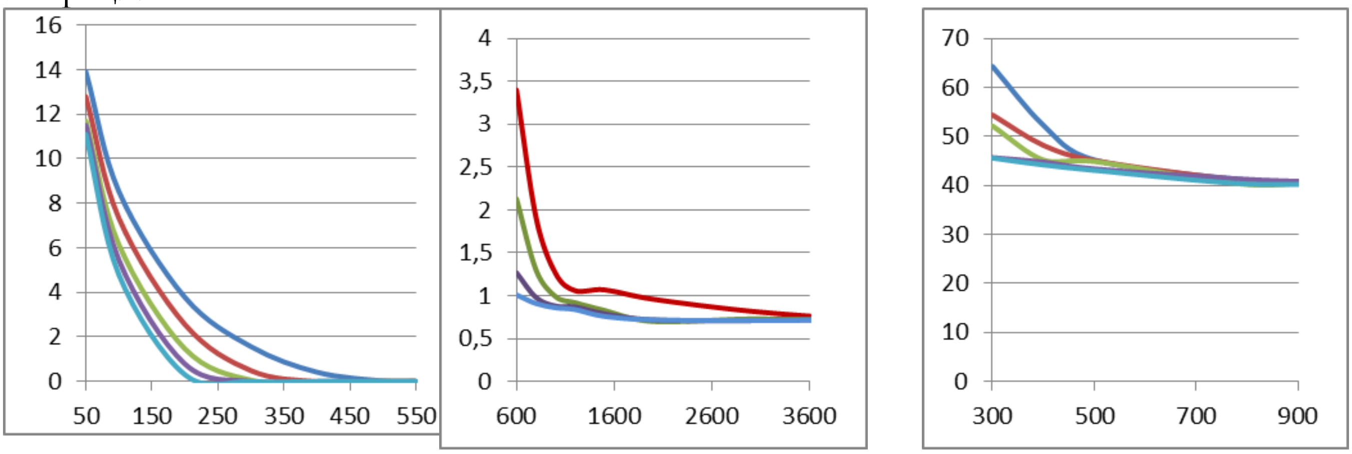


а) б) в)

Рисунок 4. *Зависимость средней ошибки от популяции и количества итераций*

для функций 8-ми переменных: а) Де-Джонг, б) Розенброк, в) Стыбинский-Танг

Итерации: 50 100 200 300



а) б) в)

Рисунок 5. *Зависимость средней ошибки от популяции и количества итераций для функций 10-ти переменных: а) Де-Джонг, б) Розенброк, в) Стыбинский-Танг*

Итерации: 30 40 50 60 70

Таблица 1. *Время работы (time) алгоритма BA для функций 8 переменных*

Функция	Популяция (P)	Количество итераций (i)				
		50	100	200	300	400
Де-Джонга	50	0,04	0,07	0,15	0,22	0,28
	75	0,06	0,11	0,21	0,32	0,44
	100	0,07	0,15	0,28	0,45	0,58
	150	0,11	0,22	0,44	0,6	0,86
	200	0,15	0,29	0,58	0,89	1,2
	300	0,22	0,45	0,79	1,34	1,8
	400	0,3	0,51	1,2	1,8	2,2

Розенброка	50	0,04	0,09	0,17	0,26	0,33
	75	0,06	0,12	0,22	0,37	0,49
	100	0,08	0,17	0,29	0,50	0,66
	150	0,13	0,25	0,49	0,78	0,98
	200	0,17	0,33	0,68	1,10	1,31
	300	0,25	0,51	1,04	1,52	1,97
	400	0,34	0,67	1,68	1,98	2,69
Стыбинского-Танга	50	0,05	0,09	0,19	0,26	0,34
	75	0,06	0,13	0,27	0,43	0,52
	100	0,09	0,18	0,38	0,56	0,67
	150	0,11	0,25	0,52	0,79	0,95
	200	0,16	0,34	0,76	1,06	1,38
	300	0,24	0,52	1,06	1,72	2,04
	400	0,33	0,70	1,43	2,33	2,76

Таблица 2. *Примерные интервалы удовлетворительных значений параметров*

Кол-во переменных	Параметр	Функции		
		Де-Джонг	Розенброк	Стыбинский-Танг
2	*P*	90-110	55-100	120-170
	i	20-50	40-60	30-50
4	*P*	180-250	325-450	300-450
	i	30-60	40-60	40-50
8	*P*	320-420	800-1000	600-800
	i	50-60	40-60	40-60
10	*P*	400-500	1300-1800	800-900
	i	40-60	40-60	30-60

Для нахождения оптимальных значений параметров, выделим интервалы наиболеее удовлетворительных значений (табл. 2). На основании полученных данных были выведены формулы для минимизации средней ошибки с минимальной вычислительной сложностью для разных типов функций.

Функция Де-Джонга:

$$P_D = 325 * ln(N) - 270 + \frac{280}{N!}$$
$$i_D \in [50; 60]$$

Функция Розенброка:

$$P_R = 1{,}14 * N^3 - 28{,}6 * N^2 + 374{,}3 * N - 680{,}8 + \frac{200}{N!}$$
$$i_R \in [50; 60]$$

Функция Стыбинского-Танга:

$$P_{ST} = 425{,}6 * ln(N) - 170$$
$$i_{ST} \in [50; 60]$$

Также были выведены формулы для случая, если вид целевой функции неизвестен.

$$P_{overall} = 1{,}11 * N^3 - 26{,}5 * N^2 + 334{,}6 * N - 484{,}6 + \frac{80}{N!}$$
$$i_{overall} \in [50; 60]$$

Рассчитанные показатели для исследованных функций (округлены до ближайшего целого) приведены в таблице 3.

Таблица 3. *Оптимальные значения популяции*

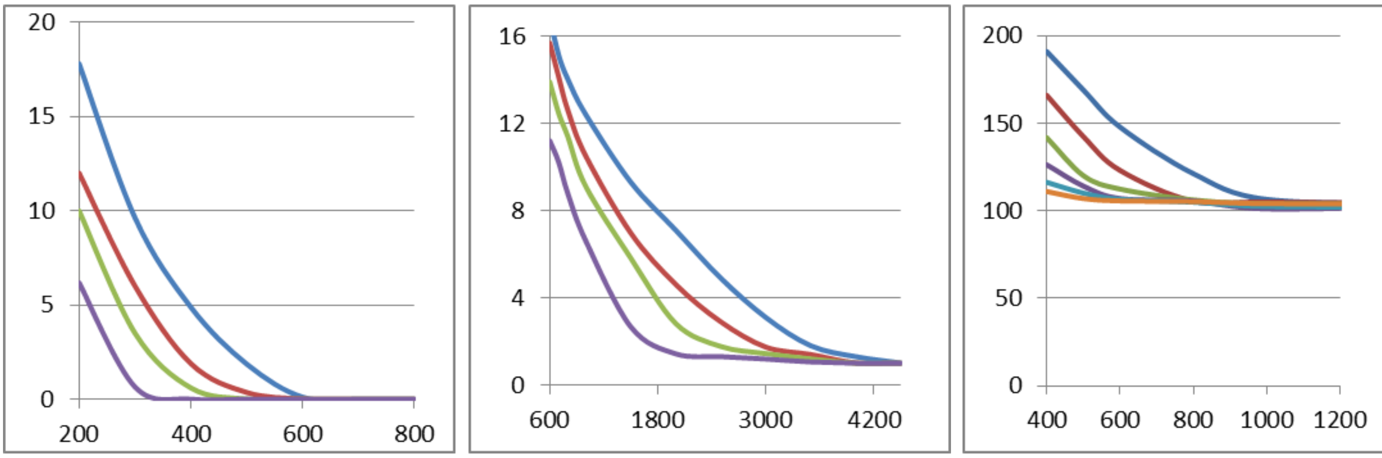
Кол-во переменных	Функция Де-Джонга	Функция Розенброка	Функция Стыбинского-Танга	Overall
2	95	62	125	125
4	192	440	420	504
8	406	1066	715	1064
10	478	1342	810	1321

Для проверки корректности формулы высчитаем оптимальные значения для функции с бóльшим количеством переменным и сверим их с результатами в серии тестов. Результаты для $N = 20$ и $N = 50$ (округлены до ближайшего целого) приведены в таблице 4.

Таблица 4. *Вычисленные оптимальные значения популяции для $N = 20$ и $N = 50$*

Кол-во переменных	Функция Де-Джонга	Функция Розенброка	Функция Стыбинского-Танга	Overall
20	704	4485	1105	4487
50	1001	89034	1495	88745

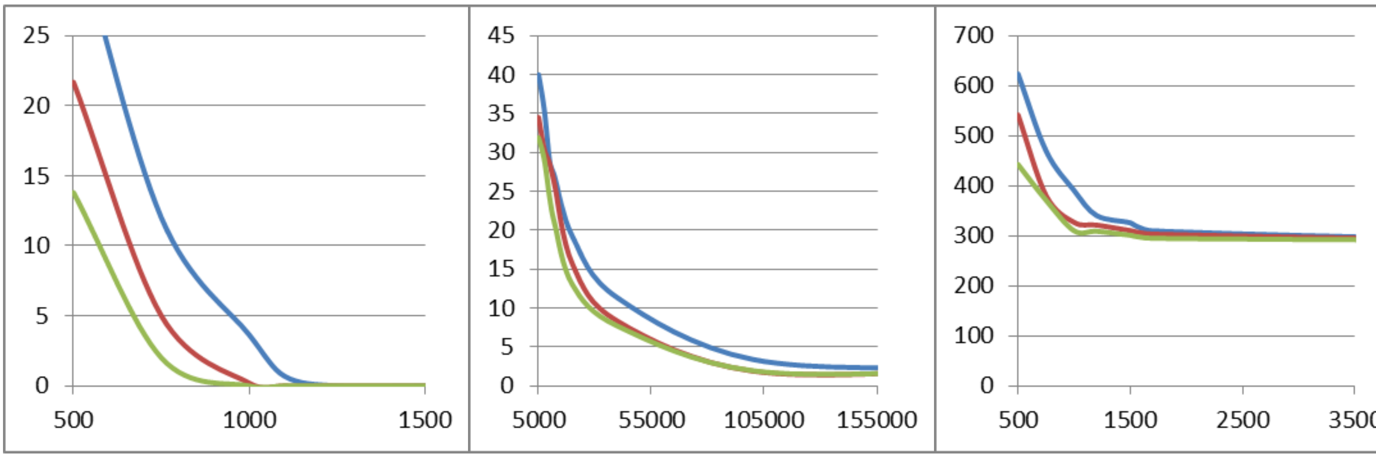
Проверим полученные значения на практике - запустим алгоритм с данными параметрами. Для сравнения, запустим алгоритм также и с несколькими ближайшими значениями данных параметров. Графики зависимостей средней ошибки для $N = 20$ и $N = 50$ представлены на рисунках 6,7.



а) б) в)

Рисунок 6. *Зависимость средней ошибки от популяции и количества итераций для функций 20-ти переменных: а) Де-Джонг, б) Розенброк, в) Стыбинский-Танг*

Итерации: 40 50 60 80



а) б) в)

Рисунок 7. *Зависимость средней ошибки от популяции и количества итераций для функций 50-ти переменных: а) Де-Джонг, б) Розенброк, в) Стыбинский-Танг*

Итерации: 40 50 60

Как видно из графиков, предложенные формулы с высокой точностью позволяют подобрать оптимальные значения параметров для алгоритма BA.

В случае же получения большей средней погрешности S чем требуется, наиболее целесообразно будет увеличивать популяцию при минимальном количестве итераций. Так, например, на расчет функции Стыбинского-Танга для 8 переменных затрачивается 0,33-0,34 сек. (табл. 1) с такими парами популяции и итерации, как: 50 и 400, 200 и 100, 400 и 50, в то время как значения средней погрешности для каждой пары параметров соответственно равны: 2.516, 2.24, 2.17.

Заключение

Высокая вычислительная трудоемкость решения оптимизационных задач со стохастическим критерием заставляет искать способы достаточно быстрого получения желаемых результатов. Наиболее радикальным направлением, сокращающим трудоемкость решения сложных вычислительных задач, стали эвристические алгоритмы, одним из которых является алгоритм летучих мышей (BA). Для эффективного использования алгоритма необходимо подбирать наиболее оптимальные его параметры, позволяющие минимизировать среднюю погрешность вычислений, не затрачивая при этом дополнительных временных ресурсов.

В рамках исследования было выявлено, что:

- наиболее оптимальное количество итераций 50-60
- в случае неудовлетворительной средней ошибки целесообразно увеличение популяции при минимальном количестве итераций

А также выведены:

- формулы нахождения оптимального значения популяции для унимодальных, овражных и мультимодальных функций.
- общая формула нахождения оптимального значения популяции, в случае, если вид функций не известен заранее.

Список литературы

1. Абрамов О.В., Катуева Я.В. Технология параллельных вычислений в задачах анализа и оптимизации / О.В. Абрамов, Я.В. Катуева // Проблемы управления. – 2003. №4. – С. 11-15.
2. Ахмедова, Ш.А. Коллективный самонастраивающийся метод оптимизации на основе бионических алгоритмов [Текст]: дис. канд. тех. наук: 05.13.01 / Ш.А. Ахмедова. - Красноярск, 2016. - 150 с.
3. Ахмедова Ш.А. Последовательный и параллельный стайный алгоритм для задач условной и безусловной оптимизации / Ш.А. Ахмедова // Актуальные проблемы авиации и космонавтики. – 2012. – Т.1. №8. – С. 289-290.
4. Брестер К.Ю. О решении задач многокритериальной оптимизации самонастраивающимся генетическим алгоритмом / К.Ю. Брестер // Актуальные проблемы авиации и космонавтики. – 2012. – Т.1. №8. – С. 290-291.
5. Диго Г.Б., Диго Н.Б., Катуева Я.В. Применение детерминированных критериев в задачах стохастической оптимизации / Г.Б. Диго, Н.Б. Диго, Я.В. Катуева // Многопроцессорные вычислительные системы. – 2006. - №2(12). - С. 82-88.
6. Кулиев Э.В., Щеглов С.Н., Пантелюк Е.А., Кулиева Н.В. Адаптивный алгоритм стаи серых волков для решения задач проектирования / Э.В. Кулиев, С.Н. Щеглов, Е.А. Пантелюк, Н.В. Кулиева // Известия ЮФУ. Технические науки. - 2017 . - №7. - С.28-38.
7. Лагунова, А.Д. Алгоритм летучих мышей (BA) для задачи глобальной безусловной оптимизации / А.Д. Лагунова // Оригинальные исследования (ОРИС). – 2019. – №6. – С.101-116.
8. Лагунова, А.Д. Алгоритм стаи серых волков (GWO) для задач оптимизации / А.Д. Лагунова // Оригинальные исследования (ОРИС). – 2019. – №4. – С.52-62.
9. Лагунова А.Д., Назаров Д.А. Параллельный алгоритм решения задачи оптимального параметрического синтеза на основе метода сеток / А.Д. Лагунова, Д.А. Назаров // Труды Международного симпозиума "Надежность и качество". - 2018. - Т.1. - С. 255-258.
10. Орловская Н.М. Анализ биоинспирированных методов глобальной оптимизации // Труды МАИ. – 2014. - №73. URL: https://mai.ru/upload/iblock/85d/85d945530545d38e6d4cbd591417766d.pdf (дата обращения: 17.05.2020).
11. Сагун А.В., Хайдуров В.В., Кунченко-Харченко В.И. Метод стаи волков и его модификация для решения задачи поиска оптимального пути / А.В. Сагун, В.В. Хайдуров, В.И. Кунченко-Харченко // Фізико-математична освіта : науковий журнал. – 2017. - №2(12). – С. 135-139.
12. Сергиенко А. Б. Тестовые функции для глобальной оптимизации / А.Б. Сергиенко. – Красноярск: Изд-во СГАУ им. М.Ф. Решетнева, 2015. – 112 с.
13. Частикова В.А., Дружинина М.А., Кекало А.С. Исследование эффективности алгоритма поиска косяков рыб в задаче глобальной оптимизации // Современные проблемы науки и образования. - 2014. - №4. - URL: http://science-education.ru/ru/article/view?id=14142 (дата обращения: 05.01.2020).
14. Частикова. В.А., Новикова Е.Ф. Алгоритм летучих мышей для решения задачи глобальной оптимизации // Научные труды КубГТУ. – 2015. - №2 - URL: https://ntk.kubstu.ru/file/348 (дата обращения: 05.01.2020).
15. Altringham, J.D. Bats: Biology and Behaviour. Oxford University Press, 1996.

16. Bossek Y. SMOOF: Single- and Multi-Objective Optimization Test Functions / Y. Bossek // The R Journal. - 2017.- Vol. 9 (1). - P. 103-113.
17. Jamil, M.; Yang, X.S. A Literature Survey of Benchmark Functions for Global Optimization Problems. Int. J. Math. Model. Numer. Optim. 2013, 4, 150–194.
18. Molga M., Smutnicki C. Test functions for optimization needs: [Электронный ресурс]. 2005. URL: https://www.vafaeijahan.com/en/wp-content/uploads/2012/02/Test-functions-for-optimization-needs.pdf (Дата обращения: 05.01.2020).
19. Yang X.S. A new metaheuristic bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization / Xin-She Yang // Springer, SCI 284. – 2010. - P. 65-74.
20. Zong W.G., Williams J.C. Ecological Optimization using Harmony Search / W.G. Zong, J.C. Williams // American conference on applied mathematics. – 2008. – P. 148-152.

References

1. Abramov OV, Katueva Ya.V. Technology of parallel computing in the tasks of analysis and optimization / O.V. Abramov, Ya.V. Katueva // Management problems. - 2003. №4. - P. 11-15 [in Russian].
2. Akhmedova, Sh.A. Collective self-adjusting optimization method based on bionic algorithms [Text]: PhD dissertation: 05.13.01 / Sh.A. Akhmedova. - Krasnoyarsk, 2016. – P. 150 [in Russian].
3. Akhmedova Sh.A. Consecutive and parallel sharing algorithm for problems of conditional and unconditional optimization / Sh.A. Akhmedova // Actual problems of aviation and cosmonautics. - 2012 - T.1. №8. - P. 289-290 [in Russian].
4. Brester K.Yu. On solving problems of multicriteria optimization of a self-adjusting genetic algorithm / K.Yu. Brester // Actual problems of aviation and astronautics. - 2012 - T.1. №8. - P. 290-291 [in Russian].
5. Digo G. B, Digo N. B, Katueva Ya.V. Application of deterministic criteria in problems of stochastic optimization / G. B. Digo, N.B. Digo, Ya.V. Katueva // Multiprocessor Computing Systems. - 2006. - №2 (12). P. 82-88 [in Russian].
6. Kuliev E.V., Shcheglov S.N., Pantelyuk E.A., Kuliyeva N.V. Adaptive algorithm for solving design problems / E.V. Kuliev, S.N. Scheglov, E.A. Pantelyuk, N.V. Kulieva // News of SFU. Technical science. - 2017 - №7. - P.28-38 [in Russian].
7. Lagunova, A.D. Bat algorithm (BA) for the problem of global unconditional optimization / A.D. Lagunova // Original Studies (ORIS). – 2019. – №6. – P. 101-116 [in Russian].
8. Lagunova, A.D. Algorithm of a pack of gray wolves (GWO) for optimization problems / A.D. Lagunova // Original Studies (ORIS). - 2019. - №4. - P.52-62 [in Russian].
9. Lagunova A.D., Nazarov D.A. Parallel algorithm for solving the problem of optimal parametric synthesis based on grid parameters / A.D. Lagunova, D.A. Nazarov // Proceedings of the International Symposium "Reliability and Quality." - 2018. - T.1. - P. 255-258 [in Russian].
10. Orlovskaya N.M. Analysis of bioinspired methods of global optimization // Proceedings of the MAI. - 2014. - №73. URL: https://mai.ru/upload/iblock/85d/85d945530545d38e6d4cbd591417766d.pdf (access date: 05.01.2020) [in Russian].
11. Sagun A.V., Khaidurov V.V., Kunchenko-Kharchenko V.I. The method of searching for optimal paths / A.V. Sagun, V.V. Khaidurov, V.I. Kunchenko-Kharchenko // Physics and Mathematics Journal of Science. - 2017. - №2 (12). - P. 135-139 [in Russian].
12. Sergienko A. B. Test functions for global optimization / A. B. Sergienko. - Krasnoyarsk: Publishing House of SSAU them. Mf Reshetnev, 2015. – 112 p. [in Russian].

13. Chastikova V.A., Druzhinina M.A., Kekalo A.S. Investigation of the effectiveness of the algorithm for searching fish schools // Modern problems of science and education. - 2014. - №4. - URL: http://science-education.ru/ru/article/view?id=14142 (access date: 17.04.2019) [in Russian].
14. Chastikova. V.A., Novikova E.F. The Bat Algorithm for solving the problem of global optimization // Scientific works of KubGTU. - 2015. - №2 - URL: https://ntk.kubstu.ru/file/348 (access date: 05.01.2020) [in Russian].
15. Altringham, J.D. Bats: Biology and Behaviour. Oxford University Press, 1996
16. Bossek, Y. SMOOF: single- and multi-purpose test optimization functions / Yu. Bossek // R. Magazine - 2017.- Tom. 9 (1). - P. 103-113
17. Jamil, M.; Yang, X.S. A Literature Survey of Benchmark Functions for Global Optimization Problems. Int. J. Math. Model. Numer. Optim. 2013, 4, 150–194
18. Molga M., Smutnitsky S. Test functions for optimization needs: [Electronic resource]. 2005. URL: https://www.vafaeijahan.com/en/wp-content/uploads/2012/02/Test-functions-for-optimization-needs.pdf (Revised: 05.01.2020)
19. Yang X.S. A new metaheuristic bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization / Xin-She Yang // Springer, SCI 284. – 2010. - P. 65-74
20. Zong W.G., Williams J.C. Environmental Optimization Using Harmony Search / W.G. Zong, J.C. Williams // American Conference on Applied Mathematics. - 2008. – P. 148-152