

<https://doi.org/10.26160/2572-4347-2020-10-10-17>

СРАВНЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМА ДИФФЕРЕНЦИАЛЬНОЙ ЭВОЛЮЦИИ И МЕТОДА РОЯ ЧАСТИЦ

Садчиков Ю.Г., Задорожная Н.М.

Ключевые слова: глобальная оптимизация, многомерная оптимизация, эвристические алгоритмы оптимизации, стохастические алгоритмы оптимизации, дифференциальная эволюция, метод роя частиц, тестовая функция «Подставка для яиц».

Аннотация. Главным недостатком классических «траекторных» алгоритмов оптимизации является то, что при оптимизации мультимодальных функций они сходятся в локальных минимумах, а также часто требуют дифференцируемость целевой функции. Для преодоления указанных недостатков используют эвристические алгоритмы. В статье описаны два метода глобальной оптимизации: метод дифференциальной эволюции и метод роя частиц. Проводится сравнение их эффективности на основе оптимизации тестовой функции «Подставка для яиц» (Eggholder function). Приводятся результаты вычислительного эксперимента, проведённого при помощи программы, написанной на языке MATLAB.

COMPARISON OF EFFICIENCY OF THE DIFFERENTIAL EVOLUTION ALGORITHM AND THE PARTICLE SWARM OPTIMIZATION

Sadchikov Yu.G., Zadorozhnaya N.M.

Keywords: global optimization, multidimensional optimization, heuristic optimization algorithms, stochastic optimization algorithms, differential evolution, particle swarm optimization, Eggholder function.

Abstract. The main disadvantage of the classical optimization algorithms is that when optimizing multimodal functions they converge at local minimum and often require differentiability of the objective function. Heuristic algorithms are used to overcome these difficulties. The article describes two methods of global optimization: the differential evolution method and the particle swarm optimization method. Their effectiveness is compared based on the optimization of the test function — Eggholder function. The results of a computational experiment are presented, conducted using a program written in MATLAB

Введение

В настоящее время во многих областях науки и техники, таких как синтез оптимального управления, автоматизированное проектирование, экономический анализ, проектирование интеллектуальных систем управления и т.п., приходится решать задачи глобальной оптимизации. Отличительными особенностями данных задач являются мультимодальность, нелинейность, недифференцируемость целевой функции, высокая размерность пространства поиска решения, плохая формализованность и др. В связи с этим становятся слабо применимыми или не применимыми вовсе точные математические методы, например, классические «траекторные» методы. Поэтому для решения данной проблемы принято использовать эвристические алгоритмы, которые основаны на имитации различных физических, биологических и других процессов. Подобные алгоритмы используют в своей работе случайные числа [1, 2].

Среди множества существующих эвристических алгоритмов можно выделить два наиболее распространённых типа: генно-эволюционные и популяционные алгоритмы. Стоит отметить, что для подобных методов не доказана сходимость к глобальному решению, однако в ходе многочисленных вычислительных экспериментов и практического применения подобных алгоритмов установлено, что в большинстве случаев они находят решение.

В работе проведено сравнение двух стохастических методов: алгоритма дифференциальной эволюции (ДЭ) и метода роя частиц (МРЧ). Первый относится к классу генно-эволюционных алгоритмов, второй – к классу популяционных алгоритмов. Оба метода были предложены в 1995 году и развивались в дальнейшем, путём их усовершенствования и разработки новых модификаций.

Поведение данных методов существенно зависит от значений параметров, выбор которых производится для конкретной задачи оптимизации. В статье в качестве целевой функции рассматривается так называемая тестовая функция «Подставка для яиц» (Eggholder function), которую необходимо минимизировать.

В работе дана постановка задачи глобальной условной минимизации и описаны алгоритм дифференциальной эволюции и метод роя частиц. При помощи вычислительного эксперимента сравнивается эффективность указанных алгоритмов.

Постановка задачи

Задача глобальной условной оптимизации может быть сформулирована следующим образом [1, 6]:

$$\min f(\bar{x}) = f(\bar{x}^*) = f^*, \quad \bar{x} \in U, \quad (1)$$

где $\bar{x}$ – вектор варьируемых параметров в n -мерном арифметическом пространстве R^n ; $f(\bar{x})$ – целевая функция, являющаяся скалярной функцией n переменных; $\bar{x}^*$ – искомое оптимальное решение задачи; $f^* = f(\bar{x}^*)$ – оптимальное значение целевой функции; $U \subset R^n$ – область поиска.

Гиперкуб, соответствующий области поиска U :

$$\Pi = \{\bar{x} | x^- \leq x_i \leq x^+, i = \overline{1:n}\}.$$

В дальнейшем предполагается, что целевая функция $f(\bar{x})$ совпадает с функцией приспособленности (fitness function).

Целевая функция

Для сравнения эффективности алгоритмов многомерной глобальной оптимизации в качестве целевой функции (1) используется тестовая функция «Подставка для яиц» (Eggholder function), обобщённая на высшие размерности:

$$f(\bar{x}) = \sum_{i=1}^{n-1} \left[- (x_{i+1} + 47) \sin \sqrt{\left| \frac{x_i}{2} + (x_{i+1} + 47) \right|} - x_i \sin \sqrt{|x_i - (x_{i+1} + 47)|} \right], \quad \bar{x} \in R^n.$$

Данная функция достигает своего глобального минимума в точке $\bar{x}^* = [512; 404,2319; 512; 404,2319; \dots; 512; 404,2319]^T$, а её значение при этом в зависимости от размерности арифметического пространства n равняется $f(\bar{x}^*) = -959,6407 \cdot n$. Вид функции при $n = 2$ представлен на рис. 1.

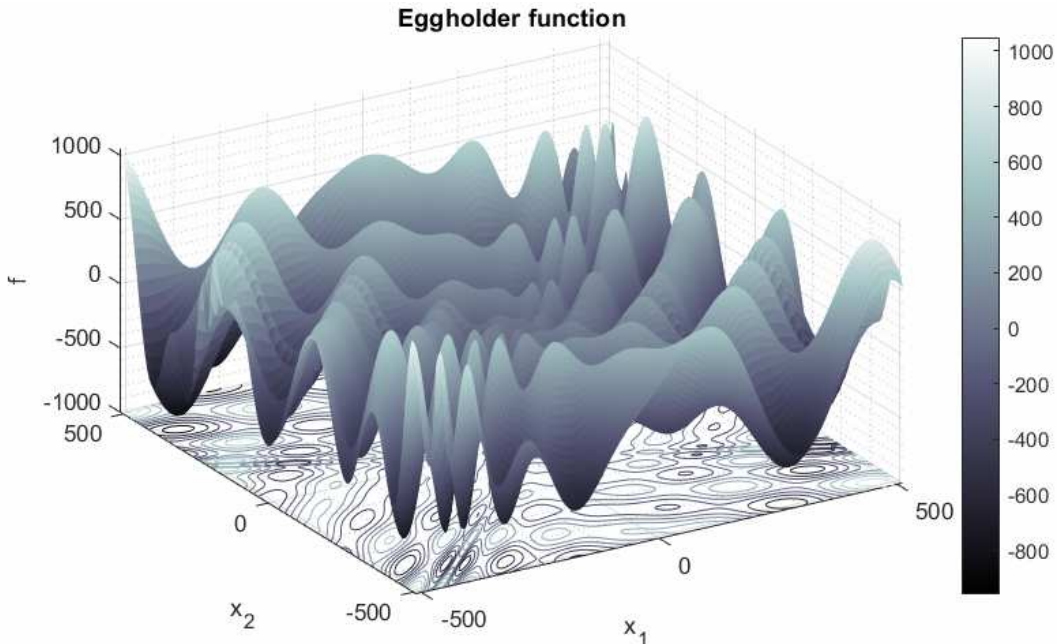


Рис. 1. Тестовая функция «Подставка для яиц» (Eggholder function) при $n = 2$

Представленная функция, как видно из её графика, является сложной функцией для оптимизации по причине наличия большого количества локальных минимумов, многие из которых незначительно отличаются по значению от глобального минимума и довольно равномерно расположены вдоль всей границы области поиска, что повышает вероятность схождения эвристических алгоритмов в локальном минимуме.

Для указанной целевой функции границы гиперкуба, задающего область поиска, составляют $x^- = -512$ и $x^+ = 512$ для каждой компоненты вектора $\bar{x}$. В случае выхода какого-либо вектора $\bar{x}$ в процессе оптимизации за границы данного гиперкуба, такой вектор проектируется на допустимое множество, то есть, если i -ая координата вектора x_i^k на k -м шаге больше x^+ , то $x_i^k = x^+$, если меньше x^- , то $x_i^k = x^-$.

Дифференциальная эволюция (ДЭ)

В работе используется базовый вариант алгоритма. Во время инициализации генерируется некоторое множество векторов, называемых поколением. Под векторами в данном случае понимаются точки n -мерного пространства, в котором определена целевая функция $f(\bar{x})$. Далее, на всех последующих итерациях генерируется новое поколение векторов. Для

каждого вектора $\bar{x}_i$ ($i = \overline{1:n}$) из старого поколения выбираются три различных случайных вектора $\bar{a}$, $\bar{b}$ и $\bar{c}$ среди векторов того же поколения, за исключением самого вектора $\bar{x}_i$, и генерируется так называемый мутантный вектор (mutant vector) по следующей формуле [1, 2, 7]:

$$\bar{c}' = \bar{c} + F(\bar{a} - \bar{b}),$$

где F – сила мутации, некоторая положительная действительная константа в интервале $[0, 2]$.

Затем над мутантным вектором $\bar{c}'$ выполняется операция «скрещивания» (crossover), то есть с некоторой вероятностью P его координаты замещаются соответствующими координатами из исходного вектора $\bar{x}_i$. Таким образом, образуется пробный вектор (trial vector). Если он оказывается лучше вектора $\bar{x}_i$, то в новом поколении исходный вектор заменяется пробным вектором, а в противном случае – остаётся $\bar{x}_i$.

Стоит отметить, что число векторов в каждом поколении одно и то же и является одним из параметров метода. Заметим, что данный алгоритм использует так называемую адаптивную мутацию, то есть когда особи популяции значительно распределены в пространстве поиска, алгоритм мутации производит их значительные вариации. В противном случае, метод реализует небольшие изменения особей.

Метод роя частиц (МРЧ)

В основе данного метода лежит поведение птиц в стае. Алгоритм моделирует многоагентную систему, где агенты-частицы двигаются к оптимальным решениям, обмениваясь при этом информацией с соседями.

Текущее состояние частицы характеризуется её координатами в пространстве решений и вектором скорости перемещения. Оба параметра выбираются случайным образом на этапе инициализации.

Для вычисления нового положения и скорости частицы на каждой итерации в каноническом методе роя частиц используются следующие формулы [1, 3]:

$$\bar{x}_i^+ = \bar{x}_i + \bar{v}_i;$$

$$\bar{v}_i^+ = w \cdot \bar{v}_i + a \cdot \bar{r}_a \otimes (\bar{p}_i - \bar{x}_i) + b \cdot \bar{r}_b \otimes (\bar{g}_i - \bar{x}_i),$$

где $\bar{x}_i = [x_1^i, x_2^i, \dots, x_n^i]^T$ – вектор, определяющий текущее положение i -ой частицы; $\bar{v}_i = [v_1^i, v_2^i, \dots, v_n^i]^T$ – вектор скорости частицы; w , a , b – свободные параметры алгоритма, которые определяют его поведение; $\bar{p}_i$ – координаты лучшей найденной i -ой частицей точки за всё время поиска, $\bar{g}_i$ – координаты лучшей точки за всё время поиска, найденной соседней с i -ой частицей; $\bar{r}_a$, $\bar{r}_b$ – случайные векторы с диапазоном чисел от 0 до 1 включительно размерностью n ; $\bar{x}_i^+$ и $\bar{v}_i^+$ – координаты и скорость i -ой частицы на

следующем шаге. Символ $\otimes$ здесь означает покомпонентное умножение элементов векторов.

Рекомендуемые значения свободных параметров алгоритма: $w = 0,7298$; $a = b = 1,49618$ [1]. Для более тонкой настройки метода роя частиц, как и других эвристических методов, для конкретной задачи используется мета-оптимизация, то есть применяется другой оптимизационный алгоритм для выбора значений свободных параметров МРЧ.

Помимо вышеназванных параметров, на эффективность МРЧ значительное влияние оказывает топология соседства частиц. Учитывая используемую целевую функцию, в работе применяется топология соседства кольцо, которая обеспечивает самую медленную скорость распространения информации о минимальном найденном значении целевой функции, достигнутом какой-либо другой частицей. При других топологиях соседства в ходе нескольких предварительных вычислительных экспериментов было обнаружено частое схождение алгоритма в локальных минимумах.

Вычислительный эксперимент

Количество векторов в поколении для дифференциальной эволюции и частиц в рое для метода роя частиц рано $s = 100$. Максимальное количество итераций алгоритма $k_{\max} = 10^4$. Исследование проводится при помощи метода мултистарта с количеством стартов $N = 100$ [4].

Для оценки эффективности представленных алгоритмов используются следующие параметры [5]:

- лучшее значение целевой функции f^* , вычисленное по следующему выражению $f(\bar{x}^*) = -959,6407 \cdot n$;

- лучшее значение целевой функции $\tilde{f}$, достигнутое по мултистарту;

- среднее достигнутое по мултистарту значение целевой функции $\bar{f}$ и оценка среднего квадратичного отклонения $\sigma(f)$ на основании несмещённой оценки дисперсии;

- потребовавшиеся средние по мултистарту количество итераций $\bar{k}$ и время $\bar{t}$, чтобы достичь минимального значения целевой функции с точностью $\varepsilon = 1$;

- оценка вероятности локализации глобального минимума с точностью $\varepsilon = 100$, P_{100} , и с точностью $\varepsilon = 1$, P_1 .

Для более корректного сравнения эффективности алгоритмов проведена их мета-оптимизация, в качестве которой использован обычный перебор. Зависимость параметров методов оптимизации от размерности пространства поиска представлена в табл. 1.

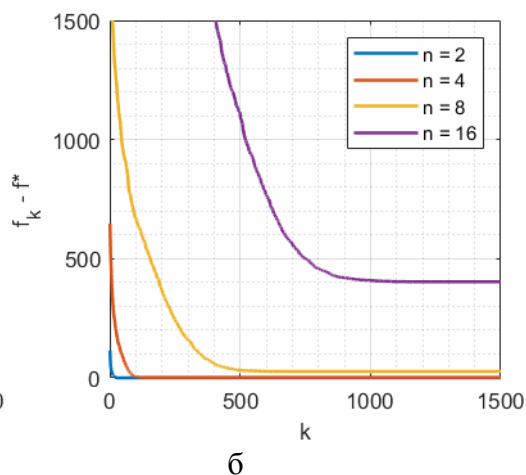
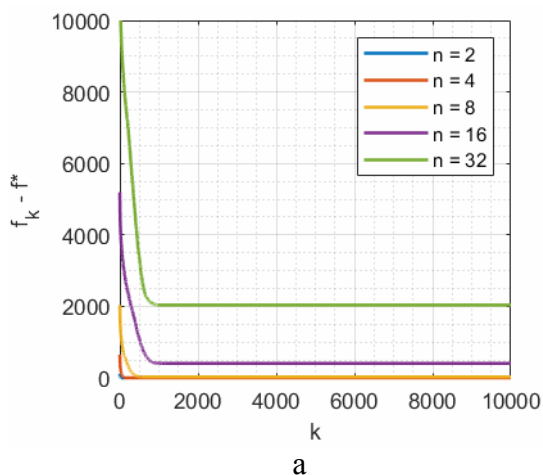
Результаты вычислительного эксперимента представлены в табл. 2 и на рис. 1 и 2.

Табл. 1. Значения параметров методов оптимизации в зависимости от размерности пространства поиска n

n	2	4	8	16	32
Дифференциальная эволюция					
F	1,2	1,1	1,1	1,0	1,0
P	1,00	0,95	0,85	0,75	0,55
Метод роя частиц					
w	0,95	0,90	0,90	0,90	0,85
a	0,95	1,15	1,20	1,35	1,45
b	1,25	1,35	1,40	1,45	1,55

Табл. 2. Результаты вычислительного эксперимента для двух методов

n	f^*	$\tilde{f}$	$\bar{f}$	$\sigma(f)$	$\bar{k}$	$\bar{t}$, с	P_{100}	P_1
Дифференциальная эволюция								
2	-959,641	-959,640	-959,320	0,286	17	0,007	1,00	1,00
4	-1919,281	-1919,279	-1917,413	9,490	286	0,119	1,00	0,98
8	-3838,563	-3838,424	-3811,856	45,912	3282	1,646	0,87	0,70
16	-7677,126	-7537,605	-7274,135	145,270	10000	6,369	0,00	0,00
32	-15354,251	-14338,194	-13317,697	501,237	10000	9,137	0,00	0,00
Метод роя частиц								
2	-959,641	-959,640	-959,355	0,293	19	0,019	1,00	1,00
4	-1919,281	-1919,270	-1918,761	0,298	489	0,549	1,00	1,00
8	-3838,563	-3838,456	-3744,718	78,549	8956	10,872	0,54	0,25
16	-7677,126	-7333,056	-6895,615	196,308	10000	14,570	0,00	0,00
32	-15354,251	-13472,886	-12593,480	343,055	10000	19,955	0,00	0,00

Рис. 2. Сходимость метода дифференциальной эволюции при k , изменяющемся от 0 до $k_{\max}$ (а), и в увеличенном масштабе (б)

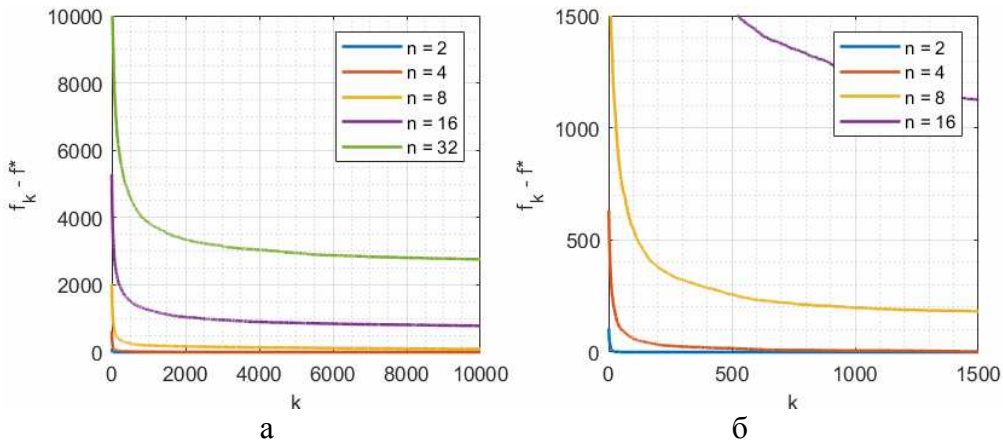


Рис. 3. Сходимость метода роя частиц при k , изменяющемся от 0 до $k_{\max}$ (а), и в увеличенном масштабе (б)

Заключение

Как видно из проведенного вычислительного эксперимента и полученных по его итогам результатов, при размерности пространства поиска равном $n=2$ между методами практически не наблюдается никакой разницы. При $n=4$ происходит незначительное ухудшение показателей алгоритмов, но при этом метод роя частиц, в отличие от метода дифференциальной эволюции, сохраняет среднее квадратичное отклонение ($\sigma(f)=0.298$ против 0,293 при $n=2$, а у дифференциальной эволюции 9,490 против 0,286). Однако при дальнейшем увеличении размерности пространства поиска метод ДЭ начинает в целом показывать результаты лучше, чем МРЧ. При этом, как видно из значений среднего времени, необходимого для достижения глобального минимума целевой функции, одна итерация МРЧ занимает значительно больше времени, чем одна итерация метода ДЭ.

Анализируя значения параметров исследуемых методов, полученные при помощи мета-оптимизации (табл. 1), стоит отметить, что по мере увеличения размерности пространства поиска для более эффективной работы алгоритмов уменьшается случайная составляющая и увеличивается влияние уже достигнутых к i -ой итерации наилучших значений целевой функции.

В итоге можно сделать вывод о том, что для целевых функций, обладающих схожей структурой с тестовой функцией «Подставка для яиц» (Eggholder function), метод дифференциальной эволюции при значительном размере пространства поиска (от 8 и более) оказывается более эффективным, чем метод роя частиц.

Список литературы

1. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновлённые природой. М.: Изд-во МГТУ им. Н.Э. Баумана, 2017. 446 с.
2. Lopez Cruz I.L. PhD-Thesis: Efficient Evolutionary Algorithms for Optimal Control. Wageningen Univ., Netherlands, 2002. 126 p.

3. Карпенко А.П., Селиверстов Е.Ю. Обзор методов роя частиц для задачи глобальной оптимизации (Particle Swarm Optimization) // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2009. № 3. DOI: 10.7463/00309.0116072
4. Горнов А.Ю. Реализация метода случайного мультистарта для задачи оптимального управления // Ляпуновские чтения: Тез. Докл. Иркутск: ИДСТУ СО РАН, 2003. С. 38.
5. Козов А.В. Сравнение эффективности некоторых модификаций алгоритма эволюционной стратегии. Политехнический молодёжный журнал, 2018, №5(22). DOI: 10.18698/2541-8009-2018-5-309
6. Аттетков А.В., Галкин С.В., Зарубин В.С. Методы оптимизации: Учебник для ВУЗов. М.: Издательство МГТУ им Н.Э. Баумана, 2001. 439с.
7. Курейчик В.В., Курейчик В.М., Родзин С.И. Теория эволюционных вычислений. М.: ФИЗМАТЛИТ, 2012. 260 с.

References

1. Karpenko A.P. Modern search engine optimization algorithms. Algorithms inspired by nature. Moscow: Publ. house of BMSTU, 2017. 446 p.
2. Lopez Cruz I.L. PhD-Thesis: Efficient Evolutionary Algorithms for Optimal Control. Wageningen Univ., Netherlands, 2002. 126 p.
3. Karpenko A.P., Seliverstov E.Yu. A review of particle swarm methods for the global optimization task (Particle Swarm Optimization) // Science and Education. BMSTU, Electron. Journal, 2009, no 3. DOI: 10.7463/00309.0116072
4. Gornov A.Yu. Implementation of the random multi-start method for the optimal control problem // Lyapunov readings: Thesis, Report, Irkutsk: IDSTU SB RAS, 2003. P.38.
5. Kozov A.V. Comparison of the effectiveness of some modifications of the evolutionary strategy algorithm // Politechnical student journal, 2018, no. 5 (22). DOI: 10.18698 / 2541-8009-2018-5-309
6. Attetkov A.V., Galkin S.V., Zarubin V.S. Optimization Methods: Textbook for High Schools. Moscow: Publishing House of BMSTU, 2001. 443 p.
7. Kureichik V.V., Kureichik V.M., Rodzin S.I. Theory of evolutionary computing. Moscow: FIZMATLIT, 2012. 260 p.

Садчиков Юрий Григорьевич – студент, yury.sadchickov@yandex.ru	Sadchikov Yuri Grigoryevich – student, yury.sadchickov@yandex.ru
Задорожная Наталья Михайловна – научный руководитель, кандидат технических наук, доцент, доцент кафедры «Системы автоматического управления»	Zadorozhnaya Natalya Mikhailovna – scientific advisor, candidate of technical sciences, associate professor, Department “Automatic Control Systems”
Московский государственный технический университет имени Н.Э. Баумана, Москва, Российская Федерация	Bauman Moscow State Technical University, Moscow, Russian Federation

Received 10.05.2020