



УДК 519.8

## **Разработка программного обеспечения метода глобальной оптимизации, имитирующего поведение стаи серых волков**

***Пантелеев А.В.\****

Московский авиационный институт  
(национальный исследовательский университет),  
г. Москва, Российская Федерация  
ORCID: <https://orcid.org/0000-0003-2493-3617>  
e-mail: [avpanteleev@inbox.ru](mailto:avpanteleev@inbox.ru)

***Беляков И.А.\*\****

Московский авиационный институт  
(национальный исследовательский университет),  
г. Москва, Российская Федерация  
ORCID: <https://orcid.org/0000-0003-3457-9914>  
e-mail: [vanbelyakov@yandex.ru](mailto:vanbelyakov@yandex.ru)

В данной статье рассматривается разработка программного обеспечения, позволяющего смоделировать алгоритм поиска экстремума функций многих переменных при параллелепипедных ограничениях, имитирующий поведение стаи серых волков. Для анализа эффективности работы алгоритма создано

### **Для цитаты:**

*Пантелеев А.В., Беляков И.А.* Разработка программного обеспечения метода глобальной оптимизации, имитирующего поведение стаи серых волков // Моделирование и анализ данных. 2021. Том 11. № 2. С. 59–73. DOI: <https://doi.org/10.17759/mda.2021110204>

**\*Пантелеев Андрей Владимирович**, доктор физико-математических наук, профессор, заведующий кафедрой математической кибернетики института «Информационные технологии и прикладная математика» Московского авиационного института (национального исследовательского университета), г. Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0003-2493-3617>, e-mail: [avpanteleev@inbox.ru](mailto:avpanteleev@inbox.ru)

**\*\*Беляков Иван Андреевич**, студент бакалавриата института «Информационные технологии и прикладная математика» Московского авиационного института (национального исследовательского университета), г. Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0003-3457-9914>, e-mail: [vanbelyakov@yandex.ru](mailto:vanbelyakov@yandex.ru)



программное обеспечение, позволяющее варьировать параметры метода. В статье приведены примеры работы программы на стандартном общепринятом наборе тестовых функций. Целью работы программы является сбор и анализ статистических результатов, позволяющих оценить точность нахождения экстремума и характер сходимости. Программа позволяет построить линии уровня целевых функций и продемонстрировать работу алгоритма по шагам. Анализ статистических данных позволил сформулировать обоснованные рекомендации по выбору параметров алгоритма.

**Ключевые слова:** алгоритм глобальной оптимизации, метаэвристический алгоритм, программное обеспечение, метод имитации поведения стаи серых волков

## 1. ВВЕДЕНИЕ

Метаэвристические алгоритмы хорошо зарекомендовали себя как эффективный инструмент поиска глобального экстремума. Обширное множество алгоритмов делят на несколько групп, отличающихся по принципу работы: эволюционные методы, биоинспирированные методы, алгоритмы роевого интеллекта, методы, имитирующие физические процессы, мультистартовые методы и т.д. Детальное описание перечисленных методов можно найти, например, в [1–6]. Метод имитации стаи серых волков [7] является оригинальным методом поиска глобального экстремума, он сочетает в себе как элементы роевого интеллекта, проявляемые в обмене информацией между особями, так и элементы биоинспирированных алгоритмов, так как в основе работы метода лежит математическая модель поведения волков в стае.

Метод серых волков (Grey Wolf Optimizer – GWO) имитирует охоту стаи серых волков за жертвой. Он относится к методам роевого интеллекта, в которых используется иерархия лидерства в стае и особый механизм охоты, заключающийся в отслеживании и приближении к жертве, ее последующем окружении и финальном нападении. Положение каждой особи основывается на трех лидерах, которые владеют наиболее полной информацией о положении жертвы (точке экстремума). В соответствии с положением лидеров стаи остальные особи двигаются, постепенно окружая их. Для предотвращения стагнации в точках локального экстремума алгоритм использует стохастические коэффициенты, вынуждающие волков осуществлять поиск на всем множестве допустимых решений. Таким образом, можно сформулировать основные шаги работы метода:

- 1) создание начальной популяции;
- 2) выявление наиболее приспособленных особей – лидеров стаи;
- 3) перемещение остальных особей в пространстве допустимых решений относительно лидеров;
- 4) проверка условия окончания алгоритма.

Существует также модификация алгоритма, в основе которой лежит подбор оптимального параметра, отвечающего за нападение волка на жертву [8].



Программное обеспечение позволяет смоделировать работу алгоритма с возможностью оперативно изменять характерные параметры метода. Полученные данные позволяют провести анализ результатов с целью выработки общих рекомендаций по выбору гиперпараметров алгоритма, которые можно использовать при решении прикладных задач.

## 2. ПОСТАНОВКА ЗАДАЧИ

Дана целевая функция  $f(x) = f(x_1, x_2, \dots, x_n)$ , определенная на множестве допустимых решений  $D \subseteq R^n$ .

Требуется найти условный глобальный максимум функции  $f(x)$  на множестве  $D$ , т.е. такую точку  $x^* \in D$ , что

$$f(x^*) = \max_{x \in D} f(x),$$

где  $x = (x_1, x_2, \dots, x_n)^T$ ,  $D = \{x \mid x_i \in [a_i, b_i], i = 1, 2, \dots, n\}$ .

Задача поиска минимума функции  $f(x)$  сводится к задаче поиска максимума путем замены знака перед функцией на противоположный:  $f(x^*) = \min_{x \in D} f(x) = -\max_{x \in D} [-f(x)]$ . Функция  $f(x)$  может быть многоэкстремальной, поэтому искомое решение в общем случае неединственное.

## 3. СТРАТЕГИЯ ПОИСКА РЕШЕНИЯ

В начале работы метода случайным образом, используя предположение о равномерном распределении, на множестве допустимых решений  $D$  генерируется некоторый набор начальных точек (волков в стае):  $I = \{x^j = (x_1^j, x_2^j, \dots, x_n^j)^T, j = 1, \dots, NP\} \subset D$ , где  $x^j$  – вектор координат волка с номером  $j$ ,  $NP$  – количество волков в стае. Поскольку в процессе охоты положение жертвы точно не известно вследствие ее постоянного движения (а в задаче оптимизации не известно положение точки экстремума), то члены стаи ориентируются на лидеров, полагая, что они обладают большей информацией о положении жертвы (точке экстремума).

В стае волков, где каждый волк характеризуется своей позицией в области допустимых решений, выбираются три последовательно лучших  $(\alpha, \beta, \gamma)$  по величине целевой функции  $f(x)$ :  $x^\alpha, x^\beta, x^\gamma$ . Все волки в стае меняют свое положение с учетом сравнения своей текущей позиции с этими тремя наилучшими:

$$x^j(k+1) = \frac{x^{j,1}(k+1) + x^{j,2}(k+1) + x^{j,3}(k+1)}{3},$$

$$x^{j,1}(k+1) = x^\alpha(k) - A_\alpha^j \otimes D_\alpha^j(k),$$

$$x^{j,2}(k+1) = x^\beta(k) - A_\beta^j \otimes D_\beta^j(k),$$

$$x^{j,3}(k+1) = x^\gamma(k) - A_\gamma^j \otimes D_\gamma^j(k),$$



$$D_{\alpha}^j(k) = \left| C_{\alpha}^j \otimes x^{\alpha}(k) - x^j(k) \right|,$$

$$D_{\beta}^j(k) = \left| C_{\beta}^j \otimes x^{\beta}(k) - x^j(k) \right|,$$

$$D_{\gamma}^j(k) = \left| C_{\gamma}^j \otimes x^{\gamma}(k) - x^j(k) \right|,$$

где  $\otimes$  – операция поэлементного произведения векторов по Адамару,  $k$  – номер итерации,  $x^j(k+1), x^j(k)$  – следующее и текущее положения волков,  $j=1, \dots, NP$ ;  $A_{\alpha}^j, A_{\beta}^j, A_{\gamma}^j$  – векторы, определяемые по правилу  $A_m^j = 2a \otimes r_1 - a$ ,  $m = \alpha, \beta, \lambda$ ;  $r_1$  –  $n$ -мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке  $[0, 1]$ ;  $a$  – вектор с одинаковыми компонентами, уменьшающимися линейно по закону  $a_i = 2(1 - \frac{k}{K})$ ,  $i = 1, \dots, n$ ,  $K$  – максимальное число итераций;  $C_{\alpha}^j, C_{\beta}^j, C_{\gamma}^j$  – векторы, определяемые по правилу  $C_m^j = 2r_2$ ,  $m = \alpha, \beta, \lambda$ ;  $r_2$  –  $n$ -мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке  $[0, 1]$ . Имеется модификация, в которой  $a_i = 2(1 - \frac{k^2}{K^2})$ ,  $i = 1, \dots, n$  [8].

Общая схема работы метода серых волков представлена на рис. 1.

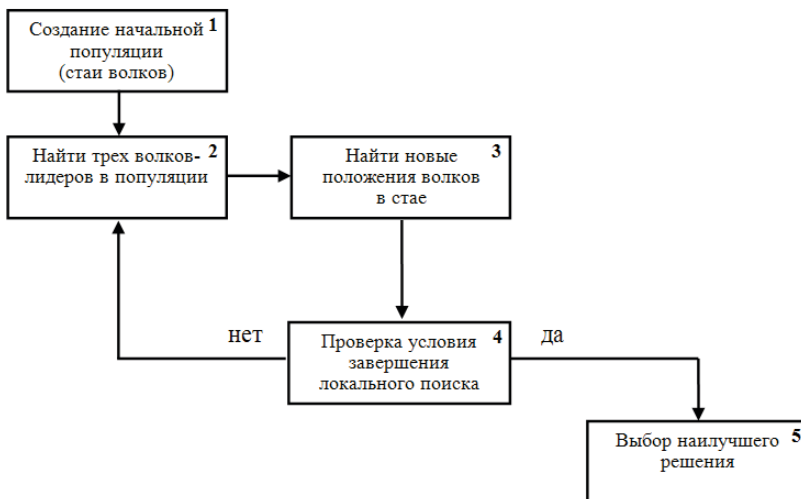


Рис.1. Общая схема работы метода, имитирующего поведение стаи серых волков

## 4. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

**Шаг 1. Генерация начальной популяции.**

Шаг 1.1. Задать параметры метода:

число элементов в популяции  $NP$ ;

максимальное число итераций  $K$ ;

Положить  $k = 1$  (счетчик числа итераций).



Шаг 1.2. Используя равномерный закон распределения на множестве, сгенерировать начальную популяцию

$$I_k = \{x^j(k) = (x_1^j(k), x_2^j(k), \dots, x_n^j(k))^T, j = 1, \dots, NP\} \subset D.$$

Шаг 1.3. Для каждого волка в стае вычислить соответствующее значение целевой функции:  $f(x^1(k)), \dots, f(x^{NP}(k))$ .

Шаг 1.4. Среди сгенерированных частиц найти три наилучших решения, которым соответствуют наибольшие значения целевой функции:

$x^\alpha = \arg \max_{j \in \{1, \dots, NP\}} f(x^j(k))$  – лучшее решение;  $x^\beta$  – второе по величине функции;  $x^\gamma$  – третье по величине функции.

### Шаг 2. Вычисление параметров.

Для каждого волка в стае с номером  $j$  найти:

- а)  $a$  – вектор с одинаковыми компонентами  $a_i = 2(1 - \frac{k}{K})$ ,  $i = 1, \dots, n$ , или  $a_i = 2(1 - \frac{k^2}{K^2})$ ,  $i = 1, \dots, n$ , в зависимости от используемой модификации;
- б)  $A_\alpha^j, A_\beta^j, A_\gamma^j$  – векторы, определяемые по правилу  $A_m^j = 2a \otimes r_1 - a$ ,  $m = \alpha, \beta, \lambda$ ; где  $r_1$  –  $n$ -мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке  $[0, 1]$ ;  $\otimes$  – операция поэлементного произведения векторов по Адамару;
- в)  $C_\alpha^j, C_\beta^j, C_\gamma^j$  – векторы, определяемые по правилу  $C_m^j = 2r_2$ ,  $m = \alpha, \beta, \lambda$ , где  $r_2$  –  $n$ -мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке  $[0, 1]$ .

### Шаг 3. Генерация новой стаи.

Шаг 3.1. Найти новые положения волков в стае

$$D_\alpha^j(k) = |C_\alpha^j \otimes x^\alpha(k) - x^j(k)|,$$

$$D_\beta^j(k) = |C_\beta^j \otimes x^\beta(k) - x^j(k)|,$$

$$D_\gamma^j(k) = |C_\gamma^j \otimes x^\gamma(k) - x^j(k)|,$$

$$x^{j,1}(k+1) = x^\alpha(k) - A_\alpha^j \otimes D_\alpha^j(k),$$

$$x^{j,2}(k+1) = x^\beta(k) - A_\beta^j \otimes D_\beta^j(k),$$

$$x^{j,3}(k+1) = x^\gamma(k) - A_\gamma^j \otimes D_\gamma^j(k),$$

$$x^j(k+1) = \frac{x^{j,1}(k+1) + x^{j,2}(k+1) + x^{j,3}(k+1)}{3}, \quad j = 1, \dots, NP.$$

Шаг 3.2. Для каждого волка в стае вычислить соответствующее ему значение целевой функции:  $f(x^1(k+1)), \dots, f(x^{NP}(k+1))$ .

Шаг 3.3. Найти новые три наилучших решения, которым соответствуют наибольшие значения целевой функции:



$$x^\alpha = \arg \max_{j \in \{1, \dots, NP\}} f(x^j(k+1)),$$

$x^\beta$  – второе по величине функции;

$x^\gamma$  – третье по величине функции.

#### **Шаг 4. Проверка условия завершения поиска.**

Если  $k = K$ , то процесс поиска завершить, перейти к шагу 5, а иначе положить  $k = k + 1$  и перейти к шагу 2.

#### **Шаг 5. Выбор решения из последней популяции.**

Закончить работу алгоритма. В качестве решения (приближенного) задачи  $f(x^*) = \max_{x \in D} f(x)$  выбрать волка с положением  $x^\alpha$ , которому соответствует наибольшее значение целевой функции.

#### Замечание.

1. Компоненты векторов  $A_m^j$ , где  $m = \alpha, \beta, \lambda$ , являются случайными на отрезке  $[-a_i, a_i]$ , при этом  $a_i$  уменьшается от 2 до нуля. При  $|A_{m_i}^j| \geq 1$  волк может удаляться от жертвы, осуществляя исследование множества допустимых решений, а при  $|A_{m_i}^j| < 1$ ,  $i = 1, \dots, n$ , волк приближается к жертве, нападая на нее.
2. Если новое положение волка на шаге 3.1 не принадлежит множеству допустимых решений, следует генерировать параметры метода заново до тех пор, пока ограничения не будут выполнены. Второй способ – если какая-то компонента вышла за границы отрезка  $[a_i, b_i]$ , то в качестве нового значения выбрать соответствующую ближайшую границу.

## 5. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

На основе изложенного алгоритма разработана программа поиска глобального экстремума функций. Для ее создания использовалась среда разработки Microsoft Visual Studio, язык программирования C# [1].

Возможности программы позволяют изучить алгоритм метода, а также влияние параметров метода на результат его работы. В список выбираемых функций включены стандартные многоэкстремальные тестовые функции двух переменных, для которых известно точное решение.

На рис. 2 представлено главное окно метода, имитирующего поведение стаи серых волков, где пользователь может выбрать вид целевой функции, задать множество допустимых решений и параметры метода, просматривать результаты работы.

Разработанная программа предусматривает возможность анализа работы метода по шагам. На рис. 3 представлено окно пошаговой работы метода, где изображена общая схема метода, отображаются результаты работы после выполнения каждого шага (график изменения наилучшего значения целевой функции, а также графическое изображение популяции) и после завершения работы метода.

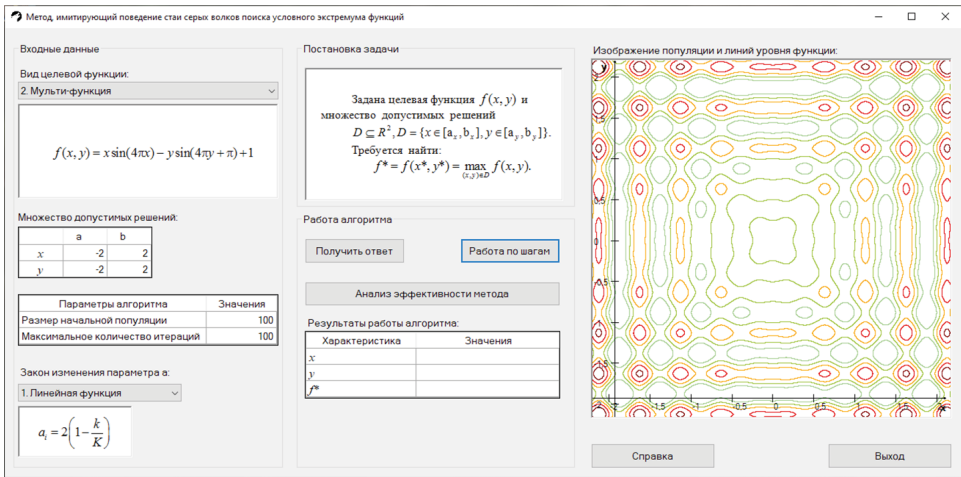


Рис. 2. Главное окно программы метода, имитирующего поведение стаи серых волков

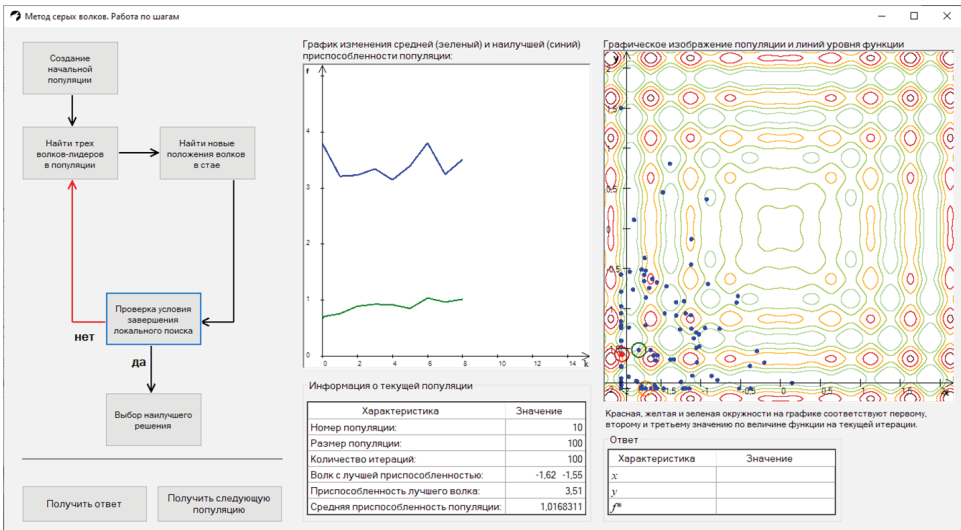


Рис. 3. Окно пошаговой работы метода, имитирующего поведение стаи серых волков

## 6. ТЕСТОВЫЕ ПРИМЕРЫ

В качестве тестовых функций использовались следующие функции (табл. 1).



Таблица 1

## Примеры стандартных тестовых функций

Название функции	Формула	Глобальный экстремум	Точки глобального экстремума
Корневая функция	$-\frac{1}{1+ z^6-1 }, z \in \mathbb{C}, z = x + iy$	1	$(-1,6288;-1,6288)^T$ ; $(1,6288;1,6288)^T$ ; $(-1,6288;1,6288)^T$ ; $(1,6288;-1,6288)^T$ .
Функция Розенброка	$-(1-x)^2 - 100(y-x^2)^2$	0	$(1;1)^T$
Функция Швевеля	$x \sin(\sqrt{ x }) + y \sin(\sqrt{ y })$	837,9657	$(420,9687;420,9687)^T$

**Пример 1.** Найдем глобальный максимум корневой функции. Зададим множество допустимых решений  $x, y \in [-2; 2]$ . Выберем следующие параметры алгоритма:

число элементов в популяции  $NP = 100$ ;

максимальное число итераций  $K_{\max} = 100$ .

На рис. 4 представлена популяция на начальной ( $k = 1$ ), промежуточных ( $k = 10$ ,  $k = 25$ ) и конечной ( $k = 200$ ) итерациях. Красная точка соответствует наиболее приспособленному решению – лидеру стаи. Красная, желтая и зеленая окружности на графике соответствуют лучшему, второму и третьему значению по величине функции на текущей итерации.

Результаты работы алгоритма:

наилучшее решение  $(x^*; y^*) = (0,9999; 0,0003)$ ;

значение целевой функции  $f(x^*, y^*) = 0,9984$ ;

отклонение от точного решения  $\Delta = 0,0016$ .

Графики изменения наилучшего и среднего значений целевой функции представлены на рис. 5.

**Пример 2.** Найдем глобальный максимум функции Розенброка. Зададим множество допустимых решений  $x \in [-3; 3]$ ,  $y \in [-1; 5]$ . Выберем следующие параметры алгоритма:

число элементов в популяции  $NP = 50$ ;

максимальное число итераций  $K_{\max} = 200$ .

На рис. 6 представлена популяция на начальной ( $k = 1$ ), промежуточных ( $k = 10$ ,  $k = 25$ ) и конечной ( $k = 200$ ) итерациях. Красная точка соответствует наиболее приспособленному решению – лидеру стаи. Красная, желтая и зеленая окружности на графике соответствуют наилучшему, второму и третьему значению по величине функции на текущей итерации.



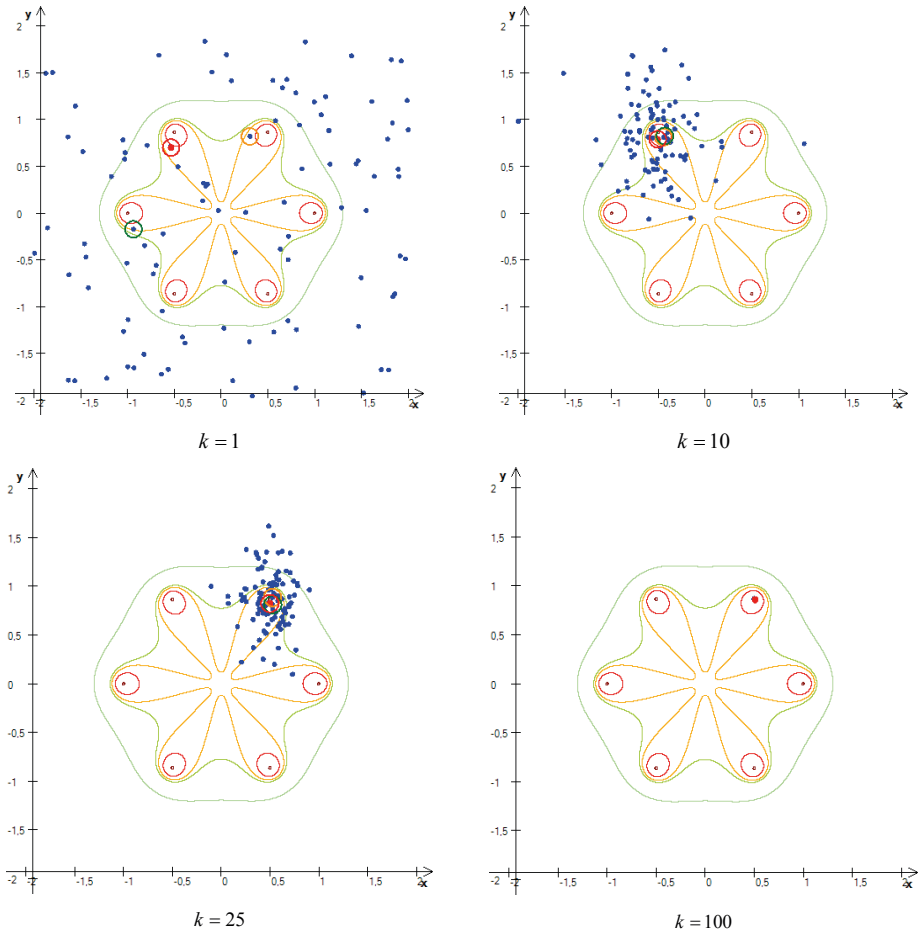


Рис. 4. Начальная, промежуточные и конечная итерации

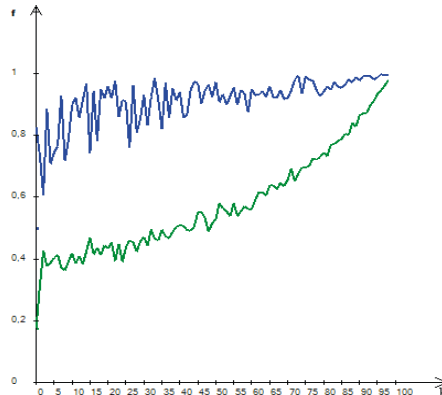


Рис. 5. Графики изменения наилучшего (синий) и среднего значения (зеленый) целевой функции



Результаты работы алгоритма:

наилучшее решение  $(x^*; y^*) = (1, 0056; 1, 0113)$ ;

значение целевой функции  $f(x^*, y^*) = 0$ ;

отклонение от точного решения  $\Delta = 0$ .

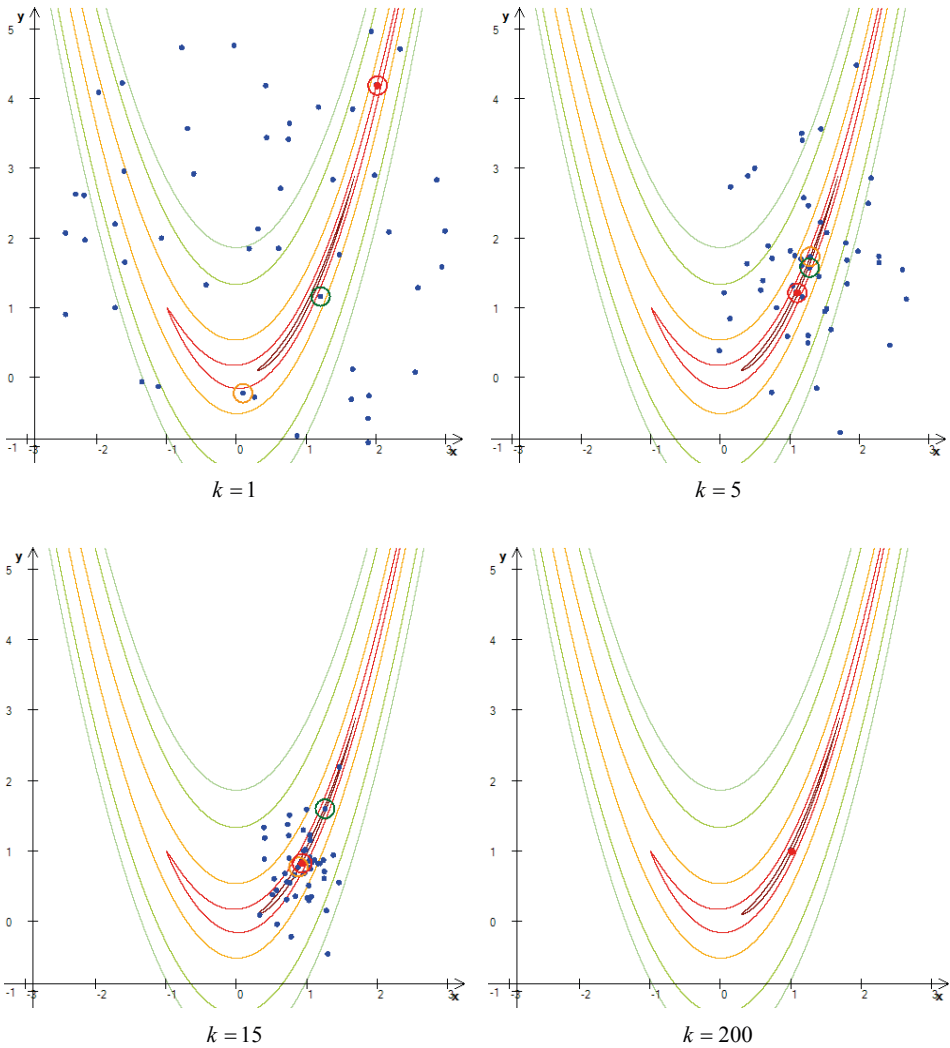


Рис. 6. Начальная, промежуточные и конечная популяции

Графики изменения наилучшего и среднего значений целевой функции представлены на рис. 7.

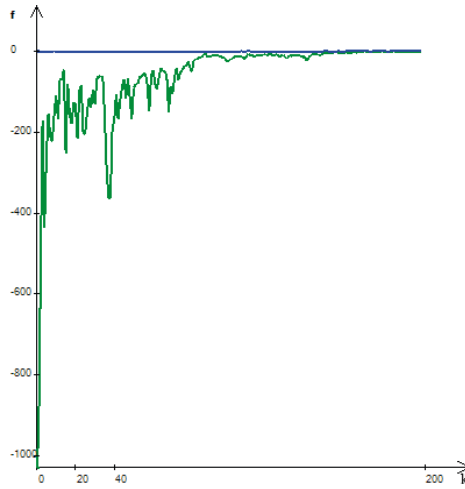


Рис. 7. Графики изменения наилучшего (синий) и среднего значения (зеленый) целевой функции

Рассмотрим подробнее процесс перехода от одной популяции к другой. Формирование новой популяции состоит из трех основных этапов. На первом этапе происходит вычисление значения целевой функции для каждого волка.

Второй этап содержит в себе выбор трех наиболее приспособленных особей. Наиболее приспособленная особь (ей соответствует наибольшее значение целевой функции) с положением  $x^{\alpha}$  окрашивается в красный цвет и выделяется красным кругом. Вторая по приспособленности особь с положением  $x^{\beta}$  выделяется оранжевым кругом. Третья по величине целевой функции особь с положением  $x^{\gamma}$  выделяется зеленым кругом.

Третий этап соответствует движению волков вокруг трех лучших особей, выделенных на втором этапе. В ходе движения, как лучшие, так и все остальные особи меняют свое положение. Процесс перехода от одной популяции к другой изображен на рис. 8.

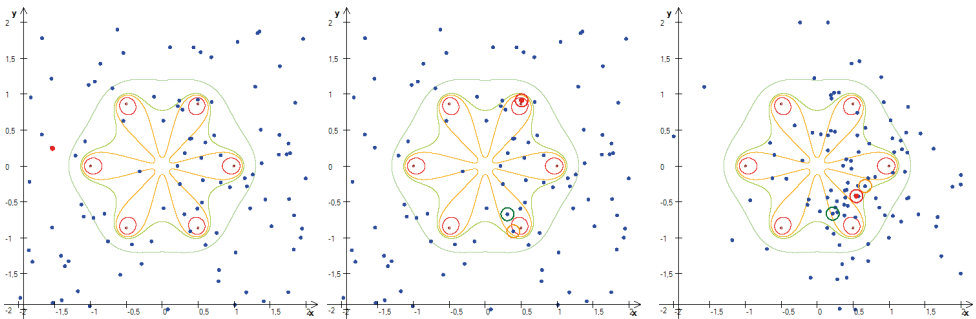


Рис. 8. Процесс перехода от одной популяции к другой



## 7. АНАЛИЗ ЭФФЕКТИВНОСТИ МЕТОДА

### 7.1. Рекомендации по выбору параметров

**Размер популяции**  $NP$  определяет количество вычислений целевой функции на каждой итерации. Для задачи с большой областью допустимых решений рекомендуется выбирать большее значение параметра  $NP$ . Рекомендуемые значения параметра  $NP \in [100; 1000]$ .

**Число итераций**  $ITER$  определяет, как долго будет продолжаться поиск новых решений. Чем больше величина  $ITER$ , тем более точным будет решение. Для рассмотренного набора стандартных тестовых функций рекомендуемые значения  $ITER \in [200; 1500]$  в зависимости от сложности целевой функции.

### 7.2. Анализ работы метода при различных значениях параметров

В данном разделе приводится статистический анализ и сравнение работы метода, имитирующего поведение стаи серых волков, при различных значениях его параметров. Исследуемый метод применялся к некоторым функциям из набора тестовых функций (табл. 1). Для каждой функции проводились серии из 100 решений одной и той же задачи с одними и теми же значениями параметров. Для полученной выборки  $\{f^1, f^2, \dots, f^{100}\}$  вычислялись среднее значение отклонения полученного решения от точного  $\bar{\Delta f} = \frac{1}{100} \sum_{i=1}^{100} \Delta f_i$ , где  $\Delta f_i = |f(x^*) - f^i|$ ; наименьшее значение отклонения  $\Delta f_{best} = \min_i \Delta f_i$ ; среднеквадратическое отклонение  $\bar{\sigma}_f = \sqrt{\bar{S}_{100}}$ , где  $\bar{S}_{100} = \frac{1}{100} \sum_{i=1}^{100} (\Delta f_i - \bar{\Delta f})^2$ ; количество успехов  $n_{усп}$  (успехом считалось попадание лучшей точки в  $\varepsilon$ -окрестность точного решения,  $\varepsilon = \frac{\max_{i=1, \dots, n} |b_i - a_i|}{1000}$ ). Результаты, полученные для каждой функции, представлены в табл. 2–3.

Таблица 2

Влияние параметров метода. Корневая функция

Параметры метода		$\bar{\Delta f}$	$\Delta f_{best}$	$\bar{\sigma}_f$	$n_{усп}$
$NP$	$ITER$				
100	100	0,005039	0,000284	0,003659	100
200	100	0,002689	0,000130	0,001656	100
200	200	0,001483	0,000057	0,001414	100
500	100	0,001567	0,000026	0,000881	100
500	500	0,000245	0,000002	0,000212	100
500	200	0,000720	0,000007	0,000606	100
1000	200	0,000440	0,000010	0,000329	100
50	50	0,012814	0,003722	0,007206	91
100	50	0,009087	0,000778	0,006744	96
80	80	0,007032	0,000430	0,004969	99



Таблица 3

**Влияние параметров метода. Функция Швепеля**

Параметры метода		$\overline{\Delta f}$	$\Delta f_{best}$	$\overline{\sigma_f}$	$n_{ycn}$
<i>NP</i>	<i>ITER</i>				
100	100	56,868254	0,000956	61,493337	53
200	100	34,356351	0,000346	53,740254	71
200	200	20,137841	0,000102	44,488405	83
500	100	8,296381	0,000041	30,218292	93
500	500	0,000235	0,000041	0,000207	100
500	200	3,554438	0,000041	20,204130	97
1000	200	0,000525	0,000041	0,000447	100
1000	1000	0,000053	0,000041	0,000026	100
500	1000	1,184456	0,000041	11,784464	99
1000	200	0,000719	0,000041	0,000680	100

Анализируя работу метода, имитирующего поведение стаи серых волков, было установлено, что алгоритм успешно находит экстремум большинства тестовых функций. Вариация параметров позволяет в серии из 100 запусков найти глобальный максимум функций с высокой точностью.

Алгоритм испытывает трудности при нахождении экстремума функций Швепеля, Шаффера и Розенброка. Однако данная проблема решается благодаря увеличению числа итераций или увеличению популяции стаи.

Проведенный анализ позволяет рекомендовать данный алгоритм для решения разнообразных задач теории управления [9].

**Литература**

1. Пantelev A.V., Skavinskaya D.V. Метаэвристические алгоритмы глобальной оптимизации. М.: Вузовская книга, 2019.
2. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. М.: Изд-во МГТУ им. Н.Э. Баумана, 2014.
3. Гладков В.А., Курейчик В.В. Биоинспирированные методы в оптимизации. М.: Физматлит, 2009.
4. Glover F.W., Kochenberger G.A. (eds.). Handbook of Metaheuristics. Boston, MA: Kluwer Academic Publishers, 2003
5. Floudas C.A., Pardalos P.M. (eds.). Encyclopedia of Optimization. Springer, 2009.
6. Clerc M. Particle swarm optimization. ISTE Ltd, 2006.
7. Mirjalili S., Mirjalili S.M., Lewis A. Grey wolf optimizer //Advances in Engineering Software. 2014. Vol. 69. P. 46–61.
8. Mittal N., Singh U., Sohi B.S. Modified grey wolf optimizer for global engineering optimization // Applied Computational Intelligence and Soft Computing. 2016. Article ID 7950348. <http://dx.doi.org/10.1155/2016/7950348>
9. Пantelev A.V. Метаэвристические алгоритмы оптимизации законов управления динамическими системами. М.: Факториал, 2020.



## Developing Software of Global Optimization Method Based on Grey Wolf Optimizer

**Andrei V. Panteleev\***

MAI (National Research University), Moscow, Russia

ORCID: <https://orcid.org/0000-0003-2493-3617>

e-mail: [avpanteleev@inbox.ru](mailto:avpanteleev@inbox.ru)

**Ivan A. Belyakov\*\***

MAI (National Research University), Moscow, Russia

ORCID: <https://orcid.org/0000-0003-3457-9914>

e-mail: [vanbelyakov@yandex.ru](mailto:vanbelyakov@yandex.ru)

This article discusses the development of software that allows to simulate the algorithm of the “Grey Wolf Optimizer” method. This algorithm belongs to the class of metaheuristic algorithms that allow finding a global extremum on a set of admissible solutions. This algorithm is being the most efficiently used in a situation where the cost function is specified in the form of a black box. The algorithm belongs to both bioinspired algorithms and to the class of algorithms of Particle Swarm Optimization. To analyze the efficiency of the algorithm, software was created that allows to vary the parameters of the method. The article contains examples of the program’s work on various test functions. The purpose of the program is to collect and analyze statistical results, making possible to evaluate the final result. The program provides to build graphs that make it possible to make a more thorough assessment of the results obtained. The program has a step-by-step function that allows one to analyze the specifics and features of the algorithm. Analysis of statistical data provides more detailed selection of the parameters of the algorithm.

**Keywords:** global optimization algorithm, metaheuristic algorithm, software, GWO, Grey Wolf Optimizer

### For citation:

Panteleev A.V., Belyakov I.A. Developing Software of Global Optimization Method Based on Grey Wolf Optimizer. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2021. Vol. 11, no. 2, pp. 59–73. DOI: <https://doi.org/10.17759/mda.2021110204> (In Russ., abstr. in Engl.).

\***Panteleev Andrei Vladimirovich**, Doctor of Physics and Mathematics, Professor, Head of the Department of Mathematics and Cybernetics, Institute of Information Technologies and Applied Mathematics, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: <https://orcid.org/0000-0003-2493-3617>, e-mail: [avpanteleev@inbox.ru](mailto:avpanteleev@inbox.ru)

\*\***Belyakov Ivan Andreevich**, Undergraduate Student of the Institute of Information Technologies and Applied Mathematics, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: <https://orcid.org/0000-0003-3457-9914>, e-mail: [vanbelyakov@yandex.ru](mailto:vanbelyakov@yandex.ru)



### **References**

1. Panteleev A.V., Skavinskaya D.V. *Metaevristichskie algoritmy global'noj optimizacii*. M.: Vuzovskaya kniga, 2019.
2. Karpenko A.P. *Sovremennye algoritmy poiskovoj optimizacii*. Algoritmy, vdohnovlennyye prirodoy. M.: MGTU im. N.E. Baumana, 2014.
3. Gladkov V.A., Kurejchik V.V. *Bioinspirirovannyye metody v optimizacii*. M.: Fizmatlit, 2009.
4. Glover F.W., Kochenberger G.A. (eds.). *Handbook of Metaheuristics*. Boston, MA: Kluwer Academic Publishers, 2003
5. Floudas C.A., Pardalos P.M. (eds.). *Encyclopedia of Optimization*. Springer, 2009.
6. Clerc M. *Particle swarm optimization*. ISTE Ltd, 2006.
7. Mirjalili S., Mirjalili S.M., Lewis A. Grey wolf optimizer // *Advances in Engineering Software*. 2014. Vol. 69. P. 46–61.
8. Mittal N., Singh U., Sohi B.S. Modified grey wolf optimizer for global engineering optimization // *Applied Computational Intelligence and Soft Computing*. 2016. Article ID 7950348. <http://dx.doi.org/10.1155/2016/7950348>
9. Panteleev A.V. *Metaevristicheskie algoritmy optimizacii zakonov upravleniya dinamicheskimi sistemami*. M.: Faktorial, 2020.