

# МЕТОД, ИМИТИРУЮЩИЙ ПОВЕДЕНИЕ СТАИ СЕРЫХ ВОЛКОВ

## 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

$$f(x^*) = \max_{x \in D} f(x), \quad (1)$$

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.

## Стратегия поиска решения

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.

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:

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

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

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

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

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

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

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

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  $A_m^j = 2a \otimes r_1 - a$ ,  $m = \alpha, \beta, \gamma$ ;  $r_1$  –  $n$ -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  $C_m^j = 2r_2$ ,  $m = \alpha, \beta, \gamma$ ,  $r_2$  –  $n$ -мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке  $[0,1]$ . Имеется модификация, в которой  $a_i = 2(1 - \frac{k^2}{K^2})$ ,  $i = 1, \dots, n$ .

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

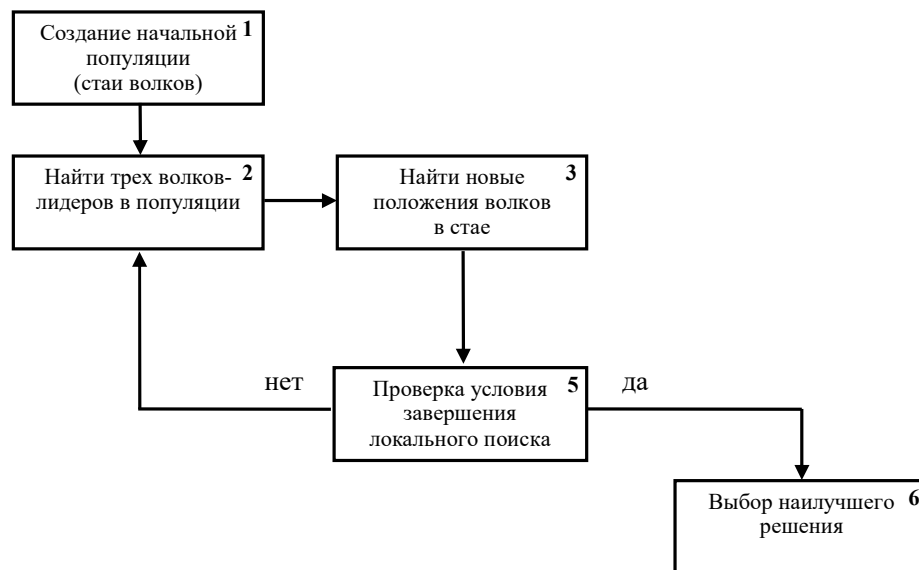


Рис. 1. Общая схема работы метода, имитирующего поведение стаи серых волков

### 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  $D$ , 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:

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

$$x^\gamma \text{ – третье по величине функции.}$$

## 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

$$A_m^j = 2a \otimes r_1 - a, \quad m = \alpha, \beta, \gamma;$$

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

$$C_m^j = 2r_2, \quad m = \alpha, \beta, \gamma,$$

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.

**З а м е ч а н и е.**

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 границу.

#### Список использованных источников

1. Mirjalili S., Mirjalili S.M., Lewis A. Grey wolf optimizer //Advances in Engineering Software. 2014. Vol. 69. P. 46–61.
2. 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>