

МЕТОД, ИМИТИРУЮЩИЙ ПОВЕДЕНИЕ СТАИ ГОРБАТЫХ КИТОВ

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.

2. Стратегия поиска решения

Метод горбатых китов (Whales Optimization Algorithm – WOA) имитирует охоту стаи горбатых китов за крилем или мелкой рыбой, используя уникальный способ преследования и окружения цели – метод пузырьковой сетки.

В начале работы метода случайным образом, используя предположение о равномерном распределении, на множестве допустимых решений D генерируется некоторый набор начальных точек (горбатых китов 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 кита с номером j , NP – количество горбатых китов 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 лидера, полагая, что они обладают большей информацией о положении жертвы (точке экстремума).

В стае китов положение каждого кита характеризуется относительно лидера стаи. В процессе реализации охоты с помощью пузырьковой сетки стая китов окружает добычу, постепенно приближаясь к жертве. Одновременно с этим совершают спиралевидное движение вокруг добычи.

На каждой итерации для всех особей генерируется число p , равномерно распределенное на отрезке $[0, 1]$.

Если $p < 0,5$, то подсчитывается $|A^j|$. В случае, когда $|A^j| < 1$, используется формула, по которой положение китов определяется относительно расположения жертвы:

$$x^j(k+1) = x^*(k) - A^j \otimes D^j(k),$$
$$D^j(k) = |C^j \otimes x^*(k) - x^j(k)|,$$

где $x^*(k)$ – лучшая особь на текущей итерации, \otimes – операция поэлементного произведения векторов по Адамару, k – номер итерации, $x^j(k+1)$, $x^j(k)$ – следующее и текущее положения китов, $j = 1, \dots, NP$; A^j – вектор, определяемые по правилу

$A^j = 2a \otimes r_1 - a$, 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^j – вектор, определяемый по правилу $C^j = 2r_2$, r_2 – n -мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0,1]$. Имеется модификация, в которой $a_i = 2(1 - \frac{k^2}{K^2})$, $i = 1, \dots, n$.

Если $|A^j| \geq 1$, то реализуется исследование на множестве допустимых решений:

$$x^j(k+1) = x_{rand}(k) - A \otimes D^j,$$

$$D^j(k) = |C^j \otimes x_{rand}(k) - x^j(k)|,$$

где $x_{rand}(k)$ – случайно выбранная особь на текущей итерации.

Если $p \geq 0,5$, то положение кита определяется по формуле:

$$x^j(k+1) = D^{*j} \cdot e^{bl} \cdot \cos(2\pi l) + x^*(k),$$

$$D^{*j}(k) = |x^*(k) - x^j(k)|,$$

где l – случайная величина, распределенная равномерно на отрезке $[-1,1]$, b – параметр формы логарифмической спирали.

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

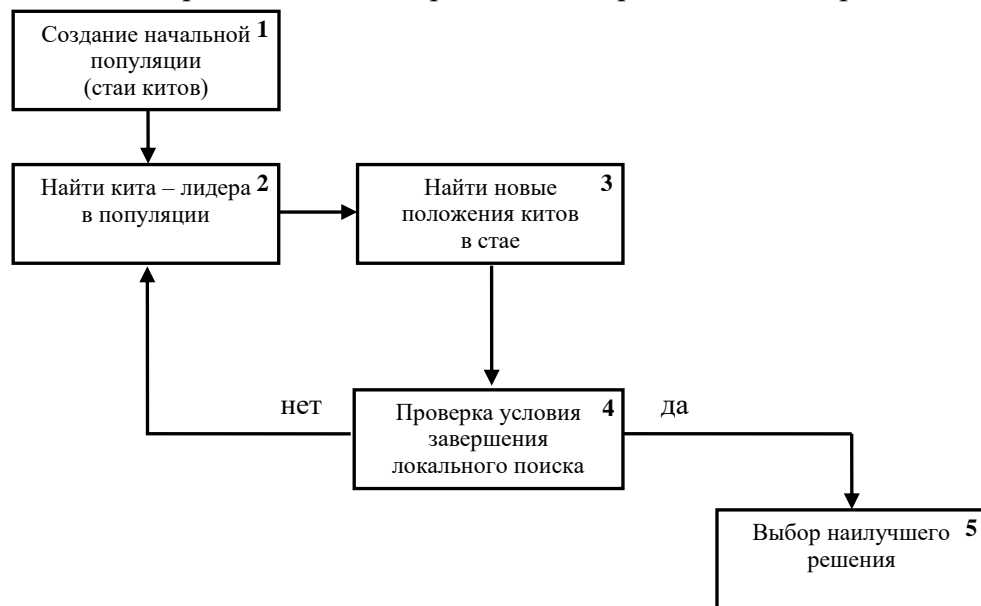


Рис. 1. Общая схема работы метода, имитирующего поведение стаи горбатых китов

3. Алгоритм решения задачи

Шаг 1. Генерация начальной популяции.

Шаг 1.1. Задать параметры метода:

- число элементов в популяции NP ;
- максимальное число итераций K ;

- параметр формы логарифмической спирали b ;

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 кита в стае вычислить соответствующее значение целевой функции: $f(x^1(k)), \dots, f(x^{NP}(k))$.

Шаг 1.4. Среди сгенерированных частиц найти наилучшее решение, которому соответствует наибольшее значение целевой функции:

$$x^* = \arg \max_{j \in \{1, \dots, NP\}} f(x^j(k)) - \text{лучшее решение}$$

Шаг 2. Вычисление параметров.

Для каждого горбатого кита 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) p – случайную величину, распределенную равномерно на отрезке $[0, 1]$;

в) если $p < 0.5$, то вычислить:

A^j – вектор, определяемый по правилу:

$$A^j = 2a \otimes r_1 - a,$$

где r_1 – n -мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0, 1]$; \otimes – операция поэлементного произведения векторов по Адамару;

C^j – вектор, определяемый по правилу

$$C^j = 2r_2,$$

где r_2 – n -мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке $[0, 1]$;

г) если $p \geq 0.5$, то сгенерировать

l – случайную величину, описанную равномерным распределением на отрезке $[-1, 1]$.

Шаг 3. Генерация новой стаи.

Шаг 3.1. Найти новые положения китов в стае

Если $p < 0.5$, то подсчитывается $|A^j|$. Если $|A^j| < 1$, то новое положение особи вычисляется по формуле:

$$x^j(k+1) = x^*(k) - A^j \otimes D^j(k),$$

где $D^j(k) = |C^j \otimes x^*(k) - x^j(k)|$, $x^*(k)$ – положение наиболее приспособленной особи.

При $|A^j| \geq 1$, положение особи определяется согласно формуле:

$$x^j(k+1) = x_{rand}(k) - A \otimes D^j,$$

где $D^j(k) = |C^j \otimes x_{rand}(k) - x^j(k)|$, $x_{rand}(k)$ – положение случайно выбранной особи на текущей итерации.

Если $p \geq 0,5$, то положение определяется с помощью формулы спиралевидного движения:

$$x^j(k+1) = D^{*j} \cdot e^{bl} \cdot \cos(2\pi l) + x^*(k),$$

где l – случайная величина, распределенная равномерно на отрезке $[-1,1]$, b – параметр формы логарифмической спирали, $D^{*j} = |x^*(k) - x^j(k)|$.

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

Шаг 3.3. Среди сгенерированных частиц найти наилучшее решение, которому соответствует наибольшее значение целевой функции:

$$x^* = \arg \max_{j \in \{1, \dots, NP\}} f(x^j(k)) \text{ – лучшее решение}$$

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 горбатого кита с положением x^* , которому соответствует наибольшее значение целевой функции.

З а м е ч а н и е.

1. Компоненты вектора A^j являются случайными на отрезке $[-a_i, a_i]$, при этом a_i уменьшается от 2 до нуля.

2. Если новое положение кита на шаге 3.1 не принадлежит множеству допустимых решений, следует генерировать параметры метода заново до тех пор, пока ограничения не будут выполнены. Вторым способом – если какая-то компонента вышла за границы отрезка $[a_i, b_i]$, то в качестве нового значения выбрать соответствующую границу.