Алгоритм дифференциальной эволюции

Было предложено реализовать алгоритм дифференциальной эволюции из следующей статьи. Этот алгоритм должен решать оптимизационную задачу поиска оптимальных значений параметров функции, чтобы её значение было максимальным или минимальным. То есть, производится поиск глобального максимума или минимума.

Оригинальный алгоритм поиска светлячком был представлен в 1995 году и использует механизм эволюции популяции, чтобы найти оптимальные значения параметров функции. Каждый кандидат представлен вектором значений параметров, которые влияют на функцию, которую мы пытаемся оптимизировать. Вначале алгоритма, популяция создаётся случайным образом в заданных границах значений параметров.

Шаги алгоритма DE:

1. Сгенерировать случайную популяцию решений в виде векторов (или матриц).
2. Выбрать три случайных решения из популяции и выполнить операцию скрещивания (комбинации).
3. Полученное решение (потомок) сравнивается с решением-родителем. Если потомок имеет лучшее значение целевой функции, то он заменяет родителя в популяции. Иначе родитель остается в популяции.
4. Повторять шаги 2 и 3 до достижения критерия останова (максимальное число итераций, достижение определённого значения функции цели и т.д.).
5. Возвращаем лучшее решение, найденное в популяции.

Операция мутации происходит вариантом “rand/1/bin”, который работает следующим образом:

1. Для каждого индивида выбираются три случайных индивида из популяции, и на их основе генерируется мутант.
2. Мутант вычисляется как сумма разности векторов двух индивидов, умноженной на коэффициент F, и третьего случайно выбранного индивида.
3. Для каждой компоненты мутанта случайно выбирается, будут ли использоваться значения из мутанта или из оригинального индивида. Это создаёт новый индивид, который затем используется для замены исходного индивида, если он оказался лучше.

Тестирование алгоритма проводилось на 8 функциях, рассмотрим каждую из них.

1. Функция Розенброка

Это функция, которая определяется следующим образом:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

где x и y - переменные функции.

Глобальный минимум функции находится в точке (x, y) = (1, 1), и его значение равно 0.

2. Функция Де Джонга

Это функция, которая определяется следующим образом:

$$f(x, y) = x^2 + y^2$$

где x и y - переменные функции.

Глобальный минимум функции находится в точке (x, y) = (0, 0), и его значение равно 0.

3. Функция Швефеля

Это функция, которая определяется следующим образом:

$$f(x_1, x_2, ..., x_n) = - \sum_i x_i \sin(\sqrt{(|x_i|)})$$

где $x_1, x_2, ..., x_n$ - переменные функции.

Глобальный минимум функции находится в точке, где $x_1 = x_2 = ... = x_n = 0$, и его значение равно 0.

4. Функция суммы разных степеней

Это функция, которая определяется следующим образом:

$$f(x_1, x_2, ..., x_n) = \sum_i |x_i|^n$$

где $x_1, x_2, ..., x_n$ - переменные функции, а n - положительное число.

Глобальный минимум функции находится в точке $(x_1, x_2, ..., x_n) = (0, 0, ..., 0)$, и его значение равно 0.

5. Функция Бута

Это функция, которая определяется следующим образом:

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

где x и y - переменные функции.

Глобальный минимум функции находится в точке (x, y) = (1, 3), и его значение равно 0.

6. Функция Била

Это функция, которая определяется следующим образом:

$$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$$

где x и y - переменные функции.

Глобальный минимум функции находится в точке (x, y) = (3, 0.5), и его значение равно 0.

7. Функция Гольдштейна-Прайса

Это функция, которая определяется следующим образом:

$$f(x, y) = [1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)][30 + (2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)]$$

где x и y - переменные функции.

Глобальный минимум функции находится в точке (x, y) = (-0.54719, -1.54719), и его значение равно 3.

8. Функция Захарова

Это функция, которая определяется следующим образом:

$$f(x_1, x_2, ..., x_n) = \sum_i x_i^2 + (0.5\sum_i ix_i)^2 + (0.5\sum_i ix_i)^4$$

где $x_1, x_2, ..., x_n$ - переменные функции.

Глобальный минимум функции находится в точке, где $x_1 = x_2 = ... = x_n = 0$, и его значение равно 0.

Результаты тестирования алгоритма, написанного на Python с использованием Numpy и Numba и на C с упаковкой в dll, которая позже запускается из Python-кода с использованием модуля ctype, на функциях, приведённых выше, показаны в первых двух секциях блокнота, вместе с сравнением времени выполнения двух реализаций. В третьей секции блокнота показано сравнение времени выполнения версии без Numba с версией, использующей Numba, а так же их действие на функциях с большим количеством измерений.