

Оптимизация методом Spider Wasp Optimizer (SWO)

Кривнюк Константин

гр. 9302

Аннотация

В данном реферате представлен алгоритм Spider Wasp Optimizer (SWO), вдохновленный охотничьим поведением пауков-ос. Рассматривается структура проекта, основные этапы алгоритма и его применение к оптимизационным задачам. Приведены псевдокод алгоритма, тестовые функции и результаты сходимости.

Содержание

1	Введение	2
2	Структура проекта	3
3	Описание алгоритма	4
3.1	Параметры алгоритма	4
3.2	Математическое описание алгоритма	4
3.3	Псевдокод	5
4	Тестовые функции	6
4.1	Примеры графиков функций	7
5	Результаты	8
5.1	Графики сходимости	8
5.2	Результаты оптимизации	8
6	Заключение	9

1 Введение

Spider Wasp Optimizer (SWO) — это инновационный метаэвристический алгоритм, имитирующий охотничьи и спаривательные стратегии пауков-ос. Его ключевые преимущества — высокая эффективность в глобальной оптимизации и гибкость для широкого класса задач.

Цель данного проекта — разработать реализацию SWO, протестировать алгоритм на стандартных тестовых функциях и проанализировать его сходимость.

2 Структура проекта

Проект имеет следующую структуру:

```
├── report
│   ├── images
│   ├── report.pdf
│   └── report.tex
├── sw_optimizer
│   ├── __init__.py
│   └── sw_optimizer.py
├── test_functions
│   ├── __init__.py
│   ├── ackley.py
│   ├── bukin_function_n6.py
│   ├── eggholder_function.py
│   ├── himmelblau.py
│   ├── rastrigin.py
│   ├── rosenbrock.py
│   ├── schwefel_function.py
│   └── sphere.py
├── utils
│   ├── __init__.py
│   ├── initialization.py
│   ├── levy_flight.py
│   └── main.py
├── README.md
└── requirements.txt
```

3 Описание алгоритма

Spider Wasp Optimizer включает в себя две основные стратегии:

- **Охота** — поведение пауков, направленное на исследование поискового пространства.
- **Спаривание** — обмен информацией для улучшения текущих решений.

3.1 Параметры алгоритма

- **Размер популяции** (`search_agents_no`) — количество особей в популяции.
- **Количество итераций** (`Tmax`) — максимальное число итераций.
- **Границы поиска** (`lb`, `ub`) — нижняя и верхняя границы пространства поиска.
- **Функция приспособленности** (`fobj`) — целевая функция оптимизации.

3.2 Математическое описание алгоритма

Алгоритм SWO основан на математическом моделировании поведения пауков-ос. Основные уравнения, используемые в алгоритме, включают:

1. ****Обновление параметров:****

$$a = 2 - 2 \left(\frac{t}{Tmax} \right)$$

$$a2 = -1 - 1 \left(\frac{t}{Tmax} \right)$$

$$k = 1 - \frac{t}{Tmax}$$

2. ****Обновление позиций:****

$$C = a(2r1 - 1)$$

$$l = (a2 - 1) \cdot \text{rand} + 1$$

$$B = \frac{1}{1 + \exp(l)}$$

$$m1 = |\text{rn1}| \cdot r1$$

$$m2 = B \cdot \cos(l \cdot 2\pi)$$

3. ****Обновление позиций агентов:****

$$\text{Positions}(i, j) = \text{Positions}(i, j) + m1 \cdot (\text{Positions}(JK(1), j) - \text{Positions}(JK(2), j))$$

$$\text{Positions}(i, j) = \text{Positions}(JK(i), j) + m2 \cdot (lb(j) + \text{rand} \cdot (ub(j) - lb(j)))$$

$$\text{Positions}(i, j) = \text{Positions}(i, j) + C \cdot |2 \cdot \text{rand} - 1| \cdot \text{Positions}(JK(3), j) - \text{Positions}(i, j)$$

$$\text{Positions}(i, j) = \text{Positions}(i, j) \cdot vc(j)$$

4. ****Обновление позиций при спаривании:****

$$SW_m(j) = \text{Positions}(i, j) + (\exp(l)) \cdot |\text{rn1}| \cdot v1(j) + (1 - \exp(l)) \cdot |\text{rn2}| \cdot v2(j)$$

3.3 Псевдокод

Основные шаги алгоритма SWO приведены в псевдокоде (см. Алгоритм 1).

Input: search_agents_no, Tmax, ub, lb, dim, fobj, tol, max_stall

Output: Best_score, Best_position, Convergence_curve

begin

Инициализация параметров;

 Создать начальную популяцию и вычислить значения функции приспособленности;

for $t = 1$ **do** $Tmax$ **do**

 Обновить параметры алгоритма (скорость, направление);

 Применить стратегии охоты и спаривания для обновления позиций;

 Обновить лучшее решение и значение функции;

 Если критерий остановки выполнен — выйти из цикла;

end

 Вернуть лучшее найденное решение.

end

Algorithm 1: Spider Wasp Optimizer (SWO)

4 Тестовые функции

Для тестирования SWO использовались следующие функции:

- **Ackley**: многомерная функция с большим числом локальных минимумов. Иллюстрация представлена на Рисунке 1.

$$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + \exp(1)$$

- **Bukin Function N.6**: функция с узким желобом, усложняющим поиск минимума. Иллюстрация на Рисунке 2.

$$f(x) = 100 \sqrt{|x_2 - 0.01x_1^2|} + 0.01|x_1 + 10|$$

- **Eggholder**: сложная поверхность с глубокими ямами. См. Рисунок 3.

$$f(x) = -(x_2 + 47) \sin \left(\sqrt{|x_2 + \frac{x_1}{2} + 47|} \right) - x_1 \sin \left(\sqrt{|x_1 - (x_2 + 47)|} \right)$$

- **Himmelblau**: двухмерная функция с несколькими минимумами. Рисунок 4.

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

- **Rastrigin**: функция с периодическими локальными минимумами. Рисунок ??.

$$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$$

- **Rosenbrock**: узкая долина, содержащая глобальный минимум. Рисунок ??.

$$f(x) = \sum_{i=1}^{n/2} [100(x_{2i} - x_{2i-1}^2)^2 + (x_{2i-1} - 1)^2]$$

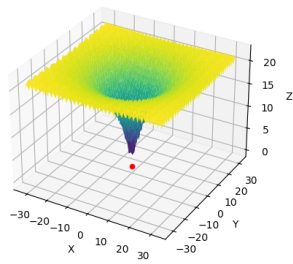
- **Schwefel Function**: функция с глубокими глобальными минимумами. Рисунок ??.

$$f(x) = 418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|})$$

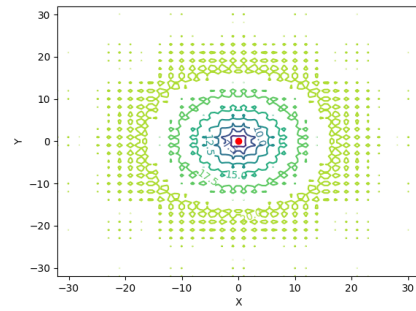
- **Sphere**: простая функция с параболической формой. Рисунок ??.

$$f(x) = \sum_{i=1}^n x_i^2$$

4.1 Примеры графиков функций



(a) Ackley function (3D).



(b) Ackley function (Contour).

Рис. 1: Ackley function visualizations.

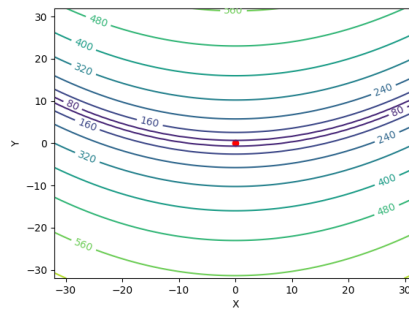


Рис. 2: Bukin Function N.6.

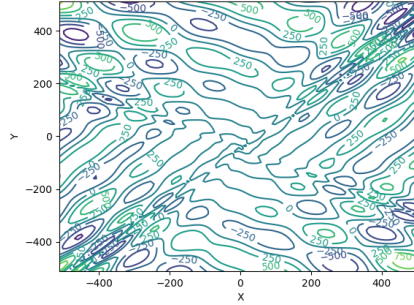
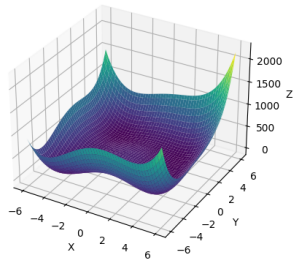
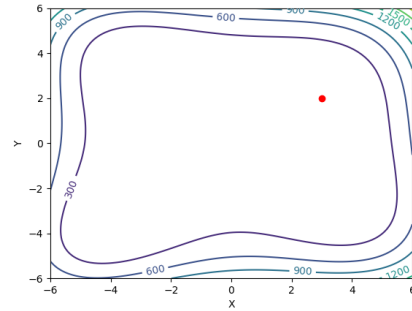


Рис. 3: Eggholder function.



(a) Himmelblau function (3D).



(b) Himmelblau function (Contour).

Рис. 4: Himmelblau function visualizations.

5 Результаты

5.1 Графики сходимости

Графики сходимости SWO на различных тестовых функциях представлены на Рисунке 5.

5.2 Результаты оптимизации

Результаты оптимизации для различных тестовых функций представлены в таблице 1.

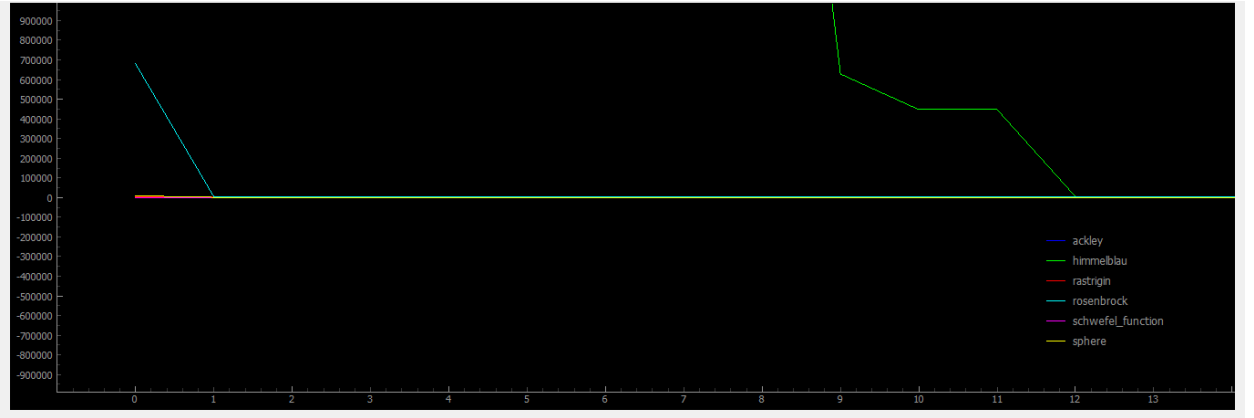


Рис. 5: Сходимость SWO на тестовых функциях.

Функция	Оптимальное значение (fmin)	Оптимальное решение (xmin)	Общее количество итераций
Sphere	1.70×10^{-59}	$[-3.69 \times 10^{-30}, 1.85 \times 10^{-30}]$	9225
Ackley	-4.44×10^{-16}	$[-3.68 \times 10^{-18}, -1.71 \times 10^{-16}]$	13585
Bukin N.6	1.00×10^{-1}	$[2.34 \times 10^{-19}, 4.83 \times 10^{-25}]$	17145
Eggholder	-894.58	$[-465.69, 385.72]$	12445
Himmelblau	4.93×10^{-17}	$[-2.81, 3.13]$	18145
Rastrigin	0.0	$[7.67 \times 10^{-10}, -4.84 \times 10^{-10}]$	10925
Rosenbrock	0.0	$[1.0, 1.0]$	16805
Schwefel	2.55×10^{-5}	$[420.97, 420.97]$	13085

Таблица 1: Результаты оптимизации для различных тестовых функций.

6 Заключение

Spider Wasp Optimizer демонстрирует высокую эффективность при решении задач оптимизации. Алгоритм успешно протестирован на различных тестовых функциях. Будущие работы направлены на улучшение адаптивности алгоритма, а также его применение в реальных задачах.