МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ.ІГОРЯ СІКОРСЬКОГО»

НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Лабораторна робота №2

«Реалізація алгоритму оптимізації роєм часток для пошуку глобального мінімуму функції»

Виконав:
Студент 3 курсу
Групи ФІ-21
Голуб Михайло

Перевірив:
Железняков. Д. О.

Київ 2025

ЗМІСТ

1. ЗАВДАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Ознайомитись з теоретичними відомостями до методу роєм часток. Розробити програмне забезпечення для розв’язання задач оптимізації (використовувати готові рішення заборонено).
2. Дослідити основні властивості алгоритму оптимізації роєм часток на прикладі функцій пристосованості:
    - Ackley's function
    - Функція Розенброка
    - Cross-in-tray function
    - Hölder table function
    - McCormick function
    - Styblinski–Tang function
    - На додаткові бали можна запропонувати свій варіант використання генетичних алгоритмів.
3. Провести експерименти з різними параметрами алгоритму оптимізації роєм часток.
4. Для кожної з функції:
    - Побудувати графічне зображення цільової функції. Можна використовувати код з попередньої лабораторної роботи.
    - Показати процес пошуку глобального екстремуму. Тобто потрібно продемонструвати положення популяції на функції на кожній ітерації (В звіт можна вставити тільки для однієї функції).
    - Отримані залежності представити в графічному вигляді (по осі абсцис відкладається номер ітерації, а по осі ординат – найкраще значення функції). Для порівняння показати результати різних екскрементів (різні значення гіперпараметрів) на одному графіку.
5. Зробити звіт
6. Захистити роботу

2. ХІД РОБОТИ

2.1. Реалізація алгоритму оптимізації роєм часток

2.1.1. Словник параметрів

Для зменшення кількості та стандартизації аргументів функцій, методів і класів алгоритмів оптимізації, аргументи які стосуються безпосередньо конкретного алгоритму оптимізації будуть передаватись класу, методу чи функції в словнику.

Для алгоритму рою часток (далі – PSO) в словнику будуть міститись наступні гіпер параметри:

- `a1` – прискорення до найкращої позиції яку пам'ятає частинка;
- `a2` – прискорення до найкращої позиції яку пам'ятає весь рій;
- `pop_size` – кількість частинок;
- `dim` – кількість вимірів пошуку;
- `pos_min` – вектор мінімальних значень координат пошуку;
- `pos_max` – вектор максимальних значень координат пошуку;
- `speed_min` – вектор мінімальної (не за модулем) швидкості за кожною з координат;
- `speed_max` – вектор максимальної (не за модулем) швидкості за кожною з координат.

Фітнес-функція та аргумент `seeking_min` передаються як окремі аргументи.

2.1.2. Клас частинки

Клас `PSOParticle` містить інформацію про поточну позицію, найкращу позицію і швидкість частинки.

Метод `update` приймає на вхід найкращу позицію популяції і два вектори `r1`, `r2`. `update` оновлює позицію частинки згідно з гіперпараметрами і аргументами.

2.1.3. Клас рою

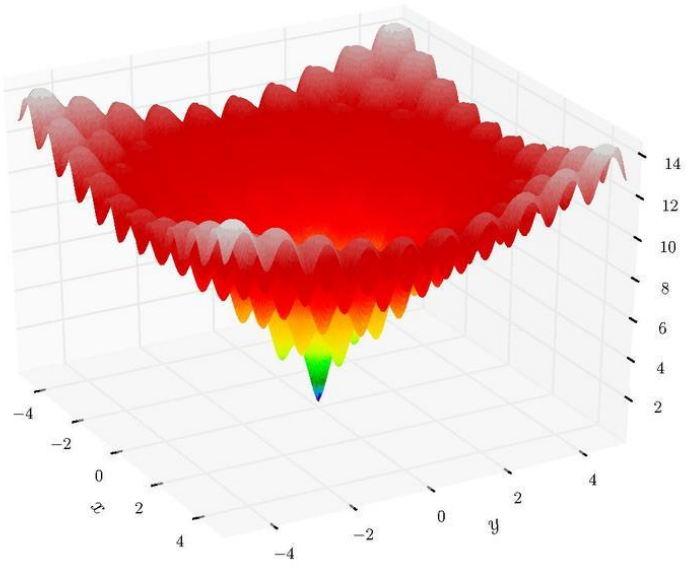
Клас PSOSolver ініціалізує необхідну кількість частинок і містить методи необхідні для пошуку оптимальних значень:

- `iter` – генерує вектори `r1`, `r2` і запускає `update` у кожній частинці рою;
- `solve` – запускає `iter` вказану кількість раз і повертає результат;
- `anisolve` – запускає `iter` вказану кількість раз, виводячи графік чи анімацію процесу пошуку (в залежності від кількості вимірів);

- `solve_stats` – запускає `iter` вказану кількість раз, повертає результат і найкраще значення на кожній ітерації;
- `solve_time` – запускає `iter` вказану кількість раз, повертає результат і час що минув після початку для кожної ітерації.

2.2. Використання алгоритму оптимізації роєм часток на функціях пристосованості

2.2.1. Ackley



(Мал. 1. Функція Ackley)

$$f(x,y) = -20\,exp\left(-0.2\sqrt{(x^2+y^2)}\right) - exp\left(\frac{\left(cos(2\pi x)+cos(2\pi y)\right)}{2}\right) + e + 20$$

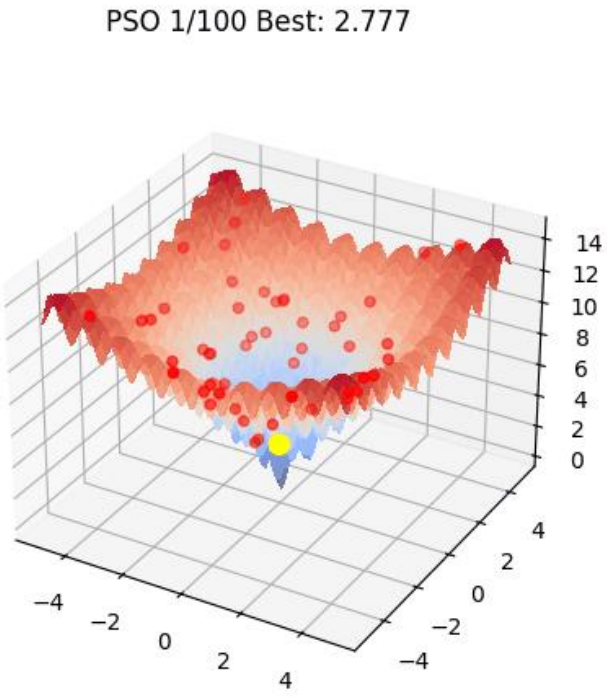
$$f_{min(0,0)} = 0$$

$$-5 \leq \text{x,y} \leq 5$$

Область пошуку невелика, тож a1, a2, speed_min і speed_max мають бути малими. Розмір популяції покладено 50.

Перший набір гіпер параметрів:

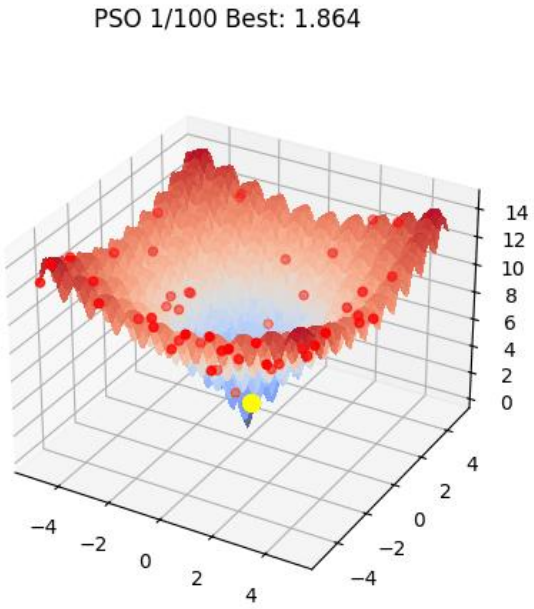
- a1 = 0.1
- a2 = 0.2
- Розмір популяції = 50
- Модуль швидкості за координатою не більше за 1



(Мал. 2. Функція Ackley, перший набір гіпер параметрів)

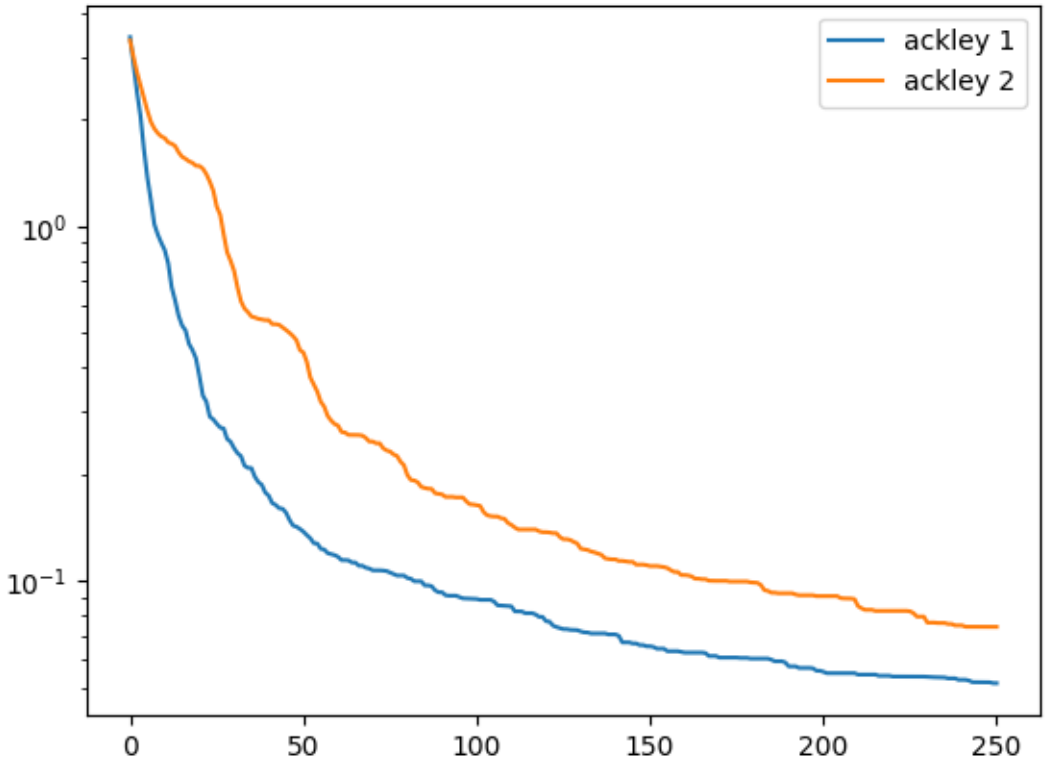
Отриманий результат має відносно низьку точність – 0.01 замість 0. Слід зменшити прискорення. Другий набір гіпер параметрів:

- a1 = 0.01
- a2 = 0.02
- Розмір популяції = 50
- Модуль швидкості за координатою не більше за 1



(Мал. 3. Функція Ackley, другий набір гіпер параметрів)

При одиничному запуску точність не покращилась.

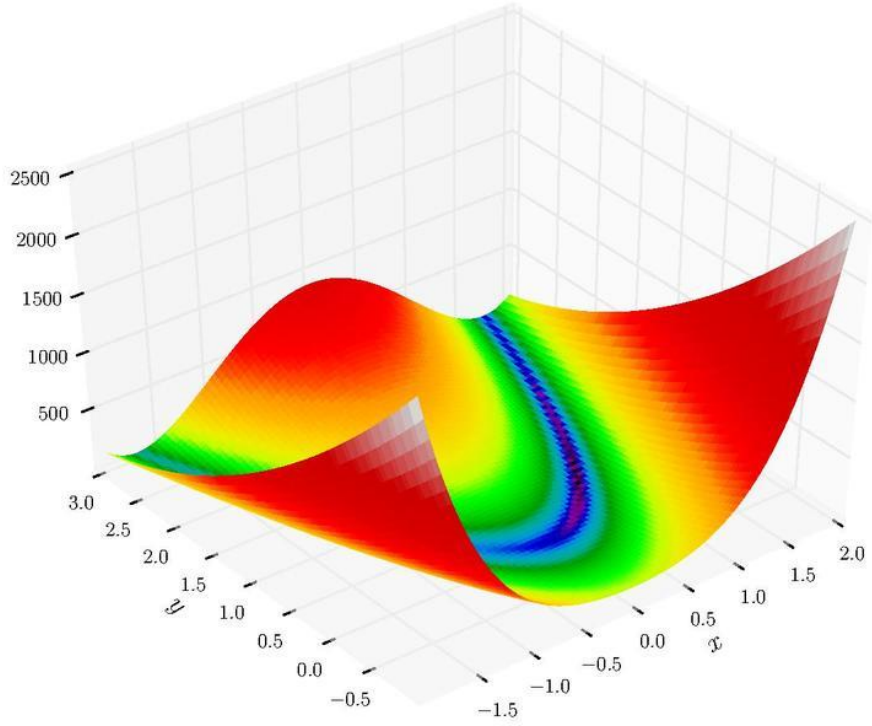


.

(Мал. 4. Функція Ackley, порівняння наборів гіпер параметрів)

Середнє арифметичне 100 тестів обох наборів гіпер параметрів показує, що значне зменшення прискорень негативно вплинуло на збіжність графіку. Близько 10ої, 40ої і 65ої ітерацій явно видно як при другому наборі гіпер параметрів найкраще значення алгоритму «зависає» в локальних екстремумах, тоді як при першому наборі швидкість збіжності стабільно приблизно експоненційно зменшується.

2.2.2. Функція Розенброка



(Мал. 5. Функція Розенброка)

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

$$f_{min}\left(\vec{x} = (1, \dots, 1)\right) = 0$$

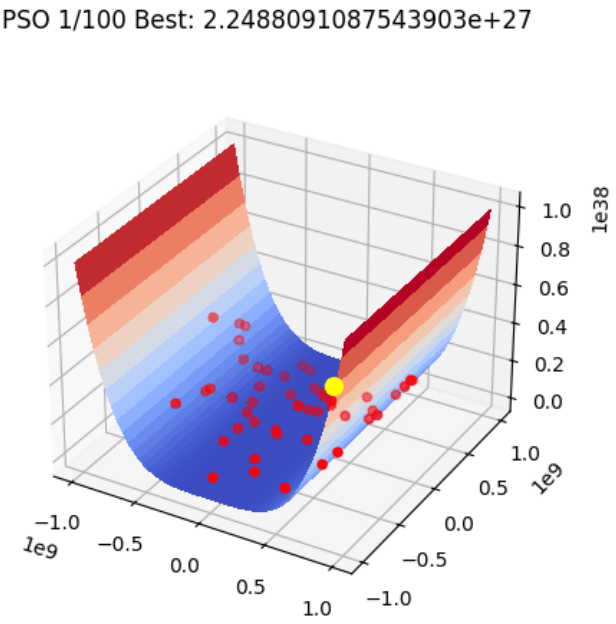
$$-\infty \leq x_i \leq \infty$$

$$1 \leq i \leq n$$

Область пошуку велика, тож a1, a2 і ліміти швидкостей мають бути великими.

Перший набір гіпер параметрів:

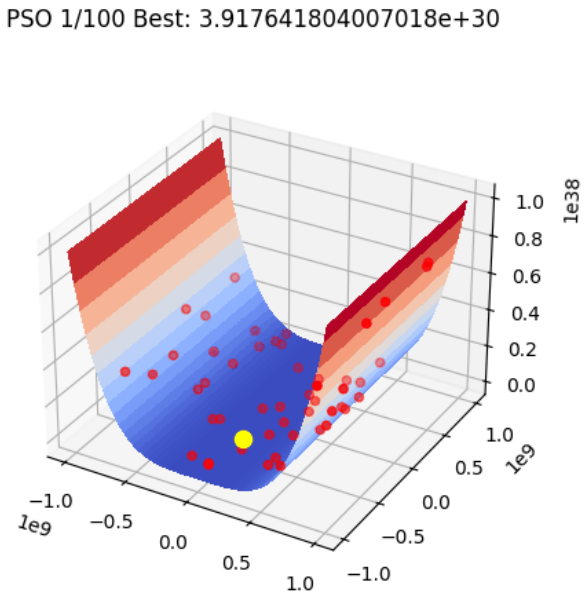
- a1 = 10^7
- a2 = 2*10^7
- Розмір популяції = 50
- Модуль швидкості за координатою не більше за 10^8
- Модуль координат не більше за 10^9



(Мал. 6. Функція Розенброка, перший набір гіпер параметрів)

Другий набір гіпер параметрів:

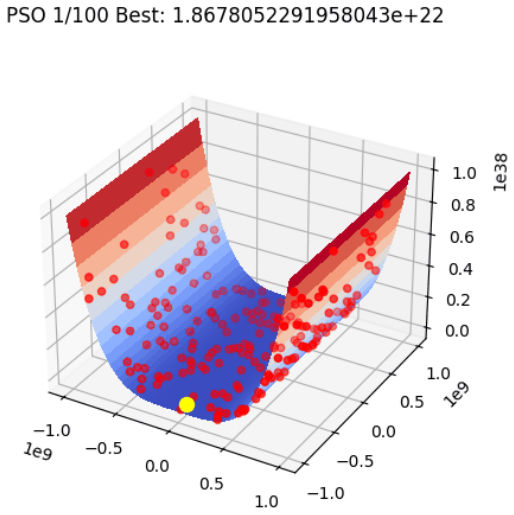
- a1 = 10^6
- a2 = $2*10^6$
- Розмір популяції = 50
- Модуль швидкості за координатою не більше за 10^7
- Модуль координат не більше за 10^9



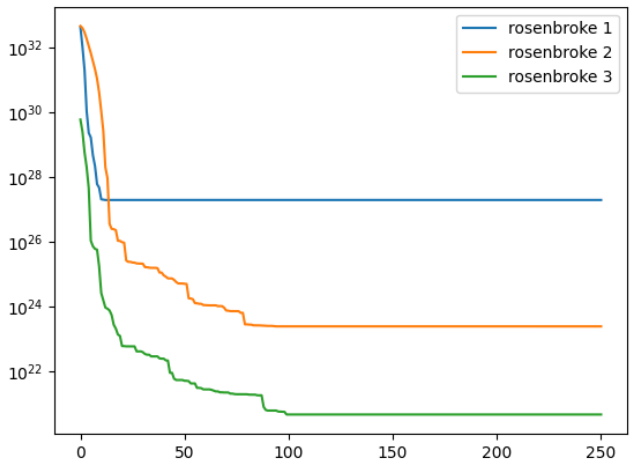
(Мал. 7. Функція Розенброка, другий набір гіпер параметрів)

Третій набір гіпер параметрів:

- a1 = 10^6
- a2 = $2*10^6$
- Розмір популяції = 200
- Модуль швидкості за координатою не більше за 10^7
- Модуль координат не більше за 10^9



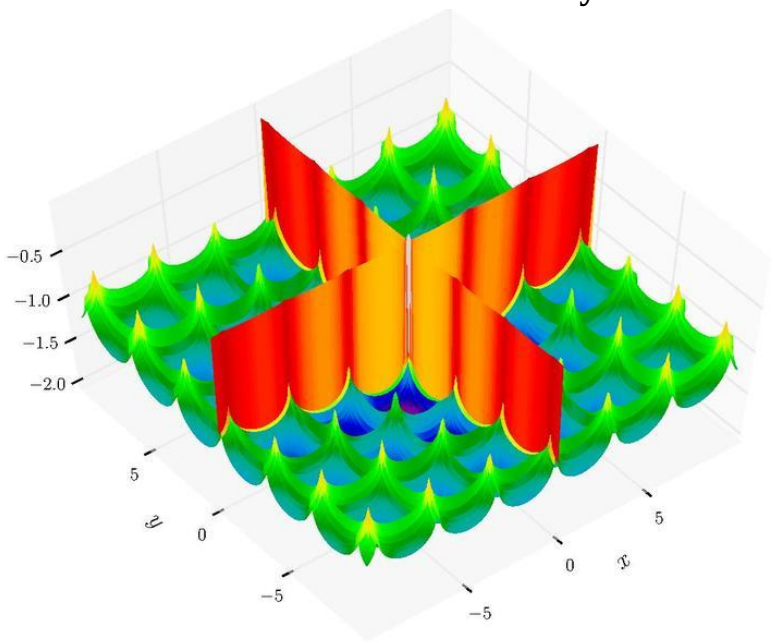
(Мал. 8. Функція Розенброка, третій набір гіпер параметрів)



(Мал. 9. Функція Розенброка, порівняння трьох наборів гіпер параметрів)

Середнє арифметичне 100 тестів трьох наборів гіпер параметрів показує, що алгоритм зависає в локальному мінімумі дуже рано і не покращується опісля.

2.2.3. Cross-in-tray



(Мал. 10. Функція Cross-in-tray)

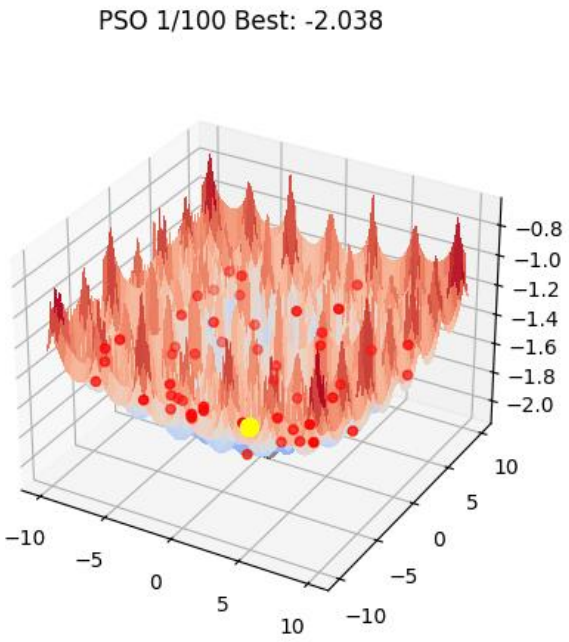
$$f(x,y) = -10^{-4} \cdot \left(\left| sin(x)\, sin(y)\, exp\left(\left| 100 - \frac{\sqrt{x^2+y^2}}{\pi} \right| \right) \right| + 1 \right)^{0.1}$$

$$f_{min}(\pm 1.34941, \pm 1.34941) = -2.06261$$

$$-10 \leq x, y \leq 10$$

Перший набір гіпер параметрів:

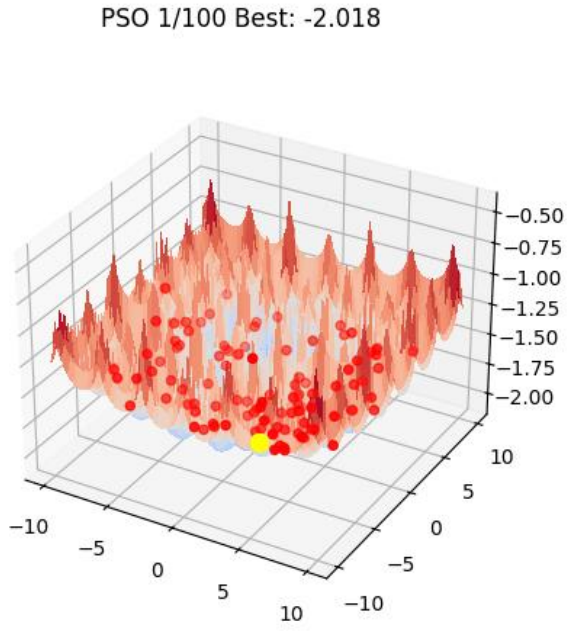
- a1 = 0.1
- a2 = 0.2
- Розмір популяції = 50
- Модуль швидкості за координатою не більше за 1



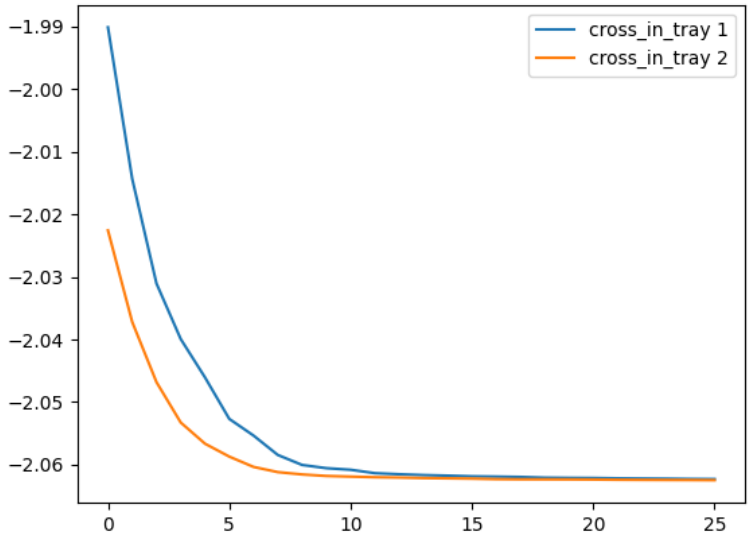
(Мал. 11. Функція Cross-in-tray, перший набір гіпер параметрів. Хрест не видно на графіку, оскільки він не співпав з wireframe)

Другий набір гіпер параметрів:

- a1 = 0.1
- a2 = 0.2
- Розмір популяції = 100
- Модуль швидкості за координатою не більше за 1



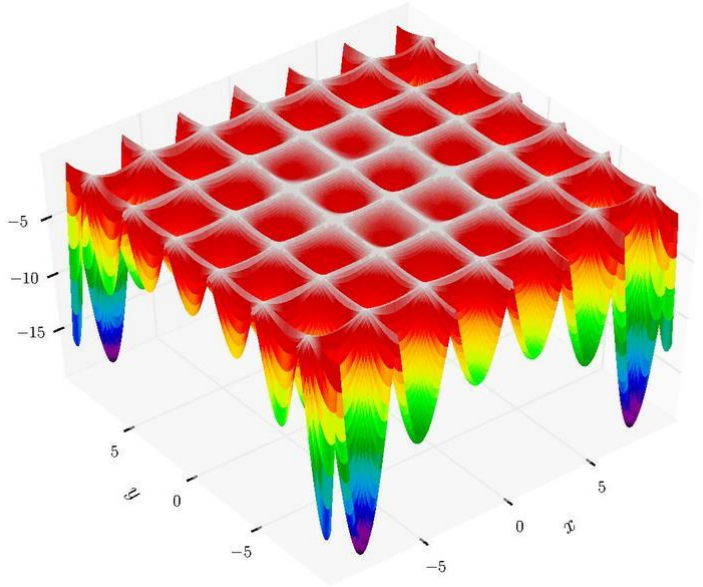
(Мал. 12. Функція Cross-in-tray, другий набір гіпер параметрів)



(Мал. 13. Функція Cross-in-tray, порівняння обох наборів гіпер параметрів)

Середнє арифметичне 100 тестів обох наборів гіпер параметрів показує, що алгоритм знаходить глобальний мінімум за менш ніж 15 ітерацій. Другий набір параметрів збігається дещо швидше через більший розмір популяції.

2.2.4. Hölder table



(Мал. 14. Функція Hölder table)

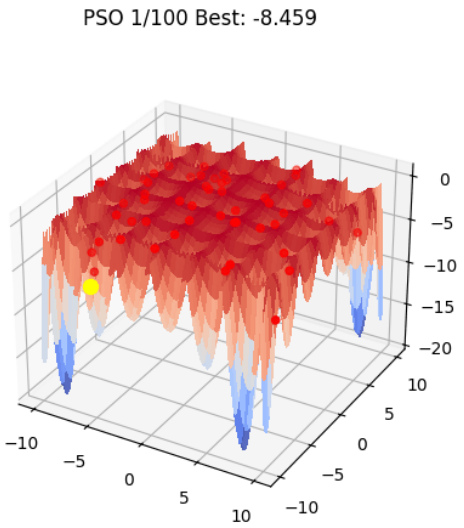
$$f(x,y) = -\left|\sin(x)\cos(y)\exp\left(\left|1 - \frac{\sqrt{x^2+y^2}}{\pi}\right|\right)\right|$$

$$f_{min}(\pm 8.05502, \pm 9.66459) = 19.2085$$

$$-10 \leq x, y \leq 10$$

Перший набір гіпер параметрів:

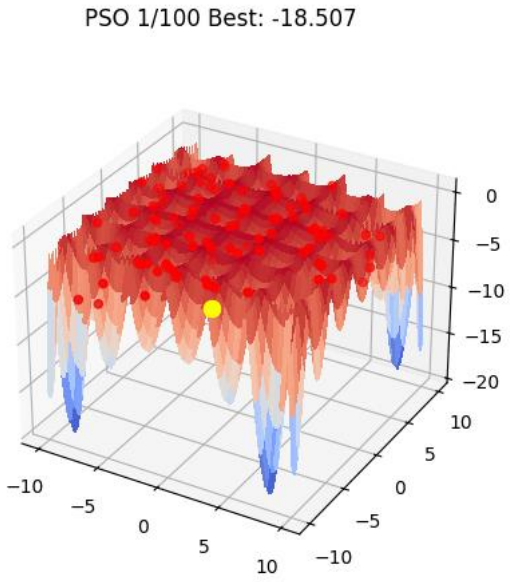
- a1 = 0.1
- a2 = 0.2
- Розмір популяції = 50
- Модуль швидкості за координатою не більше за 1



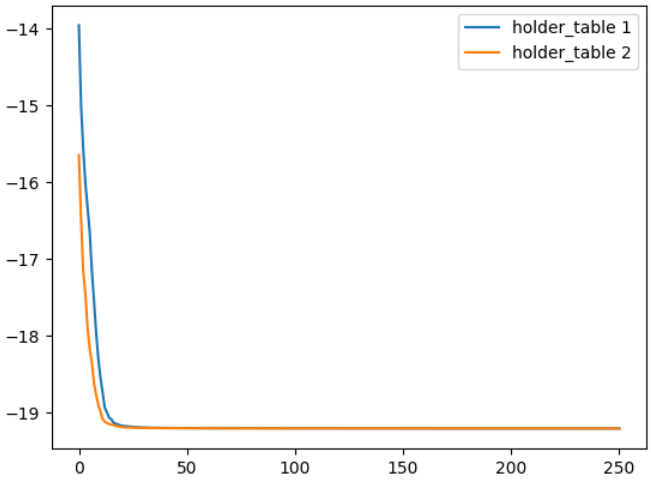
(Мал. 15. Функція Hölder table, перший набір гіпер параметрів)

Другий набір гіпер параметрів:

- a1 = 0.1
- a2 = 0.2
- Розмір популяції = 100
- Модуль швидкості за координатою не більше за 1



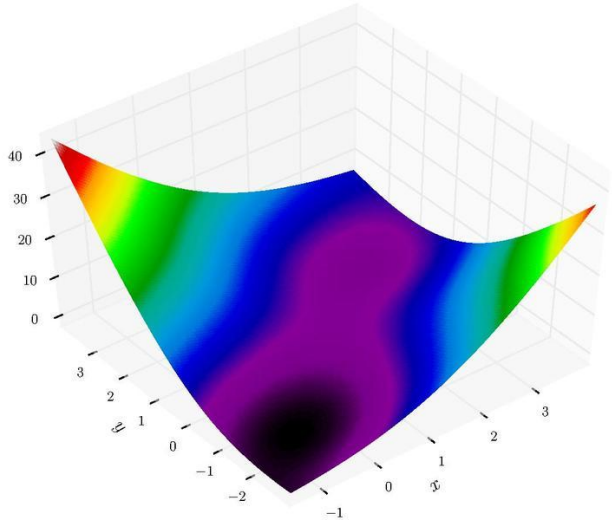
(Мал. 16. Функція Hölder table, другий набір гіпер параметрів)



(Мал. 17. Функція Hölder table, порівняння обох наборів гіпер параметрів)

Середнє арифметичне 100 тестів обох наборів гіпер параметрів показує, що алгоритм знаходить глобальний мінімум за менш ніж 25 ітерацій. Другий набір параметрів збігається дещо швидше через більший розмір популяції. Якщо врахувати, що розмір популяції збільшено вдвічі, то покращення незначне

2.2.5. McCormick



(Мал. 18. Функція McCormick)

$$f(x,y) = sin(x+y) + (x-y)^2 - 1.5x + 2.5y + 1$$

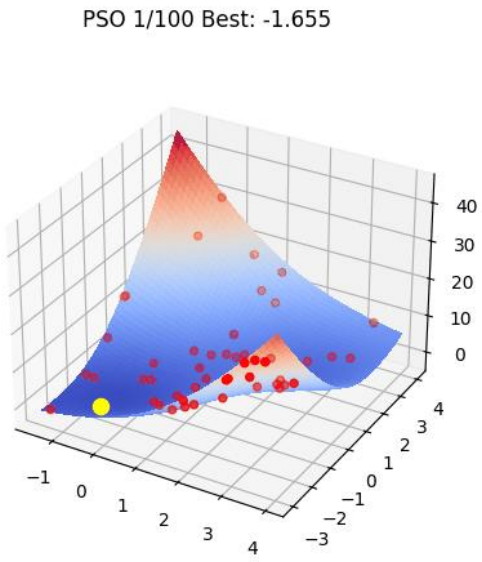
$$f_{min}(-0.54719, -1.54719) = -1.9133$$

$$-1.5 \leq x \leq 4$$

$$-3 \leq y \leq 4$$

Перший набір гіпер параметрів:

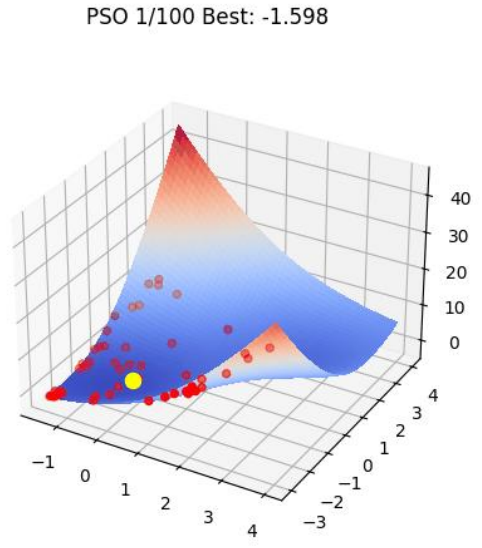
- a1 = 0.1
- a2 = 0.2
- Розмір популяції = 50
- Модуль швидкості за координатою не більше за 1



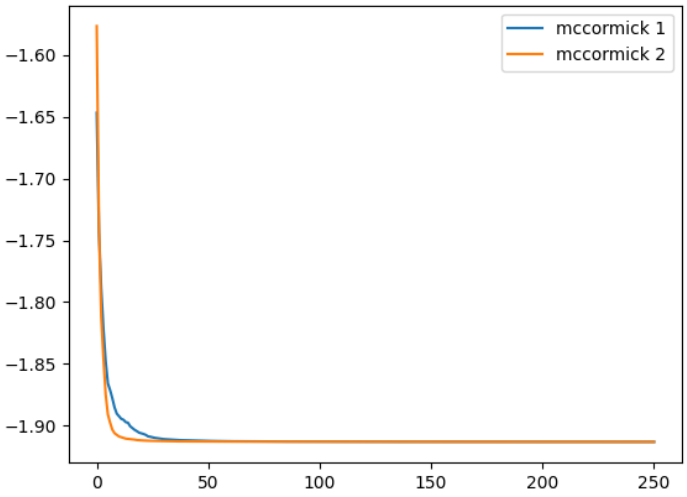
(Мал. 19. Функція McCormick, перший набір гіпер параметрів)

Другий набір гіпер параметрів:

- a1 = 1
- a2 = 2
- Розмір популяції = 50
- Модуль швидкості за координатою не більше за 1



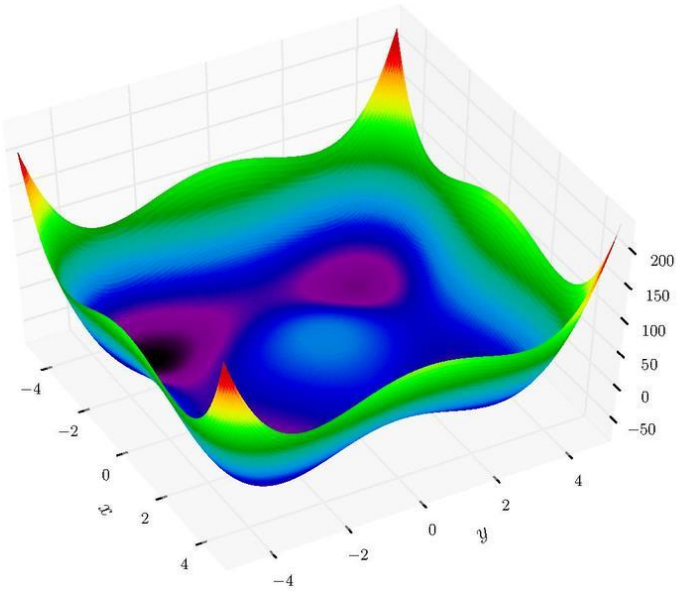
(Мал. 20. Функція McCormick, другий набір гіпер параметрів)



(Мал. 21. Функція McCormick, порівняння обох наборів гіпер параметрів)

Середнє арифметичне 100 тестів обох наборів гіпер параметрів показує, що алгоритм знаходить глобальний мінімум за менш ніж 40 ітерацій. Другий набір параметрів збігається значно швидше.

2.2.6. Styrblinski-Tang



(Мал. 22. Функція Styrblinski-Tang)

$$f(x) = \frac{\sum_{i=1}^{n}\left(x_i^4 - 16x_i^2 + 5x_i\right)}{2}$$

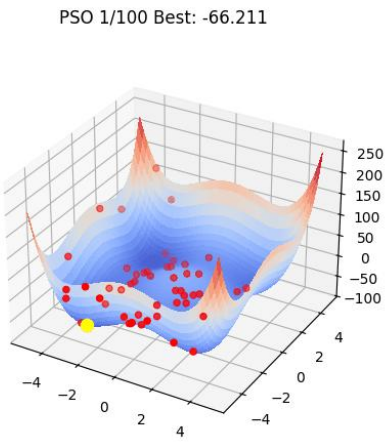
$$f_{min}\left(\overrightarrow{X_n}\right) = -39.16617n$$

$$f_{min}\left(\overrightarrow{X_2}\right) = -78.33234$$

$$-5 \leq x_i \leq 5$$

Перший набір гіпер параметрів:

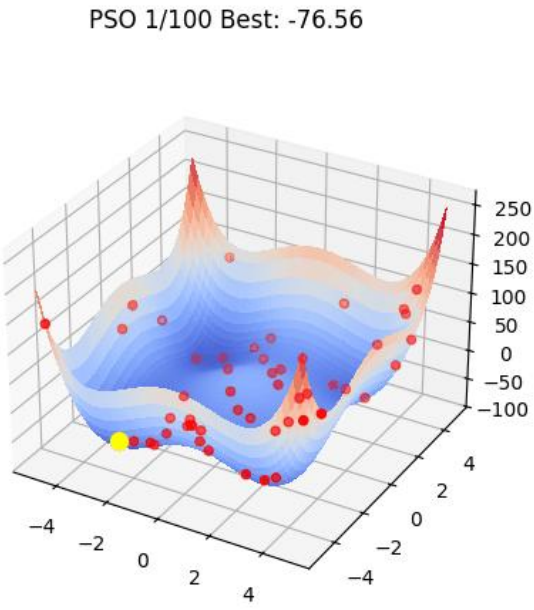
- a1 = 0.1
- a2 = 0.2
- Розмір популяції = 50
- Модуль швидкості за координатою не більше за 1



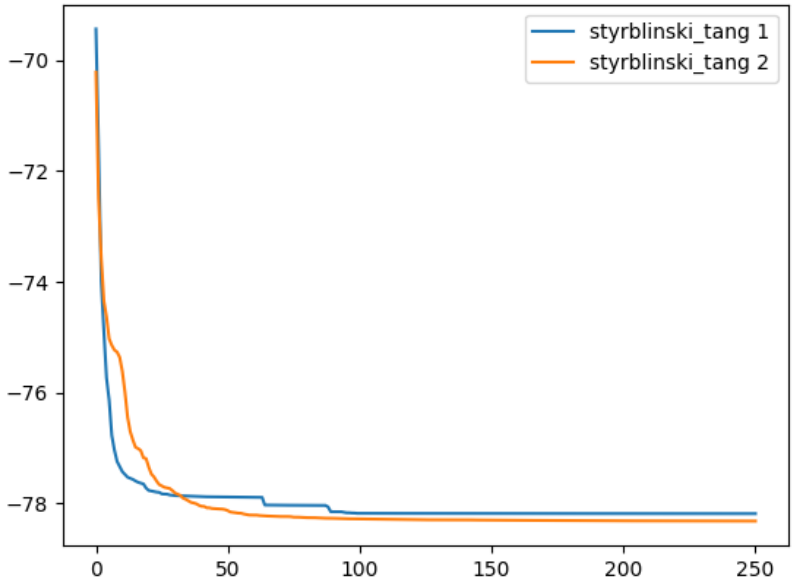
(Мал. 23. Функція Styrblinski-Tang, перший набір гіпер параметрів)

Другий набір гіпер параметрів:

- a1 = 0.2
- a2 = 0
- Розмір популяції = 50
- Модуль швидкості за координатою не більше за 1



(Мал. 24. Функція Styrblinski-Tang, другий набір гіпер параметрів)



(Мал. 25. Функція Styrblinski-Tang, порівняння обох наборів гіпер параметрів)

Середнє арифметичне 100 тестів обох наборів гіпер параметрів показує, що за відсутності комунікації між частинками, збіжність повільніша, але краща.

3. ВИСНОВКИ

- Значне збільшення розміру популяції не завжди дає значне пришвидшення збіжності.
- Відсутність прискорення в напрямку найкращої позиції може дати кращу, але повільнішу, збіжність.
- На функціях з великою областю пошуку неможливо знайти достатньо точний мінімум без поступового зменшення прискорення і значної кількості часток.