

POP - dokumentacja końcowa

Autorzy: Jakub Bąba, Aleksandra Szymańska

Treść zadania

Zadanie nr 7:

Kontrola przystosowania punktu środkowego populacji jest sposobem na poprawienie jakości działania algorytmów ewolucyjnych. Na wybranym przez siebie algorytmie ewolucyjnym zbadaj wpływ sposobu zdefiniowania punktu środkowego na działanie tego algorytmu.

Poza statystykami jak średnia arytmetyczna lub mediana zbadaj różne warianty średniej ważonej oraz statystyki odporne. Wykorzystaj wybrane funkcje testowe.

Opis problemu

Ocena przystosowania punktu środkowego populacji często okazuje się lepszym kryterium zakończenia przeszukiwania niż ocena najlepszego osobnika, ponieważ uwzględnia cały rozkład populacji. W przypadku skomplikowanych problemów, gdzie populacje zbliżają się do optimum w sposób nieregularny, statystyki odporne, takie jak mediana, mogą okazać się bardziej skuteczne niż średnia jako reprezentacja punktu środkowego.

Naszym zadaniem było zbadanie wpływu zdefiniowania punktu środkowego na wybranym algorytmie ewolucyjnym. W związku z tym musieliśmy najpierw wybrać algorytm oraz przestrzeń przeszukiwań na której będziemy testowali nasz algorytm.

Opis algorytmów

Algorytm genetyczny

Zdecydowaliśmy się na użycie następującego algorytmu genetycznego:

- Z krzyżowaniem jednopunktowym - o prawdopodobieństwie 0.7
- Z mutacją z rozkładu normalnego - **o wariancji 0.04 i prawdopodobieństwie 0.1 zamiast planowanej 0.01 ze względu na zbyt małą eksplorację przestrzeni przeszukiwań**
- Z selekcją proporcjonalną
- Z sukcesją generacyjną
- Z populacją o rozmiarze 100 elementów

Algorytm zatrzymywaliśmy, gdy różnica w ocenie przystosowania punktu środkowego w kolejnych generacjach była mniejsza niż $1e-10$ lub po osiągnięciu 10000 generacji.

Jako funkcję fitness wykorzystaliśmy odpowiednią funkcję testową, przy czym jeśli oceniany punkt nie należał do dziedziny, przypisywana mu była kara w wysokości 10000. Lepszym

osobnikiem był ten, który otrzymał niższą wartość funkcji fitness, ponieważ wybrane przez nas funkcje są stosowane w zadaniach minimalizacji. Wyniki te konwertowane są także do wag używanych w selekcji proporcjonalnej oraz punktach środkowych; obliczyliśmy je, odwracając ocenę przystosowania pomniejszoną o minimalną ocenę w populacji i powiększoną o współczynnik $a=1e-12$, aby uniknąć wartości ujemnych oraz dzielenia przez zero.

Punkty środkowe

Wykorzystaliśmy następujące sposoby definiowania punktu środkowego:

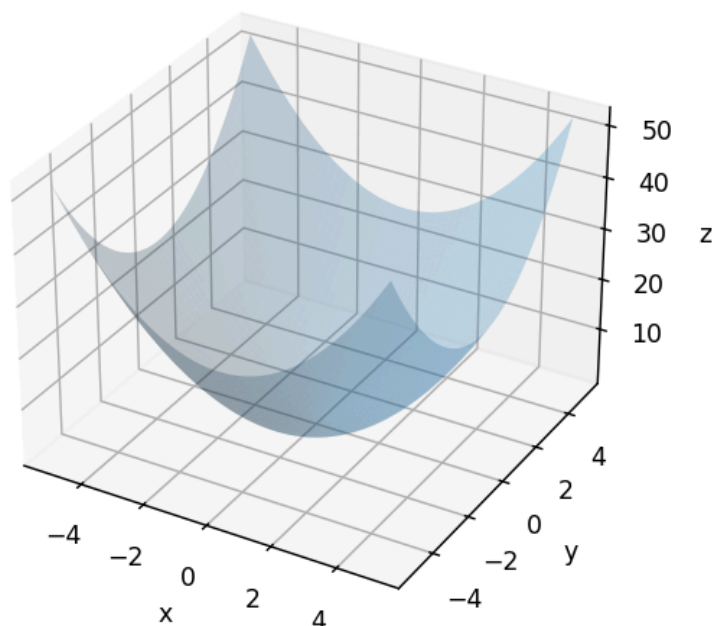
- Średnia arytmetyczna (miara wrażliwa na wartości odstające)
- Mediana
- Średnia ważona
- Średnia geometryczna ważona (aby uniknąć logarytmowania wartości niedodatnich, do obliczeń bierzemy wartość bezwzględną osobnika powiększoną o $1e-6$)
- Średnia ucinana (ucięcie po 15% skrajnych punktów, osobno dla każdej współrzędnej)
- Estymator Hubera (punktem środkowym jest punkt minimalizujący funkcję straty Hubera, przy czym początkowy punkt to średnia populacji)

Funkcje testujące

Sprawdziliśmy różnice zgodnie z planowanymi eksperymentami na trzech funkcjach testujących. Były nimi:

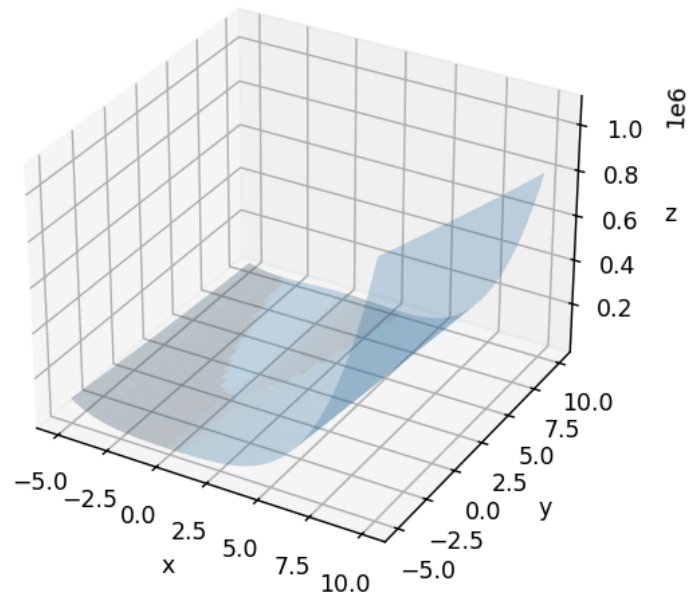
- Funkcja sferyczna:
 - domena: $[(-5.12, 5.12), (-5.12, 5.12)]$
 - minimum globalne: $(0, 0)$ o wartości 0

Sphere Function



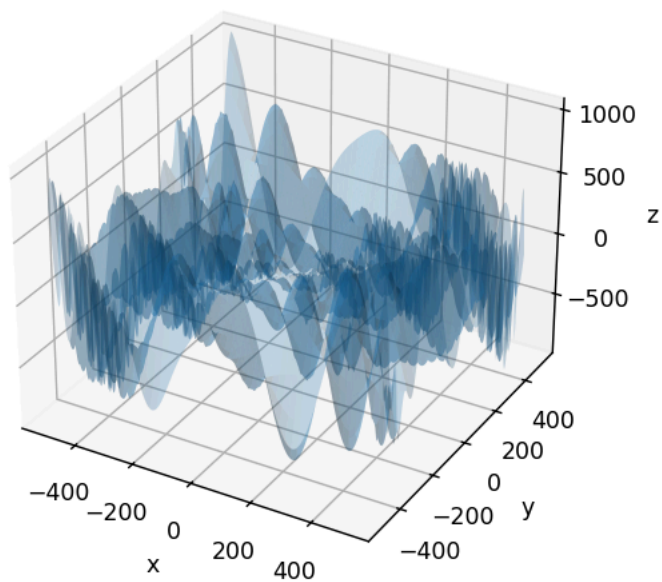
- Funkcja Rosenbrocka
 - domena $[(-5, 10), (-5, 10)]$
 - minimum globalne: $(1, 1)$ o wartości 0

Rosenbrock Function



- Funkcja Eggholder
 - domena $[(-512, 512), (-512, 512)]$
 - minimum globalne: $(512, 404.2319)$ o wartości -959.6407

Eggholder Function



Każda z tych funkcji ma trochę inną charakterystykę, więc możliwe było zbadanie wpływu różnych wyliczeń punktów środkowych dla różnych rodzajów przestrzeni. W celu łatwego porównania eksperymentów między tymi funkcjami funkcje te były wykonane w dwóch wymiarach.

Przeprowadzone eksperymenty numeryczne

Zgodnie z planem, dla każdej kombinacji funkcji testowej i metody wyznaczania punktu środkowego uruchomiliśmy zaimplementowany przez nas algorytm genetyczny trzydzieści razy, wykorzystując trzydzieści różnych ziaren losowości (1, 2, ..., 30). Zebrane dane obejmowały m.in. liczbę generacji, listę punktów środkowych i najlepsze punkty z każdej generacji, ziarno losowości oraz dziedzinę. Obliczyliśmy średnią liczbę generacji potrzebnych do zakończenia działania algorytmu dla każdej kombinacji oraz średnią odległość najlepszego znalezionej punktu od rzeczywistego optimum. Wyniki zapisaliśmy w pliku JSON zawierającym również metadane dotyczące parametrów uruchomienia algorytmu, takie jak użyta funkcja testowa, metoda definicji punktu środkowego, domena oraz wartość optymalnego rozwiązania. Odległość od minimum była obliczana w przestrzeni trójwymiarowej.

Technologie

Zarówno do wykonania zadania jak i przeprowadzenia eksperymentów użyliśmy języka Python z rozszerzeniami o biblioteki NumPy, Matplotlib oraz scipy. Zarówno wyniki, jak i ziarna do generatorów liczb losowych wyeksportowaliśmy do plików json. W celu prostego uruchomienia programu w środowisku Linux stworzony został plik run.sh, a dokładniejsza instrukcja uruchomienia znajduje się w pliku README. Cały projekt będzie znajduje się w repozytorium na serwerze GitHub.

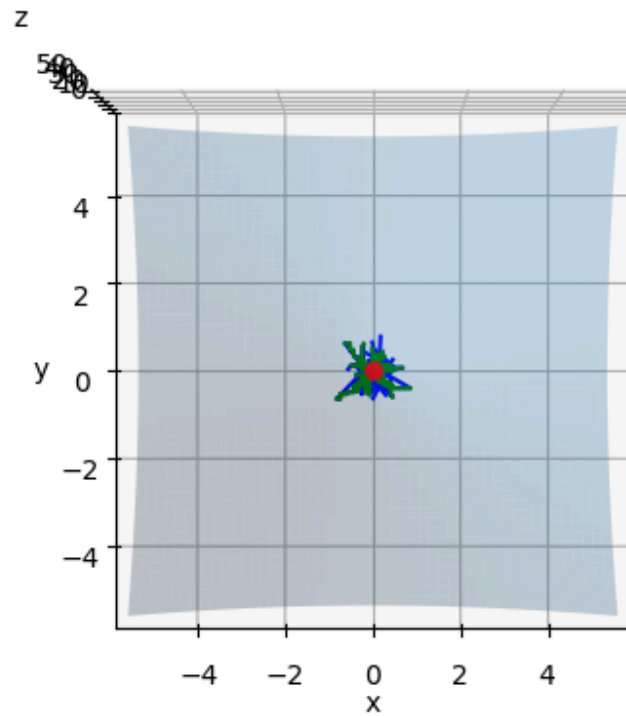
Wyniki

Tabela średniej liczby generacji potrzebnych do zakończenia przeszukiwania, odległości najlepszego punktu od optymalnego rozwiązania i różnicy w wyniku (błąd względem optymalnego rozwiązania):

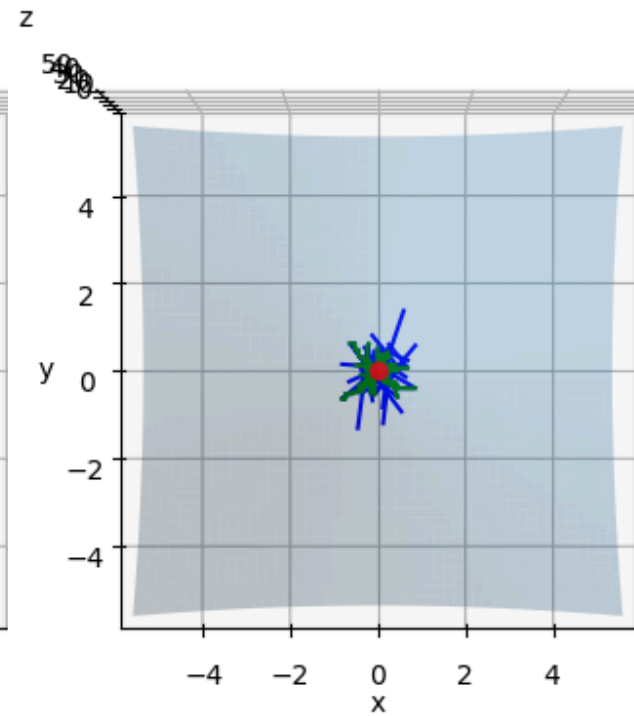
	Funkcja sferyczna	Funkcja Rosenbrocka	Funkcja Eggholder
Punkt środkowy - Średnia	Generacje: 8646.43	Generacje: 10000	Generacje: 5890.27
	Odległość: 1.32e-06	Odległość: 0.0029	Odległość: 653.21
	Błąd: 3.39e-12	Błąd: 2.74e-06	Błąd: 107.42
Punkt środkowy - Mediana	Generacje: 12.3	Generacje: 8.47	Generacje: 368.17
	Odległość: 0.0027	Odległość: 2.47	Odległość: 663.32
	Błąd: 1.05e-05	Błąd: 2.08	Błąd: 108.16
Punkt środkowy - Średnia ucinana	Generacje: 12.3	Generacje: 8.47	Generacje: 368.16
	Odległość: 0.0027	Odległość: 2.47	Odległość: 663.31
	Błąd: 1.05e-05	Błąd: 2.08	Błąd: 108.16
Punkt środkowy - Średnia ważona	Generacje: 11.73	Generacje: 7.97	Generacje: 367.67
	Odległość: 0.0032	Odległość: 2.47	Odległość: 663.32
	Błąd: 1.82e-05	Błąd: 2.08	Błąd: 108.17
Punkt środkowy - Ważona średnia geometryczna	Generacje: 11.3	Generacje: 13.67	Generacje: 374.47
	Odległość: 0.0037	Odległość: 2.43	Odległość: 663.20
	Błąd: 2.19e-05	Błąd: 1.95	Błąd: 108.13
Punkt środkowy - Estymator Hubera	Generacje: 8646.43	Generacje: 10000	Generacje: 5890.27
	Odległość: 1.32e-06	Odległość: 0.0029	Odległość: 663.21
	Błąd: 3.39e-12	Błąd: 2.74e-06	Błąd: 107.42

Dodatkowo na 18 wykresach zwizualizowaliśmy szlaki zakreślane przez punkty środkowe i najlepsze punkty w każdej generacji podczas 30 uruchomień. Kolorem zielonym oznaczone są wędrówki wszystkich najlepszych punktów, niebieskim wędrówki punktów środkowych, a czerwonym oznaczony jest najlepszy punkt dla funkcji. Przykładowo jedna prosta linia to wędrówka jednego uruchomienia.

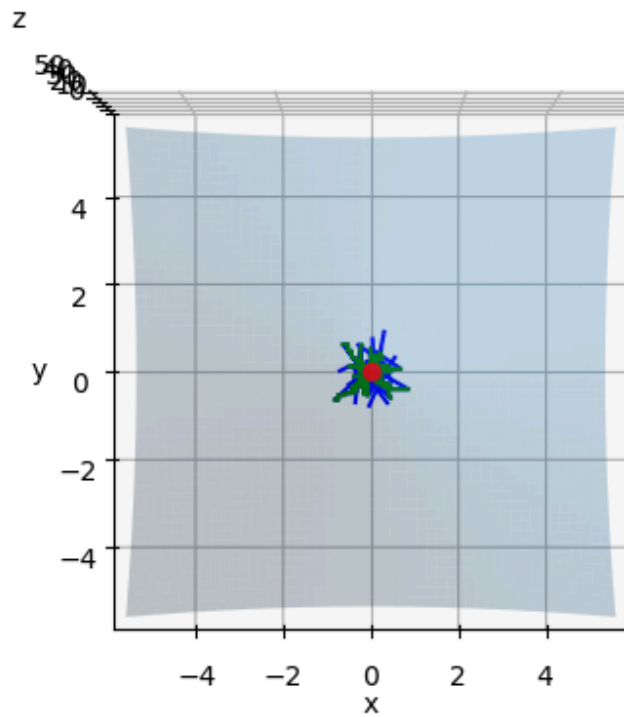
Sphere Function Mean



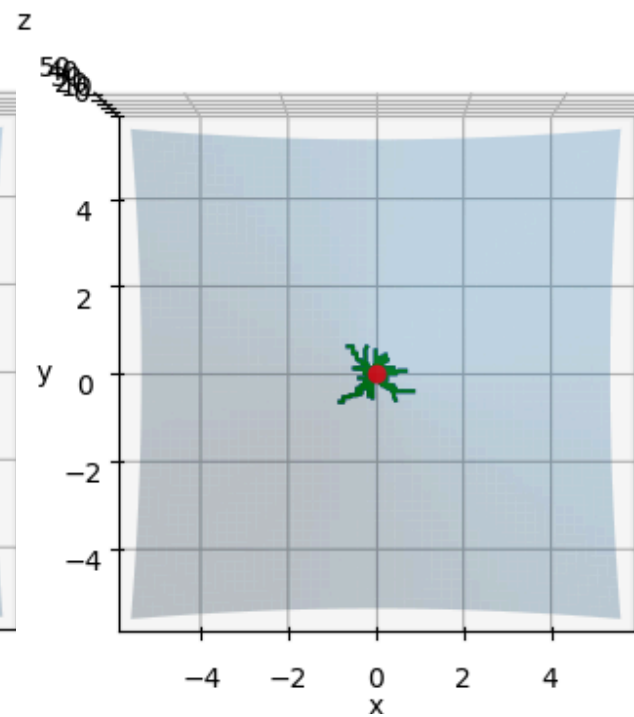
Sphere Function Median



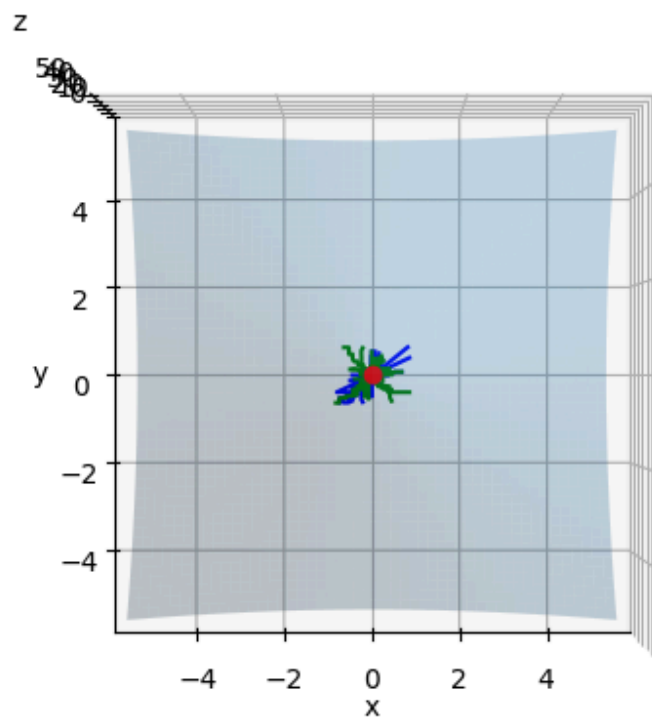
Sphere Function Trimmed Mean



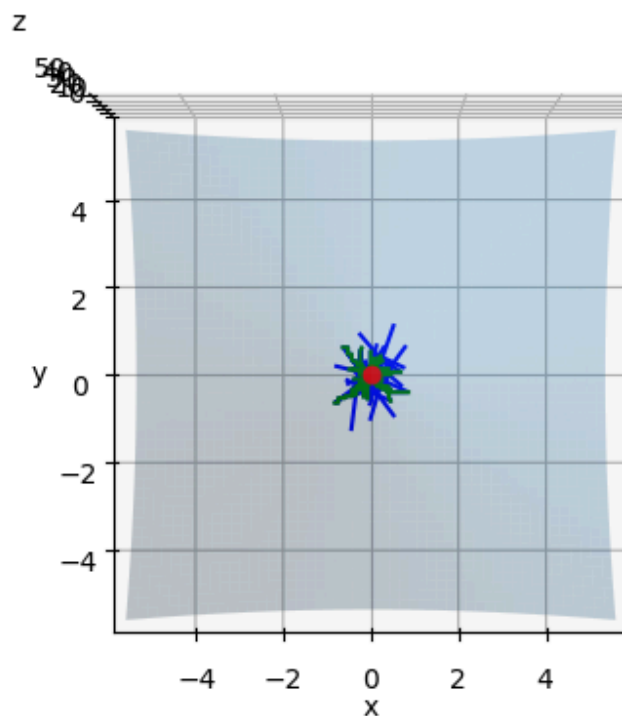
Sphere Function Weighted Mean



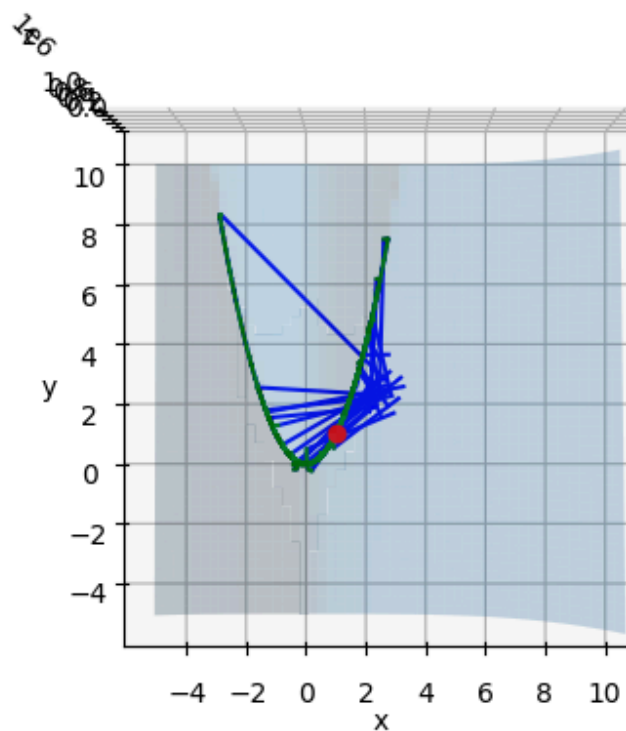
Sphere Function Weighted Geometric Mean



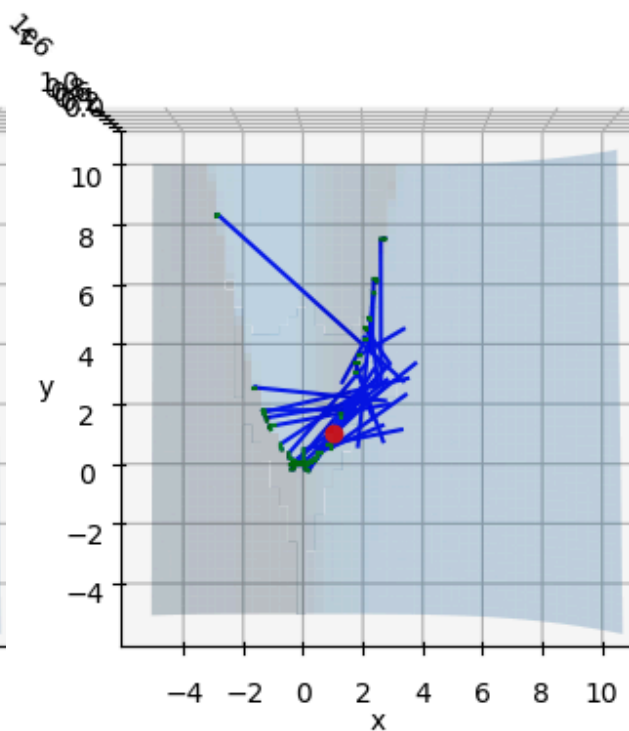
Sphere Function Huber Estimator



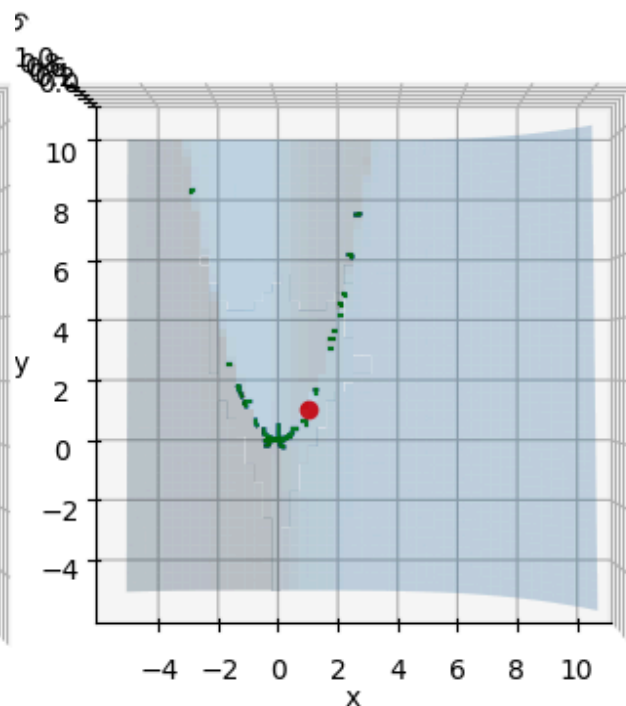
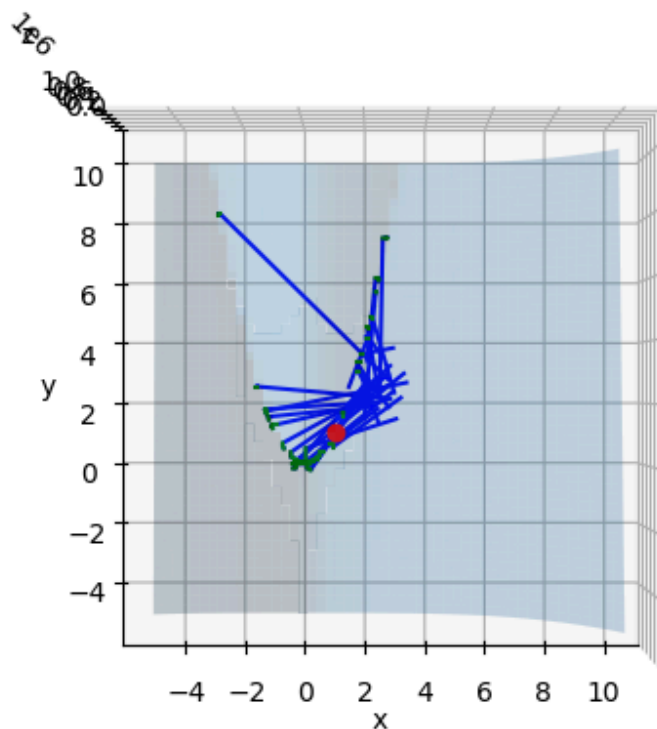
Rosenbrock Function Mean



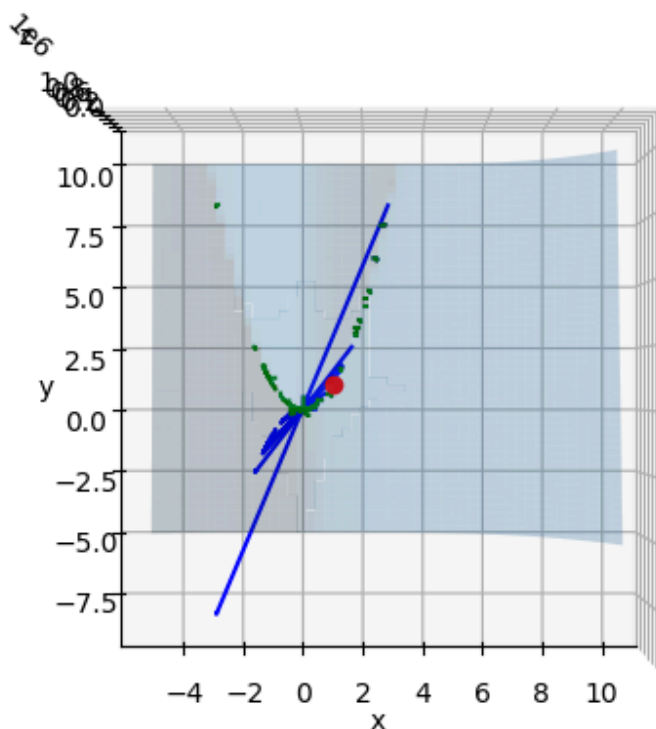
Rosenbrock Function Median



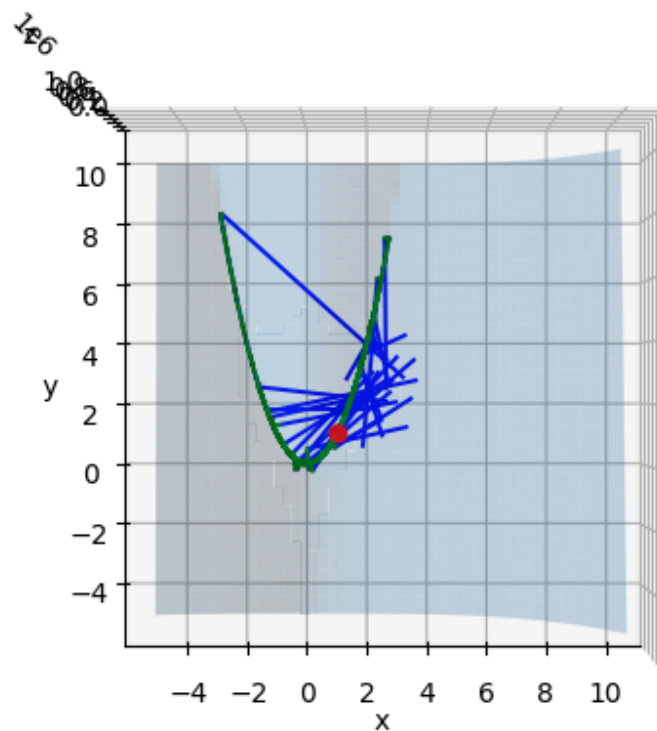
Rosenbrock Function Trimmed Mean Rosenbrock Function Weighted Mean



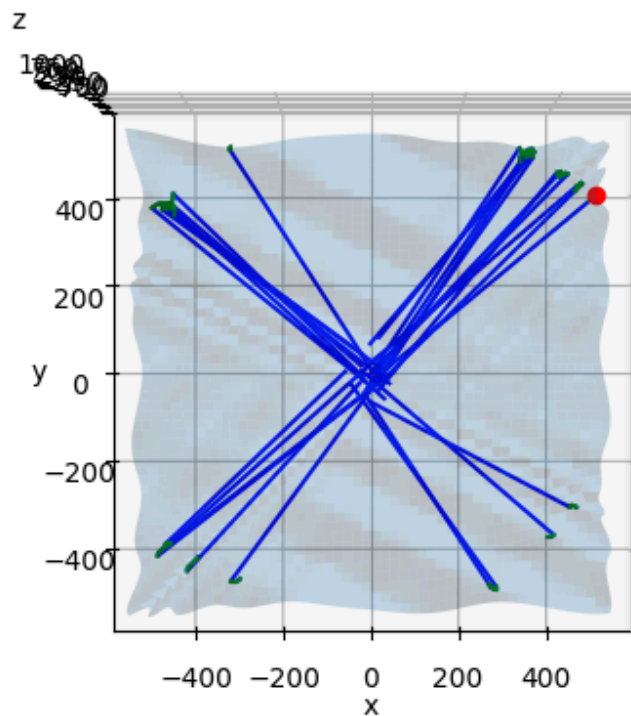
Rosenbrock Function Weighted Geometric Mean



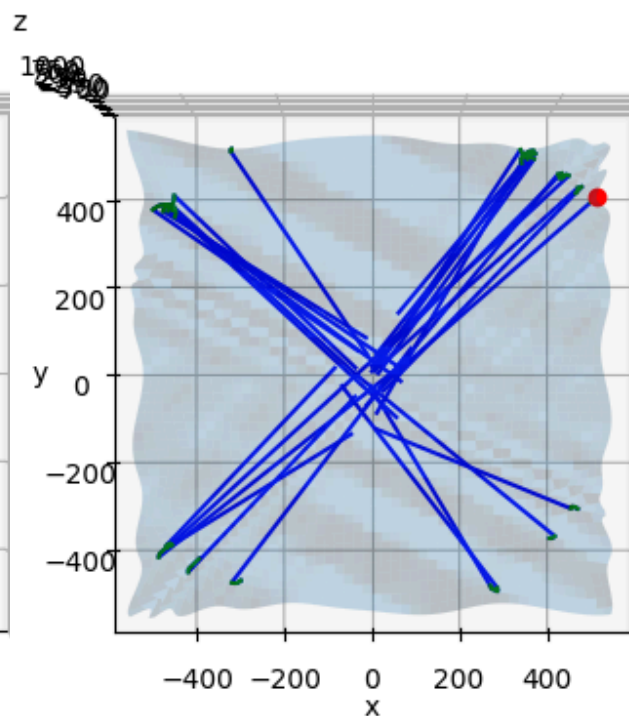
Rosenbrock Function Huber Estimator



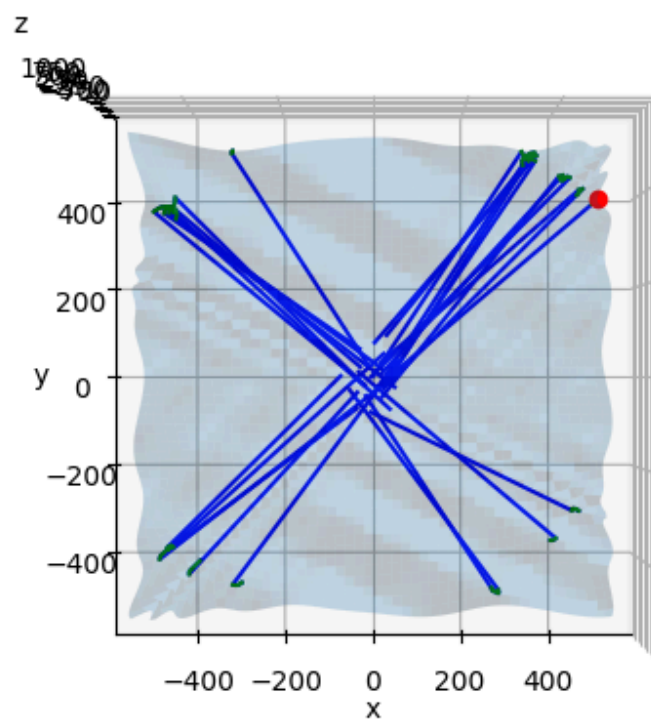
EggHolder Function Mean



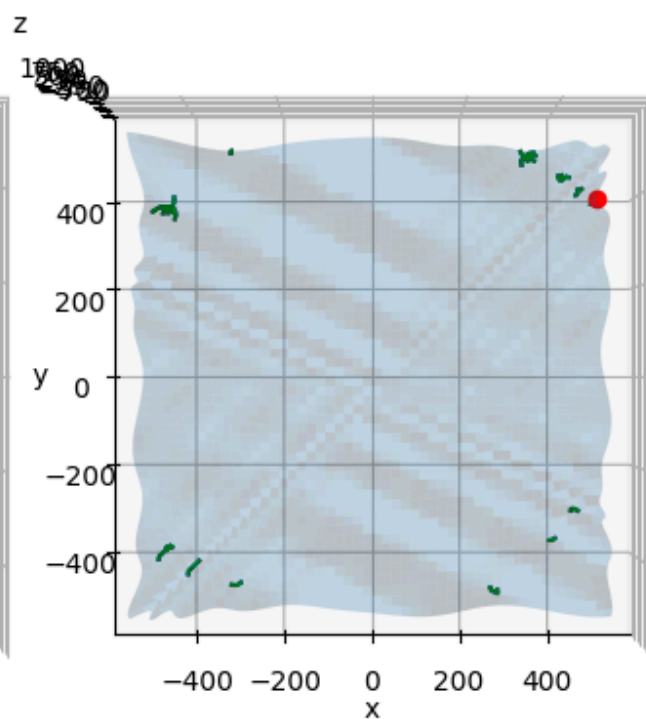
EggHolder Function Median



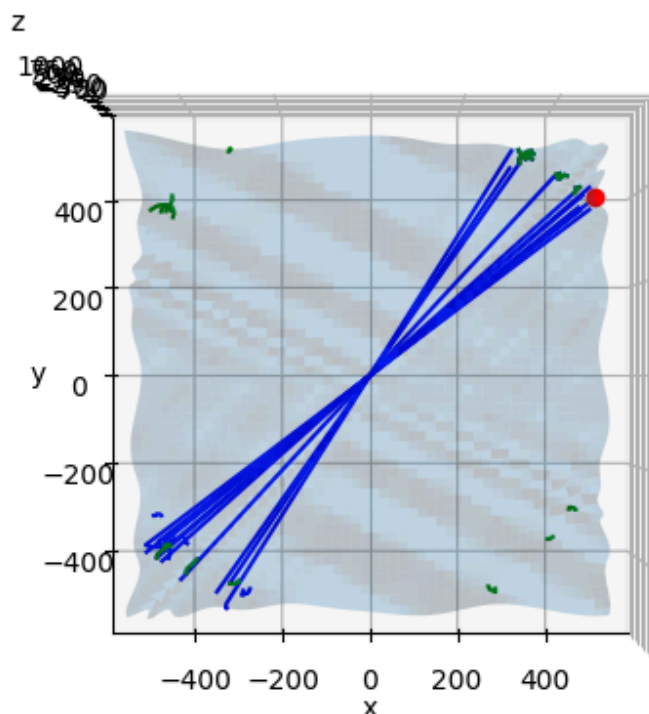
EggHolder Function Trimmed Mean



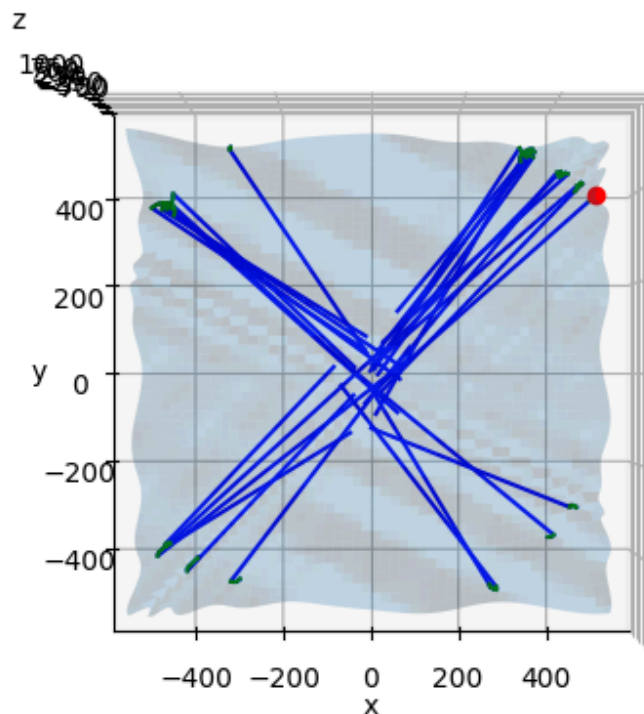
EggHolder Function Weighted Mean



EggHolder Function Weighted Geometric Mean



EggHolder Function Huber Estimator



Wnioski

Można zauważyć, że wyłączając estymator Hubera, każda z funkcji odpornych w stanie zdecydowanie zmniejszyć ilość iteracji względem wyliczania przez średnią. Jest to jednak obarczone błędem w dokładności algorytmu - odległość od punktu i różnica w wyniku jest odpowiednio większa.

Uniwersalnie można powiedzieć, że w największym stopniu ilość iteracji skraca średnia ważona, lecz wszystkie statystyki skracają czas wykonania w podobnym stopniu.

Ilość iteracji w wypadku funkcji sferycznej i Rosenbrocka zmniejsza się około 1000 krotnie, jednak jest to obarczone dużą stratą - o ile w wypadku funkcji sferycznej różnica między wartościami typu **1.05e-05** a **3.39e-12** może być tego warta (mimo dużej różnicy rzędu wielkości, obie wyznaczone wartości są bardzo małe), o tyle w wypadku funkcji Rosenbrock różnica między wartościami typu **2.08** a **2.74e-06** sprawiają wrażenie mniej opłacalnych (jedna z liczb nie jest nawet ułamkowa). Jednak w wypadku analizy na tych punktach należy rozważyć indywidualnie nasze cele i gotowość do kompromisów, w jednym problemie zmniejszenie dokładności w takim stopniu może być akceptowalna, w innym nie.

Z kolei w nieregularnej funkcji Eggholder różnica w wartości funkcji celu między funkcjami odpornymi a średnią wynosi niecały 1%, a ilość iteracji jest 10 krotnie mniejsza. Spodziewamy się, że dla większości problemów jest to bardzo dobry sposób na wcześniejsze zakończenie algorytmu.

Oglądając rysunki przejść możemy zauważyć bardzo ciekawy fakt - punkt środkowy obliczany przez średnią ważoną (generalnie najszybszą) ma bardzo małe przejścia pomiędzy wartościami. Również geometryczna średnia ważona ma z reguły trochę inne przejścia punktów środkowych. Jeśli chodzi o porównanie innych przejść punktów środkowych, główna różnica jest w stopniu odchylenia tych punktów (widocznych szczególnie w funkcji sferycznej).

Źródła

<http://www.sfu.ca/~ssurjano/optimization.html>

https://www.researchgate.net/publication/313940185_Improving_Evolutionary_Algorithms_in_a_Continuous_Domain_by_Monitoring_the_Population_Midpoint