Klasyczne algorytmy optymalizacyjne

Raport z pierwszych laboratoriów

PSIwSUM

Urszula Starowicz

407177

20.03.2022

Wprowadzenie:

Celem ćwiczenia jest implementacja klasycznych algorytmów optymalizacyjnych w zagadnieniu znajdowania minimum lokalnego dwuwymiarowej funkcji celu. Przy realizacji implementacji algorytmów korzystamy ze zbioru funkcji dostarczonego wcześniej przez prowadzącego.

Algorytm gradientowy wielostartowy

Dzięki wielostartowości algorytm może poradzić sobie z problemem minimów lokalnych. Ponieważ metoda gradientowa jest niedeterministyczna, możemy uzyskać różne wyniki przy wielu startach.

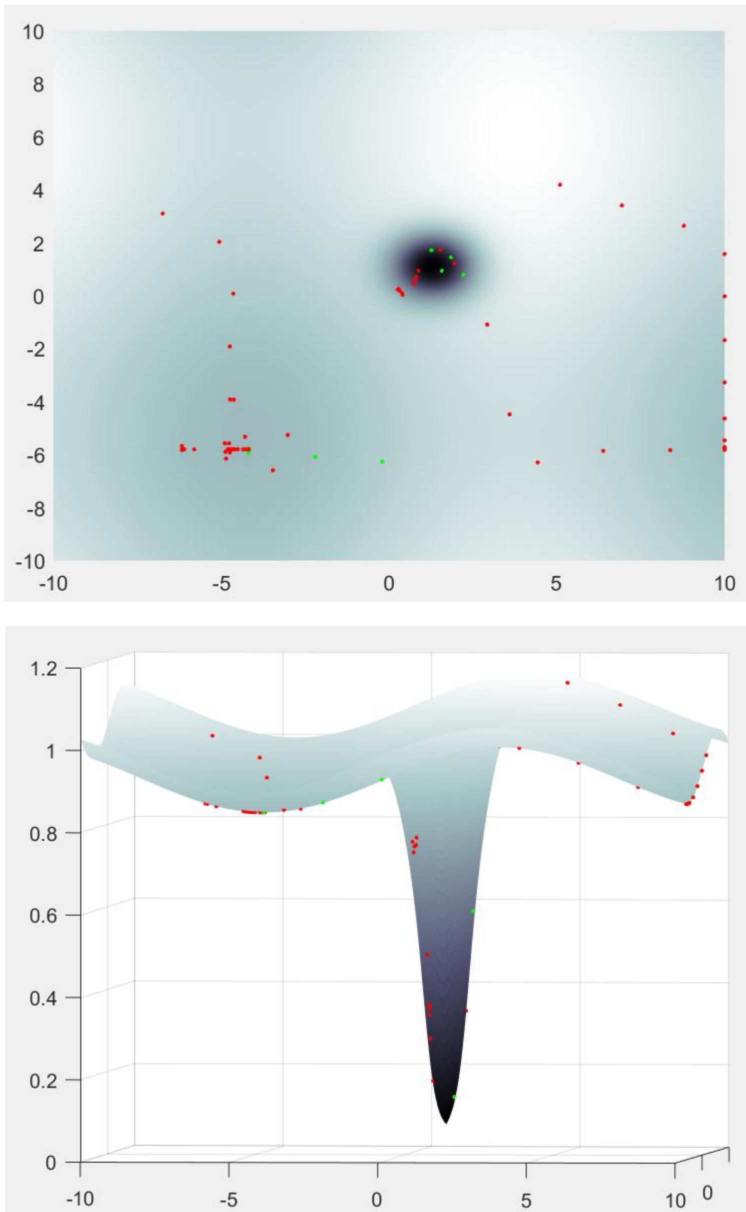
Parametry i za co odpowiadają:

- Starts – wyznacza ilość startów
- MaxSteps – ilość kroków w każdym starcie
- g_step – odległość między punktami, w których wyznaczamy gradient
- Step – długość kroku
- InitialStep – balans między eksploracją i eksploatacją
- P2 – ustala nachylenie sigmoidu
- Pozostałe parametry definiujemy jak w przypadku metody gradientowej.

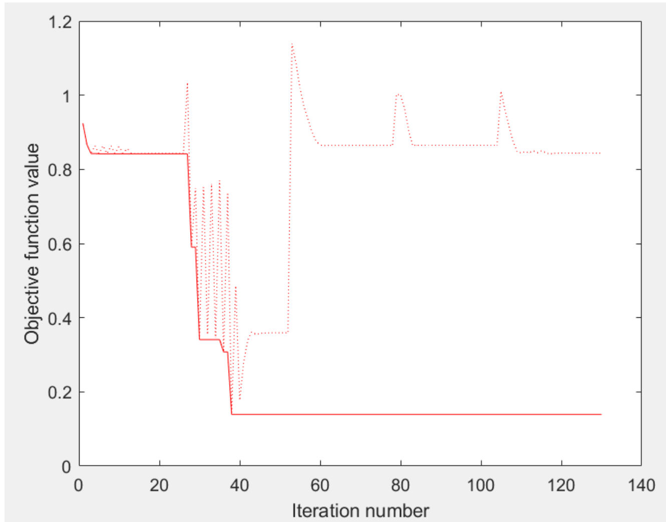
Dobrane parametry:

```
Starts=5;
g_step=1;
Step=1;
InitialStep = 2;
P1 = 2;
P2 = 1;
```

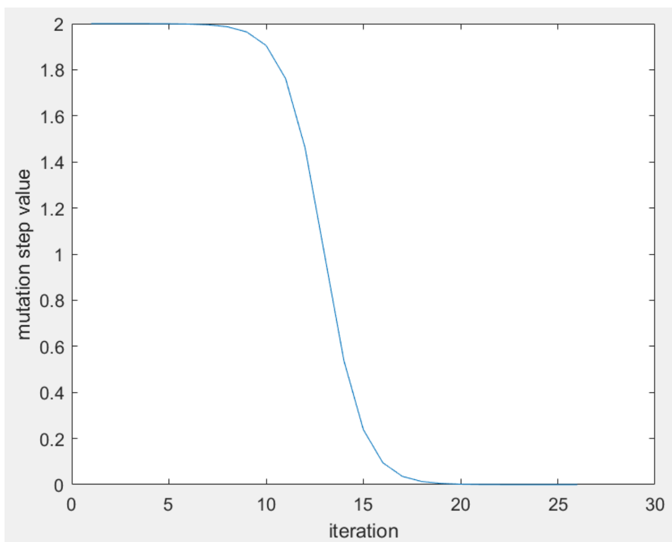
Wyniki:



Wizualizacja wykresu funkcji siódmej z algorytmem gradientowym wielostartowym



Krzywa konwergencji funkcji gradientowej wielostartowej



Sigmoid funkcji gradientowej wielostartowej

Wnioski:

Algorytm miał problem trafić w minimum globalne. Bardzo łatwo za to eksploatował minima lokalne i globalne, jeśli w nie trafił. Oznacza to, że algorytm jest podatny na minima lokalne, za to po natrafieniu na minima szybko zaczyna oscylować wyniki.

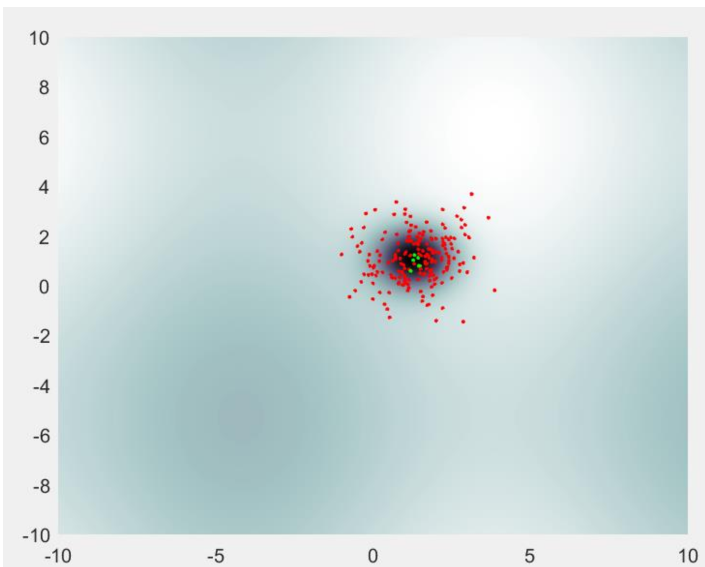
Algorytm 1+1 dla funkcji fewminima

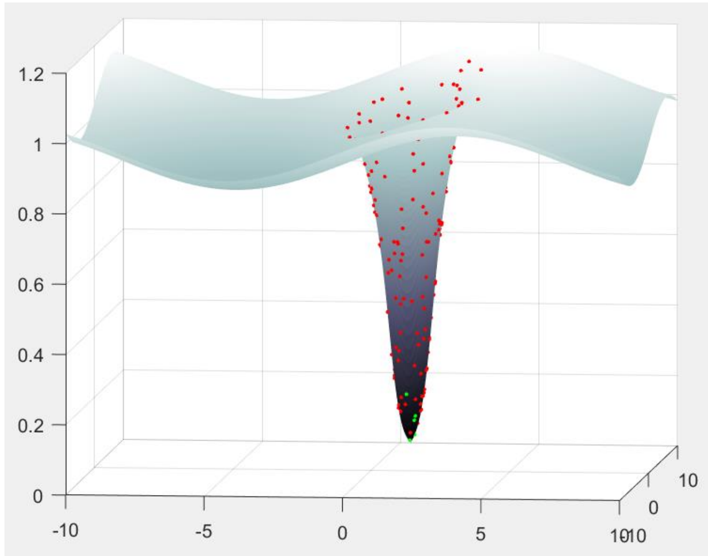
Algorytm 1+1 działa poprzez wyznaczenie losowego punktu a następnie, wykonanie w losowym kierunku kroku o określonej wartości. Sprawdzamy wartość funkcji w tym punkcie i jeżeli jest jak do tej pory najlepsza, zapamiętujemy ten punkt i od niego wykonujemy kolejne iteracje algorytmu. Jeżeli wynik będzie gorszy niż posiadany obecnie, wykonujemy krok w innym kierunku. Manipulujemy to parametrem „Step", który określa na odległość której sprawdzamy wartość punktu.

Parametry początkowe:

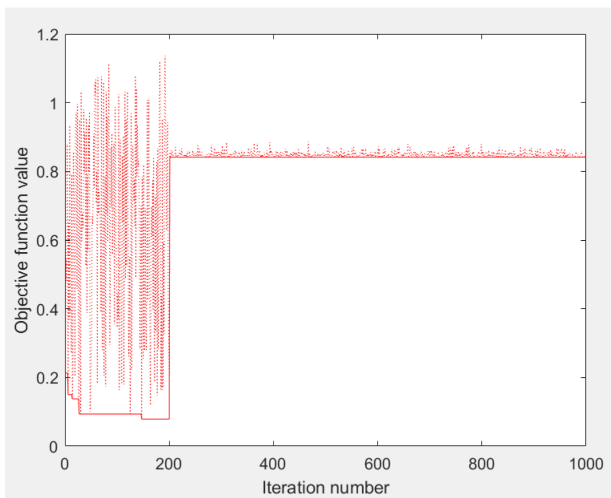
```
MaxSteps = 200;
FunctionPlot = 1;
PointPlot = 1;
ConvergenceColor = 'r';
Step = 1;
```

Wyniki:





Wizualizacja algorytmu 1+1 dla funkcji fewminima

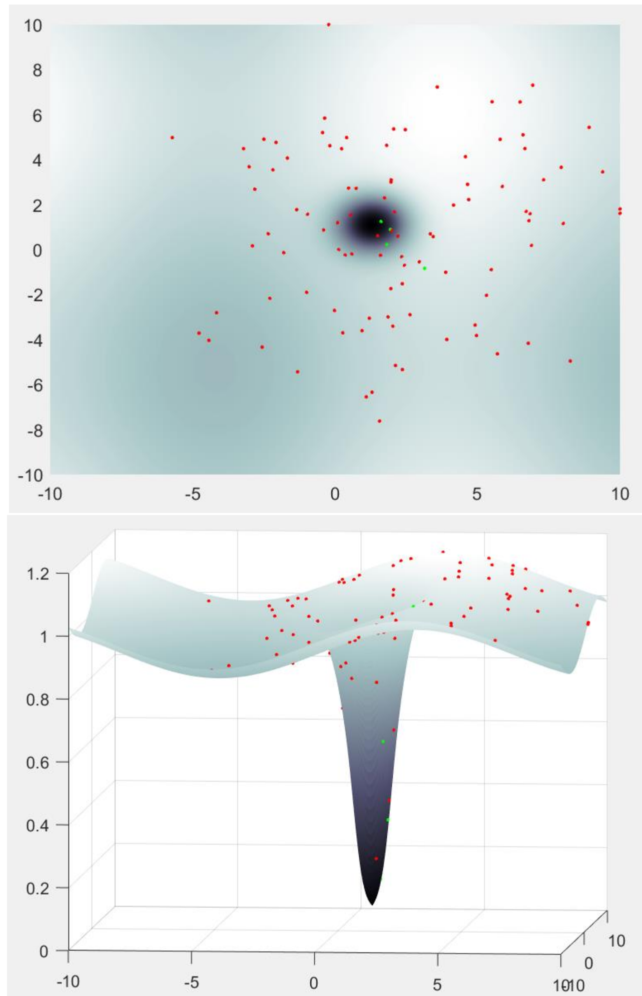


Krzywa konwergencji algorytmu 1+1 dla funkcji fewminima

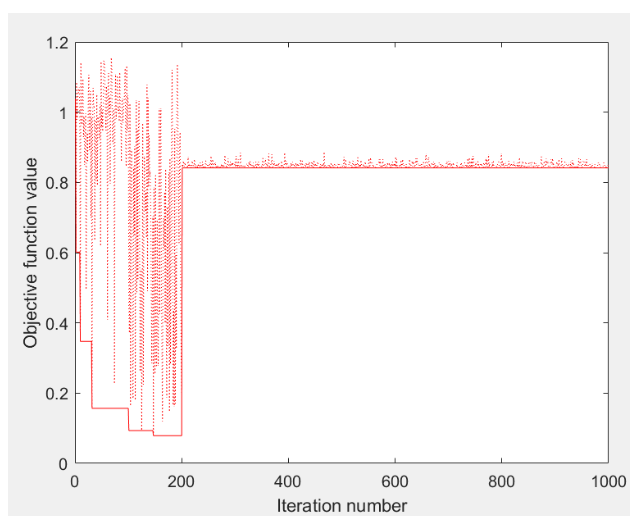
Parametry końcowe:

```
MaxSteps = 100;
FunctionPlot = 1;
PointPlot = 1;
ConvergenceColor = 'r';
Step = 4;
```

Wyniki:



Wizualizacja algorytmu 1+1 dla funkcji fewminima



Krzywa konwergencji algorytmu 1+1 dla funkcji fewminima

Wnioski:

Algorytm prawie od razu wpadł w minimum lokalne, a następnie wynik jedynie się pogarszał. Prawdopodobnie użyłam za dużej ilości iteracji, jednak ze względu na wygląd mojej funkcji postanowiłam, że większa ilość da lepsze wyniki. Ponieważ moja funkcja ma jedno główne minimum i jest mała szansa na trafienie w nie.

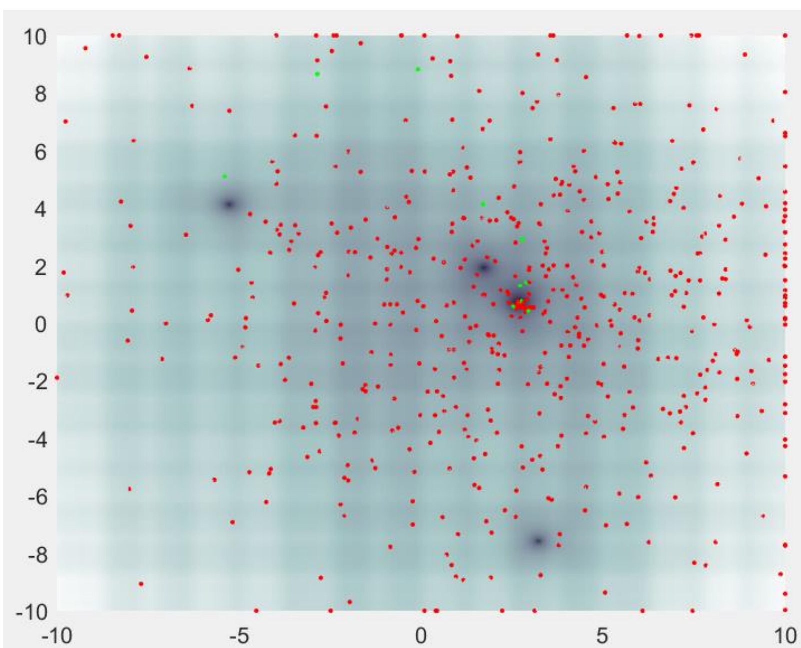
W drugim przypadku algorytm ze zwiększonym krokiem ma mniejszą szansę na dokładną eksploatację, za to znacznie zwiększyła się możliwość wejścia w obszar minimum globalnego.

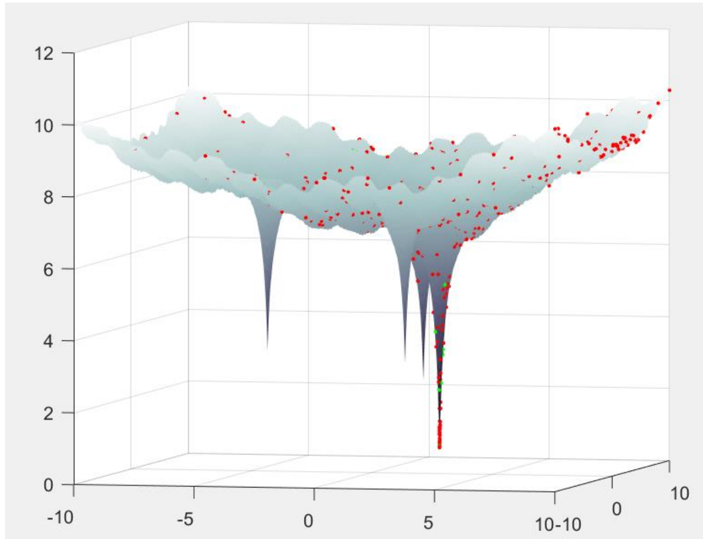
Algorytm 1+1 dla funkcji manyminima

Algorytm 1+1 ze zmiennym krokiem działa na losowym sprawdzaniu wartości w kolejnych punktach rozstawionych w niezależnych od siebie kierunkach w różnych odległościach od siebie nawzajem. Jak przy funkcji fewminima, jeśli wartość jest najlepsza do tej pory, to zostaje zapamiętana.

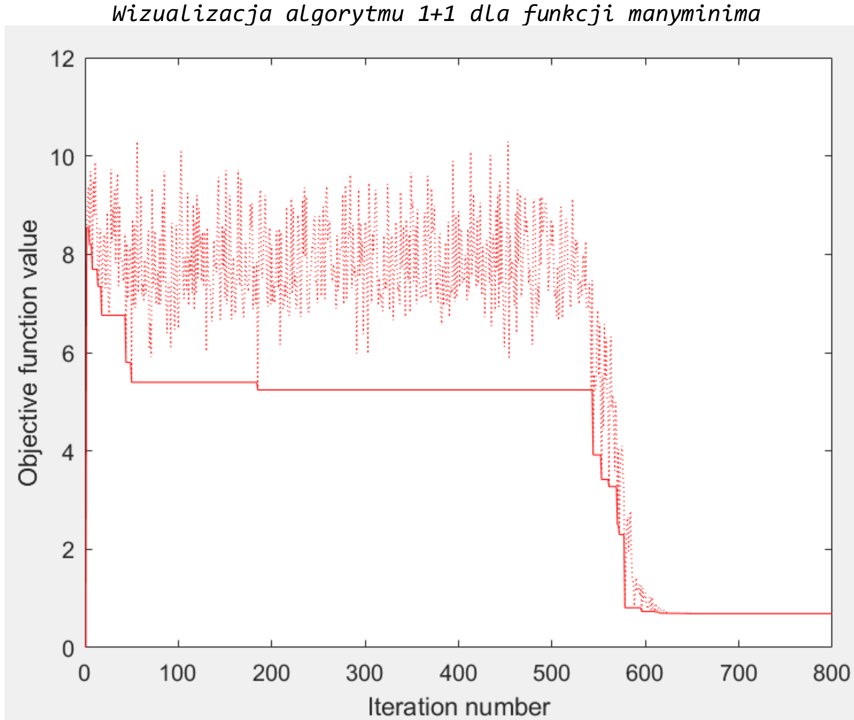
Dobrane parametry:

```
MaxSteps = 800;
InitialStep = 5
P1 = 1.5;
P2 = 10;
```





Wizualizacja algorytmu 1+1 dla funkcji manyminima



Krzywa konwergencji dla algorytmu 1+1 funkcji manyminima

Wnioski:

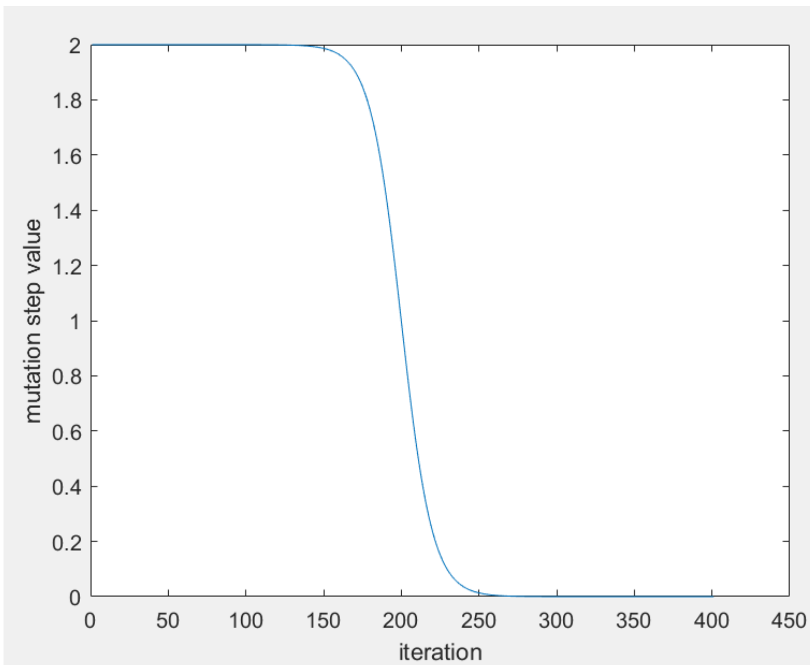
Na początku jest duże rozproszenie punktów na całej powierzchni wykresu, co gwarantuje odpowiedni dobrą eksplorację, więc i większą szansę na znalezienie minimów. Zagęszczenie w okolicy jednego z minimów świadczy o możliwości znalezienia minimum globalnego i rozpoczęciu jego eksploatacji.

Algorytm genetyczny dla funkcji fewminima

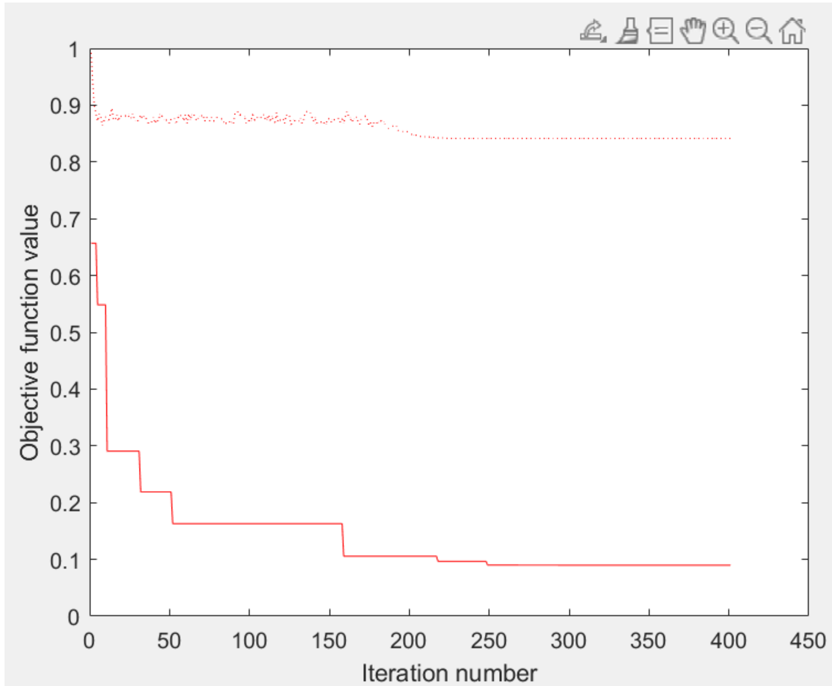
Używamy algorytmu genetycznego z selekcją n-najlepszych. Więc po wygenerowaniu rozwiązań i ocenie ich przystosowania, algorytm szereguje osobniki w pod względem ich jakości i wybiera 'n' najlepszych.

Parametry wstępne:

```
P_size = 80;
n = 35;
InitialStep = 2;
P1 = 2;
P2 = 10;
```



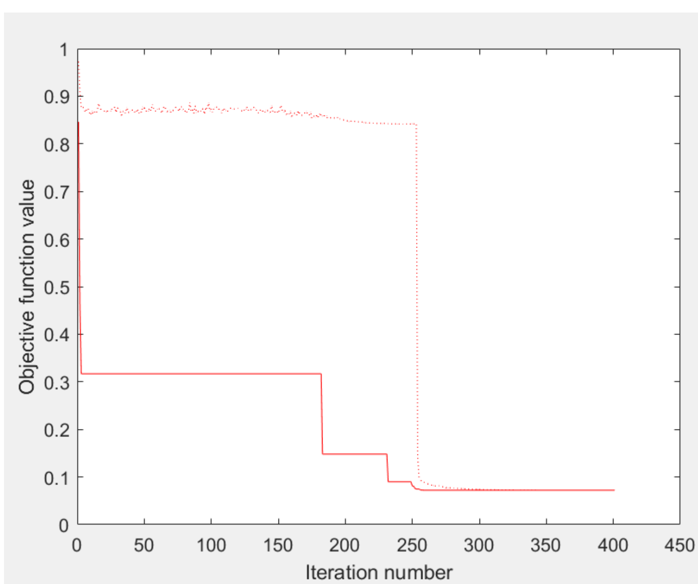
Sigmoid dla algorytmu genetycznego funkcji fewminima



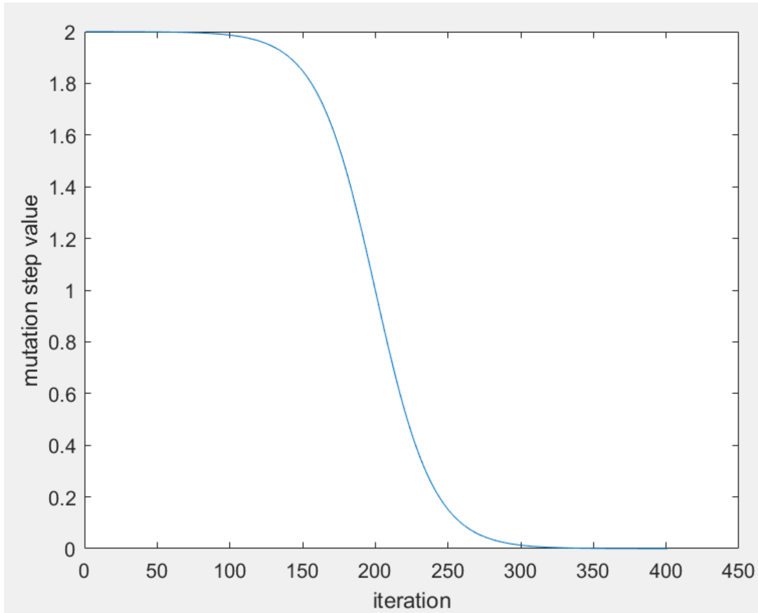
Krzywa konwergencji dla algorytmu genetycznego funkcji fewminima

Parametry końcowe:

- P_size = 100; - Zwiększyłam wielkość generacji, by wykorzystać lepiej dostępne iteracje.
- n = 25; - Zwiększyłam 'n' by zachować różnorodność i zwiększyć odporność na minima lokalne.
- InitialStep = 2;
- P1 = 2;
- P2 = 20; - Zwiększyłam by spłaszczyć sigmoid.



Krzywa konwergencji algorytmu genetycznego dla funkcji fewminima



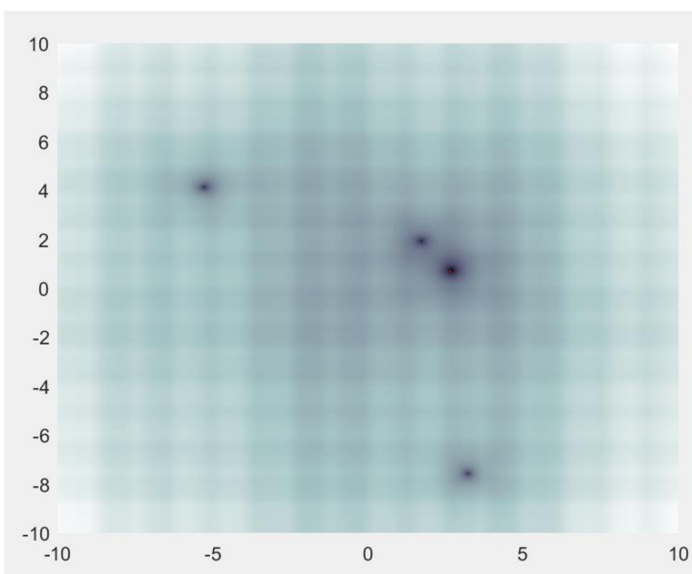
Sigmoid dla algorytmu genetycznego funkcji fewminima

Algorytm genetyczny dla funkcji manyminima

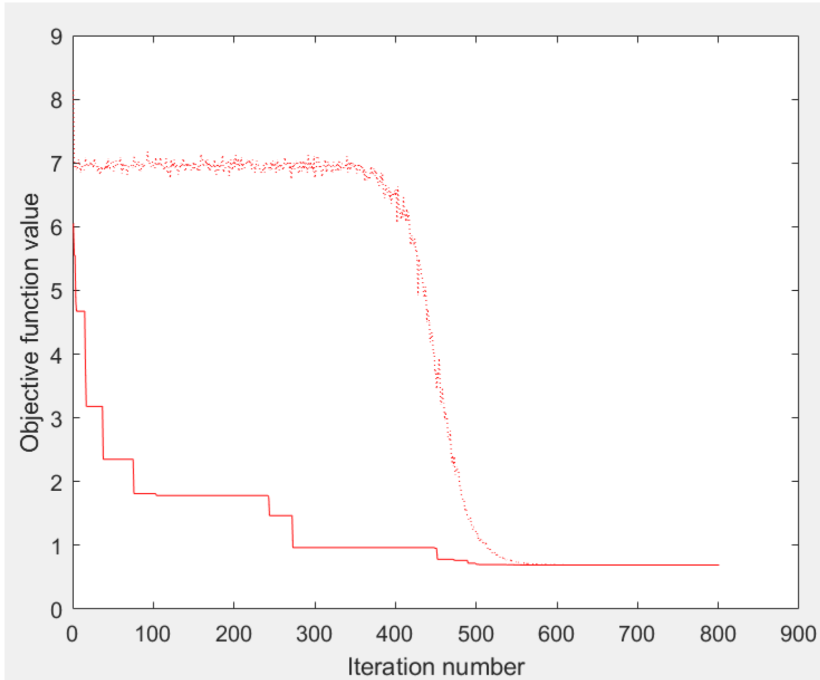
Używamy tej samej selekcji jak przy funkcji fewminima.

Parametry wejściowe:

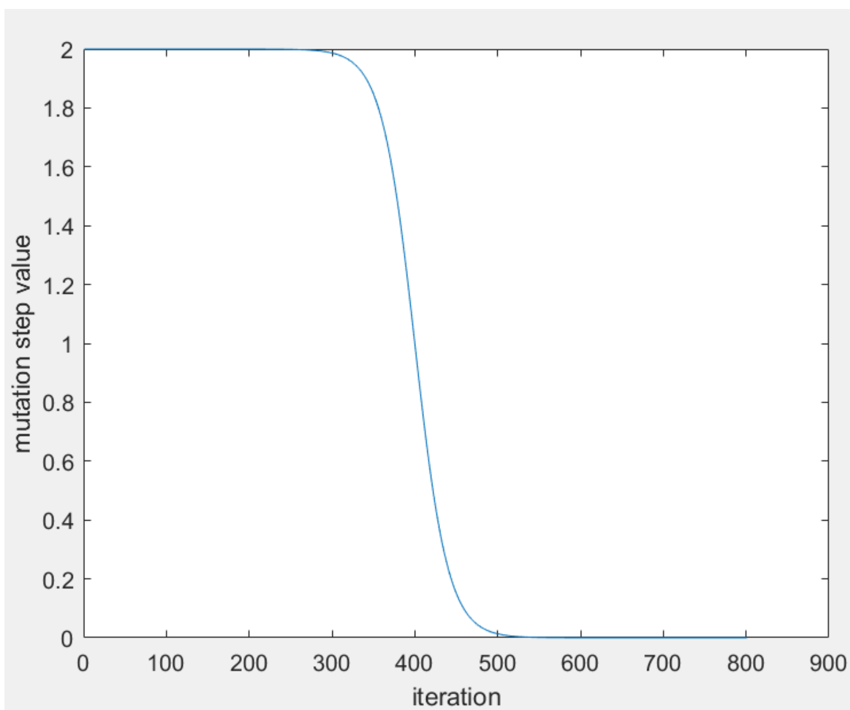
```
P_size = 50;
n = 15;
InitialStep = 2;
P1 = 2;
P2 = 20;
```



Wizualizacja algorytmu genetycznego dla funkcji manyminima



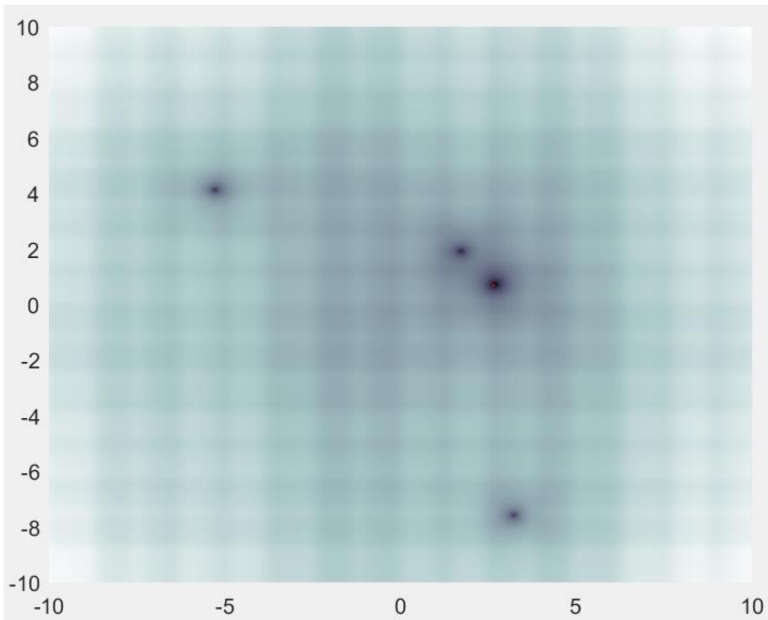
Krzywa konwergencji algorytmu genetycznego dla funkcji manyminima



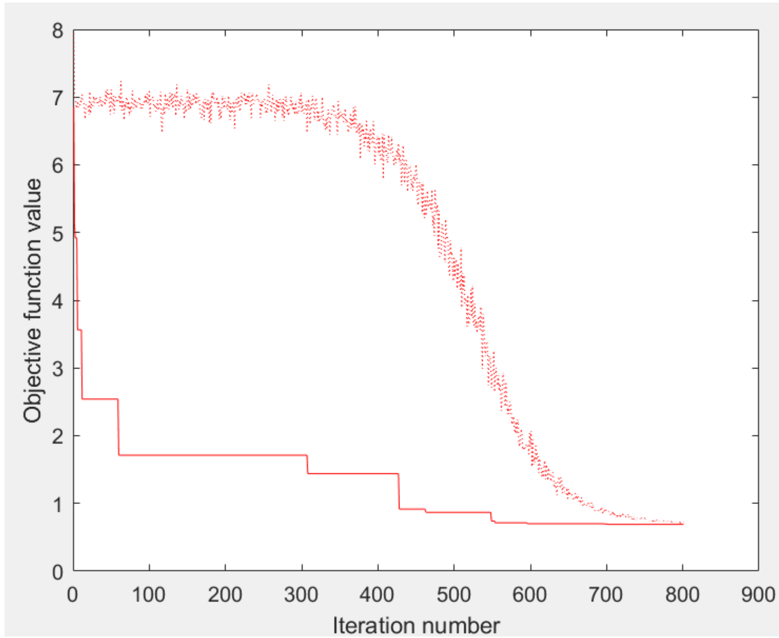
Sigmoid algorytmu genetycznego dla funkcji manyminima

Parametry końcowe:

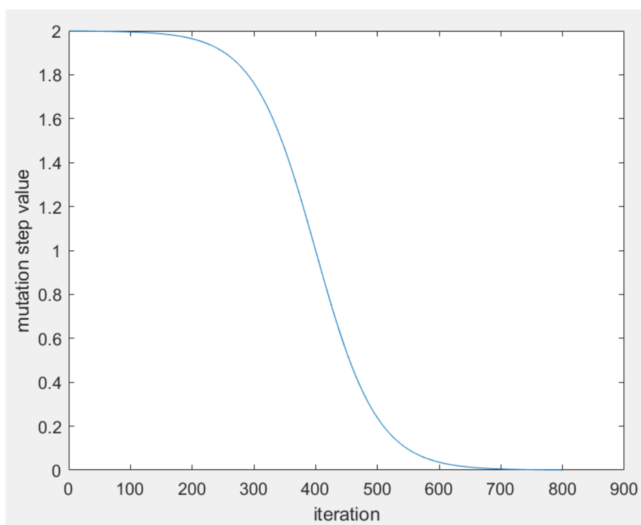
```
P_size = 20;
n = 5;
InitialStep = 2;
P1 = 2;
P2 = 50;
```



Wziualizacja algorytmu genetycznego dla funkcji manyminima



Krzywa konwergencji algorytmu genetycznego dla funkcji manyminima



Sigmoid algorytmu genetycznego dla funkcji manyminima

Wnioski:

Przy początkowych parametrach krzywa konwergencji wygląda tak samo jak przy parametrach końcowych. Oznacza to, że udało się osiągnąć taki sam wynik przy użyciu mniejszej mocy obliczeniowej. Kształt krzywej konwergencji wskazuje na szybkie wejście w minimum lokalne, czyli problem może być trywialny, a później następuje eksploatacja. Wniosek ten potwierdza animacja wizualizacji wykresu, która szybko zagęszcza się w okolicy minimum globalnego i dość długo je eksploatuje.

Statystyka

Algorytm	Gradientowy wielostartowy	1+1 dla fewminima	1+1 dla manyminima	Genetyczny dla fewminima	Genetyczny dla manyminima
	0.864837	0.072178	0.691040	0.0723	1.1189
	0.864838	0.864093	2.505060		
	0.097586	0.072179	0.691042	0.0748	1.7735
	0.841594	0.841332	0.691468		
	0.842273	0.072178	2.505938	0.0757	1.2694
	0.842027	0.072178	2.505060		
	0.841881	0.072178	0.691040	0.0722	0.7928
	0.842077	0.072178	0.691040		
	0.842195	0.072178	0.691040	0.0722	1.7209
	0.841732	0.072180	0.691088		
	0.842519	0.072178	2.505060	0.0728	1.5114
	0.087326	0.072179	0.732688		
	0.842401	0.072178	0.691040	0.0976	2.5378
	0.841536	0.072179	0.741377		
	0.152304	0.072178	0.691040	0.0722	1.2009
	0.842145	0.841332	0.691040		
	0.841531	0.072179	1.517436	0.0722	1.9796
	0.841676	0.864093	0.691040		
	0.842171	0.072178	0.691040	0.0722	1.2729
	0.841968	0.072179	0.691040		
Odchylenie	0.2686	0.3204	0.7342	0.0079	0.5019
Średnia	0.7348	0.2283	1.0998	0.0754	1.5178

Wnioski:

1. Funkcja fewminima:
    - Algorytm gradientowy wielostartowy – Analizując statystki można zauważyć, że algorytm ma duże problemy by trafić w minimum globalne i w większości przypadków zatrzymuje się w minimach lokalnych.
      W porównaniu statystyk wyników tego algorytmu z innymi algorytmami przetwarzającymi funkcję fewminima okazuje się, że działanie jest znacznie mniej skuteczny od algorytmu genetycznego. Jednak patrząc na wyniki algorytmu 1+1, to można wyciągnąć wnioski, że rozstrzał wyników w algorytmie 1+1 jest szerszy niż w gradientowym, ale częściej trafia on w minimum globalne funkcji.

- Algorytm 1+1 – Patrząc na wyniki można stwierdzić, że algorytm miewa problemy z wyjściem z minimów lokalnych, jednak o wiele rzadziej niż algorytm gradientowy wielostartowy. Nie jest jednak tak skuteczny jak algorytm genetyczny.
- Algorytm genetyczny – Nie nieporównywalnie skuteczniejszy od pozostałych dwóch algorytmów. Jego odchylenie standardowe jest minimalne, a średnia niewiele różni się od rzeczywistego wyniku.

2. Funkcja manyminima:
    - Algorytm 1+1 – Wyniki wskazują, że algoytm ten lepiej przebadał funkcję manyminima niż algorytm genetyczny. Znacznie częściej wpadał w minimum globalne, jednak też miał szerszy zakres wyników.
    - Algorytm genetyczny – Miał problemy ze znalezieniem minimum globalnego i zazwyczaj zatrzymywał się przy podobnych wartościach, co spowodowało mniejsze niż w 1+1 odchylenie standardowe. Z wyników odczytujemy, że tylko raz był blisko minimum globalnego. Kiepskie wyniki mogą być spowodowane małą populacją początkową lub za dużym naciskiem selektywnym.