

# SPRAWOZDANIE

## z ćwiczenia nr 1

### “Algorytmy ewolucyjne i genetyczne”

## Przedmiot: Wprowadzenie do Sztucznej Inteligencji

Imię i nazwisko: Viktoriia Nowotka

## Odpowiedzi na pytania:

- Trywialne rozwiązań mogłyby być ciągiem 0. Wtedy odległość od docelowego punktu będzie maksymalna (350), co daje wynik -122500.

```
Viktorias-MacBook-Pro:lab1 viktorianowotka$ python3 index.py  
[-122500.]
```

Rys. 1 – wynik trywialnego rozwiązania

- Górnym ograniczeniem wyniku może być 0, wtedy jak obiekt dotarł do punktu docelowego.
  - Przykład rozwiązania losowego oraz wynik przedstawione są na rys. 2.

```
Viktorias-MacBook-Pro:lab1 viktoriajanowotka$ python3 index.py
rozwiazanie losowe: [[1 0 1 0 1 1 0 0 1 1 1 1 0 1 0 0 1 0 1 1 0 1 0 0 0 0 1 1 0 1 1 1 0 0 0 0
1 1 0 1 1 0 0 1 0 1 1 0 1 1 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 0 1 1 0 1 1 1 1
1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0
0 0 0 1 0 0 1 0 0 1 1 1 1 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 1 1 1 0 1 0 0 1
0 0 0 0 1 1 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 1 1 1 0 1 1 1 0 0 1 0 1 1 0 1 0 1
1 1 0 0 1 0 0 1 0 1 0 1 0 0 1 1 0 1 1 0 1 1 0 0 0 0 1 0 0 0 0 1 0 1 0 1
1 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 1 0 1 0 1 1 1 1 0 1
1 0 0 0 0 1 1 1 1 1 0 1 0 1 0 0 1 0 1 0 1 1 0 0 1 0 1 0 1 0 1 1 1 0 1 1
1 1 0 0 1 0 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 1 0 0 1 1 1 0 0 1 1 1 0
0 0 1 0 0 0 0 1 1 0 0 1 0 1 1 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 0 0 1 0 0 1 0 1
1 1 0 1 1 1 0 0 1 1 0 0 0 0 1 1 1 0 1 1 0 1 1 0 1 0 1 0 1 0 0 0 0 1 0 1 0 1
1 0 0 1]]
```

Rys. 2 – rozwiązanie losowe

Podczas implementacji algorytmu genetycznego, na bazie interfejsu podanego na repozytorium GitLabie, było zrealizowano:

- selekcję ruletkową
- krzyżowanie jednopunktowe
- mutację
- sukcesję generacyjną

W trakcie wielokrotnych uruchomień algorytmu było ustalone następujące hiperparametry, które wstępnie dają dobry wynik za każdym razem:

- $m = 250$
- $p_c = 0.7$
- $p_m = 0.002$
- $t_{max} = 200$

Na początku maksymalna liczba iteracji w ewolucji była brana za 300, ale zauważono tendencje do odnajdywania najlepszego wyniku za około 50-100-150 iteracji, dlatego maksymalną liczbę zmniejszono do 200.

```
Viktoriias-MacBook-Pro:lab1 viktorianowotka$ python3 index.py
Uruchomienie optymalizacji...

Uruchomienie algorytmu: 1
Najlepsze w iteracji 0: -85130.479
Najlepsze w iteracji 50: -533.027
Najlepsze w iteracji 100: -309.196
Najlepsze w iteracji 150: -0.01

Uruchomienie algorytmu: 2
Najlepsze w iteracji 0: -71253.51
Najlepsze w iteracji 50: -356.432
Najlepsze w iteracji 100: -30.941

Uruchomienie algorytmu: 3
Najlepsze w iteracji 0: -81539.827
Najlepsze w iteracji 50: -0.244
Najlepsze w iteracji 100: -0.022
```

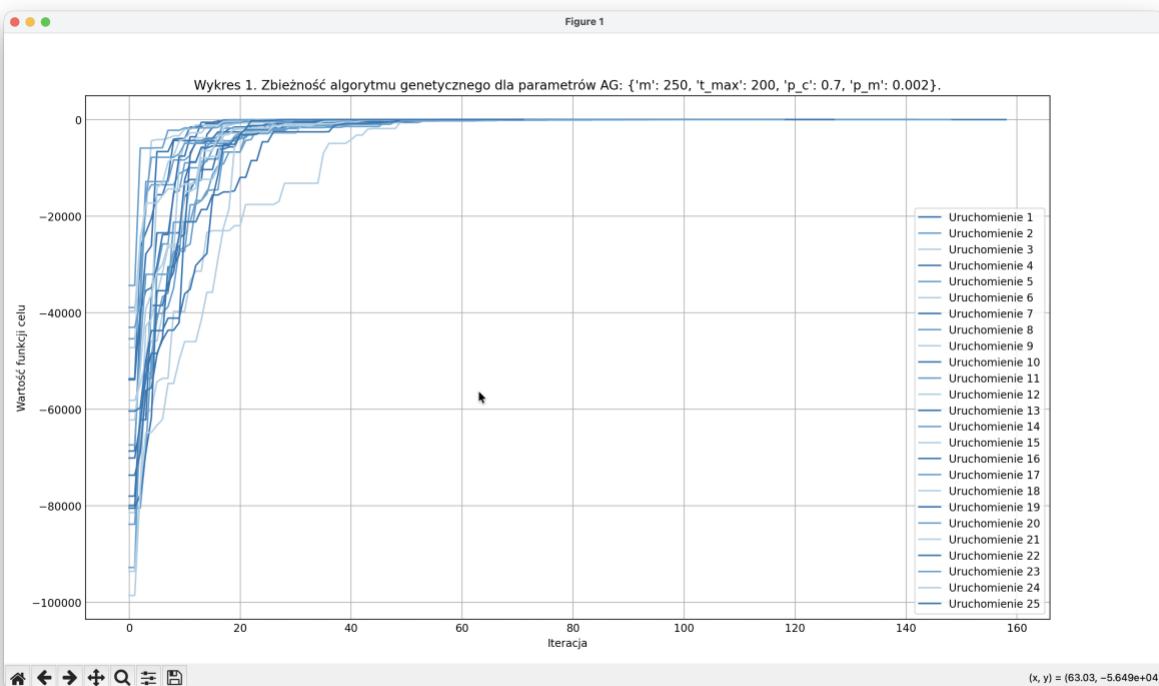
Rys. 2 – przykładowe uruchomienie algorytmu, gdzie dobry wynik znajdowany do 200ej iteracji

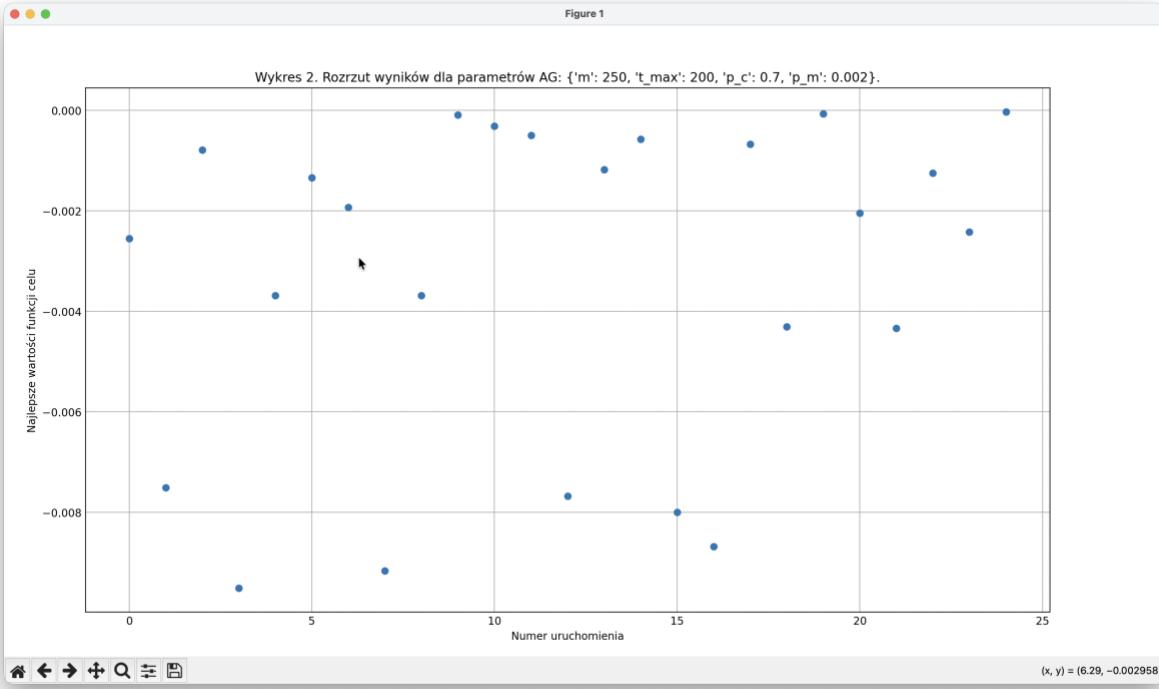
W związku z szybkim odnajdywaniem najlepszego wyniku, który zbliża się do 0.0 i pozostaje w tym punkcie dodano dodatkowy warunek stopu w ewolucji, gdyż znaleziono wystarczająco dobry wynik.

```
while t < self.t_max and o_best < -0.01:
```

Rys. 3 – umowa stopu algorytmu genetycznego

Potwierdzeniem, że dobrane hiperparametry są dobre, jest wynik uruchomienia programu 25 razy. Poniżej dodano wykresy zbieżności algorytmu oraz rozrzutu wyników, obliczone wartości średnie wyników oraz std, najlepszy i najgorszy wyniki z puli.





Zbadano wpływ wielkości populacji na działanie danej implementacji algorytmu genetycznego. W trakcie badania przeprowadzono 7 testów z różnymi wartościami parametru  $m$  - od 350 do 50. Wyniki testów podano w tabeli nr 1. Na podstawie uzyskanych danych można stwierdzić, że minimalna populacja dla rozwiązania tego problemu wynosi 150 osobników. Populacja o wielkości 250-300 osobników daje optymalne wyniki dla danego problemu.

W takim przypadku (250-300 os. w populacji) liczba iteracji skraca się w średnim znaczeniu o kilka sztuk, wtedy jak w najgorszych wypadkach różnica może sięgać 30 iteracji. Na podstawie przedstawionych wyników można uznać, że populacji w 350 osobników jest zdecydowanie zbyt duża, ponieważ wyniki są podobne do testu z populacją 300-osobowej, natomiast liczba iteracji wzrasta. Powoduje to nieefektywne wykorzystanie pamięci i mocy obliczeniowej urządzenia.

Tabela 1 – Badanie hiperparametru wielkości populacji.

| Nr | Parametr<br><b>m</b> | Najgorszy | Średni | Std  | Najlepsze | Liczba iteracji |         |     |
|----|----------------------|-----------|--------|------|-----------|-----------------|---------|-----|
|    |                      |           |        |      |           | min             | średnia | max |
| 1  | 350                  | -0.01     | 0.0    | 0.0  | 0.00      | 174             | 86      | 41  |
| 2  | 300                  | -0.01     | 0.0    | 0.0  | 0.00      | 159             | 88      | 28  |
| 3  | 250                  | -0.01     | 0.0    | 0.0  | 0.00      | 158             | 90      | 30  |
| 4  | 200                  | -0.01     | 0.0    | 0.0  | 0.00      | 174             | 102     | 43  |
| 5  | 150                  | -0.01     | 0.0    | 0.0  | 0.00      | 193             | 93      | 41  |
| 6  | 100                  | -0.08     | -0.01  | 0.02 | 0.00      | 200             | 99      | 23  |
| 7  | 50                   | -0.01     | 0.0    | 0.0  | 0.00      | 124             | 94      | 73  |

Zbadano wpływ prawdopodobieństwa mutacji na działanie danej implementacji algorytmu genetycznego. W trakcie badania przeprowadzono 7 testów z różnymi wartościami parametru **p\_m** - od 0,0005 do 0,2. Wyniki testów podano w tabeli nr 2. Na podstawie uzyskanych danych można stwierdzić, że optymalne prawdopodobieństwo mutacji dla danej implementacji wynosi 0,0005-0,002. Zwiększenie tego parametru pogarsza średnie wyniki oraz wydłuża liczbę iteracji do maksymalnej dopuszczonej wartości.

Tabela 2 – Badanie hiperparametru prawdopodobieństwa mutacji.

| Nr | Parametr<br><b>p_m</b> | Najgorszy | Średni | Std  | Najlepsze | Liczba iteracji |         |     |
|----|------------------------|-----------|--------|------|-----------|-----------------|---------|-----|
|    |                        |           |        |      |           | min             | średnia | max |
| 1  | 0.0005                 | -0.01     | 0.00   | 0.00 | 0.00      | 107             | 61      | 36  |
| 2  | 0.002                  | -0.01     | 0.00   | 0.00 | 0.00      | 200             | 94      | 39  |
| 3  | 0.005                  | -2.01     | -0.09  | 0.39 | 0.00      | 200             | 127     | 47  |
| 4  | 0.007                  | -0.50     | -0.02  | 0.10 | 0.00      | 200             | 130     | 40  |
| 5  | 0.010                  | -0.26     | -0.03  | 0.07 | 0.00      | 200             | 126     | 26  |
| 6  | 0.015                  | -0.62     | -0.04  | 0.12 | 0.00      | 200             | 142     | 32  |
| 7  | 0.020                  | -0.56     | -0.06  | 0.12 | 0.00      | 200             | 158     | 67  |

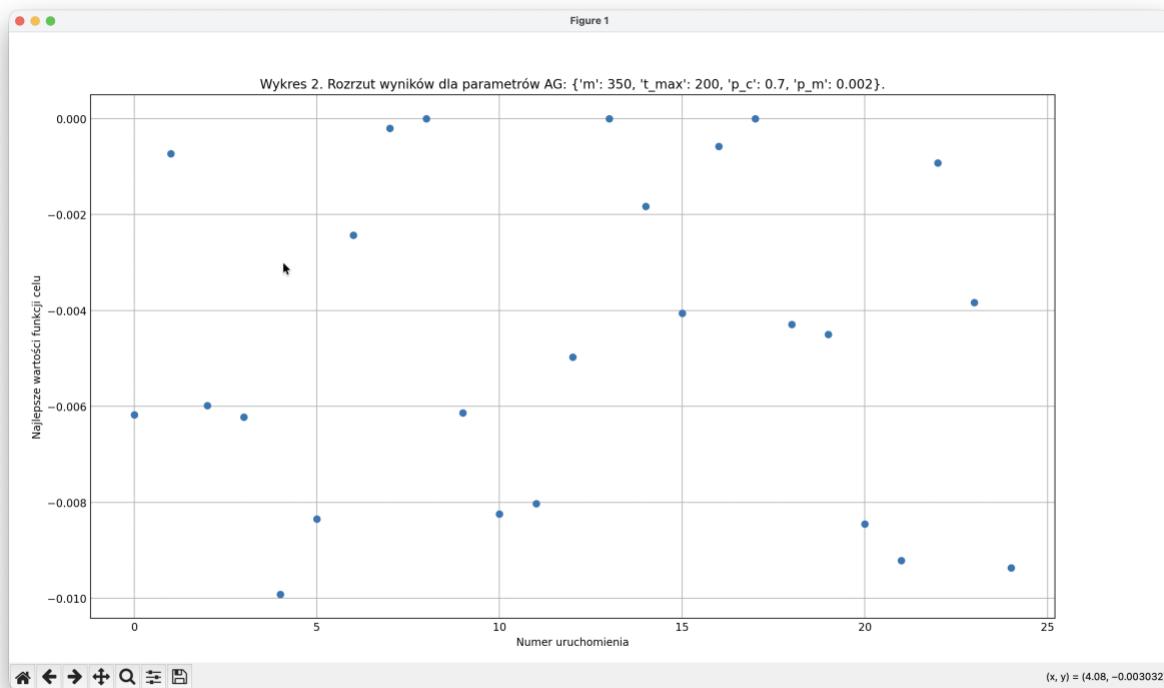
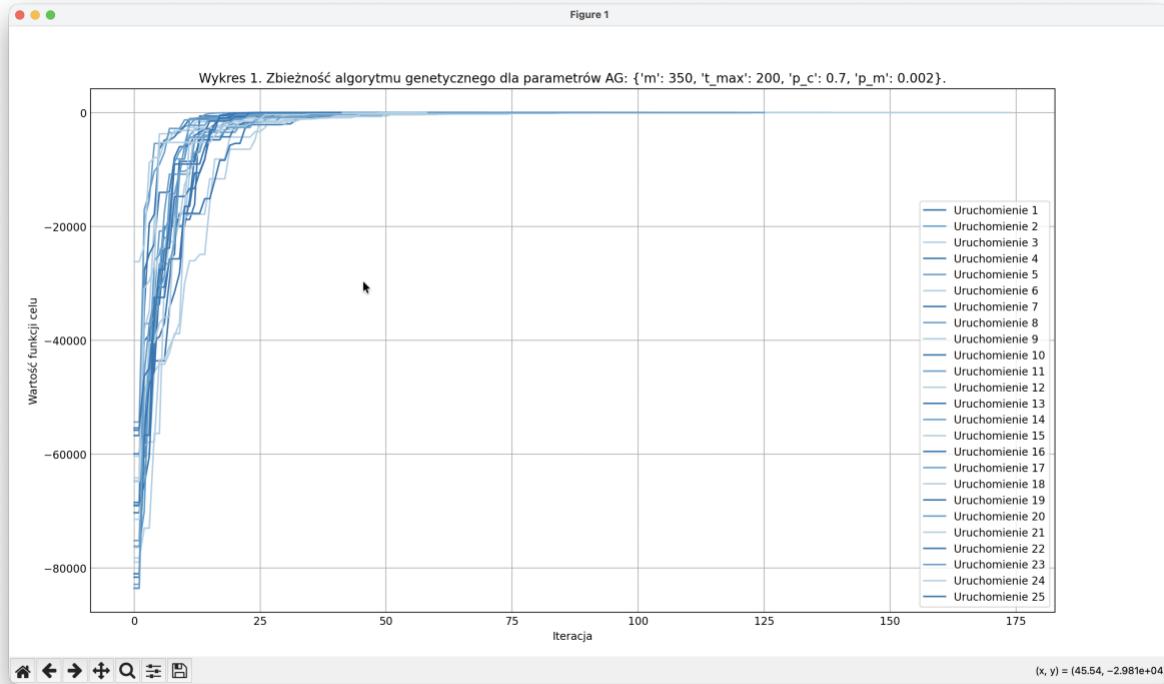
## **Wnioski**

W trakcie wykonania ćwiczenia nr 1 zaimplementowano algorytm genetyczny z selekcją ruletkową, krzyżowaniem jednopunktowym, mutacją oraz sukcesją generacyjną. Metodą ampiryczną znaleziono zestaw optymalnych wartości hiperparametrów:  $m=250$ ,  $t\_max=200$ ,  $p\_c=0,7$ ,  $p\_m=0,002$ .

Zbadano wpływ dwóch hiperparametrów na optymalizację danego problemu: liczby osobników w populacji oraz prawdopodobieństwa mutacji. Optymalna wielkość populacji mieści się w przedziale 250-300 osobników, natomiast prawdopodobieństwo mutacji w zakresie 0,0005-0,002. Zwiększenie danych hiperparametrów powoduje wzrost liczby iteracji.

Badania zostały przeprowadzone przy  $t\_max = 200$ ,  $p\_c = 0,7$ .

# TESTY dla wielkości populacji



# TESTY dla prawdopodobieństwa mutacji

