



**Wydział Elektroniki  
i Technik Informatycznych**

POLITECHNIKA WARSZAWSKA

# Wstęp do Sztucznej Inteligencji

## Ćwiczenie nr 2: Algorytmy genetyczne i ewolucyjne

Kaczmarek Robert  
293377

Warszawa 2021

## Polecenie

Zadanie polega na implementacji klasycznego algorytmu ewolucyjnego bez krzyżowania, z selekcją turniejową i sukcesją generacyjną dla problemu komiwojażera. Pan Mateusz musi objechać 30 wielkich miast, by dostarczyć szczepionki. Poszukiwany jest najkrótszy cykl, pozwalający na przejechanie przez wszystkie miasta dokładnie raz jak najkrótszą / najszybszą drogą, zaczynając i kończąc w tym samym mieście.

Należy znaleźć najlepsze rozwiązanie przy ograniczonym budżecie.

## Użyte narzędzia

Do wykonania zadania został użyty Python w wersji 3.8.2 oraz następujące biblioteki i ich wersje:

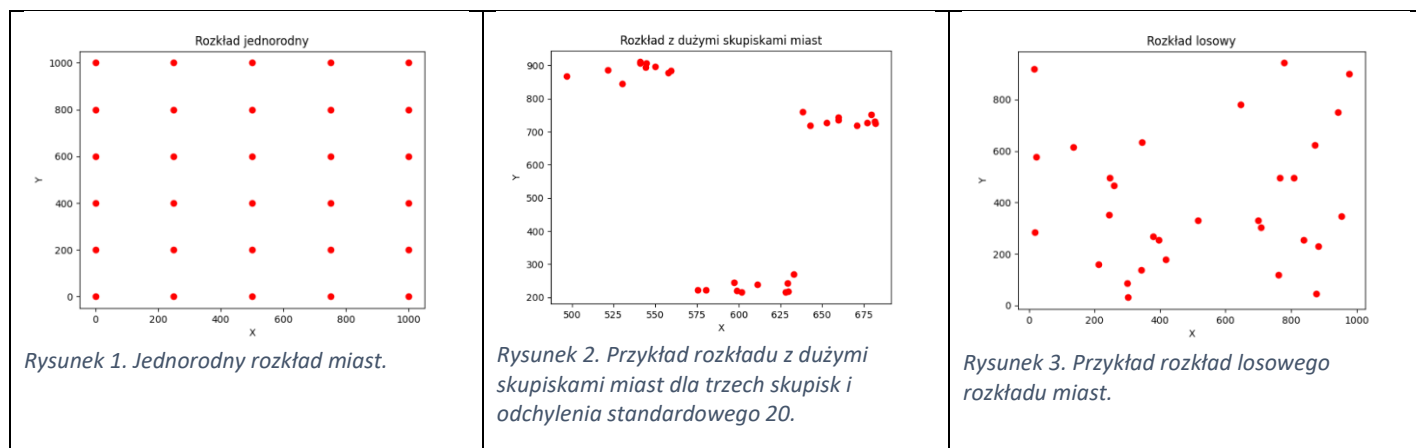
- matplotlib==3.4.3,
- numpy==1.21.4,
- random,
- sklearn==0.0,
- time.

## Wyniki

Na potrzeby zadania została stworzona lista 30 nazw miast znajdujących się w pliku *cities.txt*. Zbiór danych to współrzędne miast (x, y) w przestrzeni 1000x1000. Każdemu miastu przypisywane są współrzędne (x, y). Generacja współrzędnych miast została przeprowadzona dla trzech różnych rozkładów miast:

- rozkład jednorodny,
- rozkład z dużymi skupiskami miast,
- rozkład losowy.

Przykładowe reprezentacje wymienionych wyżej rozkładów przedstawiają rysunki 1-3.



Elementy problemu komiwojażera w kontekście algorytmu genetycznego zostały zdefiniowane następująco:

- osobnik populacji – pewien cykl przechodzący przez wszystkie miasta raz, zaczynający i kończący na tym samym,
- populacja – zbiór cykli,
- gen – pojedyncze miasto w cyklu,
- selekcja turniejowa – porównanie długości cykli losowo wybranych K osobników bez zwracania, wynik selekcji to najkrótszy cykl z wybranych osobników,
- mutacja – zamiana miejscami w cyklu dwóch losowo wybranych miast z pewnym prawdopodobieństwem zajścia mutacji.

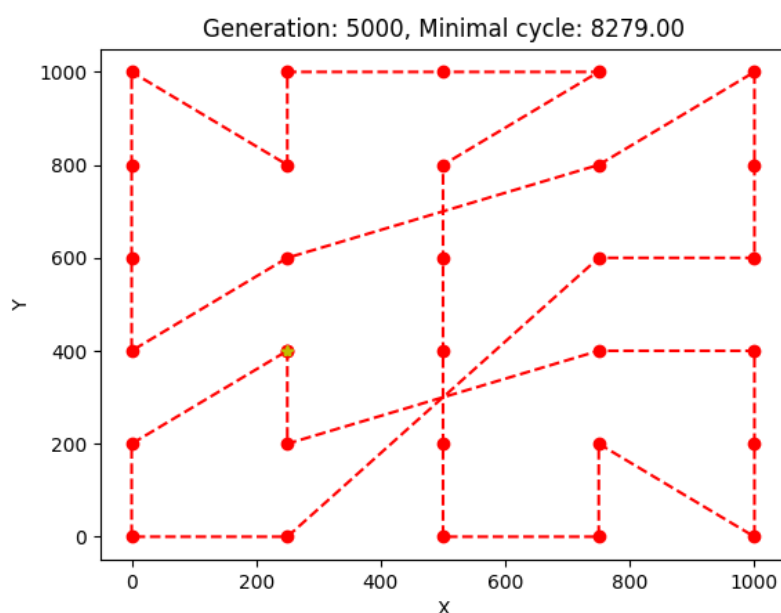
Warunek stopu algorytmu to maksymalna liczba generacji. Otrzymane wyniki zostały przedstawione w tabelach 1-3. Aby nazwy kolumn zajmowały mniej miejsca w tabeli wprowadzone zostały następujące oznaczenia:

Calls number – liczba wywołań algorytmu,  $Generation_{max}$  – maksymalna liczba generacji, Population – liczba populacji bazowej, Centres – liczba skupisk miast, Best – najkrótszy znaleziony cykl z najkrótszych,  $Bets_{avg}$  – średni najkrótszy znaleziony cykl, Std – odchylenie standardowe najkrótszych znalezionych cykli, K - liczba osobników wziętych do turnieju,  $P_m$  – prawdopodobieństwo mutacji,  $t_{avg}$  – średni czas wykonania algorytmu, Seed – ziarno generatora ustawiane poprzez `random.seed(Seed)` lub `np.random.seed(Seed)` w kolejnych wywołaniach algorytmu. Przypisanie współrzędnych do miast  $Seed = 0$ .

Tabela 1. Badanie wpływu różnych parametrów początkowych na działanie algorytmu genetycznego przy jednorodnym rozkładzie miast.

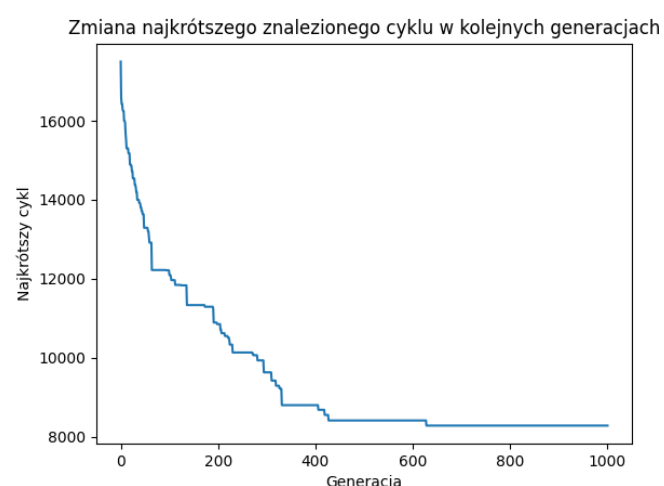
| Rozkład jednorodny |                    |            |         |              |        |   |       |               |      |
|--------------------|--------------------|------------|---------|--------------|--------|---|-------|---------------|------|
| Calls number       | $Generation_{max}$ | Population | Best    | $Best_{avg}$ | Std    | K | $P_m$ | $t_{avg}$ [s] | Seed |
| 10                 | 1000               | 10         | 8156,19 | 8689,33      | 351,81 | 2 | 0,1   | 11,81         | 1-10 |
| 10                 | 1000               | 100        | 7078,83 | 7990,27      | 482,23 | 2 | 0,1   | 109,64        | 1-10 |
| 10                 | 1000               | 250        | 7740,84 | 8396,12      | 308,34 | 2 | 0,1   | 273,01        | 1-10 |
| 10                 | 1000               | 100        | 7078,83 | 7990,27      | 482,23 | 3 | 0,1   | 111,18        | 1-10 |
| 10                 | 1000               | 100        | 7078,83 | 7990,27      | 482,23 | 4 | 0,1   | 111,42        | 1-10 |
| 10                 | 1000               | 100        | 7531,41 | 7989,01      | 246,53 | 2 | 0,5   | 113,22        | 1-10 |
| 10                 | 1000               | 100        | 7461,25 | 8015,77      | 299,30 | 2 | 0,05  | 104,44        | 1-10 |
| 1                  | 5000               | 50         | 8278,99 | -            | -      | 2 | 0,1   | 315,91        | 1    |

W przypadku rozkładu jednorodnego, zwiększenie populacji bazowej z 10 na 100 pozwoliło na uzyskanie lepszych wyników w dłuższym czasie. Dalsze zwiększenie ze 100 na 250 nie przyniosło poprawy, wręcz otrzymany wynik jest gorszy niż dla mniejszej populacji. Zmniejszyła się jedynie wartość odchylenia standardowego. Dla badanego zbioru danych zwiększenie liczebności osobników branych do turnieju nie wpłynęła na otrzymany wynik. Zwiększenie prawdopodobieństwa mutacji z 0.1 na 0.5 negatywnie wpłynęło na najlepszy uzyskany cykl. Podobnie jak ze zwiększeniem, zmniejszenie prawdopodobieństwa nie wpłynęło na znalezienie lepszego cyklu. Zmiana prawdopodobieństwa mutacji obniżyła wartość odchylenia standardowego wyników. Rysunek 4. przedstawia przykład najlepszego znalezionej cyklu, przy czym startowe miasto oznaczone zostało żółtą gwiazdką. Nie jest to najbardziej optymalne rozwiązanie problemu, ponieważ dla innych ziaren uzyskane wyniki były ponad 1000 mniejsze.

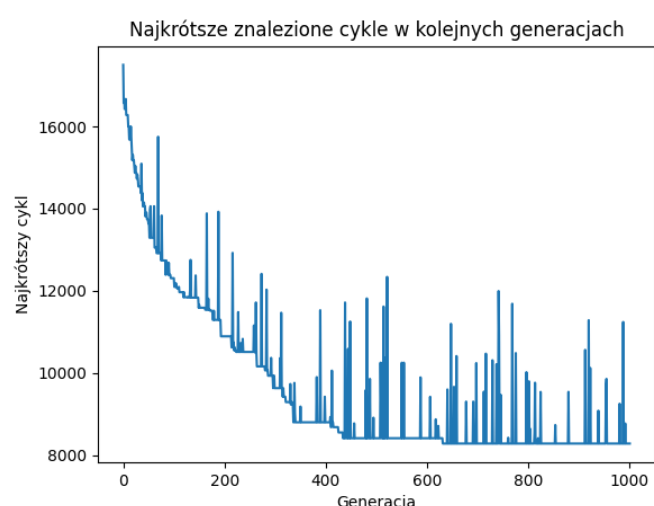


Rysunek 4. Najlepszy znaleziony cykl rozkładu jednorodnego dla  $Seed = 1$ .

W każdej iteracji algorytmu obliczany był najkrótszy cykl ze wszystkich osobników populacji, który był porównywany z dotychczas najkrótszym przechowywanym cyklem i na koniec zwracany jako wynik. Zmiana tego najkrótszego cyklu przedstawiona została na Rysunku 5. Wydawać by się mogło, że kolejne generacje mogłyby jedynie polepszyć dotychczasowy wynik. Jest to jednak błędne domniemanie co zostało zobrazowane na Rysunku 6. Powodem tego może być sposób przeprowadzanej selekcji turniejowej. W dwóch różnych turniejach mogą powtarzać się elementy, dlatego istnieje szansa, że jakiś osobnik, być może najlepszy z populacji, nie zostanie wzięty do turnieju. Oznacza to, że nie przejdzie on selekcji, więc najlepszy osobnik w nowej populacji będzie gorszy od najlepszego z poprzedniej populacji. Możliwym sposobem na wyeliminowanie takiego zjawiska jest selekcja elitarnych osobników.



Rysunek 5. Zmiana najkrótszego znalezionej cyklu w zależności od generacji.

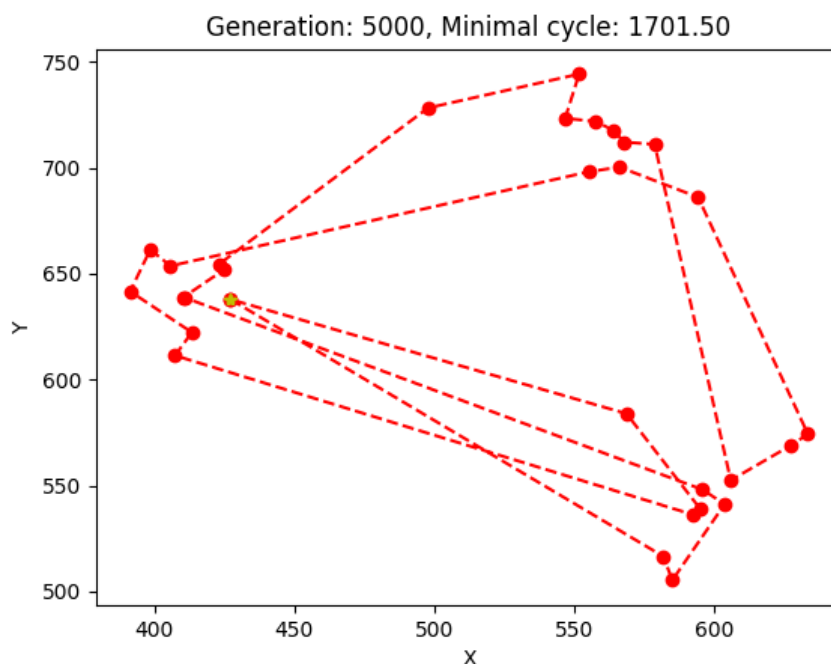


Rysunek 6. Najkrótszy cykl każdej generacji.

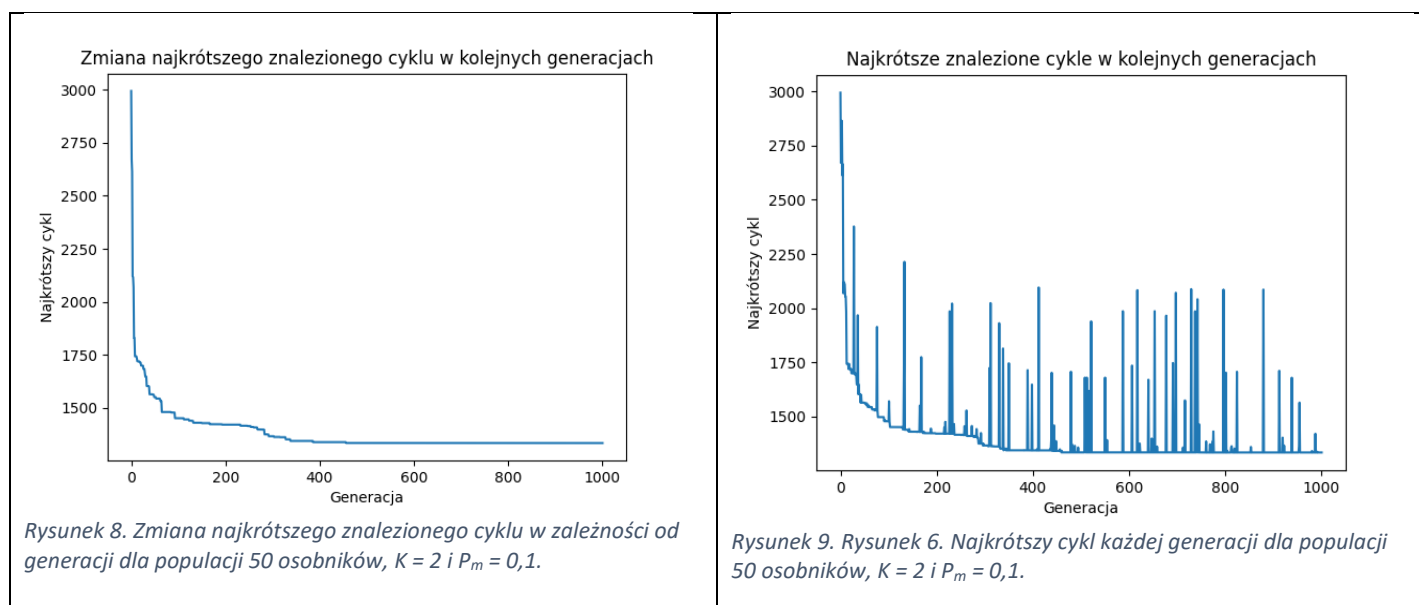
Kolejnym badanym rozkładem miast były duże skupiska. Generacja współrzędnych miast odbyła się dzięki pomocniczej funkcji `make_blobs()`, która tworzyła „plamy” odzwierciedlając skupiska miast. Na podstawie danych zawartych w Tabeli 2., można wysnuć podobne wnioski co do zmiany liczebności populacji bazowej i osobników losowanych do turnieju. Większa populacja bazowa pozwala na uzyskanie lepszych wyników, a zmiana parametru  $K$  nie wpłynęła na wyniki. Zwiększenie prawdopodobieństwa mutacji skutkowało znalezieniem krótszego cyklu niż w pozostałych przypadkach dla tej samej liczby skupisk. W obrębie jednego skupiska odległości między miastami są małe, ale odległości między skupiskami mogą być bardzo duże. Zakładając, że skupisko to punkt w przestrzeni, to aby je odwiedzić i wrócić wystarczą dwie długie drogi, czyli tam i z powrotem. Jeśli między dwoma skupiskami jest więcej takich długich dróg to z pomocą może przyjść mutacja, która może to zmienić. Im więcej jest skupisk tym większy jest najkrótszy znaleziony cykl, co wydaje się logiczne.

Tabela 2. Badanie wpływu różnych parametrów początkowych na działanie algorytmu genetycznego przy dużych skupiskach miast.

| Rozkład z dużymi skupiskami miast |                           |            |         |         |                     |        |   |                |                      |      |
|-----------------------------------|---------------------------|------------|---------|---------|---------------------|--------|---|----------------|----------------------|------|
| Calls number                      | Generation <sub>max</sub> | Population | Centres | Best    | Best <sub>avg</sub> | Std    | K | P <sub>m</sub> | t <sub>avg</sub> [s] | Seed |
| 10                                | 200                       | 10         | 2       | 958,52  | 1309,95             | 211,41 | 2 | 0,1            | 2,93                 | 1-10 |
| 10                                | 200                       | 50         | 2       | 908,46  | 1301,39             | 221,03 | 2 | 0,1            | 13,12                | 1-10 |
| 10                                | 200                       | 10         | 3       | 1626,75 | 1973,52             | 239,79 | 2 | 0,1            | 3,09                 | 1-10 |
| 10                                | 200                       | 50         | 3       | 1282,95 | 1678,02             | 200,31 | 2 | 0,1            | 13,28                | 1-10 |
| 10                                | 200                       | 50         | 3       | 1282,95 | 1678,02             | 200,31 | 3 | 0,1            | 13,26                | 1-10 |
| 10                                | 200                       | 50         | 3       | 1228,32 | 1428,10             | 156,33 | 2 | 0,5            | 12,97                | 1-10 |
| 10                                | 200                       | 50         | 3       | 1443,51 | 1763,63             | 199,49 | 2 | 0,05           | 13,21                | 1-10 |
| 10                                | 200                       | 10         | 4       | 2501,02 | 2906,23             | 273,97 | 2 | 0,1            | 2,85                 | 1-10 |
| 10                                | 200                       | 50         | 4       | 2078,72 | 2318,95             | 136,98 | 2 | 0,1            | 12,39                | 1-10 |
| 1                                 | 1000                      | 50         | 3       | 1701,50 | -                   | -      | 2 | 0,1            | 54,80                | 1    |
| 1                                 | 5000                      | 50         | 3       | 1701,50 | -                   | -      | 2 | 0,1            | 309,99               | 1    |



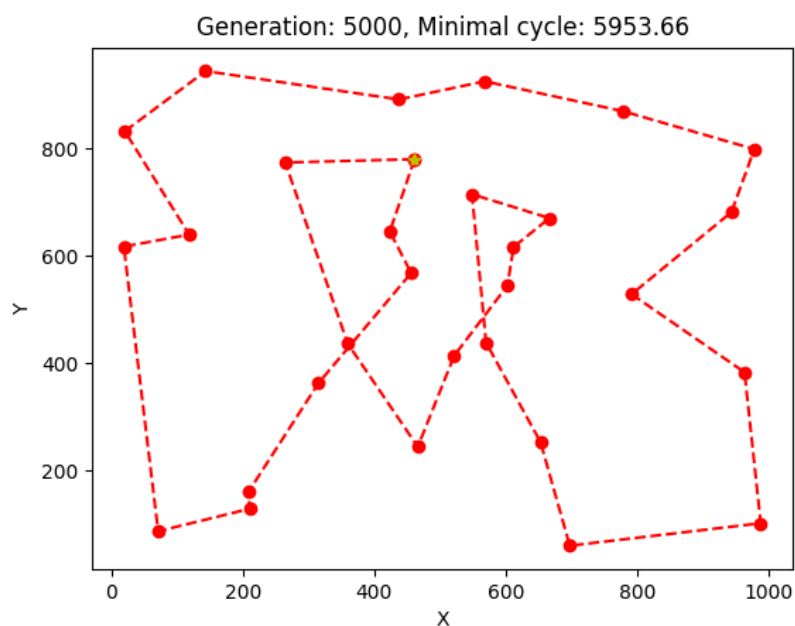
Rysunek 7. Najlepszy znaleziony cykl dla trzech dużych skupisk miast populacji 50,  $K = 2$ ,  $P_m = 0,1$  i  $Seed = 1$ .



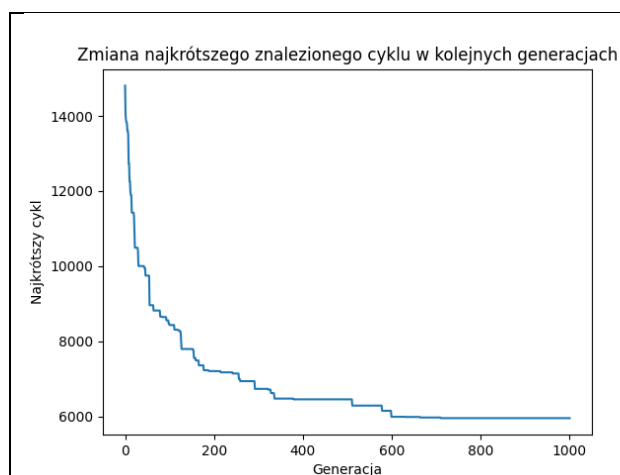
Ostatnim badanym rozkładem miast był rozkład losowy. Wyniki zostały przedstawione w Tabeli 3. W jego przypadku obserwacje i wnioski pokrywają się z tymi sformułowanymi dla poprzednich dwóch rozkładów tzn. te same parametry algorytmu tak samo wpływają na jakość uzyskanych rozwiązań.

Tabela 3. Badanie wpływu różnych parametrów początkowych na działanie algorytmu genetycznego przy losowym rozkładzie miast.

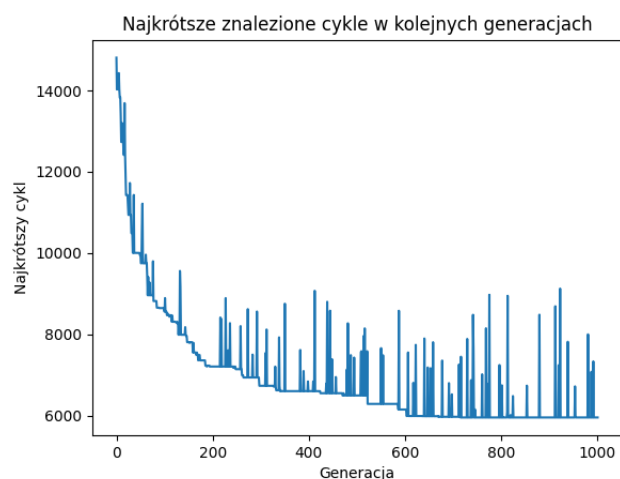
| Rozkład losowy miast |                           |            |         |                     |        |   |                |                      |      |
|----------------------|---------------------------|------------|---------|---------------------|--------|---|----------------|----------------------|------|
| Calls number         | Generation <sub>max</sub> | Population | Best    | Best <sub>avg</sub> | Std    | K | P <sub>m</sub> | t <sub>avg</sub> [s] | Seed |
| 10                   | 200                       | 10         | 8147,20 | 9076,63             | 634,95 | 2 | 0,1            | 2,62                 | 1-10 |
| 10                   | 200                       | 50         | 6927,21 | 7481,74             | 596,14 | 2 | 0,1            | 13,32                | 1-10 |
| 10                   | 200                       | 50         | 6927,21 | 7481,74             | 596,14 | 3 | 0,1            | 13,05                | 1-10 |
| 10                   | 200                       | 50         | 6926,21 | 7481,74             | 596,14 | 4 | 0,1            | 12,78                | 1-10 |
| 10                   | 200                       | 50         | 6773,14 | 7307,10             | 273,97 | 2 | 0,5            | 12,20                | 1-10 |
| 10                   | 200                       | 50         | 7493,83 | 8183,16             | 391,96 | 2 | 0,05           | 12,82                | 1-10 |
| 1                    | 1000                      | 50         | 5953,66 | -                   | -      | 2 | 0,1            | 67,06                | 1    |
| 1                    | 5000                      | 50         | 5953,66 | -                   | -      | 2 | 0,1            | 284,39               | 1    |



Rysunek 10. Najlepszy znaleziony cykl dla losowego rozkładu miast populacji 50,  $K = 2$ ,  $P_m = 0,1$  i  $Seed = 1$ .



Rysunek 11. Zmiana najkrótszego znanego cyklu w zależności od generacji dla populacji 50 osobników,  $K = 2$  i  $P_m = 0,1$ .



Rysunek 12. Najkrótszy cykl każdej generacji dla populacji 50 osobników,  $K = 2$  i  $P_m = 0,1$ .

## Podsumowanie

Dla problemu komiwojażera trudno jest znaleźć optymalne rozwiązanie przy ograniczonym budżecie. W zależności od rozkładu miast należy odpowiednio dobrać parametry takie jak liczebność populacji bazowej jak również prawdopodobieństwo mutacji. Pewne jest, że można ulepszyć zaimplementowany algorytm genetyczny na przykład poprzez wprowadzenie krzyżowania lub elity wśród osobników, ale nie jest to tematem ćwiczenia.