

Politechnika Koszalińska

Wydział Elektroniki i Informatyki

Sprawozdanie do projektu z przedmiotu:
„Zastosowania Sztucznej Inteligencji”

Semestr V, Studia Dienne

Kierunek: Informatyka

Rok akademicki 2020/2021

Temat projektu: Rozwiązanie problemu komiwojażera na przykładzie wyznaczania suboptymalnej trasy między miastami wojewódzkimi w Polsce za pomocą algorytmu genetycznego

Projekt wykonali:

Andrzej Krzyżanowski (nr indeksu U-14098)

Kamil Karmazyn (nr indeksu U-14686)

Data: 19.01.2021 r.

1.Wprowadzenie:

Projekt skupia się na problemie komiwojażera, konkretnie na znalezieniu suboptymalnej trasy między miastami wojewódzkimi w Polsce. W projekcie wykorzystywany jest algorytm genetyczny.

Założenia programu:

- Liczba pokoleń: 300
- Wielkość populacji: 1000
- Współczynnik mutacji: 0.7
- Typ krzyżowania: Krzyżowanie po połowie obojga rodziców
- Współczynnik elitarności: 10%

2.Opis rozwiązanego problemu:

Problem komiwojażera jest problemem znalezienia trasy o minimalnej długości dla odwiedzenia określonej ilości punktów przynajmniej raz, inaczej mówiąc polega na poszukiwaniu w grafie cyklu Hamiltona o najmniejszej wadze. Sam problem ma złożoność obliczeniową $(n-1)!/2$.

Czym jest cykl Hamiltona?

Cykl który odwiedza każdy wierzchołek grafu dokładnie jeden raz poza pierwszym wierzchołkiem w którym się kończy i zaczyna.

3.Opis techniki sztucznej inteligencji użytej w tworzonym projekcie:

Czym jest Algorytm Genetyczny?

Jest jedną z ewolucyjnych metod optymalizacji. Zalicza się go do klasy algorytmów heurystycznych. Polega na przeszukiwaniu możliwych rozwiązań w celu znalezienia rozwiązania najlepszego lub potencjalnie najlepszego, odbywa się za pomocą mechanizmów ewolucji oraz doboru naturalnego.

Schemat działania algorytmów genetycznych:

1.Losowanie populacji - inaczej inicjacja - Utworzenie populacji początkowej, polega na losowym wyborze określonej liczby chromosomów reprezentowanych przez ciągi cech/genów o określonej długości

Dzieli się na kolejne etapy:

a) Ustalenie podstawowych parametrów algorytmu genetycznego:

- długość chromosomu
- liczebność populacji
- Prawdopodobieństwo doboru chromosomu
- Prawdopodobieństwo krzyżowania
- Prawdopodobieństwo zajścia mutacji

b) Zasady kodowania zmiennych optymalizacji:

b1)Zbiór elementów ze zbioru rozwiązań dopuszczalnych musi być skończony.

b2)Wybrana metoda musi umożliwiać identyfikację poszczególnych elementów zbioru rozwiązań dopuszczalnych

b3)Metoda powinna eliminować możliwość wystąpienia w zbiorze rozwiązań niedopuszczalnych.

c)Wyznaczenie populacji początkowej:

Przez losowanie generowana jest populacja początkowa o określonej ilości chromosomów, chromosomy łączone są w sposób losowy.

2.Ocena osobników –polega na wybraniu odpowiednich chromosomów do bycia chromosomami rodzicielskim, czyli do stworzenia następnej generacji. Wybór ten odbywa się zgodnie z zasadą naturalnej selekcji czyli największe szanse na udział w tworzeniu nowych osobników mają te najlepiej przystosowane względem danego problemu.

Działanie oceny osobników:

Każdy osobnik w populacji jest sprawdzany pod względem konkretnych cech wymaganych dla znalezienia optymalnego rozwiązania danego problemu.

3.Selekcja – Celem selekcji jest wybór spośród chromosomów populacji poprzedniej, chromosomów które stworzą pary rodzicielskie.

Selekcja chromosomów jest losowa, a na wybór do pary rodzicielskiej ma wpływ posiadany przez chromosom pożądany zestaw cech.

Dlatego wykorzystywane są różne metody selekcji:

A.Koło ruletki - Chromosomowi przydzielany jest wycinek koła ruletki o wielkości odpowiedniej do wartości jego przystosowania. Dlatego im lepiej jest przystosowany tym większy fragment koła ruletki zajmuje. Całe koło ruletki odpowiada sumie wartości przystosowania wszystkich chromosomów danej populacji. W tym typie selekcji wygrywa chromosom należący do wylosowanego w ten sposób fragment koła ruletki.

B.Ranking – Metoda ta polega na rozłożeniu chromosomów w rankingu od najgorszego zajmującego 1 miejsce do najlepszego który zajmuje ostatnie miejsce w populacji. Do dalszego rozrodu przechodzi tylko n najlepiej przystosowanych osobników. Metoda ta charakteryzuje się lepszym uwarunkowaniem niż metoda ruletki.

C.Elitarna – Zasada elitaryzmu dotyczy wyboru najlepszych z spośród chromosomów populacji poprzedniej i skopiowaniu ich do nowej populacji. Dla zapewnienia coraz lepszego przystosowania kolejnych populacji, zakłada się że kluczowy wpływ powinny mieć najlepsze chromosomy z poprzedniej populacji. Zgodnie z tym założeniem chromosomy słabo przystosowane nie powinny trafić do nowo tworzonej populacji.

4. Krzyżowanie - Celem krzyżowania jest przekazanie cech poszczególnych par chromosomów rodzicielskich nowo utworzonym na chromosomom potomstwa. Operacja krzyżowania polega na wymianie, za pomocą współczynnika krzyżowania, genów pomiędzy chromosomami pochodzącymi od poszczególnych par rodzicielskich. W efekcie realizacji operacji krzyżowania zostają utworzone chromosomy potomków, które są połączeniem par chromosomów rodzicielskich.
Dokładny opis działania naszego krzyżowania:

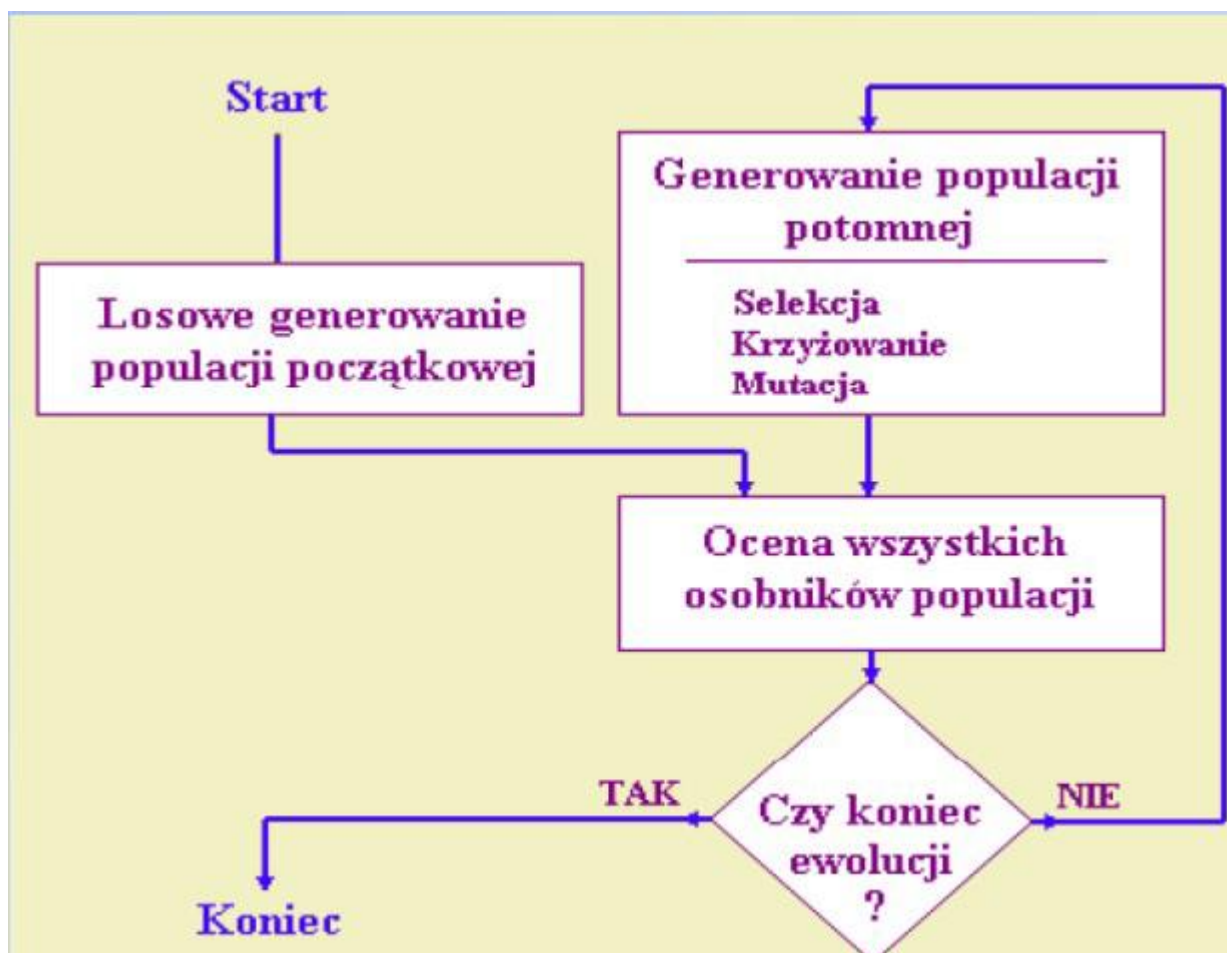
- 1/ Z puli rodziców wybieramy dwa genotypy
- 2/ Dzielimy pierwszy na pół
- 3/ Połowa punktów (genów) rodzica pierwszego stanowi pierwszą część punktów (genów) potomka
- 4/ Drugą połowę dobieramy w pętli z drugiego rodzica przestrzegając dwóch warunków:
 - punkt nie może być ani punktem startowym, ani końcowym,
 - punkt nie może już być obecny w genotypie.

5. Mutacja – jest ostatnim etapem tworzenia chromosomów nowej populacji i dotyczy zmiany genów chromosomu potomka. Zastosowanie mutacji umożliwia zaliczenie do elementów nowej populacji także tych chromosomów, które z założenia są praktycznie niemożliwe do uzyskania w wyniku samego krzyżowania.

Działanie mutacji:

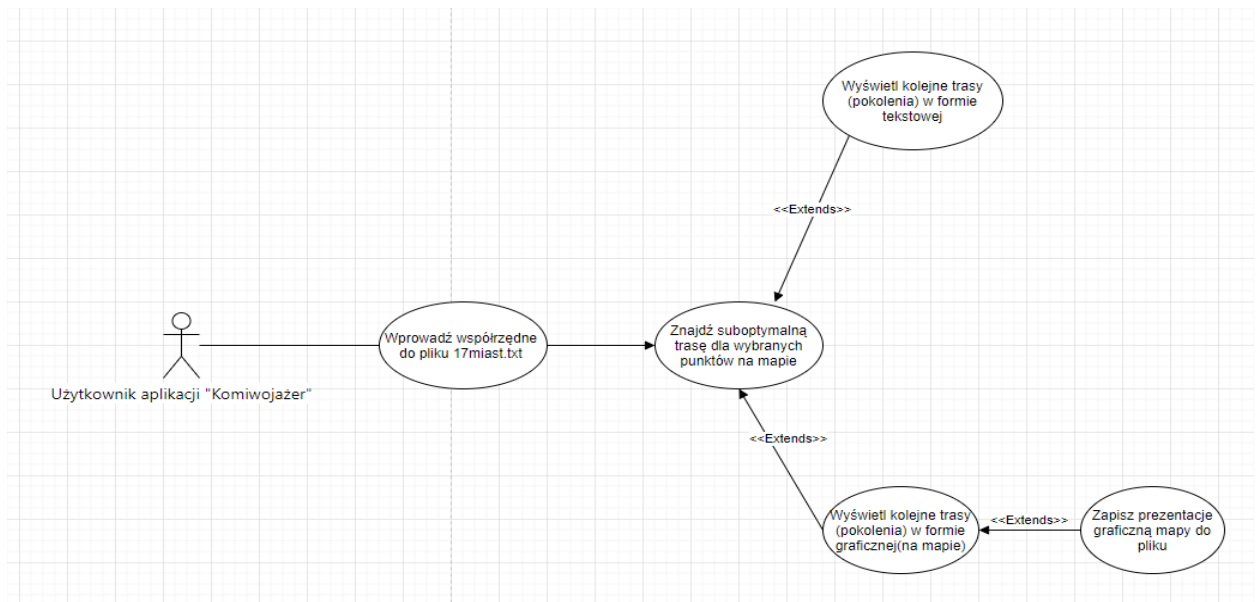
Dla każdego osobnika potomnego występuje losowanie z określoną szansą (współczynnik mutacji) na pojawienie się cech które nie wystąpiły u osobników potomnych

6. Powrót do punktu 2, tak długo póki nie zostanie ustalona suboptymalna droga.

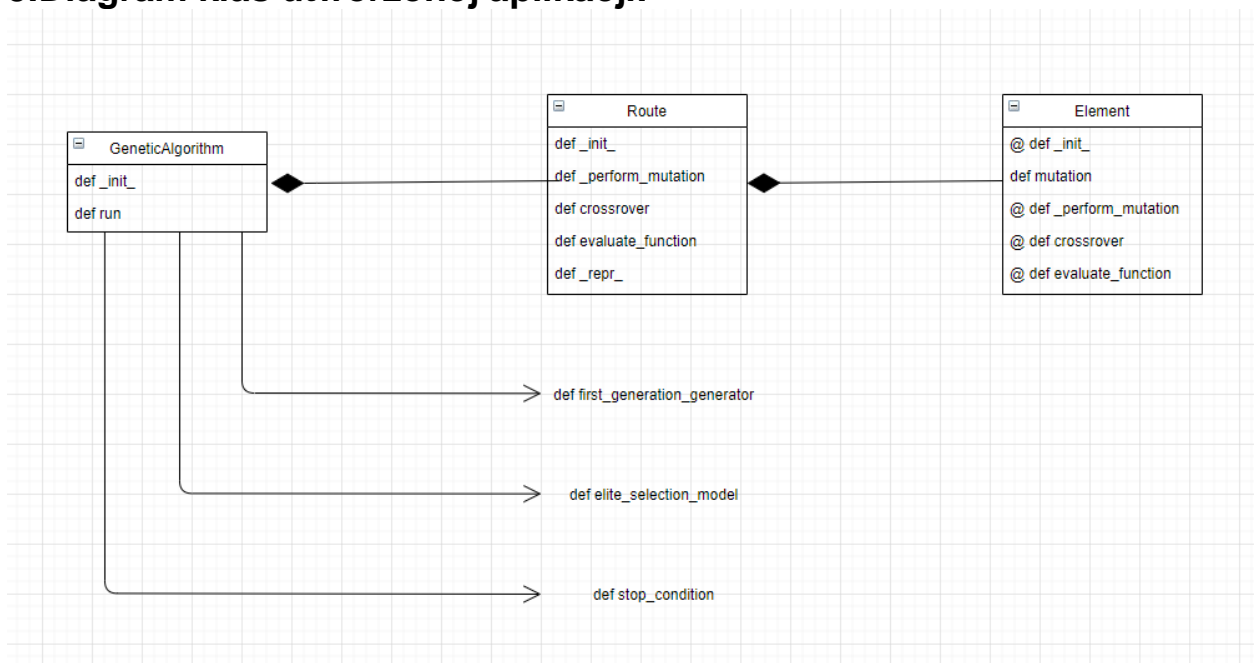


Rys. 1 - Schemat działania algorytmu genetycznego

4. Diagram przypadków użycia utworzonej aplikacji



5. Diagram klas utworzonej aplikacji:



6. Opis utworzonej aplikacji – dokumentacja użytkownika:

Użytkownik ma możliwość podania własnych współrzędnych poprzez edycję pliku 16miast.txt.

Po wprowadzeniu współrzędnych i uruchomieniu programu użytkownik ma możliwość obserwacji zmian zachodzących między kolejnymi pokoleniami zarówno na mapie w formie graficznej, w konsoli w formie tekstowej jako wyliczenie odległości w mierze punktowej.

Dodatkową funkcjonalnością jest możliwość obserwacji dynamicznego wykresu przedstawiającego postęp optymalizacji wyszukiwania trasy w czasie.

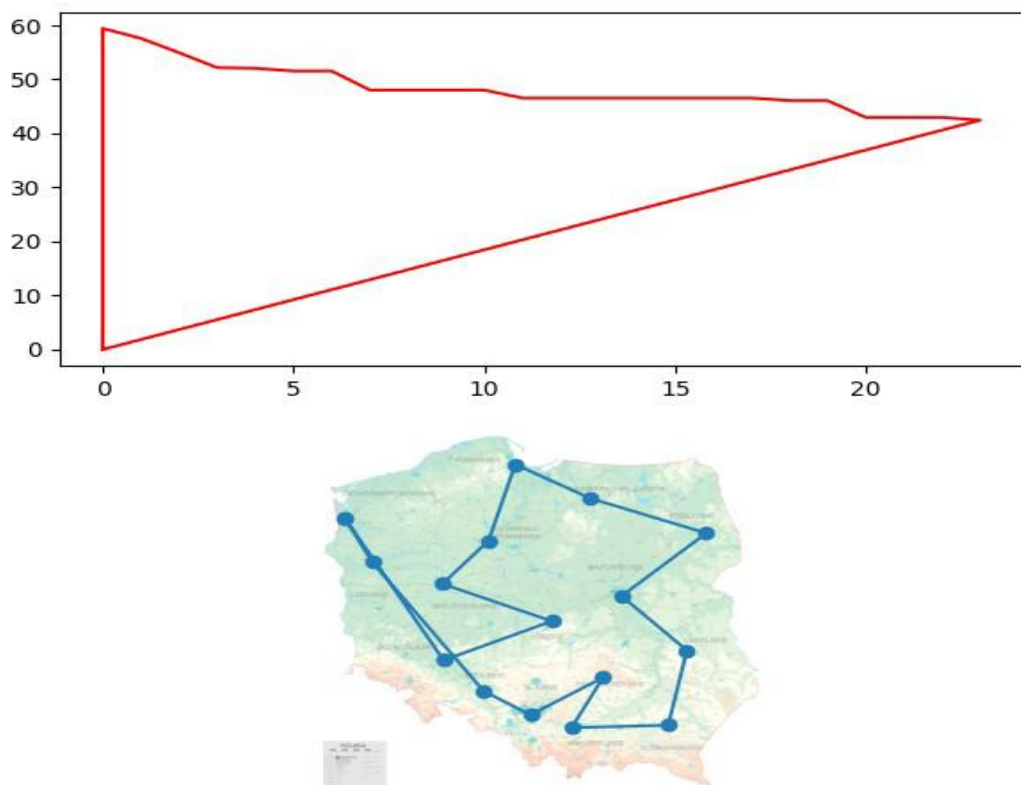
7.Podsumowanie:

W chwili obecnej można znaleźć suboptymalną drogę dla 16 miast wojewódzkich bądź podać współrzędne wybranych miast poprzez edycję pliku 16miast.txt.

Możliwy dalszy rozwój programu:

-ustawianie parametrów algorytmu przez użytkownika (liczby pokoleń,wielkości populacji, współczynnika mutacji, współczynnika krzyżowania)

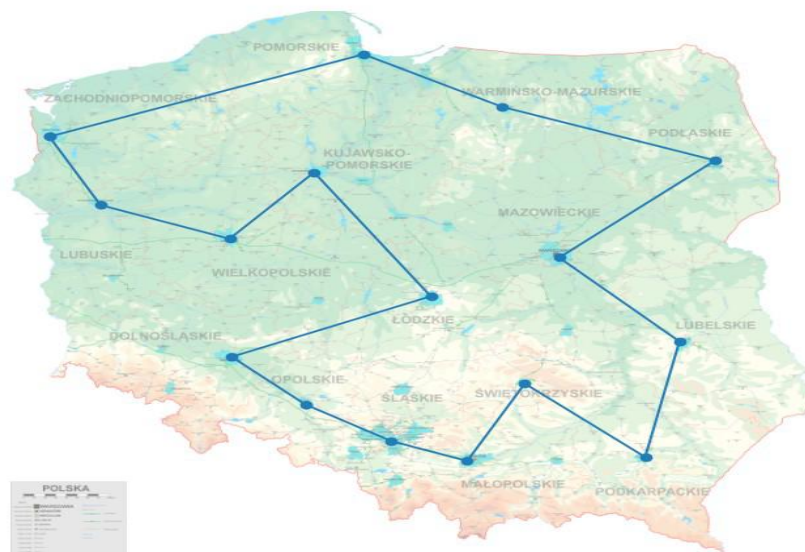
Kilka screenów z działania aplikacji:



Rys.2 - Prezentacja działania algorytmu w czasie

```
C:\WINDOWS\py.exe
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 459 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 460 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 461 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 462 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 463 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 464 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 465 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 466 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 467 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 468 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 469 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 470 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 471 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
Generation: 472 S: [(10, 10), (10, 20), (10, 30), (20, 10), (30, 10), (40, 10), (50, 10), (50, 20), (50, 30), (50, 40),
(50, 50), (40, 50), (30, 50), (20, 50), (10, 50), (10, 40), (10, 10)] fitness: 192.3606797749979
```

Rys.3 - Wyliczenie suboptymalnej trasy w odległości punktowej



Rys.4 - Potencjalnie optymalna trasa dla 16 miast wojewódzkich w Polsce

Screeny z aplikacji testującej:

Dla pierwszych danych testowych:

Wielkość populacji	Współczynnik mutacji	Liczba pokoleń	Procent elity												
500	0,0625	500	0,1												
Numer testu	1	2	3	4	5	6	7	8	9	10	suma	średnia	min	max	
Wartość początkowa	319	362	338	360	349	284	365	364	378	345	3464	346,4	284	378	
Wartość końcowa	160	240	160	224	233	222	224	160	258	220	2101	210,1	160	258	
Ostatnie ewoluujące pokolenie	246	69	115	129	65	207	172	42	85	24					

Dla drugich danych testowych:

Wielkość populacji	Współczynnik mutacji	Liczba pokoleń	Procent elity												
1000	0,4	500	0,1												
Numer testu	1	2	3	4	5	6	7	8	9	10	suma	średnia	min	max	
Wartość początkowa	359	324	371	345	275	331	365	346	359	355	3430	343	275	371	
Wartość końcowa	160	192	174	174	174	160	160	174	160	174	1702	170,2	160	192	
Ostatnie ewoluujące pokolenie	115	141	211	85	57	108	51	59	65	73					

Dla kolejnych danych testowych:

Wielkość populacji	Współczynnik mutacji	Liczba pokoleń	Procent elity												
1000	0,7	300	0,1												
Numer testu	1	2	3	4	5	6	7	8	9	10	suma	średnia	min	max	
Wartość początkowa	323	337	339	351	355	329	343	326	320	370	3393	339,3	320	370	
Wartość końcowa	174	174	174	174	174	174	192	174	160	160	1730	173	160	192	
Ostatnie ewoluujące pokolenie	103	101	86	64	71	112	103	58	104	84					

Screeny z testów aplikacji głównej:

Dla pierwszych danych testowych:

Wielkość populacji	Współczynnik mutacji	Liczba pokoleń	Procent elity												
1000	0,4	300	0,1												
Numer testu	1	2	3	4	5	6	7	8	9	10					
Wartość początkowa	52,07	60,66	54,51	59,01	58,52	58,09	58,34	58,93	54,67	55,25	570,05				
Wartość końcowa	41,06	36,4	35,3	36,4	36,45	36,09	35,86	35,86	40,26	36,59	370,27				
Ostatnie ewoluujące pokolenie	164	49	118	77	28	163	113	161	91	73					

Dla kolejnych danych tekstowych:

Wielkość populacji	Współczynnik mutacji	Liczba pokoleń	Procent elity												
1000	0,7	300	0,1												
Numer testu	1	2	3	4	5	6	7	8	9	10					
Wartość początkowa	53,64	51,86	58,14	53,65	57,54	57,79	58,69	54,42	55,93	56,5	558,16				
Wartość końcowa	36,09	36,62	36,59	36,09	36,09	35,3	35,86	35,86	36,09	35,3	359,89				
Ostatnie ewoluujące pokolenie	47	43	276	209	73	99	264	100	53	248					

Bibliografia:

Zbigniew Michalewicz: „Algorytmy genetyczne + struktury danych = programy

ewolucyjne”, Wydawnictwa Naukowo-Techniczne, Warszawa (2003), ISBN: 8320428815

[http://algorytmy.ency.pl/tutorial/problem_komiwojazera_algorytm_genetyczny?](http://algorytmy.ency.pl/tutorial/problem_komiwojazera_algorytm_genetyczny?fbclid=IwAR048NYMG00J7JpLEdj9Rm0woG3CcLRzmx-A61nA9Y5Wm7dgXP-AfBNW00)

[fbclid=IwAR048NYMG00J7JpLEdj9Rm0woG3CcLRzmx-A61nA9Y5Wm7dgXP-AfBNW00](http://www.alife.pl/gp/p/AGelem.html)

<http://www.alife.pl/gp/p/AGelem.html>

https://sound.eti.pg.gda.pl/student/isd/isd03-algorytmy_genetyczne.pdf

<https://blog.jhossa.net/2016/04/02/selekcja-w-algorytmach-genetycznych/>

https://sound.eti.pg.gda.pl/student/isd/isd03-algorytmy_genetyczne.pdf

https://eduinf.waw.pl/inf/alg/001_search/0136.php

http://algorytmy.ency.pl/artykul/problem_komiwojazera