



Dane przykładowe zawierają 66 miast, więc macierz odległości to 66x66

WSTEP

Liczba możliwości połączenia miast jest ogromna, niemożliwa do obliczenia standardowymi metodami

Należy zastosowac algorytm optymalizacyjny

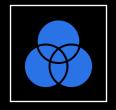
GŁÓWNĘ ZAŁOŻENIA



Program w PyQt, zapewniający interaktywność z użytkownikiem



Możliwość ustawienia parametrów dotyczących: liczby miast do zaplanowania wycieczki, maksymalny dystans trasy, miasto początkowe, liczbę potencjalnych tras, liczbę przeprowadzonych iteracji



Liczba przeprowadzonych iteracji wpływa na stopień optymalności rozwiązania, im więcej tym lepsze rozwiązania, jednakże wpływa to na czas obliczeń, należy odszukać optymalną wartość domyślną



Potencjalne trasy są posortowane od najlepszego do najgorszego wynikowego rozwiązania, poprzez porównanie stosunku city_weight do route_length WYBÓR ALGORYTMU Wybrano przeszukiwanie losowe połączone z przeszukiwaniem wyczerpującym

Umożliwia to uzyskać semioptymalne trasy

INTERFEJS

■ Optimal turistical route	×
Threshold Distance:	
1500	
Number of Cities:	
5	
Number of Iterations:	
1000	
Number of Solutions:	
5	
Start City:	
Biała Podlaska	V
Run	
Solution 1 Optimal Route: [Biała Podlaska', 'Kielce', 'Tarnobrzeg', 'Tychy', 'Legnica', 'Biała Podlaska'] Sum of Distances: 1464.0 Maximum City Weight: 60.795918367346935	
Solution 2 Optimal Route: ['Biała Podlaska', 'Jaworzno', 'Opole', 'Piotrków Trybunalski', 'Elbląg', 'Biała Podlaska' Sum of Distances: 1426.0 Maximum City Weight: 59.043083900226755]
Solution 3 Optimal Route: ['Biała Podlaska', 'Tarnobrzeg', 'Legnica', 'Tychy', 'Kielce', 'Biała Podlaska'] Sum of Distances: 1482.0 Maximum City Weight: 60.795918367346935	
Solution 4 Optimal Route: [Biała Podlaska', 'Kielce', 'Tychy', 'Legnica', 'Tarnobrzeg', 'Biała Podlaska'] Sum of Distances: 1482.0 Maximum City Weight: 60.795918367346935	
Solution 5 Optimal Route: ['Biała Podlaska', 'Elbląg', 'Piotrków Trybunalski', 'Jaworzno', 'Opole', 'Biała Podlaska' Sum of Distances: 1481.0 Maximum City Weight: 59.04308390022676]

OPTYMALNE PARAMETRY

- Metodami prób i błędów udało się ustalić liczbę 1000 iteracji jako optymalną wartość, równoważącą czas oczekiwania do jakości wyników.
- Ważnym aspektem tego algorytmu jest przemyślana struktura. Najpierw następuje losowe wyszukanie miast a następnie permutacje tych wybranych. Czasami wystarczy zmienić kolejność jednego miasta na trasie z innym aby znacząco zredukować jej długość nie tracąc przy tym na wadze miast.



import sys
from PyQt5.QtWidgets import *
from PyQt5.QtGui import QIntValidator
import pandas as pd
import random
import itertools