

Raport

Alesia Filinkova

336180

1 Treść ćwiczenia

Celem ćwiczenia było zaimplementowanie algorytmu ewolucyjnego do rozwiązania problemu komiwojażera z użyciem danych polskich miast dostępnych w pliku `cities.csv`. Algorytm wykorzystuje mechanizmy mutacji, selekcji ruletkowej, krzyżowania oraz sukcesji generacyjnej. Wynikiem działania algorytmu jest trasa o minimalnym dystansie, obejmująca wszystkie miasta dokładnie raz, zaczynając i kończąc w ustalonych punktach.

2 Cel i opis eksperymentów

Głównym celem eksperymentów było badanie wpływu różnych hiperparametrów algorytmu na jakość optymalizacji oraz czas zbieżności do dobrego rozwiązania. Badane hiperparametry to:

- Wielkość populacji,
- Ilość generacji

2.1 Zbiór danych

Do testów użyto pliku `cities.csv`, zawierającego macierz odległości pomiędzy wybranymi polskimi miastami. Trasa wyznaczona przez algorytm zaczynała się i kończyła w ustalonych miastach, których wybór był niezmienny przez wszystkie testy. Zaczynam zawsze od 'Łomża' i kończę w 'Częstochowa'.

2.2 Metryka oceny jakości

Ocena jakości trasy odbywała się na podstawie łącznej długości przebytej trasy. Krótsza trasa oznacza lepsze rozwiązanie. Jakość wyników algorytmu oceniano na podstawie średniej długości trasy oraz odchylenia standardowego z zestawu losowych uruchomień.

2.3 Kroki eksperymentu

1. Uruchomić skrypt implementujący algorytm ewolucyjny.
2. Wybrać odpowiednie hiperparametry do testów.
3. Powtórzyć eksperyment dla różnych wartości wybranego hiperparametru (Wielkości populacji i/lub Ilości generacji), notując średnia długość trasy i odchylenie standardowe.
4. Zapisać najlepszą trasę oraz uzyskane metryki.

3 Instrukcja odtworzenia wyników

3.1 Przygotowanie środowiska i Uruchomienie:

Skrypt można uruchomić przez terminal za pomocą polecenia:

1. `git clone https://gitlab-stud.elka.pw.edu.pl/afilinko/wsi.git`
2. `python3 -m venv venv`
3. `source venv/bin/activate`
4. `cd /lab2`
5. `pip install -r requirements.txt`
6. `python3 main.py --cities-path data/cities.csv --start <nazwa-miasta> --finish <nazwa-miasta> --problem-size <problem-size> --seed 42`

4 Wyniki

Poniżej przedstawiono wyniki eksperymentów w formie tabelarycznej oraz wykresów.

4.1 Wielkość populacji i Wizualizacja wyników dla 5 miast

W przypadku, gdy wybieramy mało miejsc, średnie znaczenie równa się długości trasy (w tym przypadku 1128.577), a odchylenie średnie równa się 0. Najlepsza lista miast: ['Łomża', 'Zamość', 'Kielce', 'Wrocław', 'Częstochowa']

Wielkość populacji	Ilość generacji	Długość trasy
50	200	1128.57
100	200	1128.577
10	10	1128.577

Table 1: Wpływ wielkości populacji i ilości generacji na jakość rozwiązania dla 5 miast

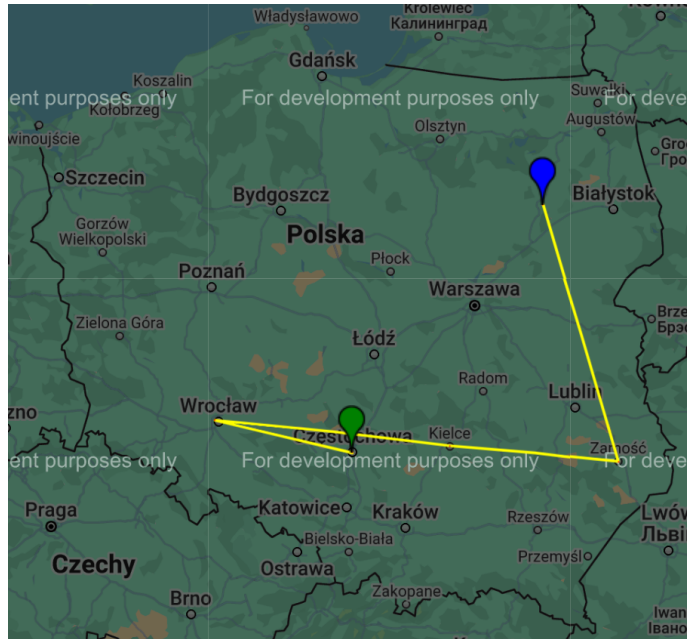


Figure 1: Najlepsza trasa znaleziona przez algorytm dla 5 miast

4.2 Wielkość populacji i Wizualizacja wyników dla 25 miast

Średnia wartość: 5324.684; Odchylenie standardowe: 1240.115; Najlepsza wielkość populacji: 200; Najlepsza ilość generacji: 300; Najlepsza długość trasy: 4504.819; Najlepsza lista miast: ['Łomża', 'Ostrołęka', 'Chełm', 'Zamość', 'Radom', 'Włocławek', 'Bydgoszcz', 'Gdańsk', 'Gorzów Wielkopolski', 'Koszalin', 'Toruń', 'Piła', 'Wrocław', 'Sieradz', 'Piotrków Trybunalski', 'Kraków', 'Zakopane', 'Krosno', 'Kielce', 'Tarnów', 'Katowice', 'Opole', 'Cieszyn', 'Leszno', 'Czestochowa']. W tym przypadku znacznie już różni się kolejność miast, dlatego różna się długości.

Wielkość populacji	Ilość generacji	Długość trasy	lista miast
50	200	6985.049	['Łomża', 'Toruń', 'Zamość', 'Ostrołęka', 'Cieszyn', 'Opole', 'Chełm', 'Kielce', 'Kraków', 'Zakopane', 'Tarnów', 'Gorzów Wielkopolski', 'Piotrków Trybunalski', 'Krosno', 'Radom', 'Gdańsk', 'Koszalin', 'Leszno', 'Wrocław', 'Katowice', 'Sieradz', 'Włocławek', 'Piła', 'Bydgoszcz', 'Częstochowa']
100	200	5192.11	['Łomża', 'Ostrołęka', 'Zamość', 'Chełm', 'Tarnów', 'Krosno', 'Kielce', 'Kraków', 'Opole', 'Piotrków Trybunalski', 'Cieszyn', 'Zakopane', 'Wrocław', 'Sieradz', 'Piła', 'Toruń', 'Leszno', 'Koszalin', 'Gorzów Wielkopolski', 'Włocławek', 'Gdańsk', 'Bydgoszcz', 'Katowice', 'Radom', 'Częstochowa']
150	300	4873.309	['Łomża', 'Ostrołęka', 'Toruń', 'Koszalin', 'Gorzów Wielkopolski', 'Piła', 'Bydgoszcz', 'Włocławek', 'Gdańsk', 'Leszno', 'Piotrków Trybunalski', 'Chełm', 'Zakopane', 'Cieszyn', 'Wrocław', 'Sieradz', 'Kielce', 'Radom', 'Katowice', 'Tarnów', 'Krosno', 'Zamość', 'Kraków', 'Opole', 'Częstochowa']
200	300	4504.819	['Łomża', 'Ostrołęka', 'Chełm', 'Zamość', 'Radom', 'Włocławek', 'Bydgoszcz', 'Gdańsk', 'Gorzów Wielkopolski', 'Koszalin', 'Toruń', 'Piła', 'Wrocław', 'Sieradz', 'Piotrków Trybunalski', 'Kraków', 'Zakopane', 'Krosno', 'Kielce', 'Tarnów', 'Katowice', 'Opole', 'Cieszyn', 'Leszno', 'Częstochowa']
300	300	5068.133	['Łomża', 'Ostrołęka', 'Zamość', 'Krosno', 'Tarnów', 'Chełm', 'Gdańsk', 'Koszalin', 'Wrocław', 'Gorzów Wielkopolski', 'Bydgoszcz', 'Piła', 'Leszno', 'Włocławek', 'Toruń', 'Kielce', 'Piotrków Trybunalski', 'Opole', 'Cieszyn', 'Katowice', 'Radom', 'Zakopane', 'Kraków', 'Sieradz', 'Częstochowa']

Table 2: Wpływ wielkości populacji i ilości generacji na jakość rozwiązania dla 25 miast

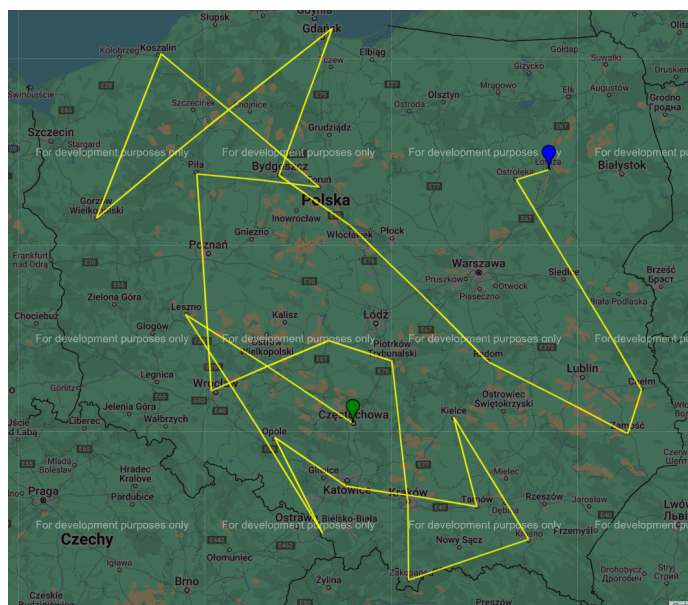


Figure 2: Najlepsza trasa znaleziona przez algorytm dla 25 miast

4.3 Wielkość populacji i Wizualizacja wyników dla wszystkich miast

Średnia wartość: 15768.311; Odchylenie standardowe: 1317.516; Najlepsza wielkość populacji: 100; Najlepsza ilość generacji: 100; Najlepsza długość trasy: 14633.564; Najlepsza lista miast: ['Łomża', 'Zakopane', 'Cieszyn', 'Suwałki', 'Ciechanów', 'Olsztyn', 'Tarnobrzeg', 'Chełm', 'Legnica', 'Sieradz', 'Bielsko - Biała', 'Skierniewice', 'Piotrków Trybunalski', 'Włocławek', 'Wrocław', 'Tarnów', 'Kielce', 'Łódź', 'Płock', 'Bydgoszcz', 'Zielona Góra', 'Gorzów Wielkopolski', 'Jelenia Góra', 'Poznań', 'Piła', 'Elbląg', 'Biała Podlaska', 'Gdańsk', 'Konin', 'Koszalin', 'Ślusk', 'Opole', 'Siedlce', 'Wałbrzych', 'Katowice', 'Szczecin', 'Toruń', 'Kraków', 'Zamość', 'Krosno', 'Przemyśl', 'Radom', 'Kalisz', 'Leszno', 'Rzeszów', 'Nowy Sacz', 'Ostrołęka', 'Lublin', 'Białystok', 'Warszawa', 'Częstochowa']. W tym przypadku można zobaczyć, że kolejność miast różni się już od drugiego miasta, dużo różni się długości tras.

Wielkość populacji	Ilość generacji	Długość trasy	lista miast
200	300	15149.139	[‘Łomża’, ‘Białystok’, ‘Lublin’, ‘Jelenia Góra’, ‘Legnica’, ‘Suwałki’, ‘Koszalin’, ‘Ślupsk’, ‘Gdańsk’, ‘Poznań’, ‘Tarnobrzeg’, ‘Siedlce’, ‘Szczecin’, ‘Bydgoszcz’, ‘Skierniewice’, ‘Kielce’, ‘Zamość’, ‘Warszawa’, ‘Wrocław’, ‘Rzeszów’, ‘Cieszyn’, ‘Tarnów’, ‘Przemyśl’, ‘Kraków’, ‘Katowice’, ‘Biała Podlaska’, ‘Wałbrzych’, ‘Konin’, ‘Olsztyn’, ‘Krosno’, ‘Bielsko - Biała’, ‘Ciechanów’, ‘Leszno’, ‘Kalisz’, ‘Płock’, ‘Sieradz’, ‘Piotrków Trybunalski’, ‘Chełm’, ‘Toruń’, ‘Łódź’, ‘Radom’, ‘Ostrołęka’, ‘Wrocław’, ‘Nowy Sacz’, ‘Opole’, ‘Piła’, ‘Zielona Góra’, ‘Gorzów Wielkopolski’, ‘Elbląg’, ‘Zakopane’, ‘Częstochowa’]
50	50	16676.483	[‘Łomża’, ‘Ostrołęka’, ‘Chełm’, ‘Toruń’, ‘Konin’, ‘Przemyśl’, ‘Olsztyn’, ‘Kraków’, ‘Wrocław’, ‘Koszalin’, ‘Ciechanów’, ‘Zielona Góra’, ‘Szczecin’, ‘Bielsko - Biała’, ‘Leszno’, ‘Lublin’, ‘Katowice’, ‘Wałbrzych’, ‘Sieradz’, ‘Opole’, ‘Piła’, ‘Kalisz’, ‘Tarnów’, ‘Zamość’, ‘Suwałki’, ‘Gdańsk’, ‘Bydgoszcz’, ‘Gorzów Wielkopolski’, ‘Ślupsk’, ‘Jelenia Góra’, ‘Łódź’, ‘Cieszyn’, ‘Płock’, ‘Rzeszów’, ‘Siedlce’, ‘Krosno’, ‘Biała Podlaska’, ‘Kielce’, ‘Skierniewice’, ‘Warszawa’, ‘Zakopane’, ‘Nowy Sacz’, ‘Legnica’, ‘Wrocław’, ‘Piotrków Trybunalski’, ‘Radom’, ‘Elbląg’, ‘Białystok’, ‘Poznań’, ‘Tarnobrzeg’, ‘Częstochowa’]
50	200	17268.596	[‘Łomża’, ‘Poznań’, ‘Wrocław’, ‘Chełm’, ‘Radom’, ‘Łódź’, ‘Tarnobrzeg’, ‘Ślupsk’, ‘Katowice’, ‘Bielsko - Biała’, ‘Szczecin’, ‘Białystok’, ‘Wałbrzych’, ‘Cieszyn’, ‘Sieradz’, ‘Zakopane’, ‘Kielce’, ‘Elbląg’, ‘Biała Podlaska’, ‘Olsztyn’, ‘Kraków’, ‘Tarnów’, ‘Nowy Sacz’, ‘Suwałki’, ‘Ostrołęka’, ‘Zielona Góra’, ‘Koszalin’, ‘Bydgoszcz’, ‘Kalisz’, ‘Toruń’, ‘Piotrków Trybunalski’, ‘Legnica’, ‘Skierniewice’, ‘Krosno’, ‘Opole’, ‘Konin’, ‘Przemyśl’, ‘Wrocław’, ‘Gdańsk’, ‘Rzeszów’, ‘Warszawa’, ‘Jelenia Góra’, ‘Siedlce’, ‘Lublin’, ‘Zamość’, ‘Gorzów Wielkopolski’, ‘Leszno’, ‘Piła’, ‘Ciechanów’, ‘Płock’, ‘Częstochowa’]
100	200	15113.772	[‘Łomża’, ‘Olsztyn’, ‘Płock’, ‘Elbląg’, ‘Skierniewice’, ‘Siedlce’, ‘Gdańsk’, ‘Konin’, ‘Piotrków Trybunalski’, ‘Łódź’, ‘Bydgoszcz’, ‘Poznań’, ‘Kielce’, ‘Ciechanów’, ‘Suwałki’, ‘Wrocław’, ‘Warszawa’, ‘Ślupsk’, ‘Wrocław’, ‘Sieradz’, ‘Koszalin’, ‘Leszno’, ‘Kalisz’, ‘Toruń’, ‘Bielsko - Biała’, ‘Zakopane’, ‘Przemyśl’, ‘Gorzów Wielkopolski’, ‘Katowice’, ‘Szczecin’, ‘Legnica’, ‘Zielona Góra’, ‘Chełm’, ‘Nowy Sacz’, ‘Lublin’, ‘Radom’, ‘Jelenia Góra’, ‘Wałbrzych’, ‘Opole’, ‘Białystok’, ‘Zamość’, ‘Kraków’, ‘Krosno’, ‘Tarnów’, ‘Piła’, ‘Ostrołęka’, ‘Tarnobrzeg’, ‘Rzeszów’, ‘Biała Podlaska’, ‘Cieszyn’, ‘Częstochowa’]
100	100	14633.564	[‘Łomża’, ‘Zakopane’, ‘Cieszyn’, ‘Suwałki’, ‘Ciechanów’, ‘Olsztyn’, ‘Tarnobrzeg’, ‘Chełm’, ‘Legnica’, ‘Sieradz’, ‘Bielsko - Biała’, ‘Skierniewice’, ‘Piotrków Trybunalski’, ‘Wrocław’, ‘Wrocław’, ‘Tarnów’, ‘Kielce’, ‘Łódź’, ‘Płock’, ‘Bydgoszcz’, ‘Zielona Góra’, ‘Gorzów Wielkopolski’, ‘Jelenia Góra’, ‘Poznań’, ‘Piła’, ‘Elbląg’, ‘Biała Podlaska’, ‘Gdańsk’, ‘Konin’, ‘Koszalin’, ‘Ślupsk’, ‘Opole’, ‘Siedlce’, ‘Wałbrzych’, ‘Katowice’, ‘Szczecin’, ‘Toruń’, ‘Kraków’, ‘Zamość’, ‘Krosno’, ‘Przemyśl’, ‘Radom’, ‘Kalisz’, ‘Leszno’, ‘Rzeszów’, ‘Nowy Sacz’, ‘Ostrołęka’, ‘Lublin’, ‘Białystok’, ‘Warszawa’, ‘Częstochowa’]

Table 3: Wpływ wielkości populacji i ilości generacji na jakość rozwiązania dla wszystkich miast

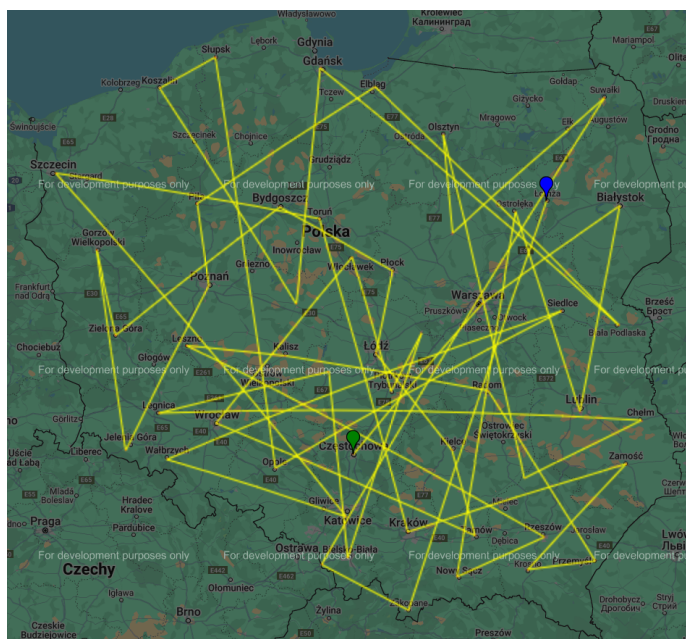


Figure 3: Enter Caption

Figure 4: Najlepsza trasa znaleziona przez algorytm dla wszystkich miast

5 Wnioski

Z przeprowadzonych eksperymentów wynika, że wielkość populacji ma znaczący wpływ na jakość wyników. Większa populacja nie zawsze przyczynia się do lepszej optymalizacji i z czasem przynosi malejące korzyści. Dla każdej ilości miast trzeba szukać najlepszej kombinacji wartości za pomocą eksperymentów. Algorytm ewolucyjny jest skutecznym narzędziem do rozwiązania problemu komiwojażera, ale osiąga dobre wyniki kosztem zwiększonej liczby iteracji.