Algorytmy Metaheurystyczne
Lista 2 - TSP Simulated Annealing and Tabu Search

Rafał Włodarczyk

12.05.2026

Spis treści

1 Model

Model TSP zakłada graf pełny $G = (V, E)$ z wagami.

2 Źródło dla danych

Dane do programu są wzięte ze strony internetowej:

`https://www.math.uwaterloo.ca/tsp/world/countries.html`

oraz są dostępne do pobrania w formacie `.tsp` za pomocą skryptu `get_data.sh`.

3 Zadanie 1 - Simulated Annealing

Symulowane wyżarzanie (*Simulated Annealing*) to algorytm metaheurystyczny inspirowany fizycznym procesem wyżarzania metali - startujemy z temperaturą T, którą następnie w każdym kroku (epoch) obniżamy o współczynnik α ($0 < \alpha < 1$) zgodnie ze wzorem

$$T_{new} = \alpha \cdot T_{old}.$$

Następnie w każdym kroku (n - liczba kroków) przeprowadzamy kilka prób (k - liczba prób w kroku) i wybieramy najlepszą z nich.

3.1 Wyznaczenie współczynników

Wyznaczenie współczynników T, α, n, k zostanie zgodnie z wymaganiami wykonane na jednym z mniejszych problemów TSP, a następnie współczynniki te zostaną wykorzystane do policzenia wyników dla reszty zestawów danych.
Zbiór testowy TEST $= T \times \alpha \times n \times k$ to iloczyn kartezjański następujących zbiorów:

1. $T = \{10.0, 50.0, 100.0, 250.0, 500.0, 1000.0, 2000.0\}$
2. $\alpha = \{0.90, 0.95, 0.97, 0.99, 0.995, 0.997\}$
3. $n = \{100, 250, 500, 1000\}$
4. $k = \{5, 10, 25, 50\}$

Następnie dla każdego egzemplarza t ze zbioru testowego $t \in$ TEST wyznaczana jest najlepszy wynik z $l = 10$ prób. Najmniejszy z nich decyduje o najlepszym egzemplarzu.

```
Temperature     : 1000
Alpha           : 0.995
Epochs          : 1000
Tries per epoch : 50
```

Widzimy, że wzrost T, α nie zawsze powoduje usprawnienie wyniku. Wzrost n, k zwiększa szanse na uzyskanie lepszego wyniku, ale tylko do pewnego momentu - po którym również zmiany nie są na tyle istotne aby go poprawić - dla dużego n mamy bardzo małe T, więc niewielką szanse na przeskok z minimum lokalnego na globalnie niższą wartość.
Wyniki prezentują się następująco:

Tabela 1: Symulowane Wyżarzanie

Plik	Liczba Miast	Średni dystans	Najlepszy dystans	OPT	Błąd Względny
File: western_sahara.tsp	29	28259.79	27603	27603	0.00%
File: djibouti.tsp	38	7732.69	7347	6656	10.38%
File: qatar.tsp	194	16076.63	15584	9352	66.64%
File: uruguay.tsp	734	318875.67	317750	79114	301.63%
File: zimbabwe.tsp	929	465948.39	457583	95345	379.92%
File: oman.tsp	1979	1218694.34	1195007	86891	1275.30%
File: canada.tsp	4663	30919498.49	30633201	1290319	2274.08%
File: tanzania.tsp	6117	9826150.00	9723972	394718	2363.53%
File: egypt.tsp	7146	4885386.74	4882675	172386	2732.41%
File: ireland.tsp	8246	5353907.10	5320736	206171	2480.74%

Warto zauważyć że moglibyśmy dalej optymalizować wybierając dobry przypadek startowy - np. wariant 2OPT oraz zwiększając n, k w celu uzyskania większej dokładności

4 Zadanie 2 - Tabu Search

Tabu search to algorytm metaheurystyczny, który zakłada stworzenie listy Tabu, która zawiera reprezentacje poprzednich rozwiązań - tak aby docelowo nie dało się zostać w minimum lokalnym, a zamiast tego wyjść z niego w poszukiwaniu lepszego rozwiązania.
Kluczowym dla tabu search jest maksymalna długość listy L, najlepiej rozumiana jako zależna od długości danych wejściowych (liczby miast n). Algorytm potrzebuje również sposobu wyboru kolejnego rozwiązania oraz dobrego warunku stopu, który pozwala ocenić kiedy rozwiązanie jest wystarczające. W tym przypadku będzie to maksymalna liczba iteracji I,
Zbiór testowy TEST $= I \times L$ to iloczyn kartezjański następujących zbiorów:

1. $I = \{10, 20, 50, 100, 200, 500, 1000, 2000\}$
2. $L = \left\{\frac{n}{20}, \frac{n}{10}, \frac{n}{4}, \frac{n}{2}, \frac{3n}{4}, n\right\}$

Następnie dla każdego egzemplarza t ze zbioru testowego $t \in$ TEST wyznaczana jest najlepszy wynik z $l = 5$ prób. Najmniejszy z nich decyduje o najlepszym egzemplarzu.

```
Max Iterations : 2000
Tabu Size      : n/10
```

Im większa liczba maksymalnych iteracji $i \in I$ tym dokładniejszy wynik, przy czym rozmiar listy tabu jest odpowiedni do $\frac{n}{10}$. Większa lista nie powoduje znalezienia lepszego rozwiązania. Mając na względzie że jest to rozmiar bardzo duży należy przemyśleć ekstrapolowanie tej wartości na większe problemy. Bezpiecznie założyć będzie istotnie dużą wartość, niebędącą jednak stricte zależną od danych, a raczej od możliwości komputera. Do algorytmu należało dołożyć lepsze wybieranie sąsiada, w celu osiągnięcia zadowalających wyników. Sposób wyboru sąsiada opiera się o zamianie 2 krawędzi z najbliższym pod względem dystansu kartezjańskiego miastem w danej turze, tak aby *rozplątać* ścieżkę. Wyniki prezentują się następująco:

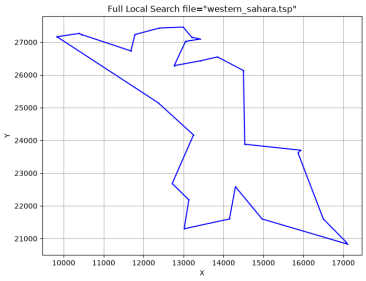
Tabela 2: Tabu Search

Plik	Liczba Miast	Średni dystans	Średnia liczba kroków	Najlepszy dystans	OPT	Błąd Względny
File: western_sahara.tsp	29	16	26150	10550	27603	61.78%
File: djibouti.tsp	38	11	11461.1111	43252.7778	6656	549.83%
File: qatar.tsp	194	51	25286.1111	51504.1667	9352	450.73%
File: uruguay.tsp	734	25	30733.3333	57316.6667	79114	27.55%
File: zimbabwe.tsp	929	63	17333.3333	28933.3333	95345	69.65%
File: oman.tsp	1979	34	17068.0556	54540.8333	86891	37.23%
File: canada.tsp	4663	52	42900	78933.3333	1290319	93.88%
File: tanzania.tsp	6117	25	1100	34050	394718	91.37%
File: egypt.tsp	7146	1	22216.6667	31516.6667	172386	81.72%
File: ireland.tsp	8246	6	51470	9416.3889	206171	95.43%

Tabu search wykazuje mniejsze względne zbliżenie do wartości optymalnych, ale zachowuje swoją przewagę dla większych egzemplarzy, tam gdzie symulowane wyżarzanie zostaje w minimum lokalnym, tabu search znajduje lepsze rozwiązania, choć nadal dalekie od optymalnych.

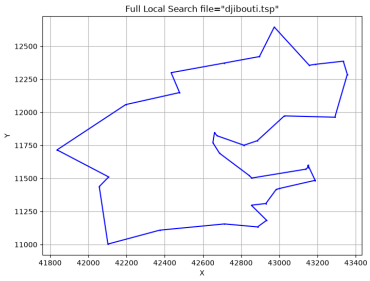
5 Wykresy

5.1 western_sahara.tsp



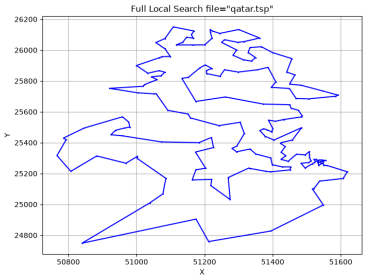
task2(western_sahara.tsp)

5.2 djibouti.tsp



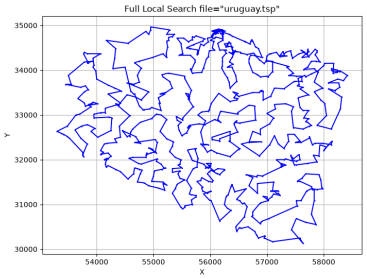
task2(djibouti.tsp)

5.3 qatar.tsp



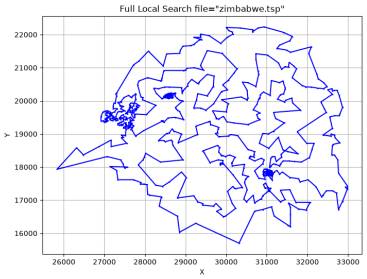
task2(qatar.tsp)

5.4 uruguay.tsp



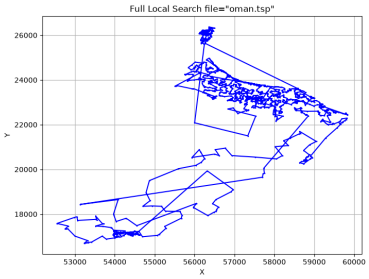
task2(uruguay.tsp)

5.5 zimbabwe.tsp



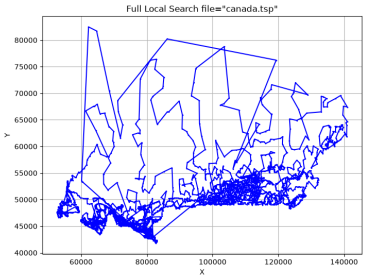
task2(zimbabwe.tsp)

5.6 oman.tsp



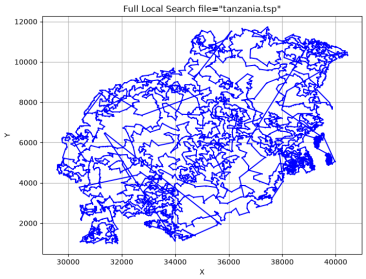
task2(oman.tsp)

5.7 canada.tsp



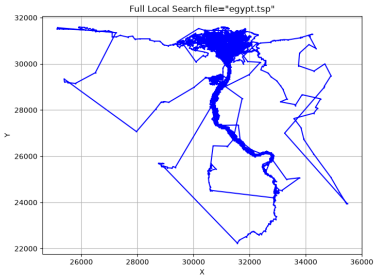
task2(canada.tsp)

5.8 tanzania.tsp



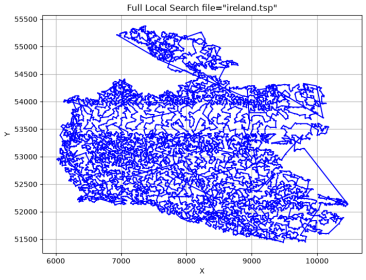
task2(tanzania.tsp)

5.9 egypt.tsp



task2(egypt.tsp)

5.10 ireland.tsp



task2(ireland.tsp)

6 Wnioski

Zgodnie z `No Free Lunch Theorem` nie istnieje uniwersalna metoda, która będzie najlepsza dla wszystkich problemów optymalizacyjnych. Wyniki pokazują, że dla mniejszych problemów TSP (do około 100 miast) Symulowanie Wyżarzanie pozwala znaleźć wyniki bliskie wartościom optymalnym, podczas gdy wyniki dla większych miast znacząco odbiegają od optymalnych. Wynika to z faktu, iż dostosowanie parametrów algorytmu nie przenosi się dobrze na większe problemy.
Dla dobranych wcześniej parametrów i znanych danych oba algorytmy w oczekiwany sposób dochodzą do rozwiązań, która są podobne do tych z poprzedniej listy. Dla większych problemów algorytm symulowanego wyżarzania zostaje w minimum lokalnym, podczas gdy tabu search znajduje nieco lepsze rozwiązania.
Dostrojenie parametrów obu algorytmow nie jest proste, ale dostatecznie szczegółowe powinno pozwolić na osiągnięcie zbliżonych wyników dla obu algorytmów. Wysoką przewagą tych algorytmów nad poprzednimi jest szybki czas wykonania oraz możliwość wyskoczenia z minimum lokalnego.