

Algorytmy Metaheurystyczne - lab 3

Jakub Musiał 268442

Styczeń 2024

1 Opis problemu

Wyznaczyć przybliżenie optymalnego cyklu komiwożera grafu pełnego używając algorytmu genetycznego.

Opis algorytmu

Schemat działania algorytmu genetycznego jest następujący:

1. Wyznaczenie populacji początkowej
2. Selekcja "rodziców", którzy będą brali udział w procesie krzyżowania, z obecnej populacji
3. Krzyżowanie - tworzenie "potomków" na podstawie genów "rodziców"
4. Mutacja - z pewnym prawdopodobieństwem geny potomków mogą zostać poddane mutacji
5. Powtórzenie kroków 2-4, jeśli nie osiągnięto warunku stopu

Poniżej przedstawiony jest pseudokod opisanego algorytmu. Użyte oznaczenia:

- G - graf wejściowy
- M - maksymalna liczba iteracji wyrażona jako procent liczby wierzchołków grafu
- n_p - rozmiar populacji
- *crossover* - metoda krzyżowania (zmienna funkcyjna)
- p_m - prawdopodobieństwo mutacji

Dodatkowo by usprawnić działanie algorytmu zastosowałem wyspowy wariant algorytmu, w którym w tym samym czasie istnieje kilka odizolowanych populacji, w których wymiana informacji jest możliwa co p pokoleń (w implementacji użyłem $p = 25$).

Algorithm 1 Genetic algorithm

```
1: procedure genetic_algorithm( $G, M, n_p, crossover, p_m$ )
2:    $P \leftarrow generate\_population(G, n_p)$  ▷ Initial population
3:
4:   for  $i \leftarrow 1$  to  $M \cdot |V_G|$  do
5:      $p \leftarrow select\_parents(P)$ 
6:      $P_n \leftarrow \{\}$  ▷ Empty population
7:
8:     for  $i_p \leftarrow 1$  to  $|p|$  do
9:        $c_a \leftarrow crossover(p[i_p].first, p[i_p].second)$ 
10:       $c_b \leftarrow crossover(p[i_p].second, p[i_p].first)$ 
11:      if  $random\_prob() < p_m$  then
12:         $c_a \leftarrow mutate(c_a)$ 
13:      if  $random\_prob() < p_m$  then
14:         $c_b \leftarrow mutate(c_b)$ 
15:       $push(P_n, [c_a, c_b])$ 
16:
17:    $P \leftarrow P_n$ 
18:    $\pi_b \leftarrow argmin(P)$  ▷ Population member with the minimum weight
19: return  $\pi_b$ 
```

Dobór parametrów

By określić najlepsze parametry przeprowadziłem eksperyment polegający na sprawdzeniu wyników kombinacji z losowej próbki wszystkich możliwych kombinacji poniżej określonych parametrów, a następnie znajdując taką, która generuje najmniejszy błąd względny.

Badane parametry:

- M - maksymalna liczba iteracji (jako procent liczby wierzchołków grafu): $\{10, 25, 50\}$
- n_p - rozmiar populacji: $\{50, 100\}$
- $crossover$ - metoda krzyżowania (zmienna funkcyjna): $\{simple, single_point\}$
- p_m - prawdopodobieństwo mutacji: $\{0.1, 0.2, 0.3\}$

Najlepszą znaną kombinacją parametrów jest:

$$M = 50 \wedge n_p = 100 \wedge crossover = single_point \wedge p_m = 0.2$$

dla której otrzymałem średni błąd względny $\delta = 0.126002$.

Wyniki

Poniższa tabela oraz wykresy przedstawiają wyniki uzyskane dla wszystkich grafów testowych dla znalezionej kombinacji parametrów.

Dane wejściowe	$ V $	$avg(w(TSC))$	$min(w(TSC))$	$w(TSC_{opt})$
xqf131.tsp	131	629	599	564
xqg237.tsp	237	1126	1102	1019
pma343.tsp	343	1522	1482	1368
pka379.tsp	379	1480	1442	1332
bcl380.tsp	380	1839	1788	1621
pbl395.tsp	395	1467	1428	1281
pbk411.tsp	411	1518	1474	1343
pbn423.tsp	423	1561	1544	1365
pbm436.tsp	436	1630	1588	1443
xql662.tsp	662	2847	2818	2513
xit1083.tsp	1083	4079	4073	3558
icw1483.tsp	1483	5067	4944	4416
djc1785.tsp	1785	7054	6917	6115
dcb2086.tsp	2086	7587	7447	6600
pds2566.tsp	2566	8889	8741	-

Table 1: Wyniki dla wszystkich danych wejściowych dla algorytmu genetycznego wraz z wagą optymalnych ścieżek

Dane wejściowe	$ V $	$w_{C(MST)}$	avg_{LS}	avg_{SA}	avg_{TS}	avg_{GA}
xqf131.tsp	131	742	621	576	614	629
xqg237.tsp	237	1424	1118	1049	1108	1126
pma343.tsp	343	1876	1497	1380	1485	1522
pka379.tsp	379	1799	1450	1341	1447	1480
bcl380.tsp	380	2337	1817	1663	1802	1839
pbl395.tsp	395	1893	1456	1312	1431	1467
pbk411.tsp	411	1930	1489	1380	1485	1518
pbn423.tsp	423	1921	1533	1403	1528	1561
pbm436.tsp	436	2095	1629	1481	1611	1630
xql662.tsp	662	3692	2822	2583	2816	2847
xit1083.tsp	1083	-	4021	3655	3936	4079
icw1483.tsp	1483	-	4986	4532	4886	5067
djc1785.tsp	1785	-	6878	6273	6837	7054
dcb2086.tsp	2086	-	7463	6810	7342	7587
pds2566.tsp	2566	-	8695	7898	8685	8889

Table 2: Porównanie wyników dla wszystkich badanych algorytmów

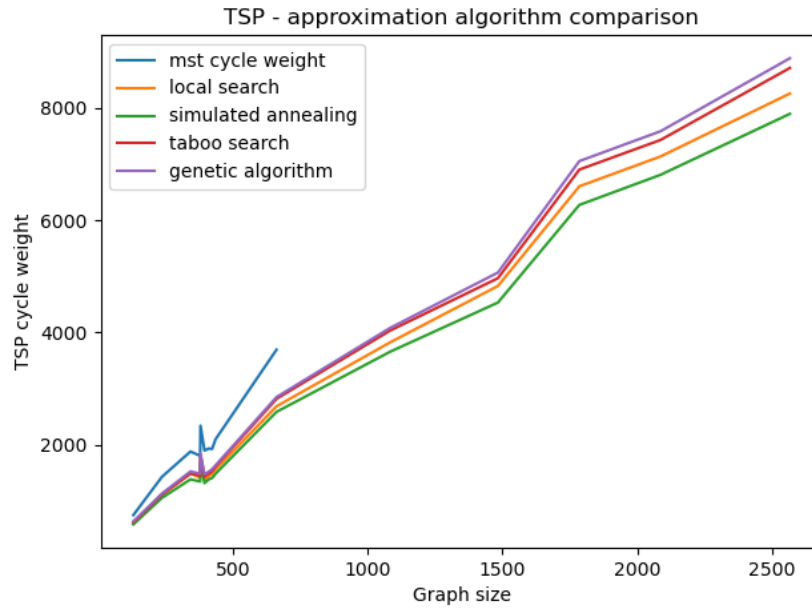
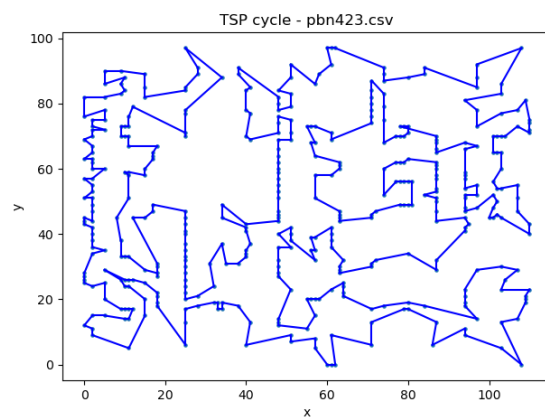
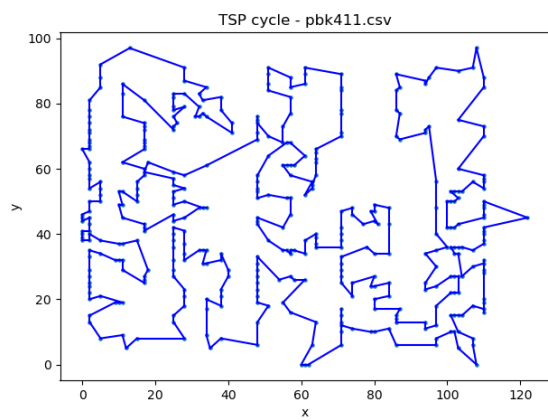
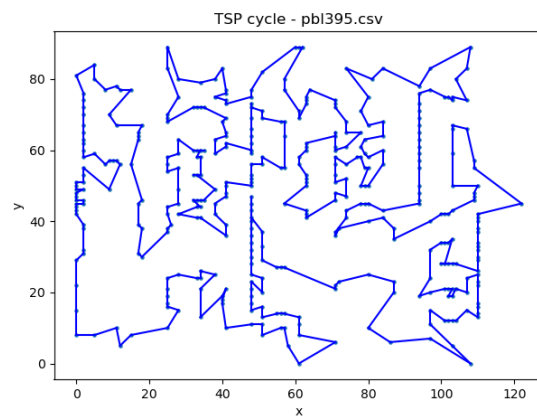
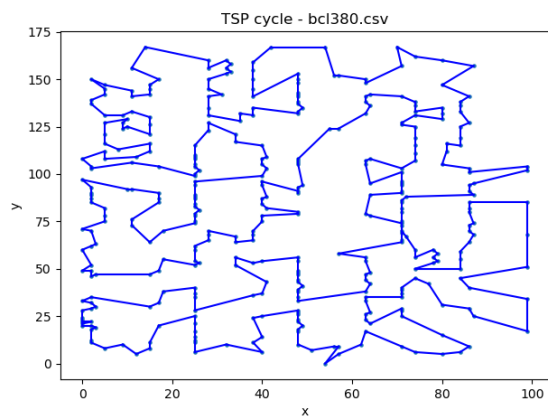
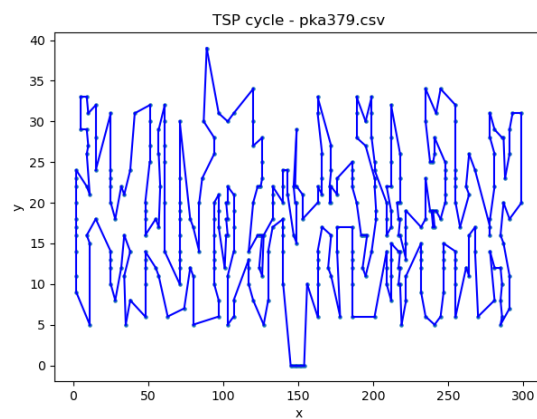
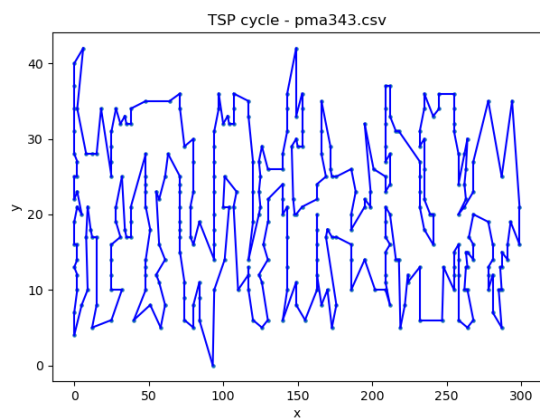
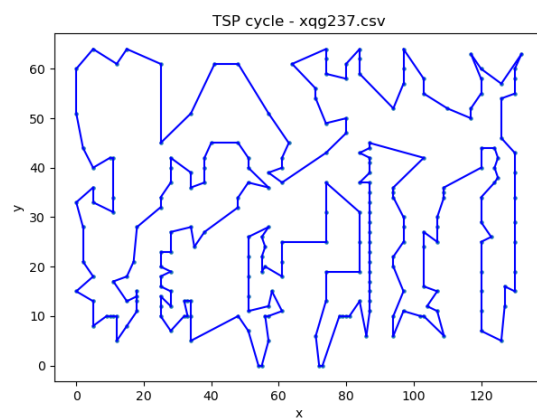
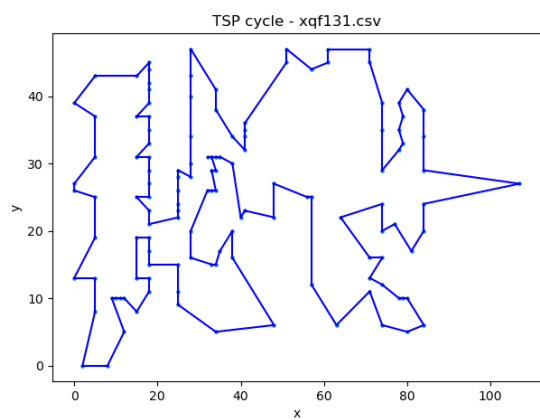


Figure 1: Porównanie wyników dla wszystkich badanych algorytmów

Na podstawie powyższej tabeli oraz wykresu możemy zauważyć, że dla zadanego problemu najlepsze wyniki zwróciła metoda symulowanego wyżarzania.



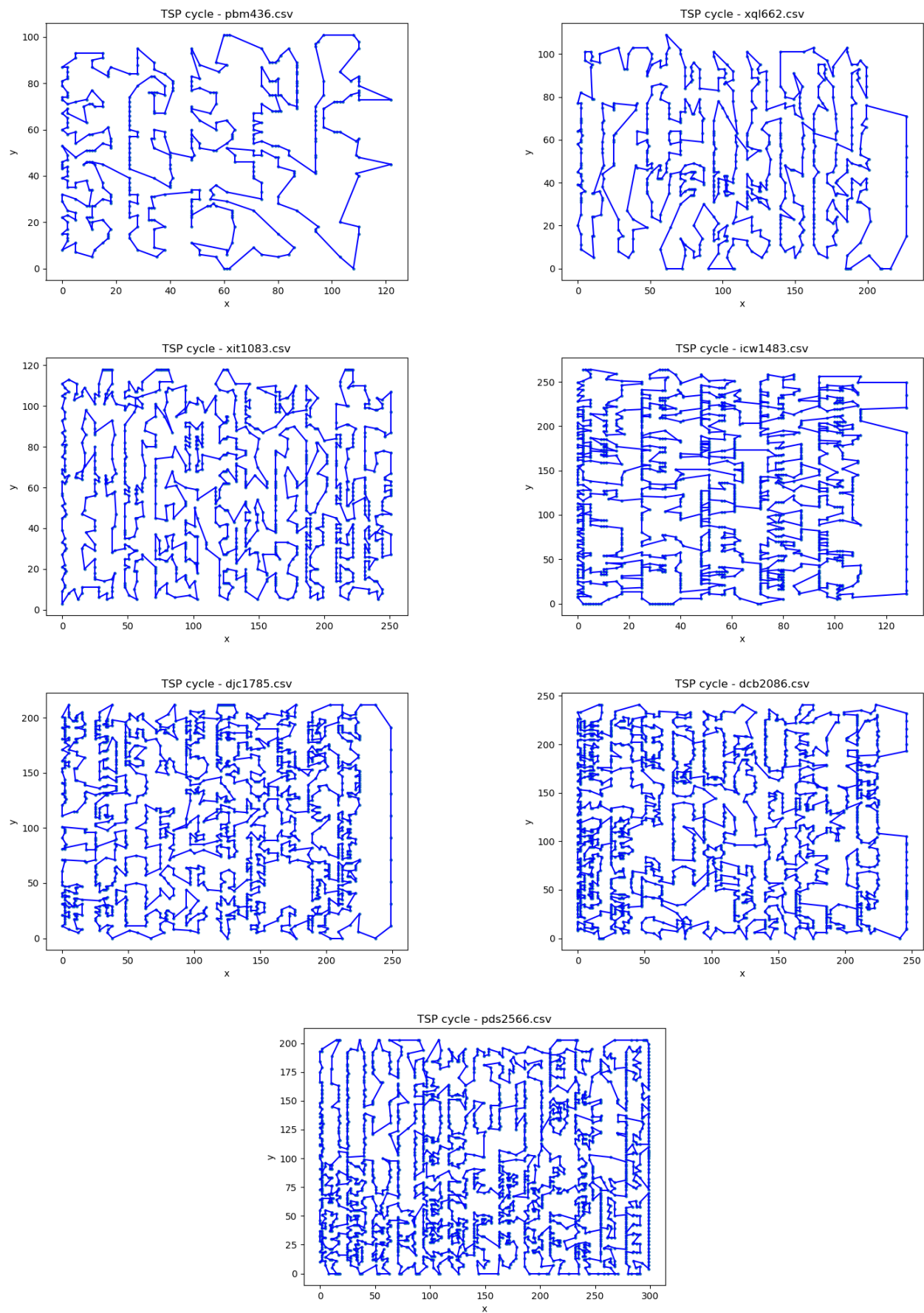


Figure 3: Genetic algorithm: wizualizacja wyznaczonych cykli komiwojażera