

AEM - Algorytm ewolucyjny

Bartosz Mila 131804

18 maja 2020

1. Krótki opis zadania.

Celem jest implementacja hybrydowego algorytmu ewolucyjnego i porównanie go z metodami MSLS i ILSx zaimplementowanymi w poprzednim zadaniu.

Proponowane parametry algorytmu: populacja elitarna o wielkości 20, algorytm steady state, w populacji nie mogą się znajdować kopie tego samego rozwiązania.

Proponowany operator rekombinacji: Wyszukujemy wszystkie wspólne podścieżki w obu rodzicach. Podścieżka może być pojedynczym wierzchołkiem, pojedynczą krawędzią, lub zbiorem połączonych krawędzi. Jeżeli suma wierzchołków we wspólnych podścieżkach jest mniejsza niż zadana liczba wierzchołków, to dodajemy losowo wierzchołki, które należały do jednego z rodziców. Podścieżki łączymy w sposób losowy, tak aby powstał cykl hamiltona.

Parametry eksperymentu i dane do sprawozdania takie same jak w przypadku ILSx.

2. Opis zaimplementowanych algorytmów w pseudokodzie.

(HEA) Hybrid Evolutionary Algorithm:

- Utwórz populację początkową bez kopii - 20 losowych, odmiennych ścieżek.
- Dopóki nie zostanie przekroczony średni czas MSLS:
 - Wylosuj dwóch rodziców - 2 losowo wybrane rozwiązania z populacji.
 - Wyszukaj wszystkie wspólne podścieżki - niezależnie od kierunku ich zapisania w rodzicach (zgodny / odwrotny).
 - Dopóki znaleziona liczba wierzchołków jest mniejsza niż zadana:
 - Losuj kolejny wierzchołek znajdujący się w jednym rodzicu i dołącz go do znalezionych wierzchołków jako ścieżkę 1-elementową.
 - Skonstruuj rozwiązanie potomne - połącz wszystkie znalezione podścieżki w sposób losowy.
 - Poddaj rozwiązanie potomne algorytmowi lokalnego przeszukiwania.
 - Sprawdź otrzymany rezultat - jeśli rozwiązanie potomne jest lepsze od najgorszego rozwiązania z populacji - usuń to rozwiązanie i na jego miejsce wstaw rozwiązanie potomne.
- Zwróć jako najlepsze rozwiązanie najlepszy wynik z populacji.

3. Wyniki eksperymentu obliczeniowego.

Do eksperymentu wykorzystano **poprawioną wersję zadania 4**. Stąd różnice w wynikach dla MSLS, ILS1 i ILS2 (względem sprawozdania 4) oraz znacznie większy ich wpływ na optymalizację.

Instancja A	MSLS	ILS1	ILS2	HEA
Najkrótsza ścieżka	14668	13072	12765	13298
Średnia ścieżka	17021.80	13290.4	13223.8	13362.33
Najdłuższa ścieżka	19657	13402	13453	13467
Minimalny czas [s]	798.93	834.53	834.54	834.73
Średni czas [s]	834.49	834.77	834.66	835.55
Maksymalny czas [s]	859.95	834.99	834.82	836.65

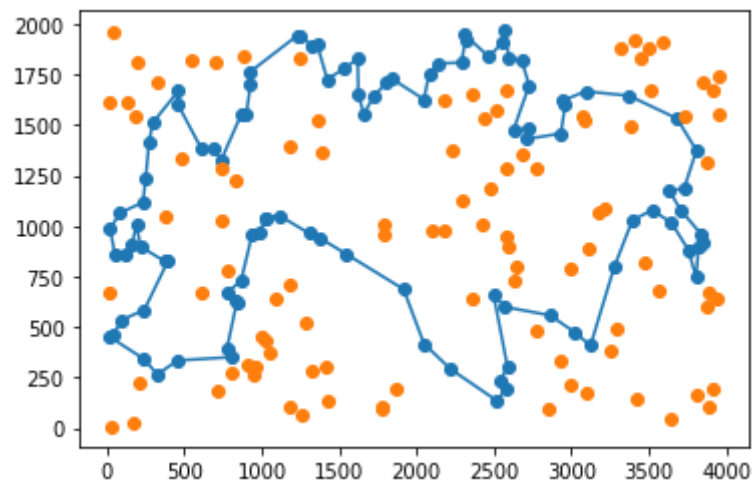
Tabela 1. Wyniki długości ścieżek i czasu działania dla wykonywanych algorytmów dla zbioru A

Instancja B	MSLS	ILS1	ILS2	HEA
Najkrótsza ścieżka	14982	12533	12828	13004
Średnia ścieżka	17037.94	13160.0	13276.8	13505.42
Najdłuższa ścieżka	19565	13521	13493	13770
Minimalny czas [s]	816.72	830.27	830.34	834.61
Średni czas [s]	830.25	830.37	830.54	835.5
Maksymalny czas [s]	863.97	830.57	830.70	836.87

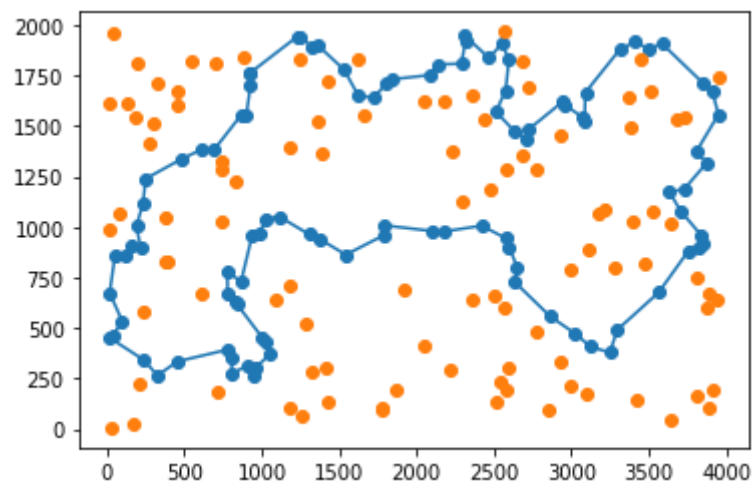
Tabela 2. Wyniki długości ścieżek i czasu działania dla wykonywanych algorytmów dla zbioru B

4. Wizualizacje najlepszych rozwiązań dla każdej kombinacji.

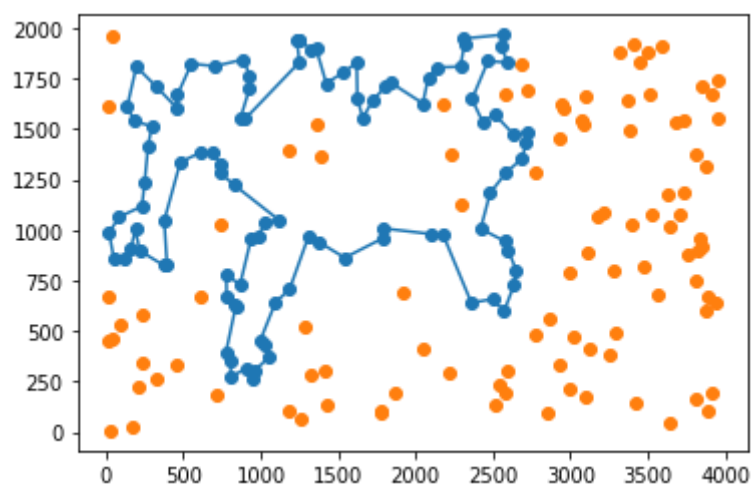
Instancja A:



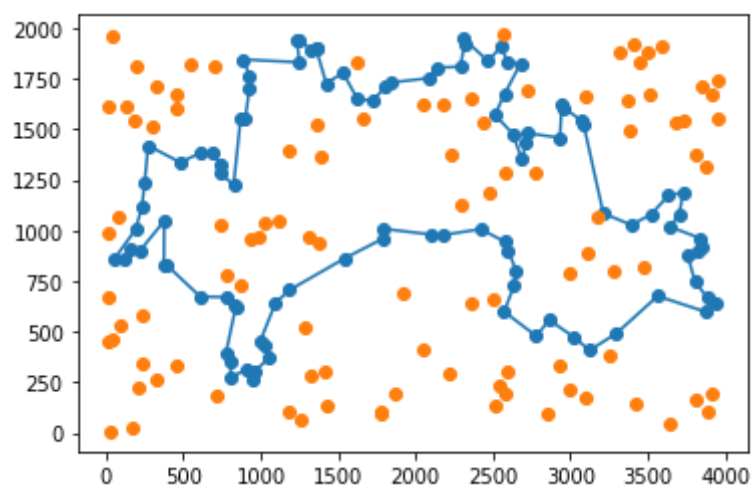
Rysunek 1. MSLS



Rysunek 2. ILS1

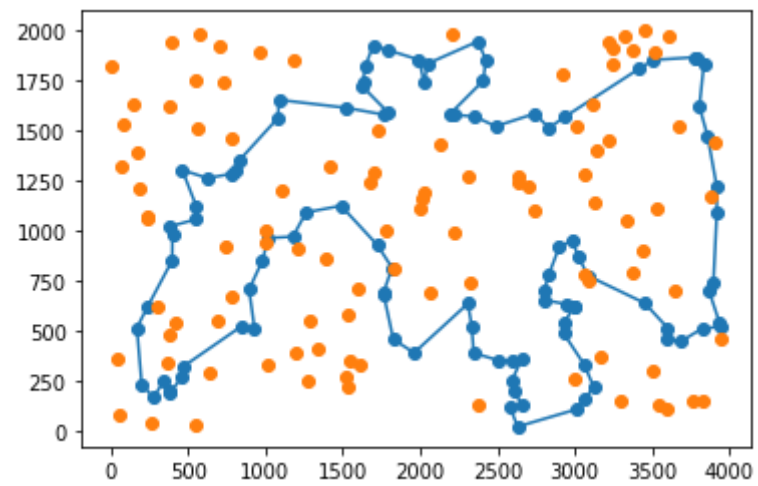


Rysunek 3. ILS2

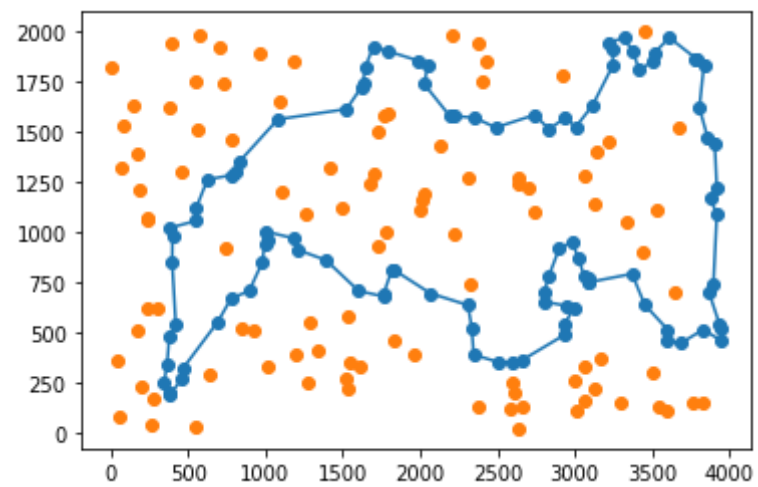


Rysunek 4. HEA

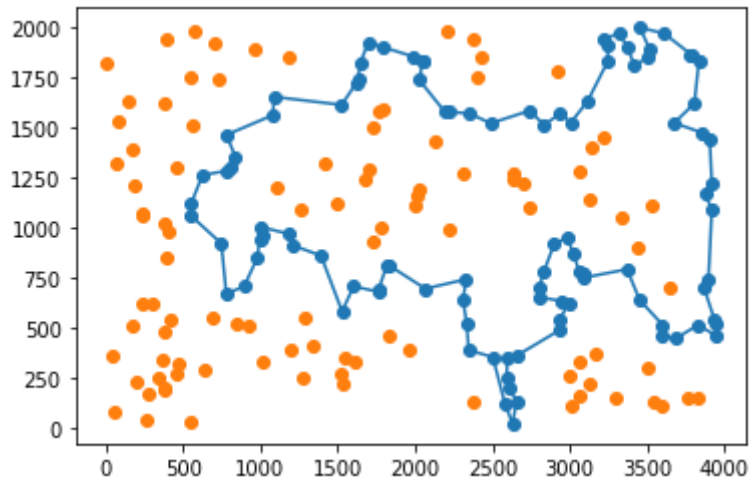
Instancja B:



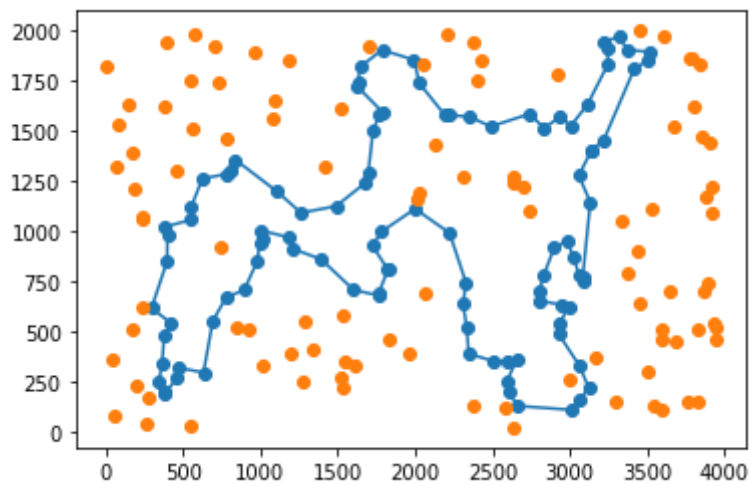
Rysunek 5. MSLS



Rysunek 6. ILS1



Rysunek 7. ILS2



Rysunek 8. HEA

6. Wnioski.

Poprawienie algorytmów z zadania 4 poskutkowało znacznym wzrostem optymalności znajdowanych rozwiązań. Algorytmy ILS1 i ILS2 znajdują zauważalnie lepsze rozwiązania. Z kolei zaimplementowany algorytm ewolucyjny radzi sobie nieznacznie gorzej od ILS1 i ILS2, jednak wciąż znacząco lepiej od MSLS.

7. Kod programu (np. w postaci linku).

<https://github.com/Kurkum/AEM>