Techniki optymalizacji

Sprawozdanie nr 1,2.

Marcin Jabłoński 117270

Kamil Kaźmierczak 117293

# Opis zadania

Zadanie polega na znalezieniu rozwiązania, zmodyfikowanego problemu komiwojażera, w którym nie istnieje konieczność przejścia po wszystkich dostępnych wierzchołkach, oraz za każdy odwiedzony wierzchołek dostajemy pewien zysk, który należy maksymalizować, a krawędzie obarczone są pewnymi kosztami.

Do rozwiązania zadania wykorzystane mają zostać następujące algorytmy:

* Nearest Neighbour (NN)
* Greedy Cycle
* Greedy Cycle with Regret

Następnie dla tak wygenerowanych rozwiązań zastosować algorytm Local Search dla trzech ruchów

* Dodanie

# Pseudokody algorytmów

1. Nearest Neighbour (NN)

**Dodaj do** listy wierzchołków wierzchołek startowy

**Dopóki** istnieje wierzchołek, którego odwiedzenie nie przynosi straty

**Dodaj do** listy wierzchołek, który przynosi największy zysk

**Dodaj** do listy wierzchołków wierzchołek startowy

1. Greedy Cycle

**Dodaj do** listy wierzchołków wierzchołek startowy

**Dodaj do** listy wierzchołek który przynosi największy zysk

**Dodaj do** listy wierzchołków wierzchołek początkowy, aby utworzyć cykl

**Dopóki** istnieje wierzchołek, taki że usunięcie pewnej krawędzi, w celu powiększenia cyklu o ten wierzchołek nie przynosi straty

**Dodaj do** cyklu ten wierzchołek, w miejsce dawnej krawędzi

1. Greedy Cycle with Regret

**Dodaj do** listy wierzchołków wierzchołek startowy

**Dodaj do** listy wierzchołek który przynosi największy zysk

**Dodaj do** listy wierzchołków wierzchołek początkowy, aby utworzyć cykl

**Dopóki** znaleziony nowy wierzchołek polepsza rozwiązanie

**Znajdź** wierzchołek, którego żal (różnica pomiędzy najlepszym, a kolejnym wstawieniem pomiędzy krawędź) jest największy

**Dodaj do** rozwiązania ten wierzchołek

1. Random Solution

**Wylosuj** liczbę wierzchołków

**Wylosuj** cykl o wylosowanej wcześniej długości

1. Local Search

**Dopóki** delta > 0

**Oblicz** deltę każdego z trzech możliwych popraw lokalnych (dodaj wierzchołek, usuń wierzchołek, zamień łuki)

**Wybierz** i zastosuj poprawę, z największą deltą

# Rozwiązania problemu

1. Nearest Neighbour (NN)

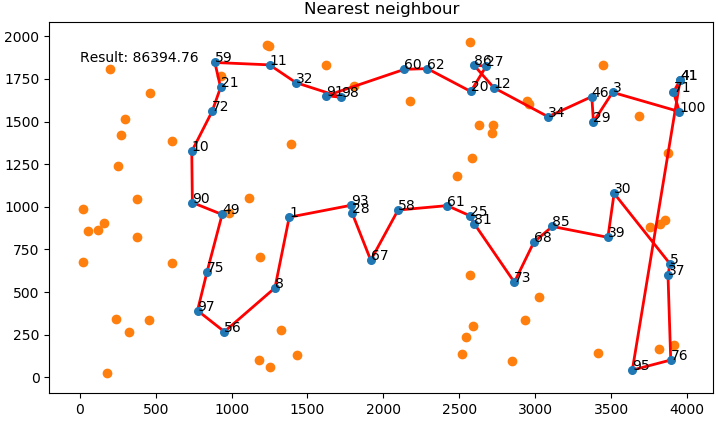
**Wartości:**

* Min - 6345.0
* Avg - 54318.2648681796
* Max - 86394.76344206632

**Rozwiązanie maksymalne:**

41, 71, 100, 3, 29, 46, 34, 12, 86, 27, 20, 62, 60, 91, 98, 32, 11, 59, 21, 72, 10, 90, 49, 75, 97, 56, 8, 1, 93, 28, 67, 58, 61, 25, 81, 73, 68, 85, 39, 30, 5, 37, 76, 95, 41

**Wizualizacja:**



1. Greedy Cycle (GC)

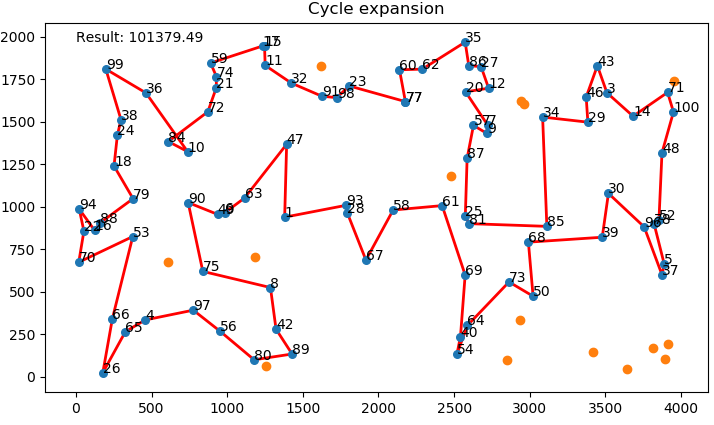
**Wartości:**

* Min - 2520.7500616818897
* Avg - 34976.450180616375
* Max - 101379.4942203332

**Rozwiązanie maksymalne:**

77, 60, 62, 35, 86, 27, 12, 20, 7, 9, 57, 87, 25, 81, 85, 34, 29, 46, 43, 3, 14, 71, 100, 48, 52, 78, 5, 37, 96, 30, 39, 68, 50, 73, 64, 54, 40, 69, 61, 58, 67, 28, 93, 1, 47, 63, 6, 49, 90, 75, 8, 42, 89, 80, 56, 97, 4, 65, 26, 66, 53, 70, 22, 94, 16, 88, 79, 18, 24, 38, 99, 36, 10, 84, 72, 21, 74, 59, 17, 15, 11, 32, 91, 98, 23, 77

**Wizualizacja:**



1. Greedy Cycle with Regret (GC with regret)

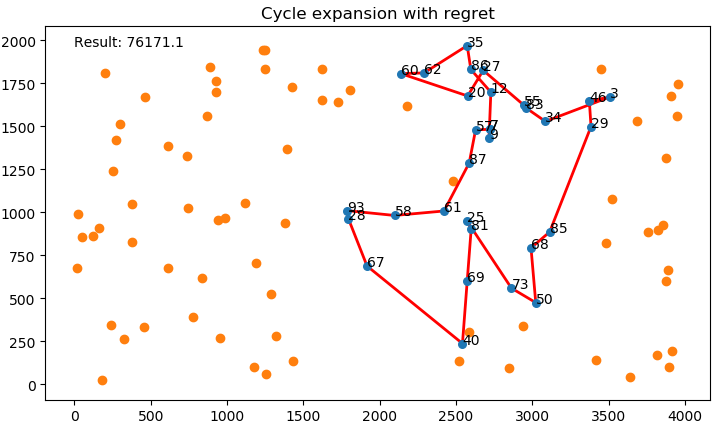
**Wartości:**

* Min - 3211.1684793212
* Avg - 28027.778602113467
* Max - 76171.10357857909

**Rozwiązanie maksymalne:**

7, 57, 87, 61, 58, 93, 28, 67, 40, 69, 81, 25, 73, 50, 68, 85, 29, 46, 3, 34, 83, 55, 27, 20, 60, 62, 35, 86, 12, 9, 7

**Wizualizacja:**



1. Nearest Neighbour (NN) (with Local Search)

**Wartości:**

* Min -
* Avg -
* Max -

**Czasy:**

* Min -
* Avg -
* Max -

**Rozwiązanie maksymalne:**

**Wizualizacja:**

1. Greedy Cycle (GC) (with Local Search)

**Wartości:**

* Min -
* Avg -
* Max -

**Czasy:**

* Min -
* Avg -
* Max -

**Rozwiązanie maksymalne:**

**Wizualizacja:**

1. Greedy Cycle with Regret (GC with regret) (with Local Search)

**Wartości:**

* Min -
* Avg -
* Max -

**Czasy:**

* Min -
* Avg -
* Max -

**Rozwiązanie maksymalne:**

**Wizualizacja:**

1. Random Solution (RS)

**Wartości:**

* Min -
* Avg -
* Max -

**Czasy:**

* Min -
* Avg -
* Max -

**Rozwiązanie maksymalne:**

**Wizualizacja:**

1. Podsumowanie Local search

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorytm | Rozwiązanie | | | Czas[s] | | |
| Min | Avg | Max | Min | Avg | Max |
| NN |  |  |  |  |  |  |
| GC |  |  |  |  |  |  |
| GC with regret |  |  |  |  |  |  |
| RS |  |  |  |  |  |  |