

# AEM - Zadanie nr 1

Bartosz Sobkowiak 125342 Joanna Świda 138675

22.03.2020

## 1 Opis zadania

Rozważany problem to zmodyfikowana wersja problemu komiwojażera. Dany jest zbiór wierzchołków i macierz symetrycznych odległości między nimi. Celem jest znalezienie najkrótszej ścieżki zamkniętej przechodzącej przez dokładnie 50% wszystkich wierzchołków (w przypadku nieparzystej liczby wierzchołków zaokrąglamy w górę). Wybór wierzchołków do ścieżki jest elementem rozwiązania. Minimalizowane kryterium to długość zamkniętej ścieżki.

## 2 Pseudokod

**Data:** zbiór wierzchołków, macierz odległości pomiędzy wierzchołkami  
**Result:** najrósza ścieżka łącząca X% wierzchołków  
wybierz i dodaj punkt pierwszy  
znajdź punkt leżący najbliżej pierwszego i dodaj go jako punkt drugi  
**while** cykl zawiera mniej niż X% wszystkich punktów **do**  
    **for** każdy pozostały punkt p **do**  
        | oblicz długość cyklu cLen jeśli punkt p zostałby dodany do cyklu  
    **end**  
    dodaj do cyklu punkt o najmniejszej wartości cLen  
    usuń ten punkt z listy pozostałych punktów  
**end**

Algorithm 1: Greedy Cycle

**Data:** zbiór wierzchołków, macierz odległości pomiędzy wierzchołkami  
**Result:** najrósza ścieżka łącząca X% wierzchołków  
wybierz i dodaj punkt pierwszy  
znajdź punkt leżący najbliżej pierwszego i dodaj go jako punkt drugi  
**while** cykl zawiera mniej niż X% wszystkich punktów **do**  
    **for** każdy pozostały punkt p **do**  
        | oblicz długość cyklu cLen jeśli punkt p zostałby dodany do cyklu  
        | oblicz żal dla danego punktu  
    **end**  
    dodaj do cyklu rozwiązanie z największym żalem  
    usuń punkt z listy pozostałych punktów  
**end**

Algorithm 2: Greedy Cycle with K-regret

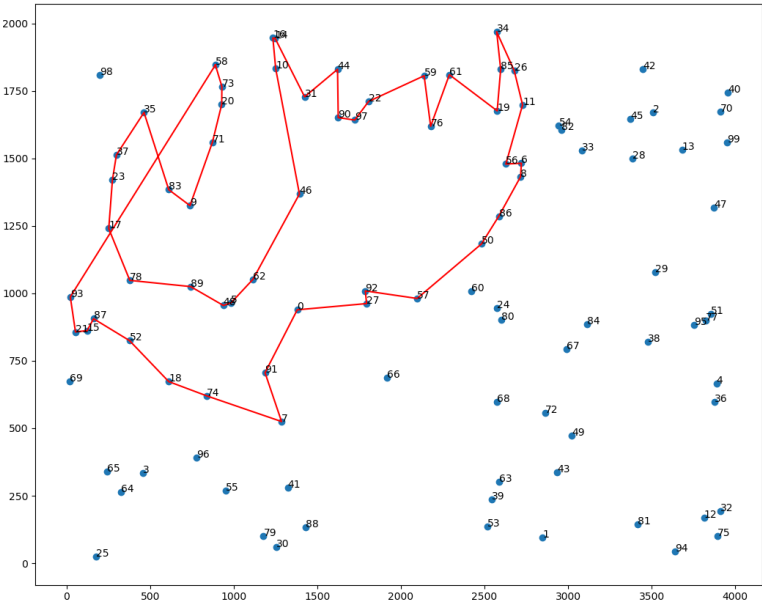
## 3 Wyniki obliczeń i wizualizacje

Zbiór	Min	Max	Avg
kroA <sub>100</sub>	11284	16157	14313
kroB <sub>100</sub>	10417	20202	14169

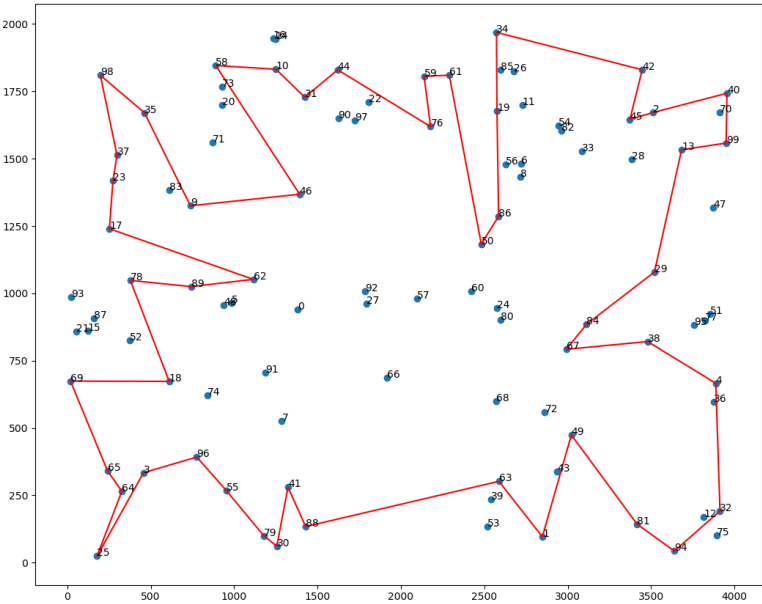
Tabela 1: Algorytm zachłanny - Greedy Cycle

Zbiór	Min	Max	Avg
kroA <sub>100</sub>	18679	24313	21710
kroB <sub>100</sub>	19830	23675	21118

Tabela 2: Algorytm z żalem - Greedy Cycle with K-regret



Rysunek 1: Algorytm zachłanny - KROA100

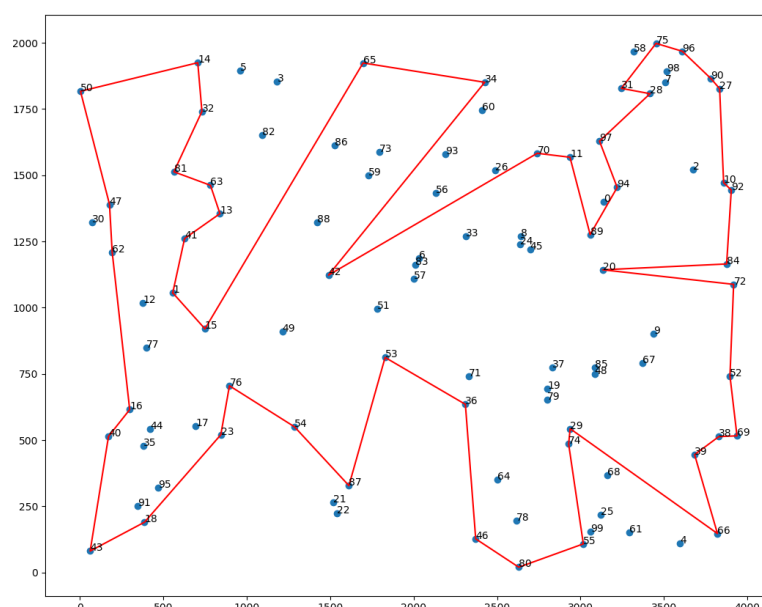
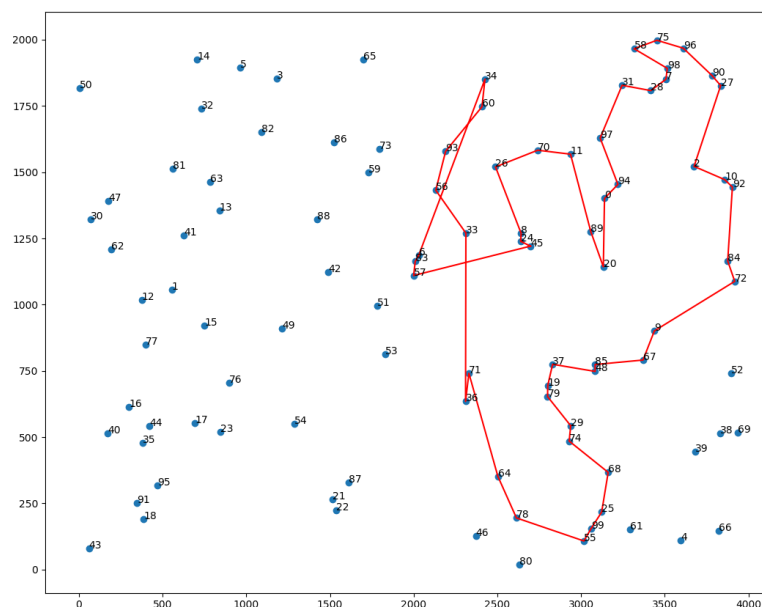


Rysunek 2: Algorytm z żalem - KROA100

4    Wizualizacje

5    Wnioski

Wyniki obliczeń jasno wskazują, że algorytm zachłanny radzi sobie znacznie lepiej przy przechodzeniu przez 50% punktów generując ścieżkę o ok. połowę krótszą. Jednakże kiedy te same algorytmy uruchomi się dla wszystkich punktów, to wyniki są zupełnie inne - algorytm z żalem wyznacza krótsze ścieżki. Dzieje się tak przy uruchamianiu algorytmów dla 90-100% punktów instancji. Spowodowane jest to tym, że algorytm z żalem wybiera początkowo "gorsze" wierzchołki (z większym żalem, często bardziej oddalone od siebie), co zaczyna przynosić profity dopiero w późniejszym etapie (do którego algorytm nie dochodzi gdyż kończy dodawanie na połowie punktów). Algorytm zachłanny nie ma takiego problemu, jego kontekst "przewidywania" jest węższy, więc przynosi tym lepsze wyniki (w porównaniu do algorytmu z żalem), im mniejsza jest liczba punktów z instancji, którą ma zawierać ścieżka.



## 6 Kod programu