Optymalizacja Kombinatoryczna - Projekt

Ant System dla problemu komiwojażera

Michał Łatka 160263
Stanisław Fiedler 160250

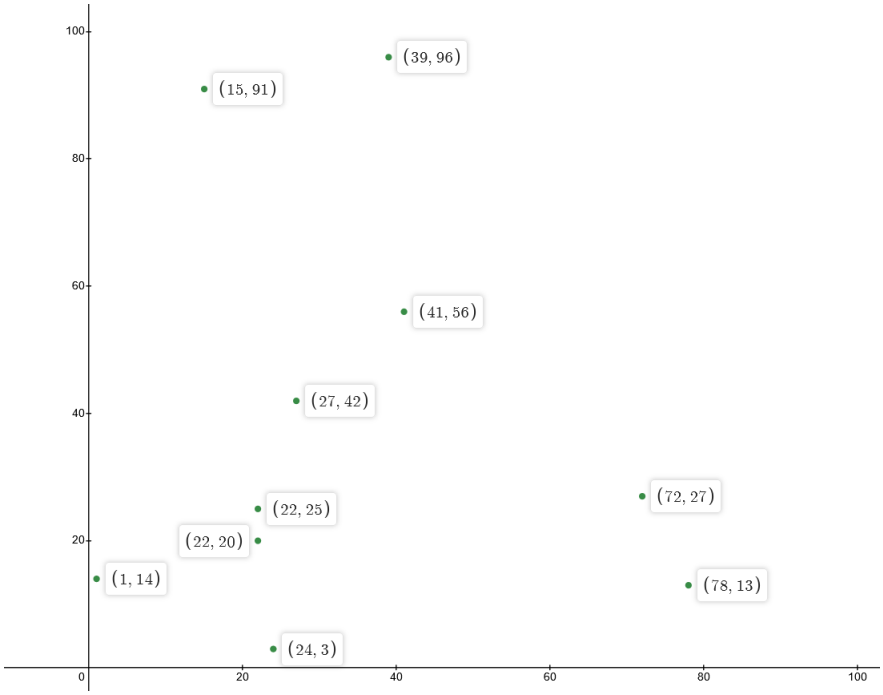
Politechnika Poznańska, Styczeń 2025

Spis treści

1 Ant System

1.1 Inicjalizacja

Instancja początkowa reprezentowana jest przez następujące dane: liczbę wierzchołków (n) oraz n par liczb całkowitych (x_i, y_i) oznaczających rozmieszczenie tych wierzchołków na płaszczyźnie kartezjańskiej.



1.2 Opis Algorytmu

Algorytmy mrówkowe są stosowane przy rozwiązywaniu niektórych problemów NP-trudnych. W przeszukiwaniu przestrzeni rozwiązań symulują one zachowanie kolonii mrówek. Każde znalezione rozwiązanie pozostawia informacje używane do budowy następnego.

Dla problemu komiwojażera w każdej iteracji wypuszczana jest pewna ilość mrówek które mają znaleźć rozwiązanie. Dla każdej mrówki, po znalezieniu ścieżki, na każdej krawędzi z której jest zbudowana, jest dodawana wartość (tzw. feromony) odwrotnie proporcjonalna do długości znalezionego rozwiązania. Mrówka tworząc trasę wybiera wierzchołki z prawdopodobieństwem zgodnym z siłą feromonów na krawędziach. Po każdej iteracji z krawędzi jest usuwany pewien procent feromonów.

Działanie algorytmu jest zależne od ustawionych parametrów:

INITIAL_FEROMONES: początkowa wartość feromonów na krawędziach

EVAPORATION_RATE: ułamek feromonów usuwany przy każdej iteracji

Q: stała określająca ilość feromonów pozostawianych przez mrówki

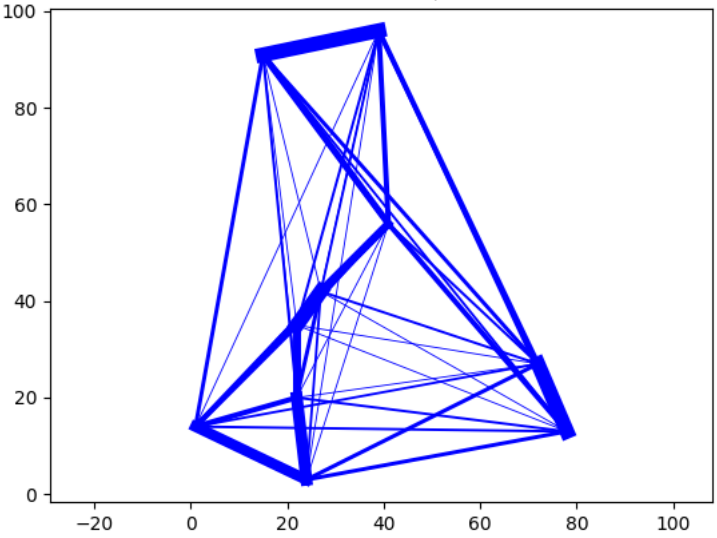
A, B: stałe używane przy wyborze następnego wierzchołka

1.3 Pseudokod

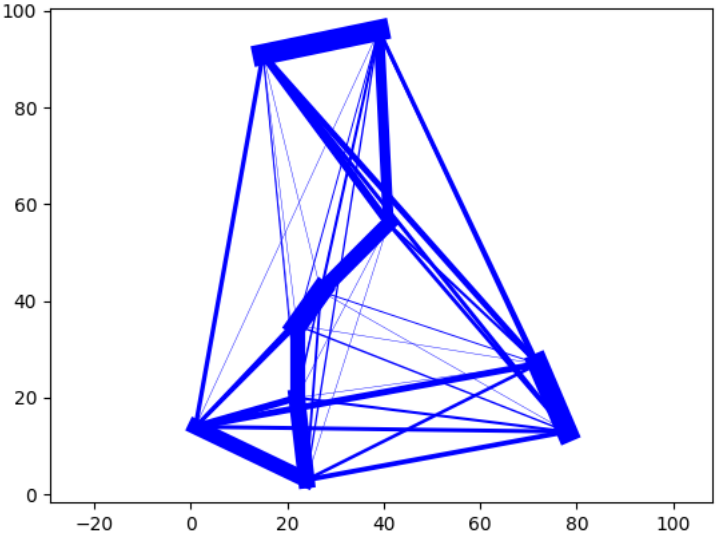
```
for each edge:
begin
    edge.lenght = distance(egde.start, edge.end);
    edge.feromones = INITIAL_FEROMONES;
end

for i in 0..ITERATION_MAX:
begin
    for each vertex:
    begin
        new ant;
        ant.visited[vertex] = true;
        ant.path += vertex;

        while ant.path < number_of_verteces:
        begin
            //find next unvisited vertex based on probabilites
            next_vertex = find_next_vertex();
            ant.path += next_vertex;
        end
    end

    for each edge:
    begin
        edge.feromones *= EVAPORATION_RATE;
    end

    for each ant:
    begin
        tour_length = length(ant.path);
        shortest_path = min(shortest_path, tour_length);
        for each edge in ant.path:
        begin
            edge.feromones += Q/tour_length;
        end
    end

end
```

1.4 Przykład obrazujący działanie

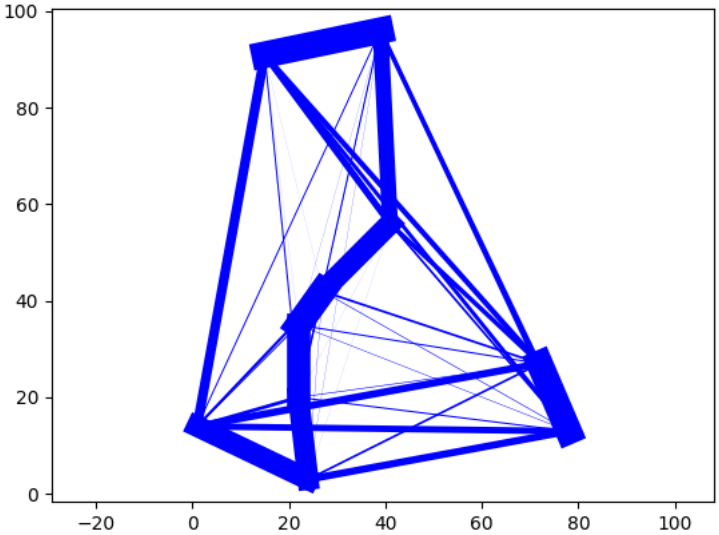
Intensywność feromonów na krawędziach po 1 iteracji:



Intensywność feromonów na krawędziach po 2 iteracjach:

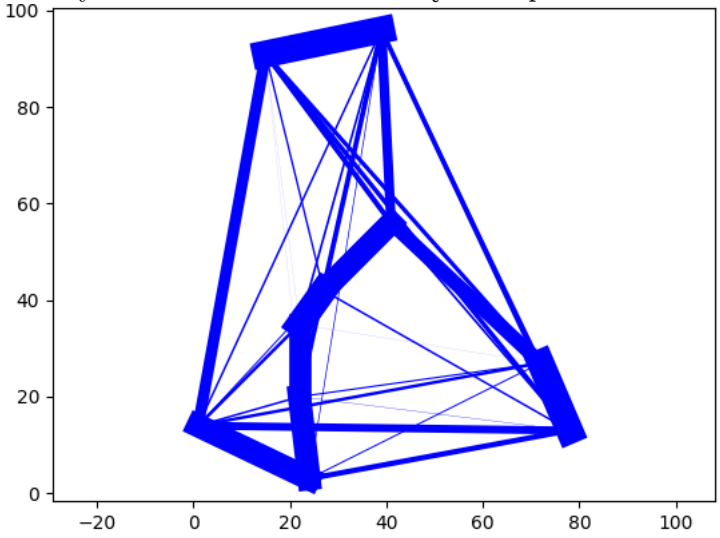


Intensywność feromonów na krawędziach po 5 iteracjach:

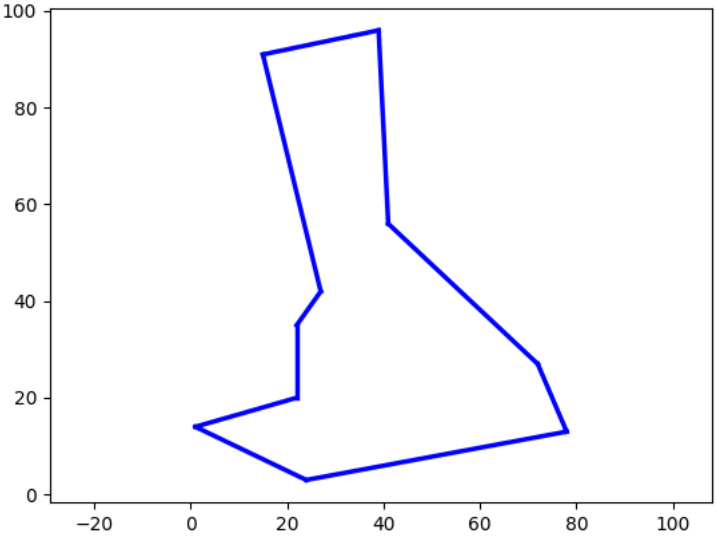


1.5 Finalizacja

Intensywność feromonów na krawędziach po zakończeniu programu:



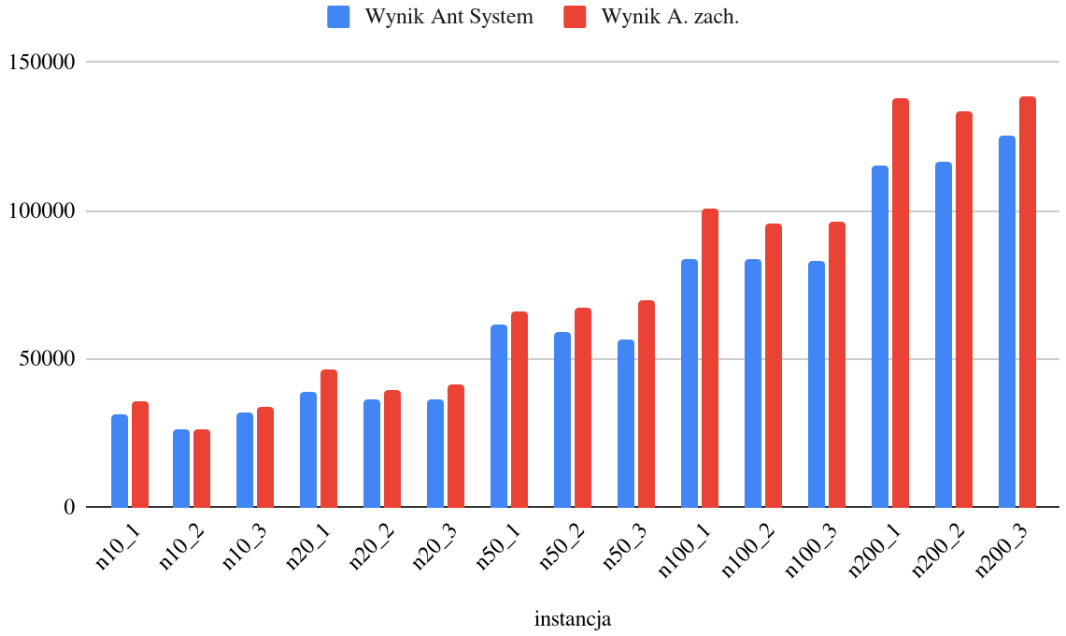
Najkrótsza znaleziona ścieżka (długości 298.551 j):



2 Wykresy

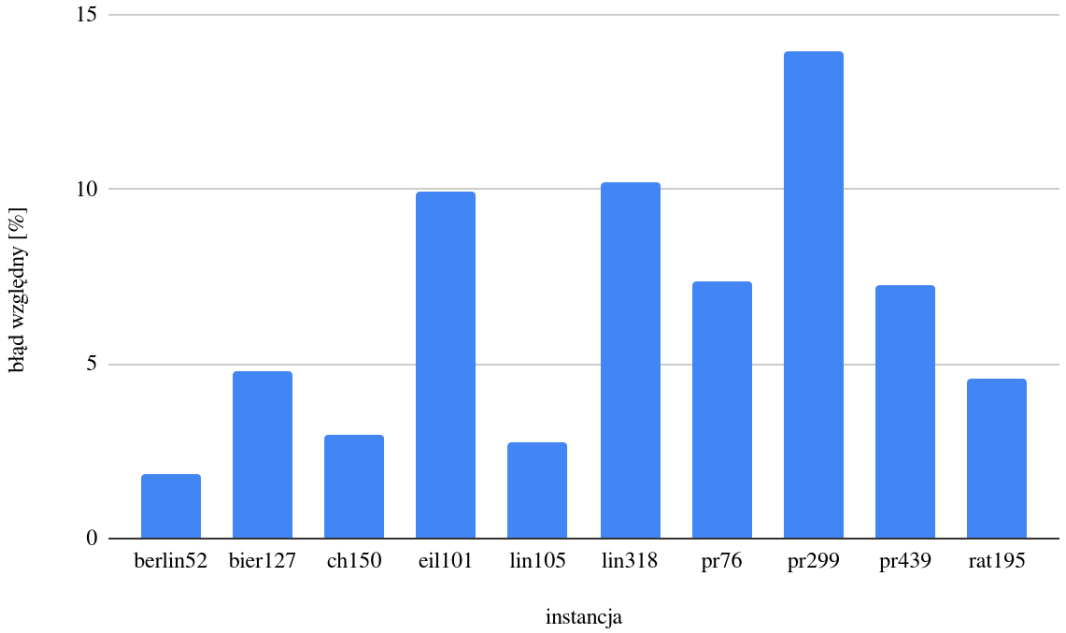
2.1 Porównanie z algorytmem zachłannym

Porównanie optymalizowanej wartości dla algorytmu Ant System z algorytmem zachłannym:



2.2 Wartości błędu do wartości optymalnej

Wykres wartości błędu względnego algorytmu Ant System w stosunku do wartości optymalnej:



2.3 Wyniki algorytmu dla instancji rankingowych

Nazwa instancji	Wynik
berlin52	7677
bier127	123903
tsp250	13414
tsp500	93377
tsp1000	27362