

# Zastosowanie algorytmu mrówkowego do rozwiązywania problemu Capacitated Vehicle Routing Problem

Raport końcowy

Jan Szablanowski

## Streszczenie

Repozytorium kodu zostało umieszczone na GitHub: <https://github.com/r-ost/MSI2-ants>

## Spis treści

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Opis problemu</b>                              | <b>2</b> |
| <b>2</b> | <b>Opis algorytmów</b>                            | <b>2</b> |
| 2.1      | Algorytm zachłanny . . . . .                      | 2        |
| 2.2      | Algorytm mrówkowy . . . . .                       | 2        |
| 2.3      | Algorytm mrówkowy z heurystyką 2-opt . . . . .    | 3        |
| 2.4      | Algorytm mrówkowy z modyfikacją Max-Min . . . . . | 3        |
| <b>3</b> | <b>Sprawdzone hipotezy</b>                        | <b>3</b> |
| 3.1      | Hipoteza 1 . . . . .                              | 4        |
| 3.2      | Hipoteza 2 . . . . .                              | 5        |
| 3.3      | Hipoteza 3 . . . . .                              | 6        |
| 3.4      | Hipoteza 4 . . . . .                              | 7        |
| <b>4</b> | <b>Wnioski</b>                                    | <b>7</b> |

# 1 Opis problemu

Capacitated Vehicle Routing Problem (CVRP) to problem optymalizacyjny, który jest rozszerzeniem problemu marszrutyzacji (Vehicle Routing Problem - VRP). W zadaniu VRP mamy dany nieskierowany graf pełny oraz ustaloną liczbę pojazdów. Wierzchołki grafu, poza jednym, oznaczają lokalizacje klientów na płaszczyźnie. Jeden wierzchołek jest wyszczególniony i oznacza magazyn. Krawędzie w grafie mają przypisane wagi, które oznaczają odległości między wierzchołkami. Celem problemu VRP jest odwiedzenie każdego klienta dokładnie raz, minimalizując całkowity koszt przebycia tras wszystkich pojazdów. Każda trasa zaczyna się i kończy w magazynie. W CVRP dodatkowo każdy klient ma określone zapotrzebowanie na jeden, określony produkt, a każdy pojazd może przewozić skończoną ilość produktu. W ramach projektu zakładamy dodatkowo, że droga, którą może przebyć każdy z pojazdów, jest ograniczona przez pewną stałą.

Problem CVRP jest NP-trudny [1]. W związku z tym dokładne rozwiązanie można wyznaczyć tylko dla małych instancji (rzędu kilku-kilkunastu wierzchołków). W praktyce problem CVRP rozwiązuje się, znajdując przybliżone rozwiązanie. W projekcie zostaną porównane cztery algorytmy znajdujące przybliżone rozwiązanie problemu CVRP:

1. Algorytm mrówkowy
2. Algorytm mrówkowy z heurystyką 2-opt
3. Algorytm mrówkowy z modyfikacją Max-Min (Max-Min ant system - MMAS)
4. Algorytm zachłanny

## 2 Opis algorytmów

### 2.1 Algorytm zachłanny

Algorytm zachłanny jest prostą heurystyką, którą można zastosować do problemu CVRP. Algorytm ten podejmuje decyzje lokalnie optymalne, licząc na uzyskanie globalnie optymalnego rozwiązania. W przypadku problemu CVRP, w każdym kroku algorytm wybiera najbliższego klienta, który jednocześnie spełnia ograniczenia pojemności pojazdu i limitu długości trasy. Procedura powtarzana jest dla kolejnych pojazdów, aż do obsłużenia wszystkich klientów.

Algorytm zachłanny podejmuje decyzje lokalnie bez uwzględnienia przyszłych konsekwencji swoich decyzji, co może prowadzić do rozwiązań słabej jakości względem algorytmu mrówkowego. Niewątpliwą zaletą algorytmu zachłannego jest za to łatwość implementacji i szybkość działania, co może mieć znaczenie przy złożonych problemach i dużych danych wejściowych.

### 2.2 Algorytm mrówkowy

Algorytm mrówkowy (Ant Colony Optimization - ACO) [2] to meta-heurystyczne podejście do rozwiązywania problemów trudnych obliczeniowo, które jest inspirowane zachowaniem kolonii mrówek. Mrówki podczas przemieszczania się pozostawiają po sobie ślady w postaci feromonów, którymi podążają inne mrówki. Z biegiem czasu najmocniejsze ślady powstają na najkrótszych ścieżkach, ponieważ mrówki mogą się nimi przemieszczać najszybciej.

Algorytm mrówkowy działa w iteracjach. Każda iteracja zawiera następujące kroki:

#### 1. Wygenerowanie pełnego rozwiązania przez każdą mrówkę na podstawie probabilistycznej funkcji przejścia

$$P_{ij} = \frac{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}{\sum_{k \in N_i} (\tau_{ik})^\alpha \cdot (\eta_{ik})^\beta} \quad (1)$$

gdzie:

$P_{ij}$  - prawdopodobieństwo przejścia z wierzchołka  $i$  do wierzchołka  $j$ ,

$\tau_{ij}$  - ilość feromonu na krawędzi  $i-j$ ,

$\eta_{ij}$  - heurystyka (odwrotność odległości między wierzchołkami  $i$  i  $j$ ),

$\alpha$  - waga feromonu,

$\beta$  - waga heurystyki,

$N_i$  - zbiór wierzchołków, do których mrówka może przejść z wierzchołka  $i$ .

2. Obliczenie jakości rozwiązania i zaktualizowania najlepszego dotychczasowego rozwiązania.

3. Zaktualizowanie śladów feromonowych, uwzględniając osłabianie się feromonów.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (2)$$

gdzie:

$\rho$  - współczynnik parowania feromonu,

$m$  - liczba mrówek,

$\Delta\tau_{ij}^k$  - ilość feromonu dodanego przez mrówkę  $k$  na krawędzi  $i-j$ .

Ilość feromonu dodanego przez mrówkę  $k$  na krawędzi  $i-j$  jest obliczana na podstawie jakości rozwiązania, które mrówka znalazła. Im lepsze rozwiązanie, tym więcej feromonu zostaje dodane do krawędzi.

$$\Delta\tau_{ij}^k = \frac{Q}{L_k} \quad (3)$$

gdzie:

$Q$  - stała,

$L_k$  - długość trasy mrówki  $k$ .

### 2.3 Algorytm mrówkowy z heurystyką 2-opt

Heurystyka 2-opt jest lokalną heurystyką, która polega na poprawie jakości rozwiązania poprzez wybór dwóch wierzchołków na trasie o indeksach  $i$ ,  $j$  i usunięcie krawędzi  $i-1-i$  oraz  $j-j+1$ . Następnie krawędzie między wierzchołkami  $i$  i  $j$  są dodawane w odwrotnej kolejności. Na koniec dodawane są krawędzie  $i-1-j$  oraz  $i-j+1$ .

Heurystyka 2-opt pozwala na usunięcie przecięcia tras, co prowadzi do skrócenia całkowitej długości trasy. Heurystyka ta jest stosowana po wygenerowaniu trasy przez algorytm mrówkowy. Algorytm mrówkowy z heurystyką 2-opt działa w następujący sposób [3]:

1. Algorytm mrówkowy generuje trasę dla każdego pojazdu.
2. Dla każdej trasy algorytm mrówkowy stosuje heurystykę 2-opt, aby poprawić jakość rozwiązania.
3. Algorytm aktualizuje feromony na krawędziach zgodnie z jakością rozwiązania.

Heurystyka 2-opt jest stosunkowo prosta do zaimplementowania i może znacznie poprawić jakość rozwiązań. W przypadku problemu CVRP, heurystyka 2-opt może być szczególnie skuteczna, ponieważ pozwala na poprawę jakości tras, które są już bliskie optymalnym.

### 2.4 Algorytm mrówkowy z modyfikacją Max-Min

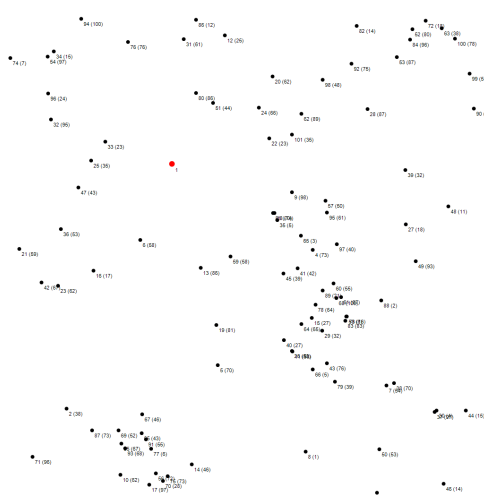
Algorytm mrówkowy z modyfikacją Max-Min (Max-Min Ant System - MMAS) [4] to rozwinięcie klasycznego algorytmu mrówkowego, które wprowadza zmiany w sposobie aktualizacji feromonu. W MMAS feromon na krawędzi jest aktualizowany tylko przez najlepszą mrówkę w danej iteracji lub globalnie. Zwiększa to koncentrację feromonu na najlepszych krawędziach, co prowadzi do szybszej konwergencji algorytmu i większej eksploatacji. Dodatkowo, w MMAS wprowadza się ograniczenia na minimalną i maksymalną ilość feromonu na krawędziach oraz maksymalną liczbę iteracji bez poprawy, po której algorytm resetuje feromony. Dzięki temu algorytm jest bardziej stabilny i mniej podatny na stagnację.

## 3 Sprawdzone hipotezy

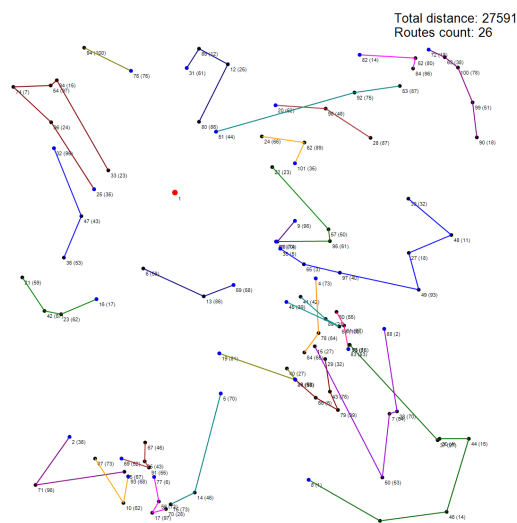
W projekcie postawiono cztery hipotezy, których celem było porównanie jakości rozwiązań uzyskiwanych przez różne algorytmy. Hipotezy były testowane na zbiorze Uchoa et al. (2014) [5] z bazy CVRLIB (Capacitated Vehicle Routing Problem Library), która zawiera 100 różnorodnych instancji problemu, w których występuje od 100 do 1000 klientów. W testowaniu hipotez wykorzystano 6 instancji problemu CVRP o różnych rozmiarach i rozkładach klientów. Przykładowa instancja została pokazana na rysunku 1, a jej rozwiązanie na rysunku 2.

Algorytmy mrówkowe wraz z modyfikacjami 2-opt i Max-Min były uruchamiane 3 razy na każdej instancji z różnymi ziarnami (wyniki były potem uśredniane). Parametry algorytmu mrówkowego i liczba

iteracji były takie same dla wszystkich instancji z wyjątkiem algorytmu mrówkowego z modyfikacją Max-Min, w którym początkową liczbę feromonów ustawiono jako maksymalną liczbę feromonu. Wartości parametrów algorytmu mrówkowego były ustalane na podstawie literatury i własnych eksperymentów i przedstawiono je w tabeli 1.



Rysunek 1: Instancja X-n101-k25 – wierzchołki oznaczono jako: IdKlienta(Zapotrzebowanie). Magazyn oznaczono na czerwono



Rysunek 2: Optymalne rozwiązanie instancji X-n101-k25

| Parametr                                   | Wartość                             |
|--|-------------------------------------|
| Liczba mrówek                              | Równa liczbie wierzchołków w grafie |
| Liczba iteracji                            | 100                                 |
| Współczynnik parowania feromonu ( $\rho$ ) | 0.2                                 |
| Waga feromonu ( $\alpha$ )                 | 1                                   |
| Waga heurystyki ( $\beta$ )                | 5                                   |
| Ilość dodawanego feromonu ( $Q$ )          | 10                                  |
| Początkowa ilość feromonu                  | 0.1                                 |

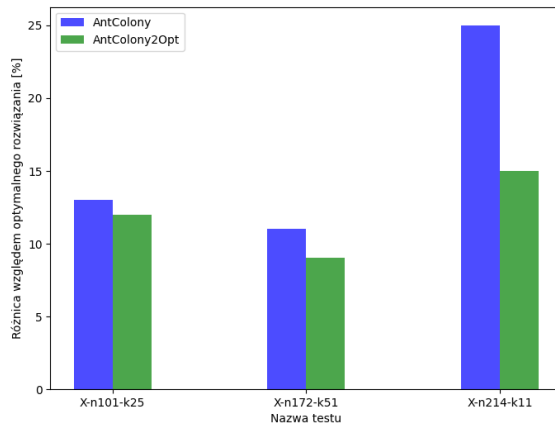
Tabela 1: Parametry algorytmu mrówkowego

### 3.1 Hipoteza 1

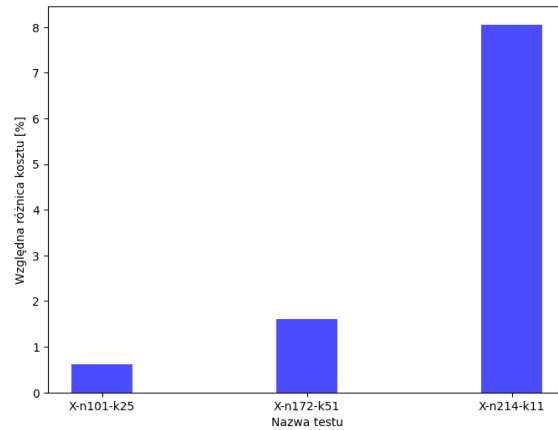
*Algorytm mrówkowy z heurystyką 2-opt znajduje trasy o co najmniej 10% mniejszym koszcie w porównaniu do klasycznego algorytmu mrówkowego.*

**Wynik:** Częściowo prawdziwa dla dużych instancji.

Celem tego eksperymentu było sprawdzenie, czy heurystyka 2-opt rzeczywiście poprawia jakość rozwiązań uzyskiwanych przez algorytm mrówkowy. W tym celu porównano wyniki uzyskane przez algorytm mrówkowy z heurystyką 2-opt z wynikami uzyskanymi przez klasyczny algorytm mrówkowy.



Rysunek 3: Długość rozwiązania ACO i ACO 2-opt względem optymalnego



Rysunek 4: O ile procent AntColony2Opt jest lepszy od AntColony?

Z rysunku 3 widać, że dla wszystkich instancji testowych algorytm mrówkowy z heurystyką 2-opt znalazł rozwiązania o niższym koszcie niż klasyczny algorytm mrówkowy. Rysunek 3.1 pokazuje, że względna różnica w długościach rozwiązań jest mniejsza niż 10%. W przypadku instancji o największej liczbie klientów (214) różnica ta wynosi około 8%, natomiast w przypadku instancji o najmniejszej liczbie klientów (101) różnica ta wynosi mniej niż 1%. Widać z tego, że różnica w długości rozwiązań rośnie wraz ze wzrostem liczby wierzchołków i hipoteza ma szansę być prawdziwa dla dużych instancji problemu CVRP.

## 3.2 Hipoteza 2

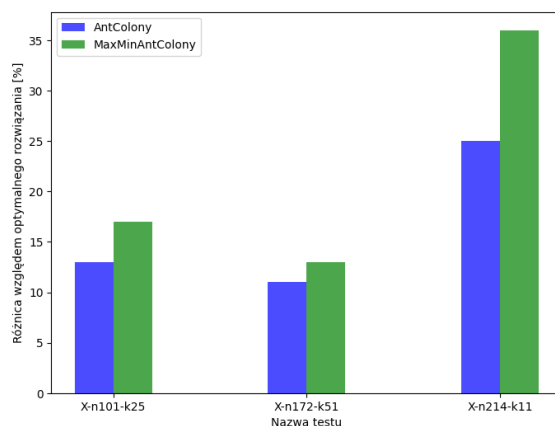
*Algorytm mrówkowy z modyfikacją Max-Min znajduje rozwiązanie o koszcie nie większym niż 5% od kosztu rozwiązania znalezione przez klasyczny algorytm mrówkowy, ale potrzebuje do tego o 10% mniejszej liczby pojazdów.*

**Wynik: Fałszywa.**

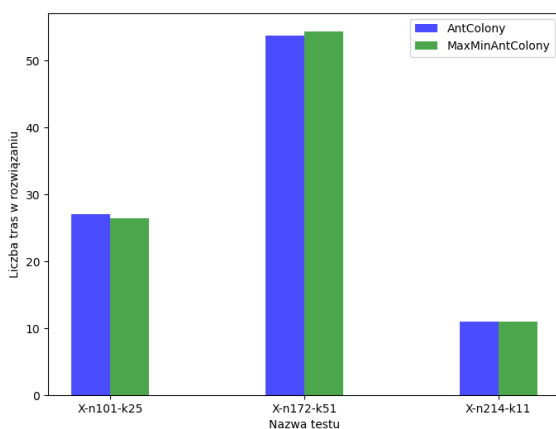
Celem tego eksperymentu było sprawdzenie, czy algorytm mrówkowy z modyfikacją Max-Min znajduje rozwiązania o większej jakości niż klasyczny algorytm mrówkowy. Jakość rozwiązania jest tutaj rozumiana jako poziom wykorzystania pojemności pojazdów i wiąże się z mniejszą liczbą pojazdów potrzebnych do obsłużenia wszystkich klientów. Użyte wartości parametrów związanych z modyfikacją Max-Min przedstawiono w tabeli 2.

| Parametr                               | Wartość |
|--|---------|
| Minimalna ilość feromonu               | 0.01    |
| Maksymalna ilość feromonu              | 10      |
| Początkowa ilość feromonu              | 10      |
| Maksymalna liczba iteracji bez poprawy | 20      |

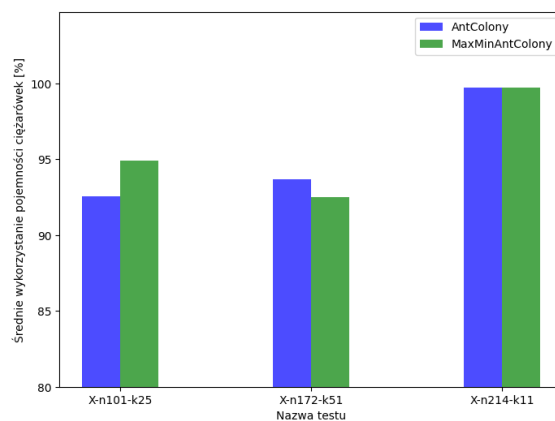
Tabela 2: Dodatkowe parametry algorytmu Max-Min



Rysunek 5: Długość rozwiązania ACO i MMAS względem optymalnego



Rysunek 6: Liczba tras dla ACO i MMAS



Rysunek 7: Średnie wykorzystanie pojemności pojazdów na trasach

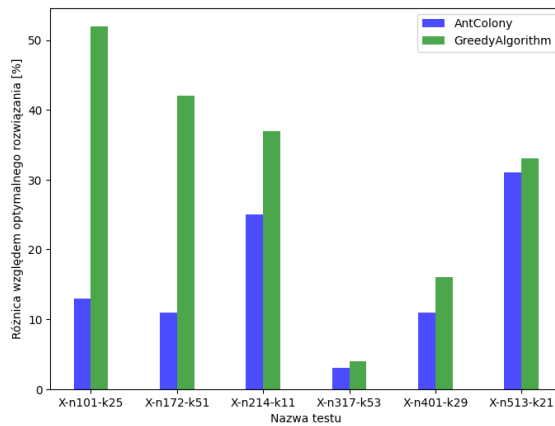
Na rysunku 5 widać, że algorytm mrówkowy z modyfikacją Max-Min znajduje rozwiązania o większym koszcie niż klasyczny algorytm mrówkowy. Różnica ta jest rzędu kilku-kilkunastu procent, co było spodziewane przy formułowaniu hipotezy. Z rysunku 6 widać jednak, że algorytm mrówkowy z modyfikacją Max-Min potrzebuje porównywalnej liczby tras jak klasyczny algorytm mrówkowy i zależy to od problemu. Podobnie jest ze średnim wykorzystaniem pojemności pojazdów na trasach, które jest porównywalne dla obu algorytmów (rysunek 7). Wynik eksperymentu, który obala hipotezę, może wynikać z nieprawidłowego doboru parametrów albo zbyt małej liczby iteracji w testach algorytmu mrówkowego z modyfikacją Max-Min.

### 3.3 Hipoteza 3

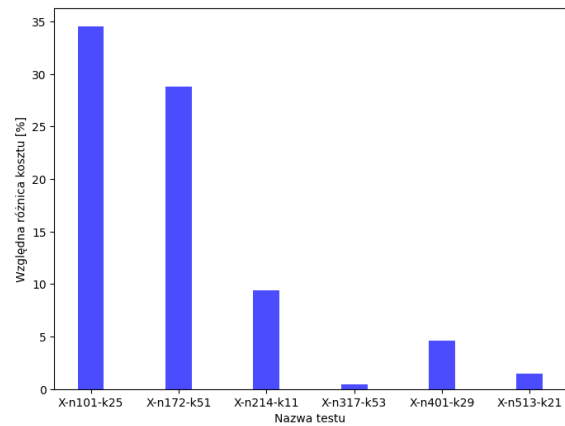
*Procentowa różnica kosztów wyznaczonych tras między algorytmem zachłannym i klasycznym algorytmem mrówkowym rośnie wraz ze wzrostem liczby klientów na niekorzyść algorytmu zachłannego.*

**Wynik: Potwierdzona hipoteza przeciwna.**

Celem eksperymentu było sprawdzenie, czy różnica w kosztach wyznaczonych tras między algorytmem zachłannym i klasycznym algorytmem mrówkowym rośnie wraz ze wzrostem liczby klientów. Hipoteza wynikała z przypuszczenia, że algorytm zachłanny nie jest w stanie znaleźć dobrego rozwiązania dla dużych instancji problemu CVRP, ponieważ podejmuje decyzje lokalnie optymalne.



Rysunek 8: Długość rozwiązania ACO i Greedy względem optymalnego



Rysunek 9: O ile procent ACO jest lepszy od Greedy?

Na rysunku 8 widać, że algorytm ACO znajduje rozwiązania o mniejszym koszcie niż algorytm zachłanny dla każdej instancji testowej, co było spodziewane. Jednakże na rysunku 9 widać, że różnica w długości rozwiązań maleje wraz ze wzrostem liczby klientów na korzyść algorytmu zachłannego, co potwierdza hipotezę odwrotną do postawionej (liczba klientów ujęta jest w nazwie instancji po literze n).

### 3.4 Hipoteza 4

*Procentowa różnica kosztów wyznaczonych tras między algorytmem mrówkowym z heurystyką 2-opt i klasycznym algorytmem mrówkowym rośnie wraz ze wzrostem rozproszenia klientów (mierzonego średnią odległością klientów od magazynu) na niekorzyść klasycznego algorytmu.*

## 4 Wnioski

### Literatura

- [1] Jan Lenstra and A. Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11:221 – 227, 10 2006.
- [2] Marco Dorigo, Mauro Birattari, and Thomas Stützle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1:28–39, 12 2006.
- [3] W.F. Tan, Lai Soon Lee, Zanariah Abdul Majid, and Hsin-Vonn Seow. Ant colony optimization for capacitated vehicle routing problem. *Journal of Computer Science*, 8:846–852, 01 2012.
- [4] Thomas Stützle and Holger Hoos. Max-min ant system. 16, 11 1999.
- [5] Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Thibaut Vidal, and Anand Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257, 08 2016.