

Zastosowanie algorytmu mrówkowego do rozwiązania problemu Capacitated Vehicle Routing Problem

Raport końcowy

Jan Szablanowski

Streszczenie

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) 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 ,

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

η_{ij} - heurystyka (odwrotność odległości między wierzchołkami i i j),

α - waga feromonu,

β - 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:

ρ - 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:

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) 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

3.1 Hipoteza 1

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

3.2 Hipoteza 2

Algorytm mrówkowy z modyfikacją Max-Min uzyska tak samo dobre rozwiązania jak klasyczny algorytm mrówkowy, ale będzie potrzebował do tego mniejszej liczby pojazdów.

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.

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] Marco Dorigo, Mauro Birattari, and Thomas Stützle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1:28–39, 12 2006.
- [2] 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.
- [3] Thomas Stützle and Holger Hoos. Max-min ant system. 16, 11 1999.
- [4] 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.