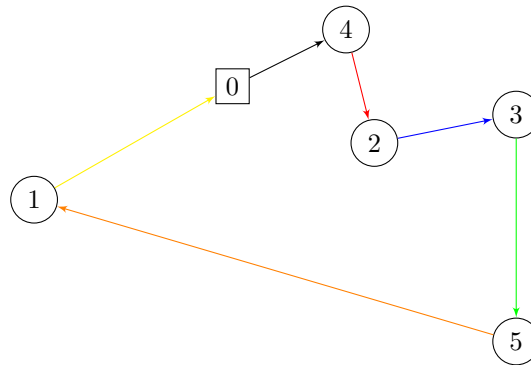


Przemysł 4.0 - Laboratorium
VRP – algorytm zachłanny

prowadzący: mgr inż. Radosław Idzikowski

1 Wprowadzenie

Celem laboratorium jest zapoznanie się z algorytmem zachłannym dla problemu marszrutyzacji pojazdów. Algorytm zachłanny na każdym kroku wykonuje lokalnie optymalny ruch, ale w ostateczności nie mamy gwarancji znalezienia optymalnego rozwiązania. Największą zaletą działania algorytmu jest krótki czas wykonywania. Idea zaproponowanego algorytmu będzie modyfikacją algorytmu najbliższego sąsiada dla problemu TSP.



Rysunek 1: Algorytm najbliższego sąsiada

2 Algorytm dla VRP

Dostosowanie algorytmu najbliższego sąsiada dla problemu marszrutyzacji pojazdów polega do dodaniu dwóch modyfikacji. Po pierwsze algorytm będzie działał osobno dla każdego pojazdu. Na początku należy wybrać najdalej położone miasto względem magazynu centralnego. Następnie dodawać najbliższe miasta dopóki ograniczenia dla jednego pojazdu są spełnione. Dla podstawowego problemu VRP możemy przyjąć dwa proste ograniczenia:

- stała liczba zleceń dla pojazdu (n/\mathcal{K}),
- jeśli kolejny najbliższy sąsiad jest dalej niż np.: $2/3$ odległości z magazynu do pierwszego miasta.

W przypadku CVRP sytuacja jest bardziej naturalna, ponieważ ograniczeniem jest ładowność pojazdu.

3 Zadanie

Należy przeprowadzić wstępne testy na mapie polski, gdzie najlepiej umiejscowić magazyn centralny. Dla uproszczenia obliczeń, przy sprawdzaniu konkretnego miasta należy je pominąć liście klientów do odwiedzenia.

Algorithm 1 Algorytm zachłanny dla VRP

```
1: procedure GREEDY( $\mathcal{D}, \mathcal{N}$ )
2:    $\mathcal{A} \leftarrow \mathcal{N}$  ▷ gdzie  $\mathcal{A}$  to zbiór dostępnych miast
3:    $\pi.$ CLEAR()
4:    $k \leftarrow 1$  ▷ gdzie  $k$  to numer ciężarówki
5:   while  $\mathcal{A} \neq \emptyset$  do
6:      $\pi.$ PUSHBACK(0)
7:      $j^* \leftarrow \operatorname{argmax}_{j \in \mathcal{A}} d_{0j}$ 
8:      $\pi.$ PUSHBACK( $j^*$ )
9:      $\mathcal{A} \leftarrow \mathcal{A} \setminus \{j^*\}$ 
10:     $x_k \leftarrow 1$  ▷ gdzie  $x_k$  to liczba zleceń obsłużonych przez pojazd  $k$ 
11:    while  $limit = true$  do
12:       $l^* \leftarrow \operatorname{argmin}_{l \in \mathcal{A}} d_{j^*l}$ 
13:      if  $x_k + 1 < \frac{n}{K}$  then
14:         $\pi.$ PUSHBACK( $l^*$ )
15:         $\mathcal{A} \leftarrow \mathcal{A} \setminus \{l^*\}$ 
16:         $x_k \leftarrow x_k + 1$ 
17:      else
18:         $limit \leftarrow false$ 
19:      end if
20:    end while
21:     $k \leftarrow k + 1$ 
22:  end while
23: end procedure
```

Algorithm 2 Algorytm zachłanny dla CVRP

```
1: procedure GREEDY( $\mathcal{D}, \mathcal{N}$ )
2:    $\mathcal{A} \leftarrow \mathcal{N}$  ▷ gdzie  $\mathcal{A}$  to zbiór dostępnych miast
3:    $\pi.$ CLEAR()
4:    $k \leftarrow 1$  ▷ gdzie  $k$  to numer ciężarówki
5:   while  $\mathcal{A} \neq \emptyset$  do
6:      $\pi.$ PUSHBACK(0)
7:      $j^* \leftarrow \operatorname{argmax}_{j \in \mathcal{A}} d_{0j}$ 
8:      $\pi.$ PUSHBACK( $j^*$ )
9:      $\mathcal{A} \leftarrow \mathcal{A} \setminus \{j^*\}$ 
10:     $weight_k \leftarrow w_{j^*}$  ▷ gdzie  $weight_k$  to suma masy ładunków dla pojazdu  $k$ 
11:     $size_k \leftarrow s_{j^*}$  ▷ gdzie  $weight_k$  to suma długości ładunków dla pojazdu  $k$ 
12:    while  $limit = true$  do
13:       $l^* \leftarrow \operatorname{argmin}_{l \in \mathcal{A}} d_{j^*l}$ 
14:      if  $weight_k + w_{l^*} < maxWeight$  and  $size_k + s_{l^*} < maxSize$  then
15:         $\pi.$ PUSHBACK( $l^*$ )
16:         $\mathcal{A} \leftarrow \mathcal{A} \setminus \{l^*\}$ 
17:         $weight_k \leftarrow weight_k + w_{l^*}$ 
18:         $size_k \leftarrow size_k + s_{l^*}$ 
19:      else
20:         $limit \leftarrow false$ 
21:      end if
22:    end while
23:     $k \leftarrow k + 1$ 
24:  end while
25: end procedure
```
