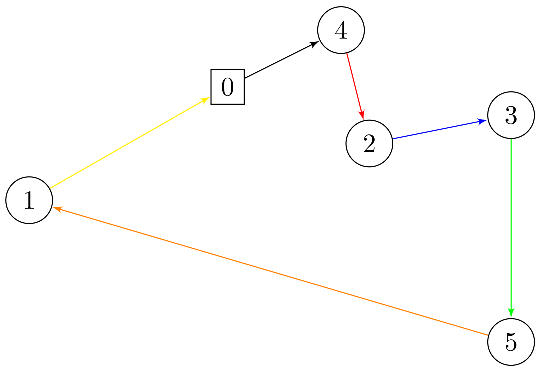
Wrocław, 20 października 2020

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 rozwiązana. Największą zaletą działania algorytmu jest krótki czas wykonywania. Idea zaproponowanego algorytmu będzie modyfikacja 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

procedure GREEDY($\mathcal{D}$,$\mathcal{N}$)
$\mathcal{A} \leftarrow \mathcal{N}$ ▷ gdzie $\mathcal{A}$ to zbiór dostępnych miast
π.CLEAR()
$k \leftarrow 1$ ▷ gdzie k to numer ciężarówki
while $\mathcal{A} \neq \emptyset$ **do**
π.PUSHBACK(0)
$j^* \leftarrow \underset{j \in \mathcal{A}}{\operatorname{argmax}} d_{0j}$
π.PUSHBACK(j^*)
$\mathcal{A} \leftarrow \mathcal{A} \setminus \{j^*\}$
$x_k \leftarrow 1$ ▷ gdzie x_k to liczba zleceń obsłużonych przez pojazd k
while $limit = true$ **do**
$l^* \leftarrow \underset{l \in \mathcal{A}}{\operatorname{argmin}} d_{j^*l}$
if $x - K + 1 < \frac{n}{\mathcal{K}}$ **then**
π.PUSHBACK(l^*)
$\mathcal{A} \leftarrow \mathcal{A} \setminus \{l^*\}$
$x_k \leftarrow x_k + 1$
else
$limit \leftarrow false$
end if
end while
$k \leftarrow k + 1$
end while
end procedure

Algorithm 2 Algorytm zachłanny dla CVRP

procedure GREEDY($\mathcal{D}$,$\mathcal{N}$)
$\mathcal{A} \leftarrow \mathcal{N}$ ▷ gdzie $\mathcal{A}$ to zbiór dostępnych miast
π.CLEAR()
$k \leftarrow 1$ ▷ gdzie k to numer ciężarówki
while $\mathcal{A} \neq \emptyset$ **do**
π.PUSHBACK(0)
$j^* \leftarrow \underset{j \in \mathcal{A}}{\operatorname{argmax}} d_{0j}$
π.PUSHBACK(j^*)
$\mathcal{A} \leftarrow \mathcal{A} \setminus \{j^*\}$
$weight_k \leftarrow w_{j^*}$ ▷ gdzie $weight_k$ to suma masy ładunków dla pojazdu k
$size_k \leftarrow s_{j^*}$ ▷ gdzie $weight_k$ to suma długości ładunków dla pojazdu k
while $limit = true$ **do**
$l^* \leftarrow \underset{l \in \mathcal{A}}{\operatorname{argmin}} d_{j^*l}$
if $weight_k + w_{l^*} < maxWeight$ or $size_k + s_{l^*} < maxSize$ **then**
π.PUSHBACK(l^*)
$\mathcal{A} \leftarrow \mathcal{A} \setminus \{l^*\}$
$weight_k \leftarrow weight_k + w_{l^*}$
$size_k \leftarrow size_k + s_{l^*}$
else
$limit \leftarrow false$
end if
end while
$k \leftarrow k + 1$
end while
end procedure