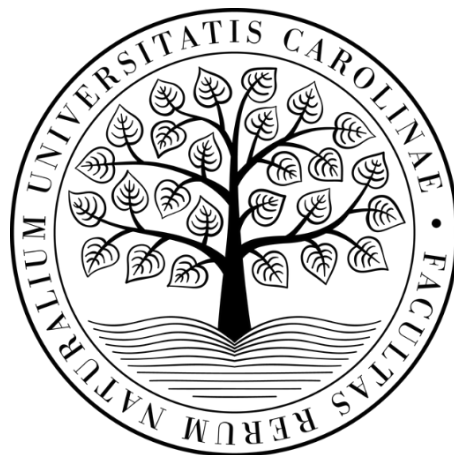


Univerzita Karlova  
Přírodovědecká fakulta



# Problém obchodního cestujícího

**Geoinformatika**

Vanda HLAVÁČKOVÁ

N-KGDPZ, 1. ročník

Praha 2023

## Zadání

*Vstup: množina uzlů  $U$  reprezentujících body.*

*Výstup: nalezení nejkratší Hamiltonovské kružnice mezi těmito uzly.*

Nad množinou  $U$  nalezněte nejkratší cestu, která vychází z libovolného uzlu, každý z uzlů navštíví pouze jedenkrát, a vrací se do uzlu výchozího. Využijte níže uvedené metody konstrukčních heuristik:

- Nearest Neighbor,
- Best Insertion.

Výsledky porovnejte s výstupem poskytovaným nástrojem Network Analyst v SW ArcMap. Otestování proveďte nad dvěma zvolenými datasety, které by měly obsahovat alespoň 100 uzlů. Jako vstup použijte existující geografická data (např. města v ČR s více než 10 000 obyvateli, evropská letiště, ...), ohodnocení hran bude představovat vzdálenost mezi uzly (popř. vzdálenost měřenou po silnici); pro tyto účely použijte vhodný GIS.

Výsledky s uvedením hodnot  $W$ ,  $k$ , uspořádejte do přehledné tabulky (metodu Best Insertion nechte proběhnout alespoň 10x), a zhodnoťte je.

Pro implementaci obou konstrukčních heuristik použijte programovací jazyk Python, vizualizaci výstupů proveďte ve vhodné knihovně, např. matplotlib.

## Zpracování

### Popis problému a výsledky

*Travelling Salesman Problem* (TSP), tedy problém obchodního cestujícího spadá mezi úlohy, jejichž řešení jsme schopni najít pouze přibližně s akceptovatelnou kvalitou – tzv. NP-úplné úlohy. Principem úlohy problému obchodního cestujícího je nalezení nejkratší (popř. nejrychlejší apod.) možné cesty mezi danými body tak, aby každý bod byl navštíven právě jednou, přičemž koncový bod je shodný s počátečním.

Úloha byla implementována pomocí programovacího jazyka Python, kde byly zpracovány dvě metody, a to Nearest Neighbor a Best Insertion. Následně byly výsledky porovnány s výstupem nástroje *Network Analyst*, který nabízí software ArcGIS Pro.

Metoda **Nearest Neighbor** je nejjednodušší z možností, jak přistupovat k problematice TSP. Nejprve jsou všechny vstupní body označeny jako nezpracované a je vybrán/zadán počáteční bod. U něj se změní stav na zpracovaný. Dalším bodem, který bude zpracováván, je pak ten, ke kterému má počáteční bod nejkratší vzdálenost. Tento proces probíhá do té doby, dokud není status všech bodů změněn na zpracovaný, čímž dojde k vytvoření Hamiltonovské kružnice – uzavření cyklu.

#### **Pseudokód** metody Nearest Neighbor

Nastav status všem bodům na N

Vytvoř seznam pro ukládání výsledků a inicializuj hodnotu W

Vyber a nastav počáteční bod, změň jeho status a přidej ho do cyklu

Dokud existují nezpracované body

Inicializuj minimální vzdálenost na nekonečno

Pro nezpracované uzly

Vypočítej vzdálenost

Pokud je vzdálenost menší než minimální vzdálenost

Nastav hodnotu minimální vzdálenosti na aktuální vzdálenost

Ulož uzel a jeho index

Přidej uzel do cyklu, nastav ho jako aktuální a změň jeho status

K délce kružnice W přičti minimální vzdálenost

Uzavři kružnici – spočítej vzdálenost mezi počátečním a koncovým bodem a přičti ji k celkové délce kružnice W

Přidej počáteční uzel do cyklu

Vrať výslednou kružnici a její délku W

**Best Insertion** představuje upravenou verzi Nearest Neighbor. Počáteční kružnice je vytvořena pomocí tří bodů. Další body jsou do kružnice přidávány tak, aby bylo navýšení cesty (tedy Hamiltonovské kružnice) minimální. Dalším rozdílem oproti Nearest Neighbor je fakt, že koncový bod není totožný se startovním, ale je místo toho přidán na libovolné místo do kružnice.

#### **Pseudokód** metody Best Insertion

Nastav status všem bodům na N

Vytvoř seznam pro ukládání výsledků a inicializuj hodnotu W

Vyber náhodné tři body ze vstupních dat

Pro startovní uzly

Změň jejich status a přidej je do kružnice

Pro startovní uzly (opakuj 3x)

Vypočítej jejich vzdálenost a přičti ji k délce kružnice

Dokud existují nezpracované body

Inicializuj minimální vzdálenost na nekonečno

Vyber náhodný bod – aktuální

Pro uzly v Hamiltonovské kružnici

Vypočítej vzdálenost pro aktuální uzel

Pokud je vypočítaná vzdálenost menší, než minimální vzdálenost

Nastav hodnotu minimální vzdálenosti na aktuální vzdálenost

Ulož si aktuální uzel

Přidej uzel do kružnice a přičti k délce kružnice minimální vzdálenost

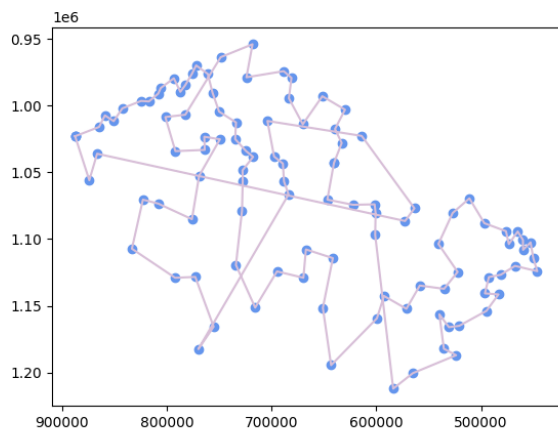
Změň status aktuálního uzlu

Přidej startovní bod na konec kružnice

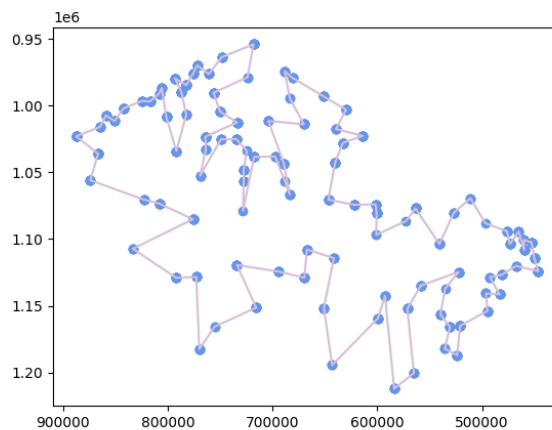
Vrať Hamiltonovskou kružnici a její délku

Problematiku obchodního cestujícího je možné řešit také v geoinformačních systémech jako je například **ArcGIS Pro**. Pomocí *Network Analyst* je možné provádět síťové analýzy. Pro testování vytvořených algoritmů byly použity dva vstupní datasety. Prvním je výběr obcí České republiky s počtem obyvatel vyšším, než je 12 tis. Druhým pak železniční stanice ve Středočeském kraji. Výsledky pro algoritmus Best Insertion proběhly dle zadání 10 a byla vybrána nejnižší hodnota (viz tabulka 1).

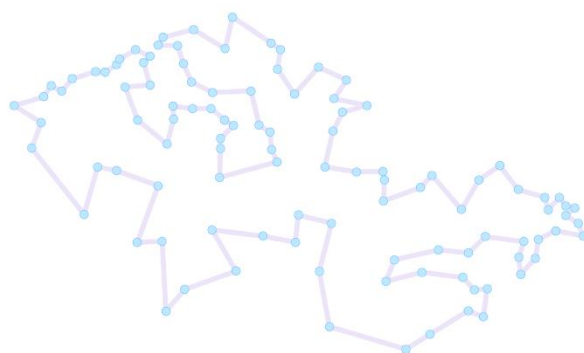
Celkové porovnání výsledků z algoritmu Nearest Neighbor, Best Insertion a výsledků ze softwaru ArcGIS Pro znázorňuje tabulka 2. Vizualizace výsledných kružnic byla provedena pomocí knihovny matplotlib – na obrázcích 1, 2 a 4, 5. Výsledky z ArcGIS Pro pak na obrázcích 3 a 6.



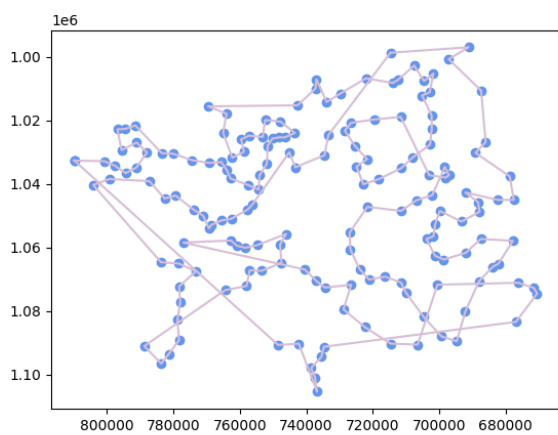
*Obrázek 1: Nearest Neighbor pro dataset obce\_12.csv*



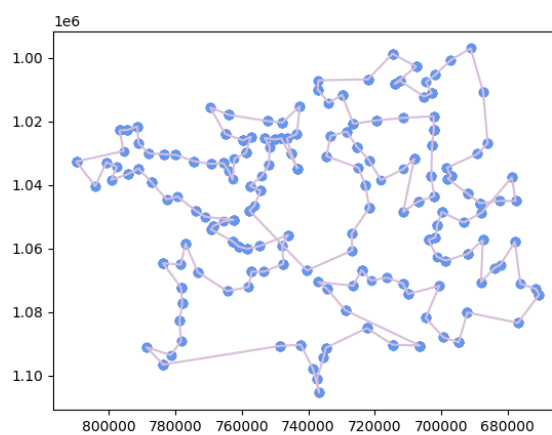
*Obrázek 2: Best Insertion pro dataset obce\_12.csv*



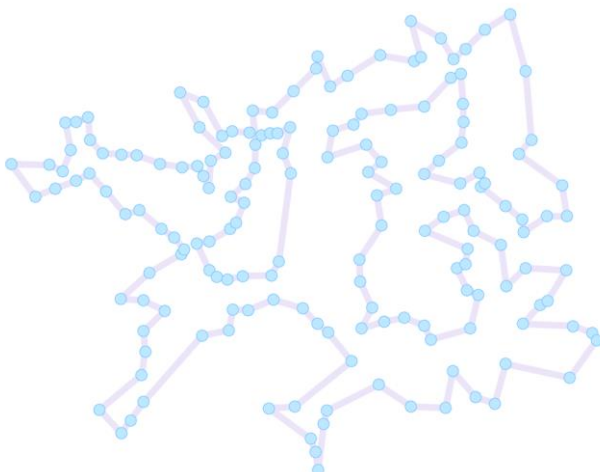
*Obrázek 3: Network Analysis v sw ArcGIS Pro pro dataset obce\_12.csv*



*Obrázek 4: Nearest Neighbor pro dataset zeleznice.csv*



*Obrázek 5: Best Insertion pro dataset zeleznice.csv*



*Obrázek 6: : Network Analysis v sw ArcGIS Pro pro dataset zeleznice.csv*

opakování	obce_12.csv	zeleznice.csv
1.	1 967 987,579	993 728,353
2.	<b>1 770 305,630</b>	<b>901 281,423</b>
3.	2 018 643,113	1 018 759,495
4.	2 037 052,930	1 059 751,442
5.	2 092 729,053	1 012 045,505
6.	1 952 448,962	989 922,151
7.	1 999 030,398	970 942,549
8.	1 853 908,979	1 000 121,918
9.	1 955 355,647	995 020,732
10.	2 073 214,960	1 047 004,610

Tabulka č. 1: Výsledky algoritmu Best Insertion pro oba datasety

	Nearest Neighbor	Best Insertion	ArcGIS Pro
zeleznice.csv	993 728,353	901 281,423	1 055 102,114
obce_12.csv	2 782 319,581	1 770 305,630	2 261 089,121

Tabulka č. 2: Celkové porovnání výsledků pro oba datasety

## Závěr

V tomto úkolu byly použity dva algoritmy pro řešení problému obchodního cestujícího a následně otestovány pomocí dvou datasetů. První dataset obsahující železniční zastávky obsahoval 168 bodů, druhý s obcemi s populací vyšší než 12 tis. obyvatel pak bodů 107. Z výsledků nejlépe vyplývá metoda Best Insertion. Výsledky algoritmu Nearest Neighbor jsou oproti Best Insertion výrazně horší.

## Zdroje

BAYER, T. (2022): Problém obchodního cestujícího, konstrukční heuristiky: stručný návod na cvičení, výukový materiál k předmětu Geoinformatika, [https://agony.natur.cuni.cz/~bayertom/images/courses/Geoinf/tsp\\_uloha.pdf](https://agony.natur.cuni.cz/~bayertom/images/courses/Geoinf/tsp_uloha.pdf) (05.01.2023).

BAYER, T. (2022): Úvod do grafových algoritmů, prezentace k předmětu Geoinformatika, [https://agony.natur.cuni.cz/~bayertom/images/courses/Geoinf/geoinf\\_grafy1.pdf](https://agony.natur.cuni.cz/~bayertom/images/courses/Geoinf/geoinf_grafy1.pdf) (05.01.2023).