Univerzita Karlova

Přírodovědecká fakulta



**Řešení problému obchodního cestujícího**

**Geoinformatika – úloha 1**

Kateřina Obrazová

1. ročník, N-GKDPZ

Praha 2021

**1 Zadání**

**Řešení problému obchodního cestující**

Vstup: množina uzlů *U* reprezentující body

Výstup: nalezení nejkratší cesty 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 (obě metody nechte proběhnout alespoň 10x) a zhodnoťte je.

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

Čas zpracování: 3 týdny

**2 Popis a rozbor problému**

**2.1 Problém obchodního cestujícího (TSP)**

Problém obchodního cestujícího (z angl. Travelling Salesman Problem) je jedním z malého souboru matematických problémů, který zachycuje představivost nejen specialisty, ale i laika, který se o něj zajímá. Může být použit jako klíčící bod pro vyšetřování na téměř jakékoli úrovni matematického vývoje. Krátce řečeno, problém spočívá v jeho základní podobě najít nejkratší cestu pro cestovatele, který musí navštívit všechna z určeného počtu center, kde jsou známé vzdálenosti mezi středisky. (Lawler a kol. 1985)

Lze jej pospat následující způsobem: Nalezněte nejkratší cestu pro obchodníka, který začíná z daného města. Obchodník navštíví každé město pouze jednou a poté se vrátí do místa, odkud začínal. Obecně je dána symetrická matice D = (dij), kdy dij představuje „vzdálenost“ od i do j, kdy body budou uspořádány do cyklického pořadí tak, aby součet dji mezi po sobě jdoucími body byl minimální. (Dantzig a kol. 1954)

**2.2 TSP v teorii grafů**

Na začátek je potřeba si připomenout definice, které napomohou k řešení problému obchodního cestujícího.

***Graf***

Definice*: „Jednoduchý graf G je uspořádaná dvojice (V; E), kde V je neprázdná množina vrcholů a E je nějaká množina dvouprvkových podmnožin množiny V. Prvkům E říkáme hrany.“ (Kolář 2013, s. 38)*

**Neorientový graf**

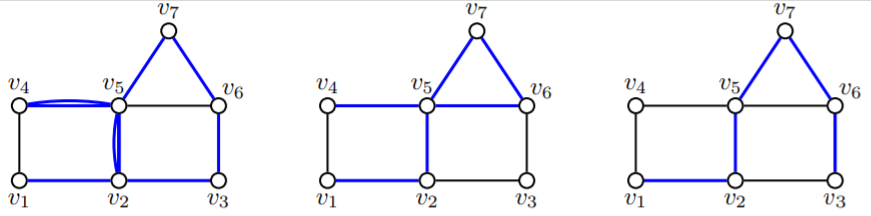
Definice: *Neorientovaný graf G je tvořený uspořádanou trojicí G = 〈H, U, ρ〉, kdy H jsou hrany grafu G, U jsou uzly grafu G a ρ je incidence grafu ρ: H → U ⊗ U, které přiřazují každé hraně h dvojici uzlů. (Kolář 2013)*

**Sled**

*Definice: Sled v grafu G je taková posloupnost vrcholů a hran (v0, e1, v1, e2, v2, . . ., en, vn), že hrana ei má koncové vrcholy vi−1 a vi pro všechna i = 1, 2, . . ., n. Sled se nazývá (v0, vn)-sled. (Kolář 2016, s. 56) (Obr.1)*

**Cesta grafem**

Definice: „*Cesta je sled, ve kterém se neopakují vrcholy.“ (Kolář 2013, s. 56) (Obr.1)*



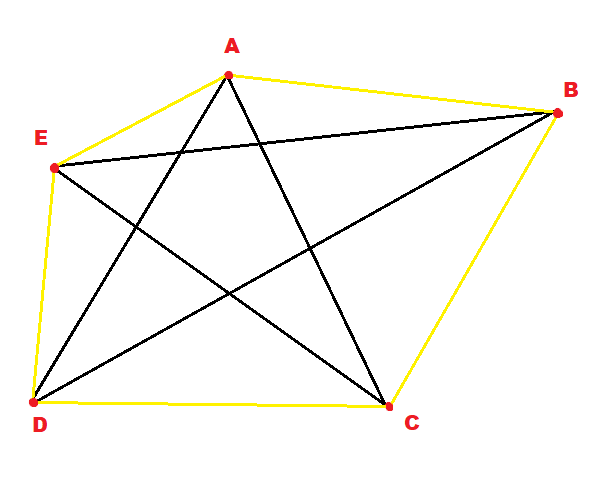
*Obr. 1 – Příklad sledu, tahu a cesty v grafu (Zdroj: Kolář 2013)*

**Kružnice grafu**

Definice: *„Jsou-li u, v sousední vrcholy v grafu G a Pn je nějaká (u, v) cesta v G, která neobsahuje hranu u, v, pak cesta Pn spolu s hranou uv tvoří cyklus Cn v grafu G.“ (Kolář 2013, s. 56)*

***Hamiltonovská kružnice, graf, cesta***

Definice: *„Cyklus, který prochází všemi vrcholy grafu G, se nazývá hamiltonovský cyklus v grafu G. Graf, který obsahuje hamiltonovský cyklus, se nazývá hamiltonovský graf. Podobně cesta, která prochází všemi vrcholy grafu G, se nazývá hamiltonovská cesta v grafu G.“ (Kolář 2016, s. 182) (Obr. 2)*



*Obr. 2 – Příklad hamiltonovské kružnice žlutě vyznačené (Zdroj: autor)*

**Problém obchodního cestujícího**

Definice: *„G je graf (řízený nebo neorientovaný) a F je rodina všech hamiltonovských cyklů (prohlídek) v G. Pro každou hranu e E je cena (váha) Ce předepsána. Problémem obchodního cestujícího je pak najít cestu (hamiltonovský cyklus) v G tak, aby součet nákladů na hrany cesty byl co nejmenší.“ (Gutin, Punnen, 2007, s. 3)*

**3 Řešení problému**

Výsledkem problému obchodního cestujícího je nejkratší cesta mezi uzly. To znamená, že výsledkem je Hamiltonova kružnice a její váha, jejíchž podmínkou je, aby byla co nejmenší.

Úloha je řešena pomocí algoritmů Best Insertion a Nearest Neighbor, které budou dále představeny. Na principu Best Insertion byl vytvořen skript *Best Insertion.py* a na principu Nearest Neighbor byl vytvořen skript *Nearest Neighbor.py*.

**3.1 Metoda Best Insertion**

Metoda Best Insertion je algoritmus, kdy na počátku jsou vybrány 3 náhodné body, z kterých je vytvořena kružnice s délkou *w.* Nyní je potřeba vybrat uzel *x,* který je nezpracovaný. Vybereme hranu mezi uzly x1 a x2 z vytvořené kružnice tak, aby vzniklá cesta přes uzel x vytvořenou kružnici co nejméně protáhla a minimalizovala velikost ∆w.

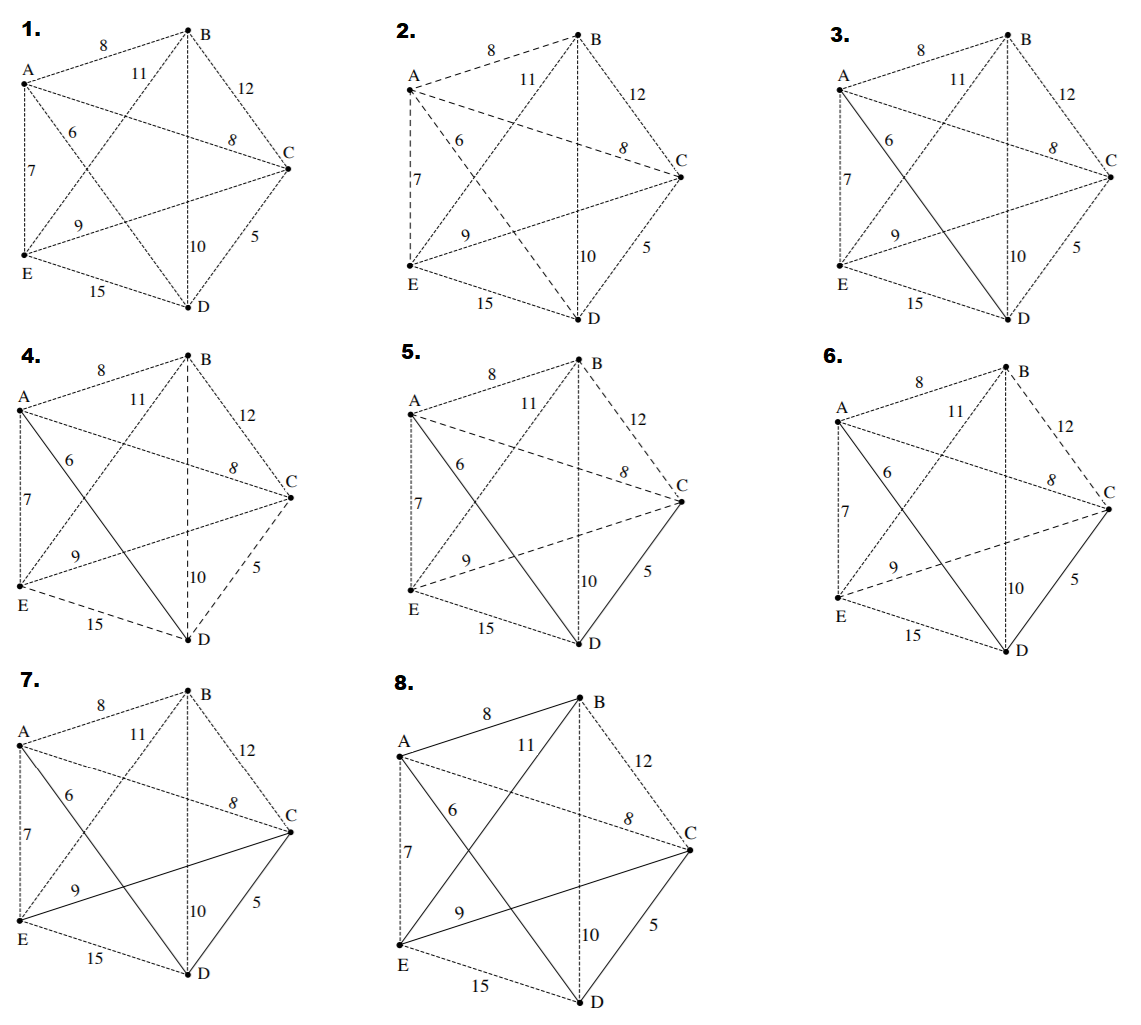
Proces od výběru nezpracovaného uzlu pokračuje, dokud nebude už žádný nezpracovaný.

**3.2 Metoda Nearest Neighbor**

Metoda Nearest Neighbor je typ *greedy* algoritmu, což znamená, že v každé fázi algoritmu vybíráme nejlepší možnou hranu, kterou lze použít. Což znamená, že v každém kroku algoritmu děláme lokálně optimální rozhodnutí. Tento počáteční vrchol budeme považovat za domov obchodního cestujícího. Další věc, kterou je třeba o tomto algoritmu poznamenat, je, že musíme vybrat počáteční vrchol a pokud vybereme různé vrcholy, můžeme získat různé Hamiltonovské cykly. (Brucato 2013)

Jde tedy o to, aby vždy obchodní cestující vždy navštívil nejbližší město. Zvolíme náhodně počáteční vrchol, který bychom mohli nazvat domovem obchodního cestujícího. Z počátečního vrcholu hledáme nejbližší nenavštívené město a tam obchodní cestující doputuje. Teď se obchodní cestující ptá, zda existují další nenavštívená města. Pokud ano, opět hledá nejbližší nenavštívené město a do něj se vydá. Pokud obchodní cestující nenalézá nenavštívená města, vrací se do svého domova, tedy do počátečního vrcholu. (Nilsson 2003)

Na Obrázku 3 vidíme možné použití algoritmu Nearest Neighbor. Jako počáteční vrchol zvolíme bod A, kde máme na výběr ze 4 hran. Z těchto 4 hran má nejmenší váhu hrana AD o váze 4, kterou si my zvolíme. Dostali jsme se do vrcholu D. Opět vybereme hranu s nejmenší váhou, tedy DC s váhou 5. Dostali jsme se do vrcholu C. Nejmenší váhu má hrana CA, jenže to bychom nesplnili hamiltonovský cyklus, a tak volíme druhou hranu s nejnižší váhou, a to CE o váze 9. Nyní jsme na vrcholu E. Předchozí situace se opakuje, volíme hranu EB o váze 11. Posledním krokem je výběr hrany BA, která přivádí obchodního cestujícího zpět domů. Tato cesta z počátečního vrcholu A má váhu 39. Pokud bychom ovšem zvolili jiný počáteční vrchol, mohli bychom dojít k lepšímu výsledku. (Nilsson 2003)



*Obr. 3 – Příklad použití Nearest Neighbor (Zdroj: Brucato 2013)*

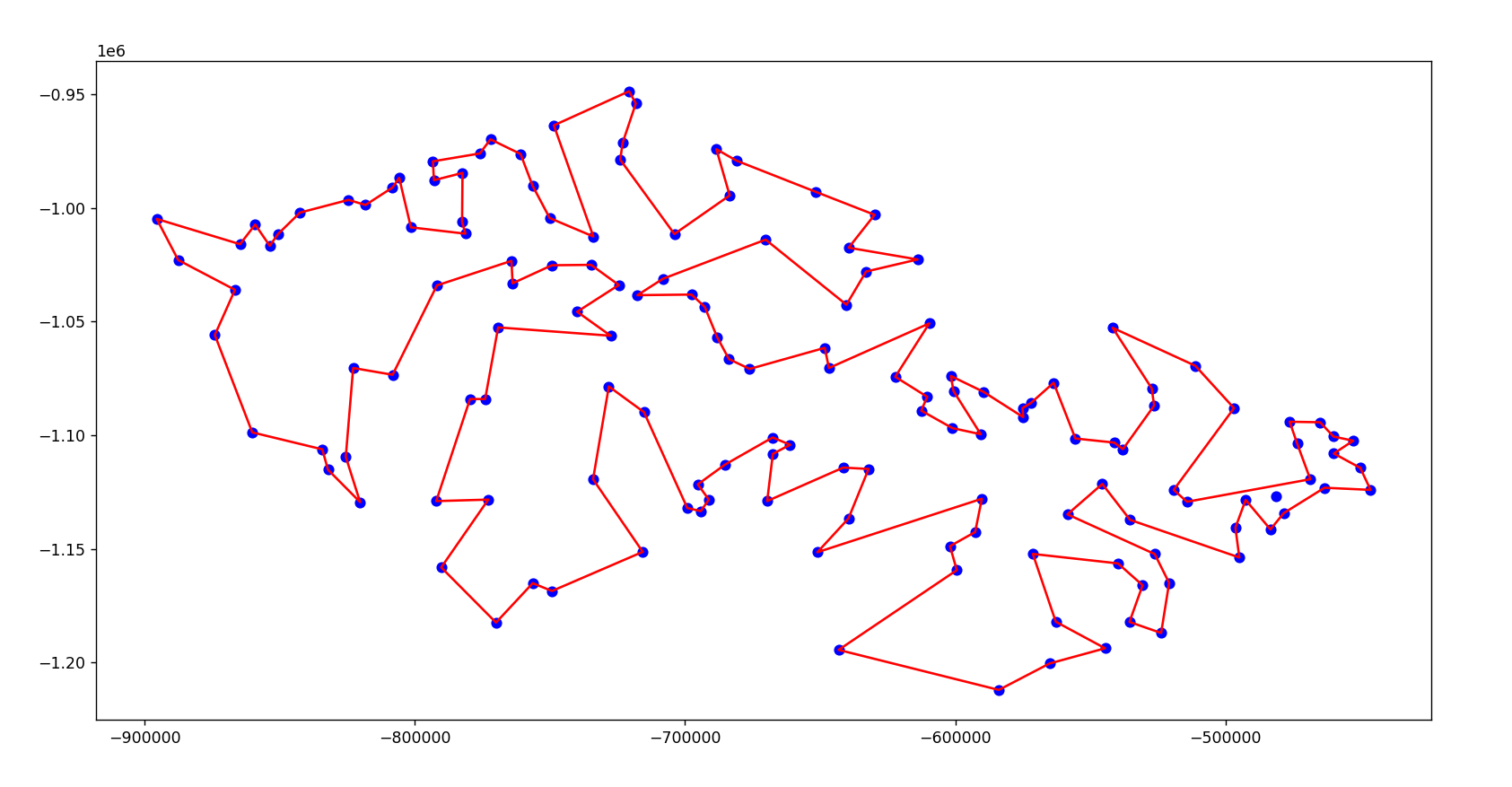
**4 Výsledky**

Pro otestování zdrojových kódů byla použita data z databáze ArcČR 500, konkrétně bodové vrstvy obcí a vrcholů ČR. Následně byly z vrstvy obcí vybrány pouze obce s počtem obyvatel nad 10 000. Z vrstvy vrcholů byly vybrány pouze vrcholy nad 1 000 m n. m. Nová vrstva obcí obsahovala 148 bodů a nová vrstva vrcholů 104 bodů. Souřadnice těchto vrstev byla následně převedeny do *.txt* souborů (konkrétně *cr\_obce.txt* a *vrcholy\_cr\_1000.txt*).

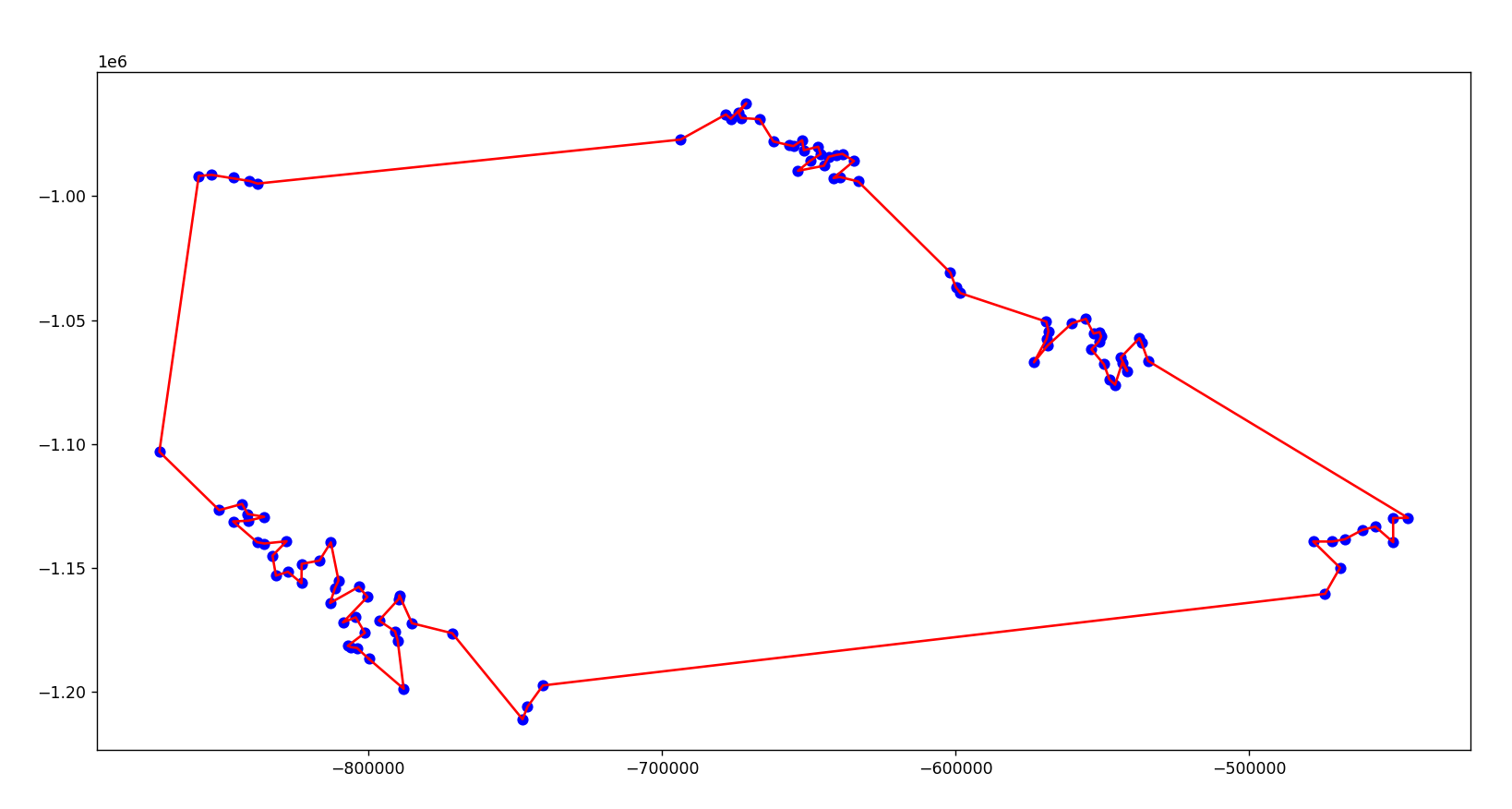
V Tabulce 1 vidíme výsledky algoritmu Best Insertion po deseti opakováních. Na konci tabulky je vypočítán průměr vzdáleností. Na Obrázku 4 a 5 jsou vizualizovány výsledky pro tento algoritmus pro obě bodové vrstvy.

|  |  |  |
| --- | --- | --- |
|  | Obce ČR nad 10 000 obyv. | Vrcholy ČR nad 1 000 m n. m. |
| 1 | 2992874,210 | 1416322,802 |
| 2 | 2810321,571 | 1434766,100 |
| 3 | 2915243,946 | 1424357,295 |
| 4 | 2811932,333 | 1433755,468 |
| 5 | 2982745,908 | 1416921,372 |
| 6 | 2807578,805 | 1401287,311 |
| 7 | 2903105,799 | 1450347,463 |
| 8 | 2987088,430 | 1406038,400 |
| 9 | 2924853,395 | 1427327,276 |
| 10 | 2797130,512 | 1420781,479 |
| průměrná vzdálenost [m] | 2893287,491 | 1423190,497 |

*Tab. 1 – Výsledné vzdálenosti algoritmu Best Insertion po 10 opakováních a dopočítanou průměrnou vzdáleností*



*Obr. 4 – Výsledek algoritmu Best Insertion pro vrstvu obcí ČR nad 10 000 obyv.*

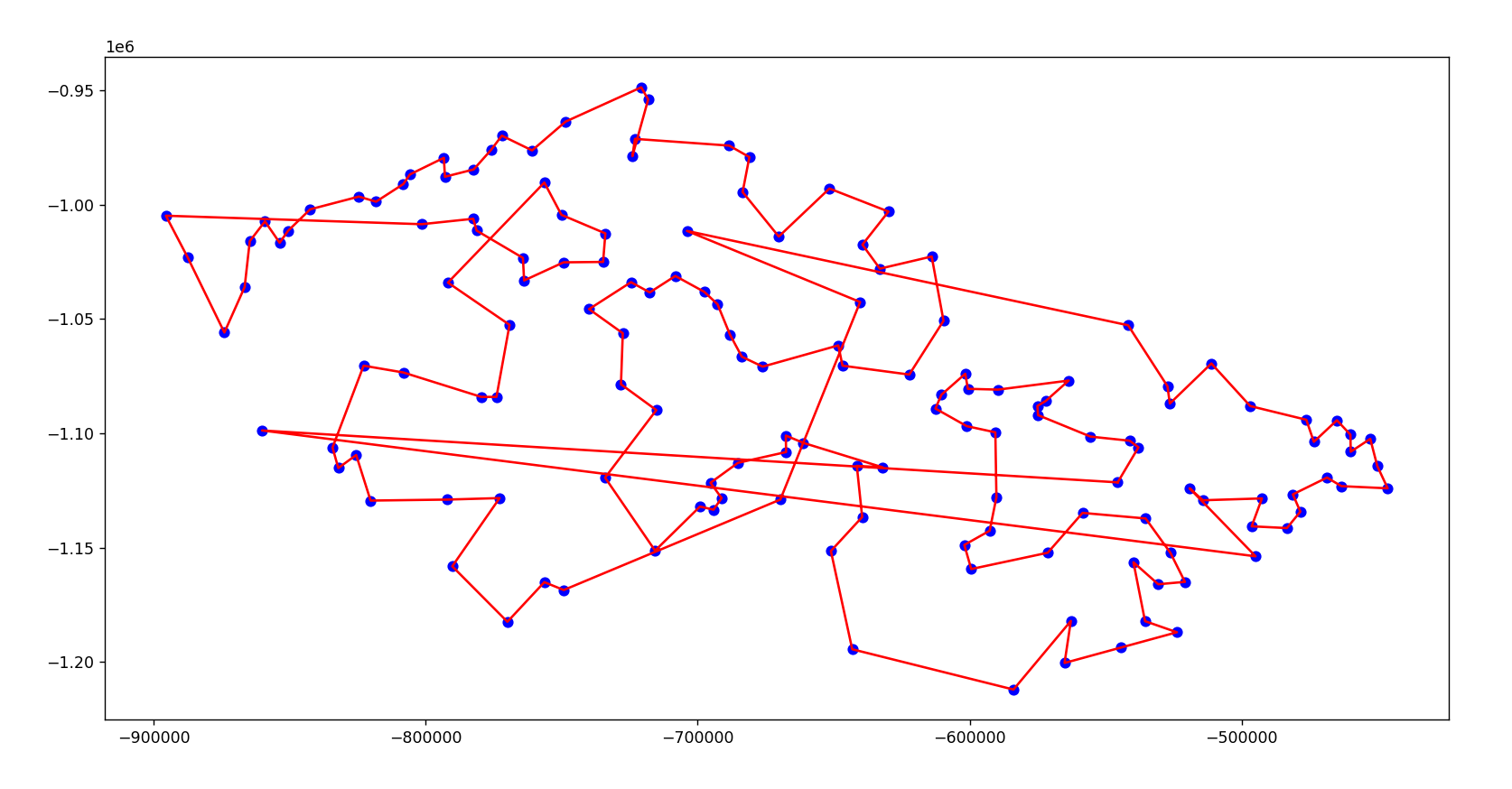


*Obr. 5 – Vizualizace výsledku algoritmu Best Insertion pro vrstvu vrcholů ČR nad 1 000 m n. m.*

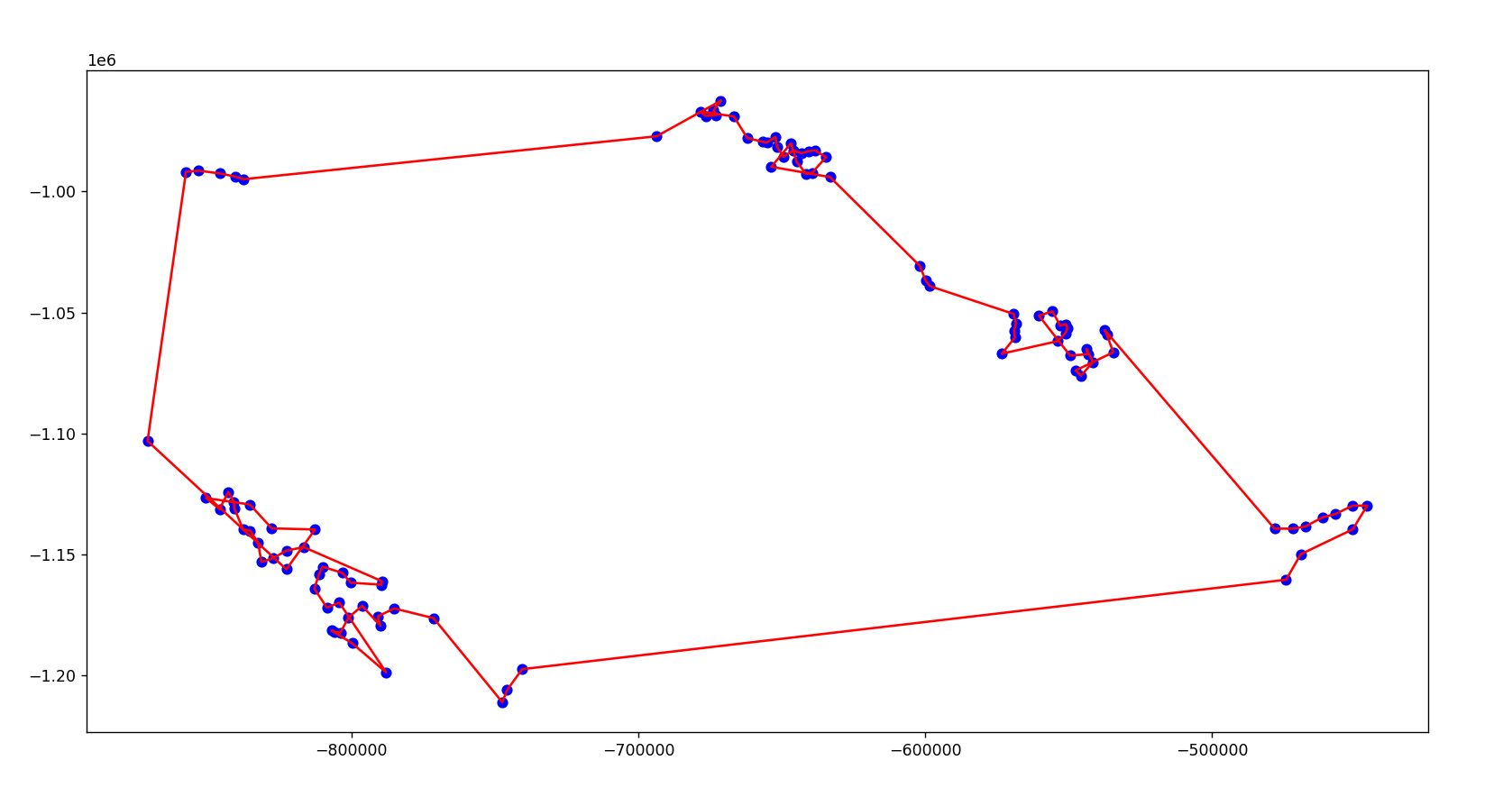
V Tabulce 2 vidíme porovnání výsledků z algoritmů Best Insertion (průměrnou hodnotu), Nearest Neighbor a výsledků ze softwaru ArcGIS Pro. Z výsledků je zřejmé, že metoda Best Insertion byla pro tento problém lepší než metoda Nearest Neighbor. Na Obrázku 6 a 7 je jsou vizualizovány výsledky pro tento algoritmus Nearest Neighbor.

|  |  |  |
| --- | --- | --- |
|  | obce ČR nad 10 000 obyv. | vrcholy ČR nad 1 000 m n. m. |
| Best Insertion (BI) [m] | 2893287,491 | 1423190,497 |
| Nearest Neighbor (NN) [m] | 3670448,641 | 1542431,350 |
| ArcGIS Pro [m] | 2671361,672 | 1630150,747 |
| BI/ArcGIS Pro [%] | 108,308 | 87,304 |
| NN/ArcGIS Pro [%] | 137,400 | 94,619 |

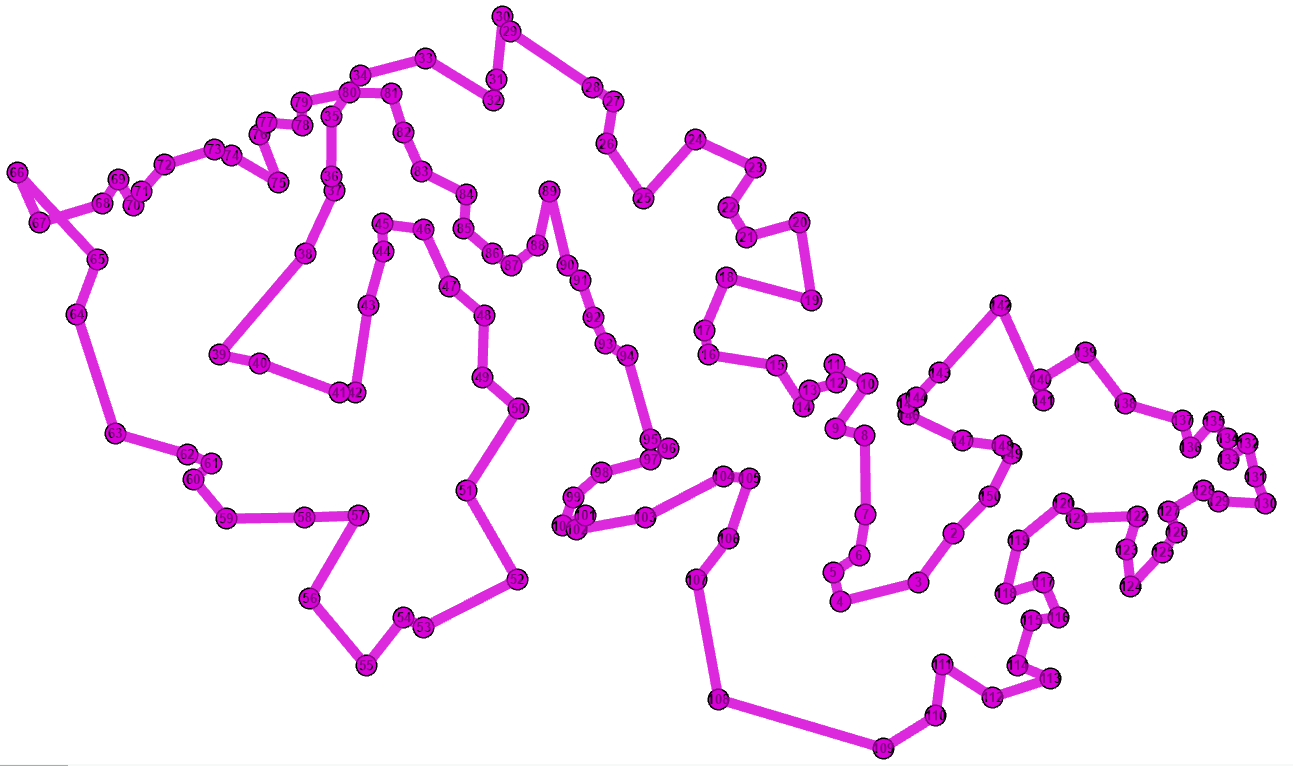
*Tabulka 2 – Porovnání výsledků Best Insertion, Nearest Neighbor a ArcGIS Pro*



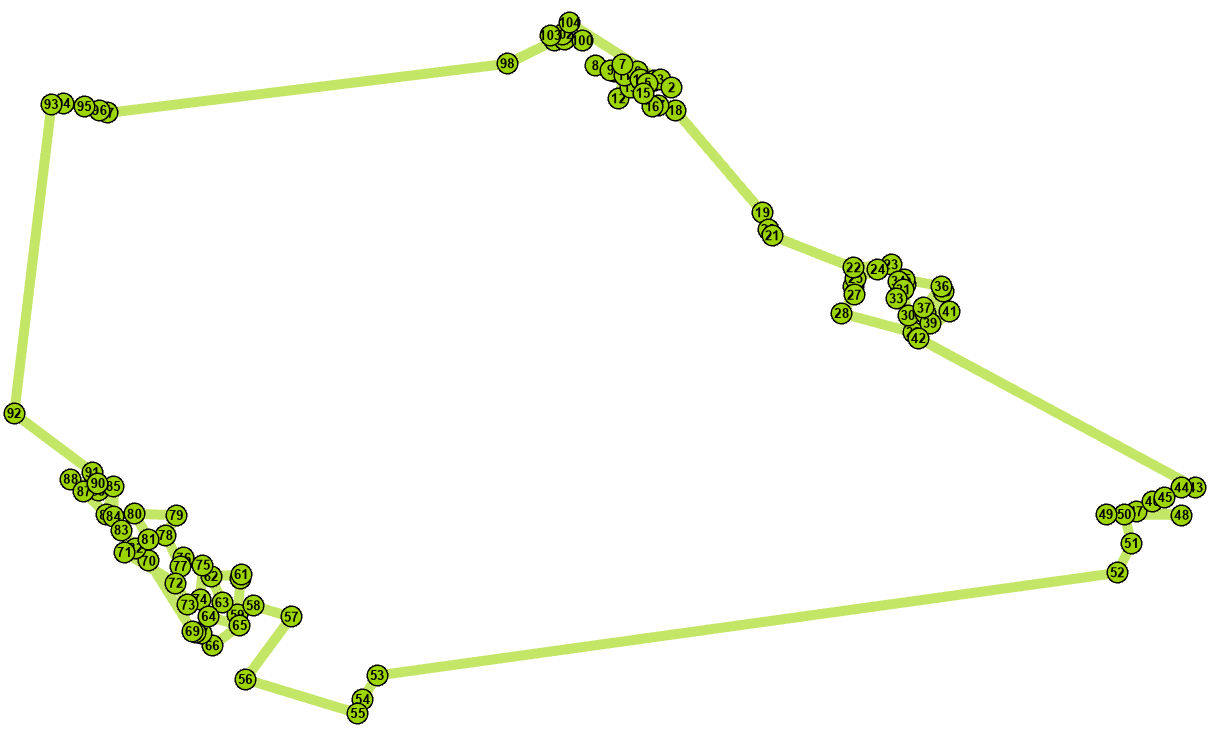
*Obr. 6 - Výsledek algoritmu Nearest Neighbor pro vrstvu obcí ČR nad 10 000 obyv.*



*Obr. 7 - Výsledek algoritmu Nearest Neighbor pro vrstvu vrcholů ČR nad 1 000 m n. m.*



*Obr. 8 - Výsledek v ArcGIS Pro pro vrstvu obcí ČR nad 10 000 obyv.*



*Obr. 9 - Výsledek v ArcGIS Pro pro vrstvu vrcholů ČR nad 1 000 m n. m.*

**5 Závěr**

Z výsledků a vizualizace je jasné, že metoda Best Insertion pracuje lépe než metoda Nearest Neighbor. Metoda v softwaru ArcGIS Pro dle mého názoru je ideální metodou pro tuto úlohu.

**6 Seznam literatury**

BRUCATO, C. (2013): The traveling salesman problem. Diplomová práce, University of Pittsburgh, Faculty of the Department of Mathematics, Pittsburgh, 57 s.

DANTZIG, G., FULKERSON, R., JOHNSON, S. (1954): Solutions of a large scale traveling salesman problem. The Rand Corporation, Santa Monica.

GUTIN, G., PUNNEN, A. P. (2007): The traveling salesman problem and its variations. Springer, USA, 132 s.

KOLÁŘ, J. (2009): Teoretická informatika. Česká technika – nakladatelství ČVUT, Praha, 206 s.

KOVÁŘ, P. (2013): Teorie grafů. Vysoká škola báňská – Technická univerzita Ostrava a Západočeská univerzita v Plzni, Ostrava, 245 s.

LAWLER, E. L., LENSTRA, J. K., RINNOOY KAN, A. H. G., R., SHMOYS, D. B. (1985): The travelling salesman problem (A quided tour of combinatorial optimisation). In: [The Mathematical Gazette](https://www.cambridge.org/core/journals/mathematical-gazette) , [Volume 70](https://www.cambridge.org/core/journals/mathematical-gazette/volume/C61B02E0DE17728A36E60579E58DDBF3) , [Issue 454](https://www.cambridge.org/core/journals/mathematical-gazette/issue/E68842DDC6DE933B90E76A52BA081B25) , December 1986, s. 327–328.

NILSSON, C. (2003) Heuristics for the Traveling Salesman problem. Linköping University, Linköping, 6 s.