Univerzita Karlova

Přírodovědecká fakulta

Řešení problému obchodního cestujícího

Geoinformatika

Tomáš Hřebec

1. N-GKDPZ

Černošice, 2022

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 konstrikčních 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.

Popis a rozbor problému:

Problém obchodního cestujícího, nebo také Travelling Salesman Problem (TSP), je diskrétní kombinatorická úloha, jejímž cílem je nalézt nejkratší cestu mezi všemi zadanými body na mapě. Jinými slovy se jedná o nalezení kružnice v úplném ohodnoceném grafu takové, že prochází všemi uzly a její cena je minimální. Kružnice, jež prochází všemi vrcholy a každý navštíví právě jednou se nazývá Hamiltonovská. Tento problém se řadí mezi takzvané NP-úplné úlohy. To zjednodušeně znamená, že u takových problémů jsem schopni získat v dostatečně krátkém čase jen řešení přiblížené.

<u>Graf</u>

Definice: Graf G (také jednoduchý graf nebo obyčejný graf) je uspořádaná dvojice G = (V, E), kde V je neprázdná množina vrcholů a E je množina hran – množina (některých) dvouprvkových podmnožin množiny V. (Kovář 2016)

Jinými graf je tvořen n vrcholy a mezi těmito vrcholy jsou hrany. Tu můžeme chápat jako spojnice či cesty mezi objekty.

Kompletní graf

Graf se nazývá kompletní, jestliže na daném počtu uzlů obsahuje všechny možné hrany. To znamená, že každý uzel je spojený se všemi ostatními uzly hranou.

Definice: Graf na n vrcholech, kde $n \in$ N, který obsahuje všech $\binom{n}{2}$ hran se nazývá úplný nebo také kompletní graf a značí se K_n. (Kovář 2016)

Ohodnocení hran

Jednotlivé hranu mezi uzly mohou mít svou váhu. Tyto váhy obvykle představují fyzikální veličiny a značí náročnost pro přesun z jednoho uzlu do druhého. V této úloze je váha hrany rovna euklidovské vzdálenosti mezi jednotlivými uzly.

Definice: Ohodnocení grafu G je funkce $w : E(G) \rightarrow$ R, která každé hraně $e \in E(G)$ přiřadí reálné číslo $w(e)$, kterému říkáme váha hrany (značení w pochází z anglického „weight“). Ohodnocený graf je graf G spolu s ohodnocením hran reálnými čísly. Kladně ohodnocený (říkáme také vážený) graf G má takové ohodnocení w, že pro každou hranu $e \in E(G)$ je její váha $w(e)$ kladná. (Kovář 2016)

Cesta grafem

Cestou grafem rozumíme průchod jeho vrcholy tak, že všechny vrcholy obsažené v této cestě byly navštíveny jen jednou. Tato cesta nemusí procházet všemi vrcholy, kterými je graf tvořen.

Kružnice (cyklus)

Jedná se o speciální případ cesty. Opět nesmí být žádný uzel grafu navštíven více než jednou a zároveň musí být cesta složena minimálně ze tří uzlů. Kružnice se od běžné cesty liší tím, že má společný začáteční a koncový bod.

Definice: Graf na n vrcholech (kde n = 3), které jsou spojeny po řadě n hranami tak, že každý vrchol je spojen s následujícím vrcholem a poslední vrchol je navíc spojen s prvním vrcholem, se nazývá cyklus na n vrcholech a značí se C_n. Číslo n je délka cyklu C_n. (Kovář 2016)

Hamiltonovská cesta

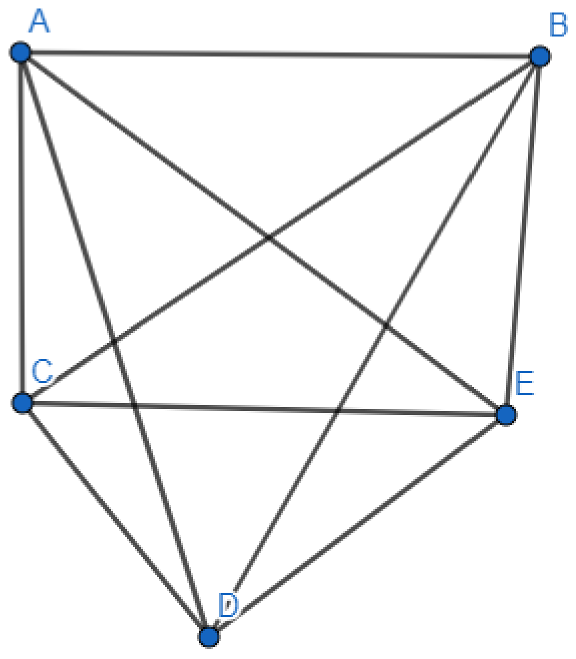
Jedná se o speciální případ cesty, která prochází všemi uzly, kterými je graf tvořen.

Hamiltonovská kružnice (cyklus)

Podobně jako u Hamiltonovské cesty se jedná o speciální případ kružnice. Tato kružnice splňuje stejné podmínky jako běžná kružnice, ale navíc musí procházet všemi uzly grafu.

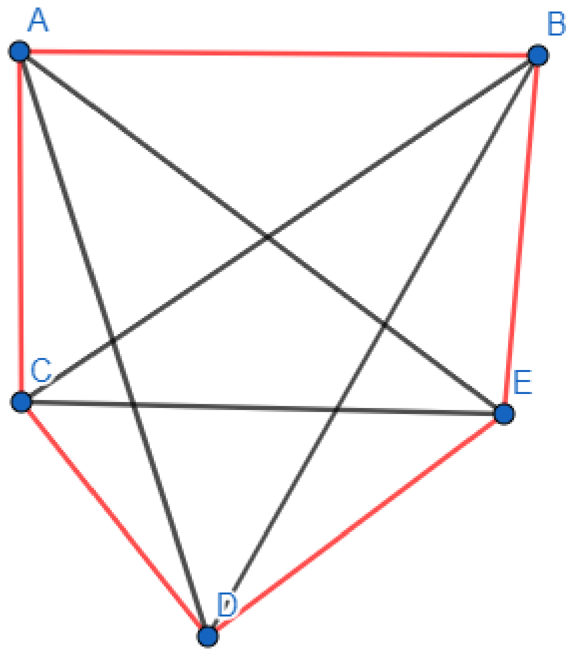
Definice: Hamiltonovský cyklus v grafu je takový cyklus, který prochází všemi vrcholy daného grafu. Graf, ve kterém existuje Hamiltonovský cyklus, se nazývá Hamiltonovský graf. (Kovář 2016)

S těmito definicemi nyní může být detailněji rozebrán řešený problém. Jako vstup je ohodnocený kompletní graf. v tomto případě jsou uzly grafu body reprezentující města. Tyto uzly mají své souřadnice. Jelikož je to kompletní graf, je počítáno s tím, že každé město má přímé spojení se všemi ostatními městy. To znamená, že z každého uzlu existuje přímá hrana do všech ostatních uzlů grafu. Ohodnocení těchto hran odpovídá euklidovské vzdálenosti mezi jednotlivými uzly. Toto ohodnocení je získáno ze souřadnic bodů. Tento graf lze vizualizovat body v prostoru (obr. 1).



Obrázek 1 Vizualizace kompletního grafu

Nyní je za úkol pomocí vybraných algoritmů tento graf projí tak, aby byly navštíveny všechny jeho uzly a zároveň výchozí bod byl i bodem konečným. Vznikne tzv. Hamiltonovská kružnice (obr. 2)



Obrázek 2 Hamiltonovská kružnice (červeně)

Výsledkem těchto algoritmů by měla být Hamiltonovská kružnice a její váha, která odpovídá součtu vah hran, ze kterých je tvořena. Tao Hamiltonovská kružnice by měla splňovat podmínku, že její váha je minimální.

$$W = \sum_{i=1}^{k-1} w(u_i, u_{i+1}) = min$$

V praxi to znamená, že hledáme nejkratší možnou cestu mezi všemi městy. To znamená, že se jedná o optimalizační úlohu.

<u>Optimalizační úloha</u>

Optimalizace se zabývá minimalizací či maximalizací funkcí mnoha proměnných. Do těchto úloh můžeme zařadit velké množství problémů. Optimalizace je část aplikované matematiky, nacházející se na pomezí matematické analýzy, lineární algebry a informatiky. (Weiner, 2011)

Vzhledem k tomu, že se jedná o NP úplný problém, je potřeba hledat jen přibližné řešení. U problému obchodního cestujícího existuje několik řešení. V této úloze je problém řešen pomocí algoritmů Nearest Neighbor a Best Insertion.

Řešení problému

Tato úloha je řešena za pomocí 2 algoritmů. První algoritmus se nazývá Nearest Neighbor.

Nearest Neighbor

Je to nejjednodušší algoritmus pro výpočet problému obchodního cestujícího. Na tomto principu byl zkonstruován skript s názvem *nearest_neighbor.py*.

Tento skript funguje následovně. Začneme v libovolném uzlu *u* grafu a označíme jej jako otevřený. Začáteční uzel *u* je na začátku definován uživatelem. Tento bod je přidán do listu, který tvoří výslednou Hamiltonovskou kružnici. Dále je od tohoto uzlu *u* zjištěna vzdálenost ke všem uzlům *v*, které ještě nebyly navštíveny. Tato vzdálenost je spočtena jako euklidovská vzdálenost. Za pomocí těchto vzdáleností je vybrán ten uzel, který má vzdálenost k uzlu *u* nejmenší. Tato vzdálenost je přičtena k celkové délce Hamiltonovské kružnice. Uzel *u* je poté nahrazen tímto uzlem *v* a celý proces se opakuje. Jakmile jsou všechny uzly označenu jako otevřené, je proces ukončen. Nyní je potřeba jen uzavřít kružnici tím, že do listu Hamiltonovské kružnice je znovu přidán startovní bod a k její délce připočtena tato vzdálenost.

Tento algoritmus je pro zadaný uzel *u* deterministický, což znamená, že při zachování počátečního uzlu vždy dojde ke stejnému výsledku.

<u>Pseudokód:</u>

```
Inicializuj začátečního uzlu u a zapamatování si ho jako start

Inicializuj stav uzlů S jako nenavštívené, vytvoř list Q, který bude odpovídat
Hamiltonovské kružnici

Přidej u na konec Q

Inicializuj délku kružnice W na 0

Změň stav uzlu u na navštívený

Dokud nejsou všechny uzly navštívené:

    Minimální vzdálenost nastav na nekonečno

    Projdi všechny uzly

        Pokud je uzel nenavštívený:

            Spočti vzdálenost aktuálního uzlu a uzlu u

            Pokud je tato vzdálenost menší než minimální vzdálenost:

                Nastav ji jako novou minimální vzdálenost

                Aktuální uzel označ jako nejbližšího souseda uzlu u
```

Přidej nejbližšího souseda uzlu u do listu Q

Nastav nejbližšího souseda uzlu u jako nový uzel u

Označ uzel u jako navštívený

K celkové délce kružnice W přičti minimální vzdálenost

Spočti vzdálenost posledního uzlu přidaného do Q k startovnímu uzlu a přičti k celkové vzdálenosti

Uzavři kružnici přidáním prvního uzlu na konec Q

Vrať celkovou vzdálenost a Hamiltonovskou kružnici

Další použitý algoritmu se nazývá Best Insertion.

Best Insertion

Tato strategie je daleko účinnější než strategie Nearest Neighbor. Na jejím principu byl vytvořen následující skript s názvem *best_insertion.py*.

Na začátku tohoto skriptu jsou vybrány 3 náhodné body, které tvoří inicializační kružnici. Je vypočtena její váha a tyto tři inicializační body jsou označeny jako otevřené. Tyto body jsou opět vloženy do listu, který reprezentuje výslednou Hamiltonovskou kružnici. V dalším kroku je vždy vybrán náhodný dosud nenavštívený uzel u a do posud vytvořené kružnice je přidán na místo, aby její váhu zvětšil o co nejméně. Toho je docíleno vztahem

$$\Delta \mathrm{w} = \min w(u_i, u) + w(u, u_{i+1}) - w(u_i, u_{i+1}),$$

Který vychází z trojúhelníkové nerovnosti. Poté je tento uzel označen jako otevřený a je vybrán další náhodný uzel z doposud nenavštívených uzlů. Toto se opakuje, dokud nebudou všechny uzly navštívené. Nakonec je opět Hamiltonovská kružnice uzavřena přidáním prvního bodu.

Tento algoritmus není deterministický a pokaždé vyjde nepatrně jinak.

<u>Pseudokód:</u>

Inicializuj stav uzlů S jako nenavštívené, vytvoř list Q, který bude odpovídat Hamiltonovské kružnici

Inicializuj délku kružnice W na 0

Vyber 3 náhodné začáteční body, které tvoří výchozí kružnici *start*

Projdi uzly kružnice *start:*

Nastav uzle jako navštívený

```
        Vypočti vzdálenost k dalšímu uzlu v pořadí
        Přičti tuto vzdálenost k délce kružnice W
        Přidej uzel do listu Q
Dokud nejsou všechny uzly navštívené:
        Vyber náhodný nenavštívený uzel u
        Minimální vzdálenost nastav na nekonečno
        Projdi doposud přidané uzly v do kružnice Q:
                Spočítej novou vzdálenost mezi uzly vi u vi+1 a odečti původní
                vzdálenost vi vi+1
                Pokud je tato vzdálenost menší než minimální vzdálenost:
                        Nastav ji jako novou minimální vzdálenost
                        Označ aktuální uzel jako potenciální pozici pro uzel u
        Nastav uzel u jako navštívený
        Vlož u do listu Q na potenciální pozici
        Přičti minimální vzdálenost k celkové vzdálenosti W
Uzavři kružnici přidáním prvního uzlu na konec Q
Vrať celkovou vzdálenost a Hamiltonovskou kružnici
```

Vstup:

Vstup pro oba skripty je nahráván ze souboru CSV a ukládán do listu, kde je informace o názvech jednotlivých uzlů a do slovníku, kde jsou k jednotlivým uzlům přiřazeny jejich souřadnice. CSV soubor musí vypadat tak, že v prvním sloupci je číslo uzlu a ve druhém a ve třetím sloupci jsou jeho souřadnice. Jednotlivé sloupce jsou odděleny středníkem.

Výstup:

Výstupem obou skriptů je list obsahující názvy uzlů uspořádané tak, jak tvoří Hamiltonovskou kružnici, hodnota této kružnice a její vizualizace.

Výsledky

Pro vstup byly použity 2 bodové datasety. 1. s názvem coord_cernosice.csv, který obsahuj souřadnice obcí spadajících do správního obvodu Černošic a Říčan. 2. datatset s názvem coord_10000.csv obsahuje souřadnice obcí s počtem obyvatel větším než 10 000. Tyto datasety byly exportovány ze shapefilu jako CSV soubor.

Výsledky algoritmu Best Insertion. Porovnání 10 opakování, ze kterých byl vybrán ten nejlepší (minimum) (tab. 1). Pro data coord_cernosice.csv se jedná o výsledek 9 a pro coord 10000.csc výsledek 8.

	coord_cernosice.csv [m]
1.	386472,385
2.	384907,604
3.	390759,486
4.	393406,179
5.	383491,278
6.	389756,778
7.	383449,671
8.	383442,786
9.	382280,704
10.	385692,632

	coord_10000.csv [m]
1.	2766303,290
2.	2722416,297
3.	2705840,856
4.	2792202,335
5.	2830534,455
6.	2798348,647
7.	2736546,666
8.	2692059,213
9.	2815756,500
10.	2790199,296

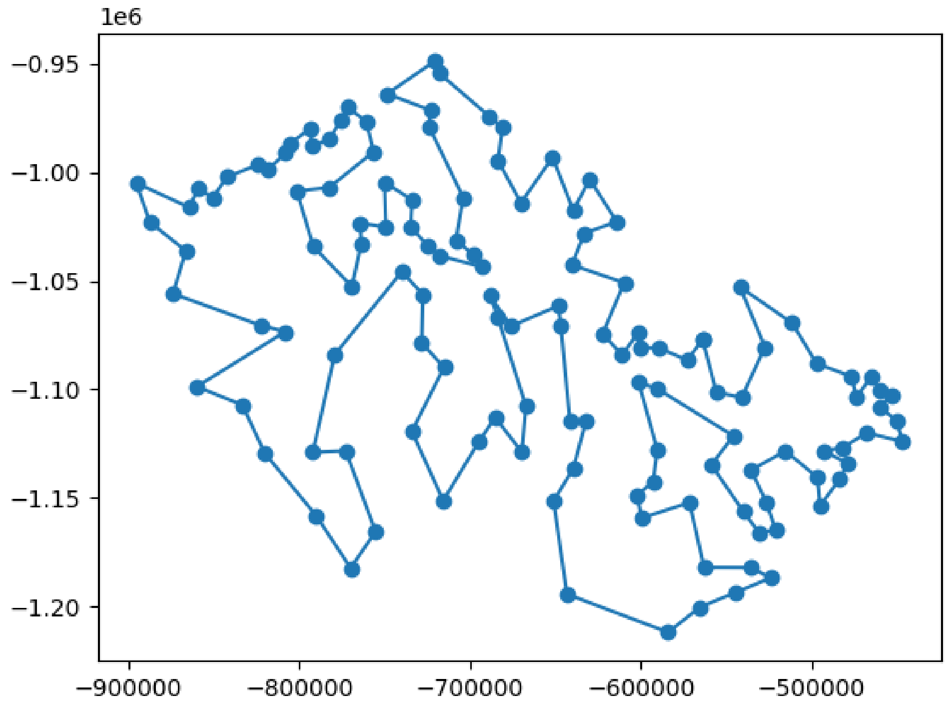
Tabulka 1 výsledky algoritmu Best Insertion pro oba datasety

Porovnání výsledků z algoritmu Nearest Neighbor, Best Insertion a výsledků ze softwaru ArcGis Pro (tab. 2). U obou datasetů vyšla nejlépe metoda používaná softwarem ArcGIS Pro. Při porovnání s touto metodou vyšla metoda Best Insertion jako lepší než metoda Nearest Neighbor. Kružnice nalezená metodou Nearest Neighbor je asi o 20 % větší než kružnice spočtená ArcGISem. Při porovnání Best Insertion a ArcGIS Pro vyšla kružnice vetší o 1,34 % a o 5,63 % delší.

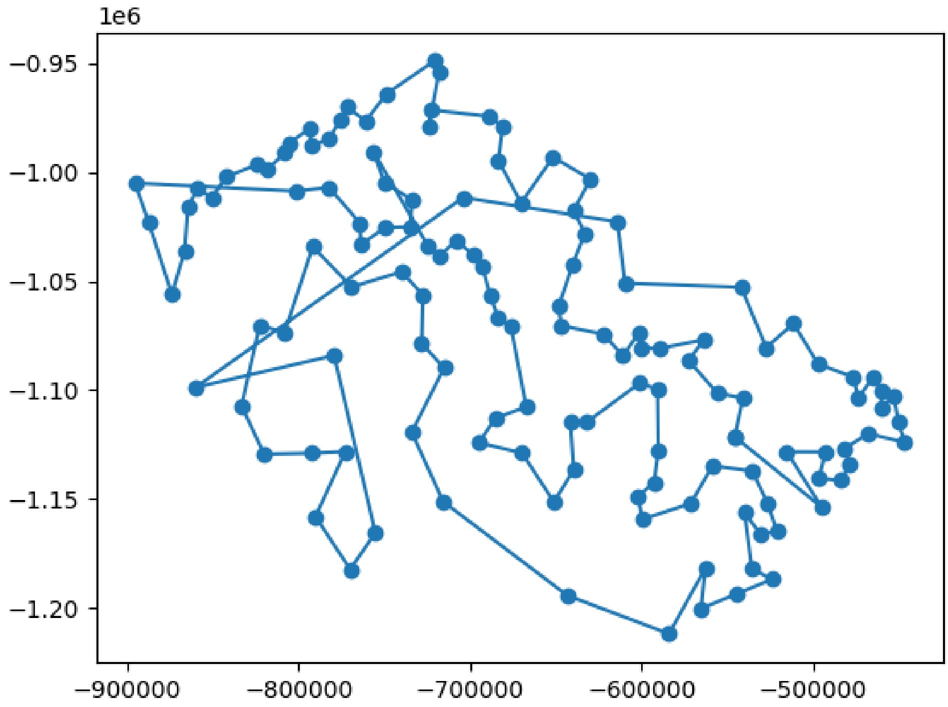
	Nearest Neighor [m]	Best Insertion [m]	ArcGIS Pro [m]	NN/ArcGIS [%]	BI/ArcGIS [%]
coord_cernosice.csv	452255,321	382280,704	377221,046	119,89	101,34
coord_10000.csv	3092515,999	2692059,213	2548559,004	121,34	105,63

Tabulka 2 porovnání výsledků

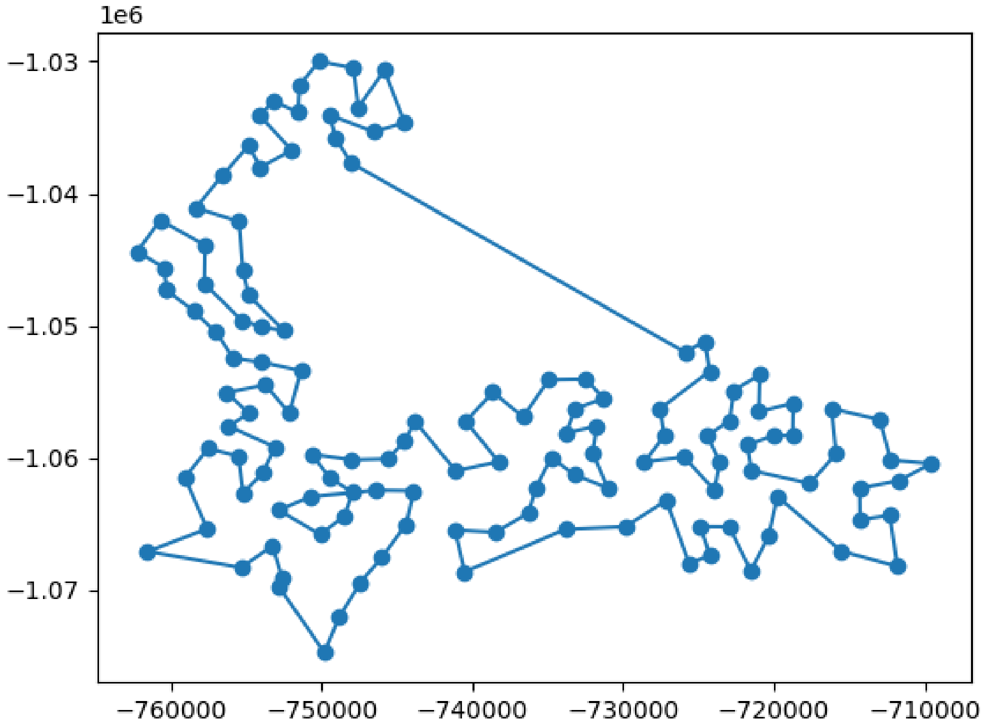
Vizualizace výsledných Hamiltonovských kružnic pomocí knihovny matplotlib jsou na obrázcích 3 až 6 a obrázcích 7 a 8 jsou vizualizovány Hamiltonovské kružnice vypočítané pomocí ArcGIS Pro.



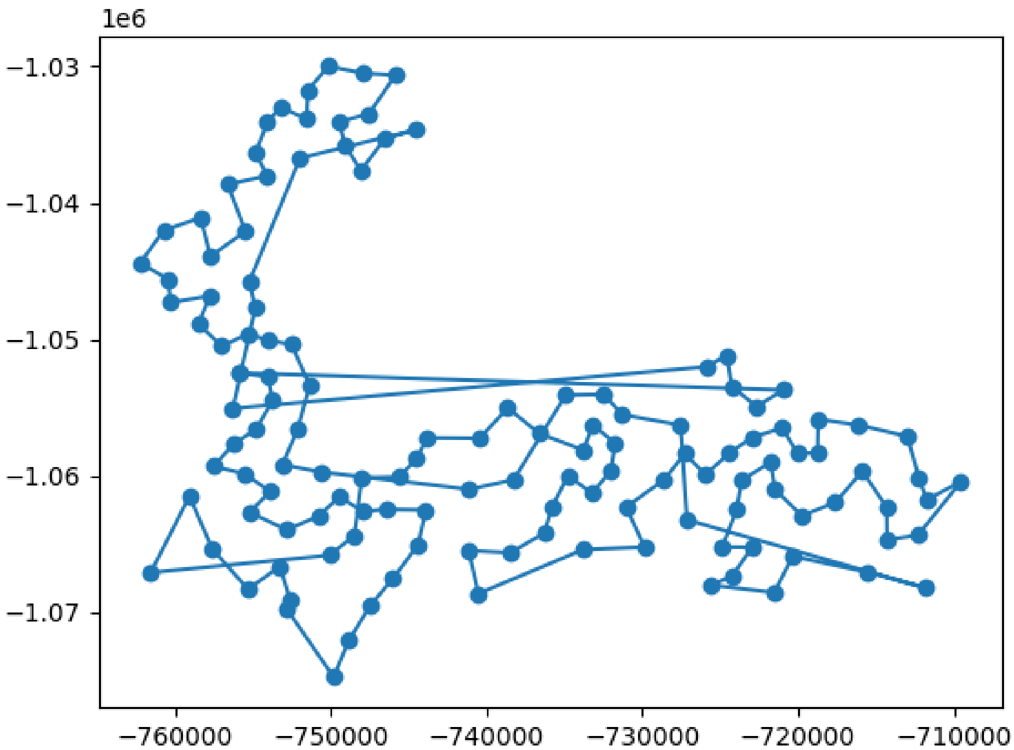
Obrázek 3 Best Insertion, dataset coord_10000.csv



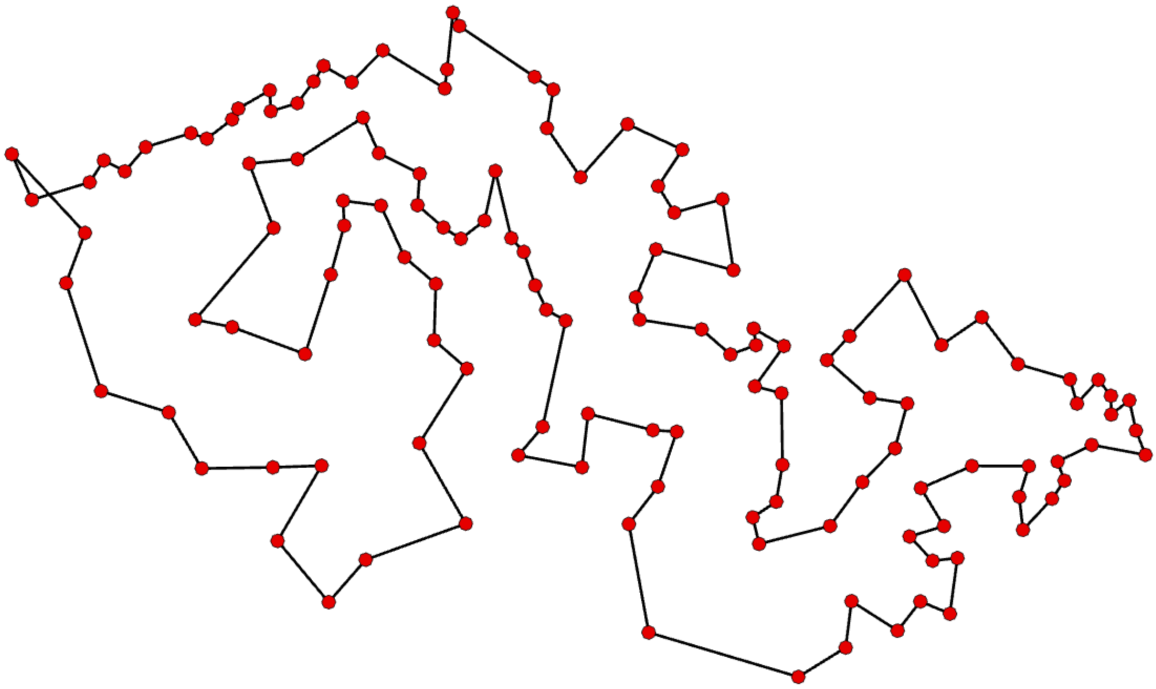
Obrázek 4 Nearest Neighbor, dataset coord_10000.csv



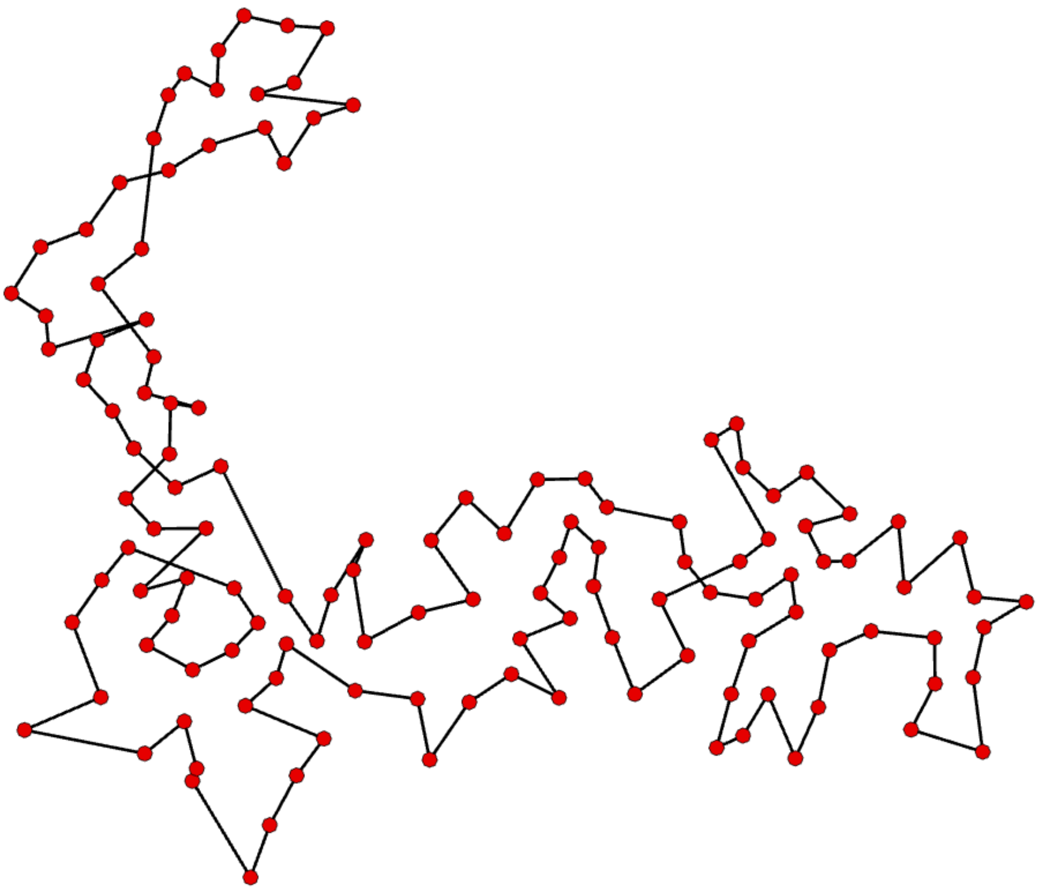
Obrázek 5 Best Insertion, dataset coord_cernosice.csv



Obrázek 6 Nearest Neighbor, dataset coord_cernosice.csv



Obrázek 7 Dataset coord_10000.csv pomocí ArcGIS Pro



Obrázek 8 Dataset coord_cernosice.csv pomoví ArcGIS Pro

Závěr

Z výsledků je patrné, že optimální metoda je využívána v softwaru ArcGIS Pro. Zde použitý skript *best_insertion.py* se ale dokáže svými výsledky přiblížit. I z obrazových výstupů je patrné, že metoda Best Insertion pracuje lépe než Nearest Neighbor. Mezi možná vylepšení těchto skriptů bych zařadil možnost vstup načítat rovnou ze shapifilu pomocí knihovny arcpy. Toto by mohlo usnadnit práci se skripty, protože by nebyl takový důraz na korektní podobu vstupu. Na druhou stranu by uživatel potřeboval přístup k této knihovně. Další vylepšení by mohlo být v oblasti výstupů. Výsledné grafy a kružnice by se nemusely jen vizualizovat ale rovnou i automaticky ukládat.

Zdroje:

Kovář, P. Úvod do teorie grafů, Západočeské univerzita v Plzni, skriptum, 2016

Weiner. T. Optimalizace, České vysoké učení technické, Fakulta elektrotechnická, skriptum, 2011