Univerzita Karlova

Přírodovědecká fakulta

GEOINFORMATIKA

Řešení problému obchodního cestujícího

Petra Krsková

1. ročník N-GKDPZ

Praha 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 konstrukčních heuristik:

- Nearest Neighbor,
- Best Insertion.

Výsledky porovnejte s výstupem poskytovaným nástrojem Network Analyt 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

Cílem problému obchodního cestujícího, známějšího pod anglickým názvem Traveling Salesman Problem (TSP), je nalezení nejkratší cesty mezi všemi zadanými body tak, aby byl každý bod navštíven právě jednou a na závěr došlo k návratu do výchozího bodu. TSP je představitelem kombinatorického optimalizačního problému a řadíme jej do kategorie těchto problémů známé jako NP-úplné. U těchto úloh nejsme schopni získat řešení v dostatečně krátkém čase, ale nalézt pouze přibližné řešení s dostatečnou kvalitou.

Základní pojmy

Graf

Graf je datová struktura tvořená uzly *U* a hranami *H*, která popisuje vztahy mezi objekty. Zpravidla je počet uzlů i hran konečný a hovoříme tak o konečném grafu. Graf představuje topologicko/geometrický model skutečnosti a lze ho znázornit nekonečně mnoha způsoby.

Neorientovaný graf

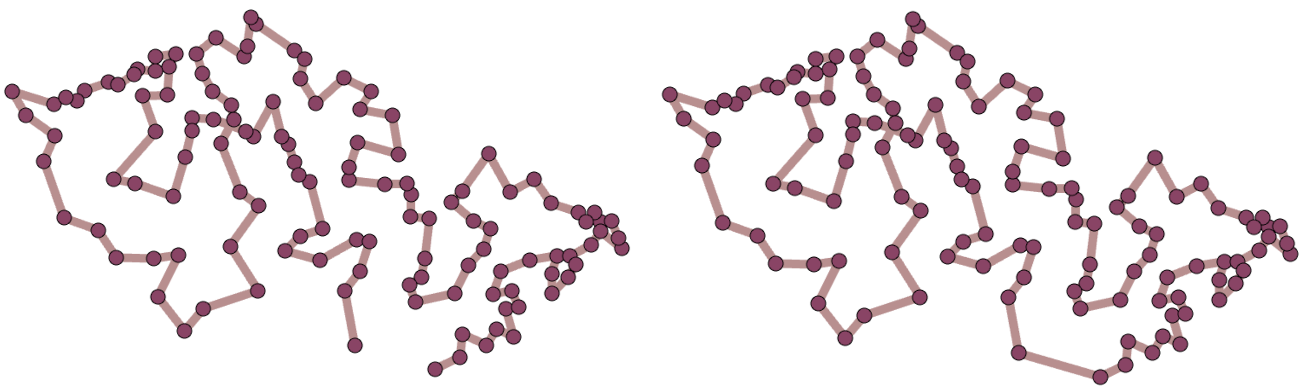
Neorientovaný graf představuje trojice disjunktních množin $G = \langle H, U, \rho \rangle$, kde *H* představují hrany, *U* uzly a ρ incidenci grafu *G*. Incidence grafu definovaná vztahem $\rho : H \rightarrow U \otimes U$ pak přiřazuje každé hraně z množiny *H* neprázdnou množinu dvojic uzlů z množiny *U*. Hranami je možné procházet oběma směry a uzly lze projít vícekrát.

Cesta grafem

Cesta grafem *C* je uspořádaná posloupnost uzlů a hran, ve které se každá hrana i uzel vyskytují nejvýše jednou.

Kružnice grafu

Kružnice *K* je taková cesta grafem, která obsahuje alespoň tři různé vrcholy a platí, že její počáteční uzel u_0 a koncový uzel u_k jsou shodné. Rozdíl mezi cestou a kružnicí grafu je zobrazen na obrázku 1.



Obr. 1 – Rozdíl mezi cestou grafem (vlevo) a kružnicí grafu (vpravo)

Hamiltonovská cesta

Hamiltonovská cesta C_h je speciální případ cesty a platí pro ni, že obsahuje všechny uzly grafu.

Hamiltonovská kružnice

Hamiltonovská kružnice K_h je kružnice obsahující všechny uzly grafu.

Problém obchodního cestujícího

V Hamiltonovské kružnici tvořené uzly $u_0, u_1, ..., u_{k-1}, u_k$ je její délka definována jako suma ohodnocení jednotlivých hran v grafu. Cílem TSP je pak nalézt v zadaném grafu Hamiltonovskou kružnici s minimální délkou

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

Řešení problému

Problém obchodního cestujícího je implementován v programovacím jazyce Python a řešen pomocí metod Nearest Neighbor a Best Insertion. Výsledky jsou pak porovnány s výstupem nástroje *Network Analyst* v SW ArcGIS Pro.

Nearest Neighbor

Tato metoda představuje nejjednodušší konstrukční heuristiku, kterou je možné použít pro řešení TSP. Na počátku jsou všechny uzly označeny jako nezpracované a z nich je náhodně vybrán jeden počáteční bod, případně je použit bod zadaný. Tento bod je označen jako zpracovaný a jako další je vybrán ten bod, ke kterému je z aktuálního bodu nejkratší vzdálenost. Tento proces je opakován do té doby, než dojde k vytvoření Hamiltonovské kružnice, tedy všechny body neoznačíme jako zpracované.

Pseudokód

Nastav stav všech uzlů jako nezpracovaný

Inicializuj délku kružnice W na hodnotu 0

Vyber náhodný uzel u, nastav ho jako aktuální uzel u_i a označ ho i jako startovní

Přidej uzel u_i do Hamiltonovské kružnice K

Nastav stav uzlu u_i jako zpracovaný

***Dokud** existuje nezpracovaný uzel*

  Minimální vzdálenost inicializuj na nekonečno

  ***Pro** všechny zbylé nezpracované uzly u*

    Spočítej vzdálenost mezi u a u_i

    ***Pokud** je vzdálenost menší než minimální vzdálenost*

Vzdálenost přiřaď do minimální vzdálenosti

Zapamatuj si uzel u jako nejbližšího souseda

Nejbližšího souseda přiřaď do kružnice K

K délce kružnice W přičti minimální vzdálenost

Nejbližšího souseda nastav jako aktuální uzel u_i

Aktualizuj stav uzlu u_i *na zpracovaný*

Spočítej vzdálenost mezi posledním uzlem přidaným do kružnice K a startovním uzlem

Tuto vzdálenost přičti k délce kružnice W

Startovní uzel přidej na konec kružnice K

Vrať Hamiltonovskou kružnici K a její délku W

Best Insertion

Tato metoda je vylepšením algoritmu Nearest Neighbor, protože pracuje od začátku s kružnicí a v každém kroku vybírá takový uzel, aby byl přírůstek délky kružnice minimální. Takový uzel pak nepřipojuje na konec, ale do libovolného místa kružnice, tím pádem by nemělo docházet ke křížení linií. Minimalizační kritérium vychází v tomto případě z trojúhelníkové nerovnosti.

Pseudokód

Nastav stav všech uzlů jako nezpracovaný

Inicializuj délku kružnice W na hodnotu 0

Vyber tři náhodné uzly a přidej je do kružnice start

***Pro** všechny uzly v kružnici start*

Nastav stav uzlu na zpracovaný

Přidej ho do kružnice K

***Pro** tři opakování*

Spočítej vzdálenost mezi aktuálním uzlem v kružnici start a následujícím uzlem

Tuto vzdálenost přičti k délce kružnice W

***Dokud** existuje nezpracovaný uzel*

Minimální přírůstek inicializuj na nekonečno

Vyber náhodný nezpracovaný uzel a nastav ho jako aktuální uzel u

***Pro** všechny uzly* u_i *v kružnici K*

Spočítej přírůstek délky hrany $\Delta w = w(u_i, u) + w(u, u_{i+1}) - w(u_i , u_{i+1})$

***Pokud** je přírůstek menší než minimální přírůstek*

Přírůstek přiřaď do minimálního přírůstku

Zapamatuj si uzly u, u_i

Uzel u přidej do kružnice K mezi body u_i a u_{i+1}

Nastav stav uzlu u na zpracovaný

Přičti minimální přírůstek k délce kružnice W

Startovní uzel přidej na konec kružnice K

Vrať Hamiltonovskou kružnici K a její délku W

Struktura programu

Program se skládá z 218 řádek včetně komentářů a odsazení a obsahuje šest funkcí.

První funkce *load_data* slouží k načtení dat ze vstupního souboru a uložení souřadnic jednotlivých bodů do seznamu. Pomocí *try* a *except* bloku je odchytáván špatně zadaný název souboru či cesta k jeho umístění a další případné chyby spojené s otevíráním obrázku. V případě chyby je do konzole vypsána chybová hláška definující problém a program skončí. Na vstupu má parametr *file*, který představuje cestu ke vstupnímu .csv souboru, který obsahuje souřadnice bodů v atributech *coord_X* a *coord_Y*. Funkce pak vrací vytvořený seznam s body.

Funkce *distance* přebírá na vstupu dva uzly a vrací vypočítanou vzdálenost mezi nimi.

Ve třetí funkci *nearest_neigh*bor je na základě řešení a pseudokódu popsaného dříve implementována metoda Nearest Neighbor (NN). Na vstupu přebírá tato funkce seznam uzlů *nodes* a paramert *start_node*. Ten představuje index počátečního uzlu a defaultně je nastavený na hodnotu -1, díky čemuž dochází k náhodnému výběru počátečního uzlu. Funkce vrací vytvořenou Hamiltonovskou kružnici v podobě seznamu seřazených uzlů a dále hodnotu délky této kružnice.

Čtvrtá funkce *best_insertion* opět vychází z dříve popsaného řešení a pseudokódu pro metodu Best Insertion (BI). Na vstupu přebírá seznam uzlů a navrací pak vytvořenou Hamiltonovskou kružnici v podobě seznamu seřazených uzlů a dále hodnotu délky této kružnice.

Funkce *plot_cycle* přebírá na vstupu seznam uzlů tvořících kružnici a slouží k jejímu vykreslení s využitím knihovny matplotlib.

Šestá funkce *tsp* pak v sobě kombinuje všechny předchozí funkce a slouží k opakovaným výpočtům Hamiltonovské kružnice s využitím vybrané metody a k vizualizaci a exportu výsledků. Na vstupu má funkce následující parametry:

- *inpurt_file* – csv soubor se vstupními body, který obsahuje jejich souřadnice v atributech *coord_X* a *coord_Y*
- *algorithm* – volba použité metody, defaultně 'NN' pro Nearest Neighbor, dále 'BI' pro Best Insertion
- *repetitions* – celé číslo udávající kolikrát má daná metoda proběhnout, defaultně 10

- *start_node* – celé číslo definující index počátečního uzlu pro metodu NN, defaultně -1 pro náhodný výběr
- *plot* – bool, vykreslení nejlepší kružnice, defaultně True
- *print_result* – bool, vypsání délky všech vypočítaných kružnic do konzole, defaultně False
- *export_result* – bool, uložení délky všech vypočítaných kružnic do nově vytvořeného souboru, defaultně True

Funkce pak vrací seznam uzlů nejlepší kružnice, její délku, seznam všech vypočítaných kružnic a seznam délek všech kružnic.

Jednotlivé metody řešící TSP je možné zavolat pomocí připravených funkcí pro každou metodu anebo pomocí komplexní funkce *tsp* a průběh a vizualizace výsledku upřesnit nastavením jejích parametrů.

Network Analyst v ArcGIS Pro

Pro porovnání výsledků byla pro oba vstupní datasety vytvořena nejkratší Hamiltonovská kružnice i s využitím nástroje *Network Analyst* v SW ArcGIS Pro. Nejprve byla funkcí *Generate Near Table* vytvořena tabulka obsahující souřadnice všech dvojic uzlů a funkcí *XY To Line* pak byly vytvořeny linie spojující všechny body. Takto vytvořená *Feature Class* byla uložena do samostatného *Feature Datasetu,* na který byla zavolána funkce *Create Network Dataset* a *Build Network*, čímž došlo k vybudování sítě a výpočtu vah jednotlivých hran, tedy vzdálenosti mezi body.

V nástroji *Network Analyst* pak byla vybrána varianta *Route* a pro zadané vstupní body byl nástroj spuštěn s nastavením sekvence na *Find Best*. Výsledkem této operace je ale pouze Hamiltonovská cesta, nikoliv kružnice, zato ale s nalezeným ideálním počátečním bodem. Tento bod tak byl zkopírován na konec sekvence jednotlivých bodů a po opětovném spuštění s nastavením na *Preserve First & Last Stop* došlo k zakončení cesty v počátečním bodě a tím pádem k vytvoření kružnice.

Vstupní data a výsledky

Pro otestování vytvořeného programu byly vybrány dva datasety tvořené existujícími geografickými daty. První dataset *ČR_obce_10_tis.csv* obsahuje 126 bodů, které odpovídají obcím v Česku s počtem obyvatel vyšším než 10 tisíc. Druhý dataset *LIB_post_office.csv* tvoří 156 bodů představující pošty v Libereckém kraji. Oba datasety byly získány z dat OpenStreetMap (OSM 2022).

Na oba vstupní datasety byla desetkrát spuštěna každá z implementovaných metod a vypočtené hodnoty délek kružnic jsou uvedeny v tabulkách 1 a 2. Tabulka 3 pak uvádí srovnání nejlepších výsledků obou metod s délkou kružnice vypočítané nástrojem *Network Analyst* v SW ArcGIS Pro. Obrázky 2–7 zobrazují vizualizace nejlepších výsledků obou metod a výsledku z ArcGIS Pro.

Obce ČR nad 10 tis. obyv.		
Opakování	Nearest Neighbor	Best Insertion
	W [m]	W [m]
1	3609169,255	2806475,466
2	3092122,379	2610373,746
3	3433095,179	2751470,222
4	3300856,629	2754386,345
5	3400668,685	2711093,189
6	3555507,686	2635265,739
7	3583731,710	2641895,308
8	3599845,313	2659277,379
9	3514206,751	2751043,419
10	3336220,276	2664174,730
min	**3092122,379**	**2610373,746**

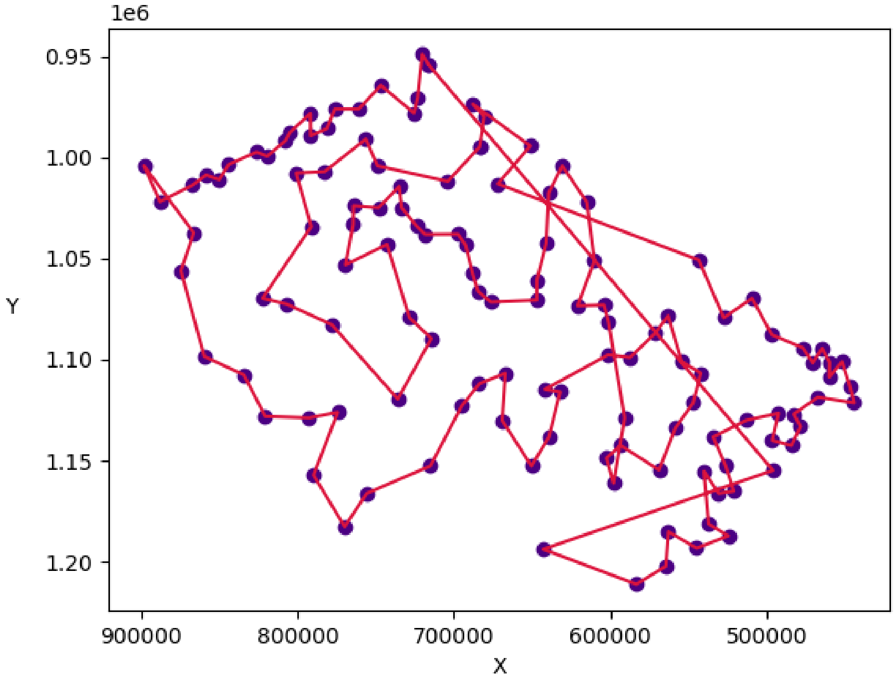
Tab. 1 – Výsledky pro obě metody pro dataset obcí ČR

Pošty v Libereckém kraji		
Opakování	Nearest Neighbor	Best Insertion
	W [m]	W [m]
1	753968,614	632295,631
2	723698,092	638598,975
3	701653,749	627385,492
4	748617,362	613487,624
5	745734,528	623041,392
6	763519,408	635342,954
7	685523,392	623740,890
8	745936,404	614362,860
9	758097,261	630479,598
10	675751,576	619833,295
min	**675751,576**	**613487,624**

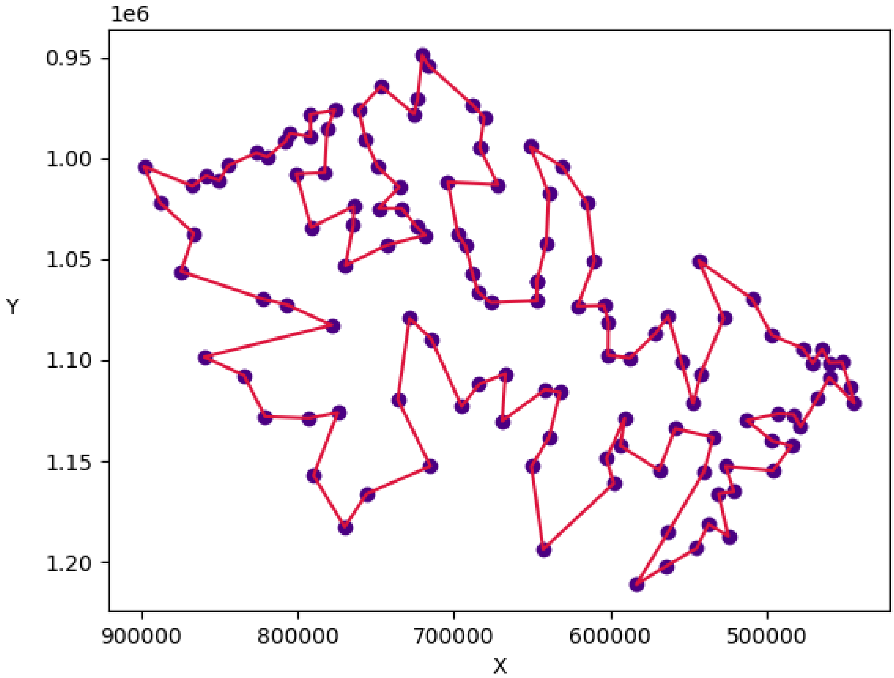
Tab. 2 – Výsledky pro obě metody pro dataset pošt v Libereckém kraji

Dataset \ W	Nearest Neighbor [m]	Best Insertion [m]	ArcGIS Pro [m]
Obce ČR	3092122,379	2610373,746	2496265,367
Pošty v Lib. kraji	675751,576	613487,624	577418,781

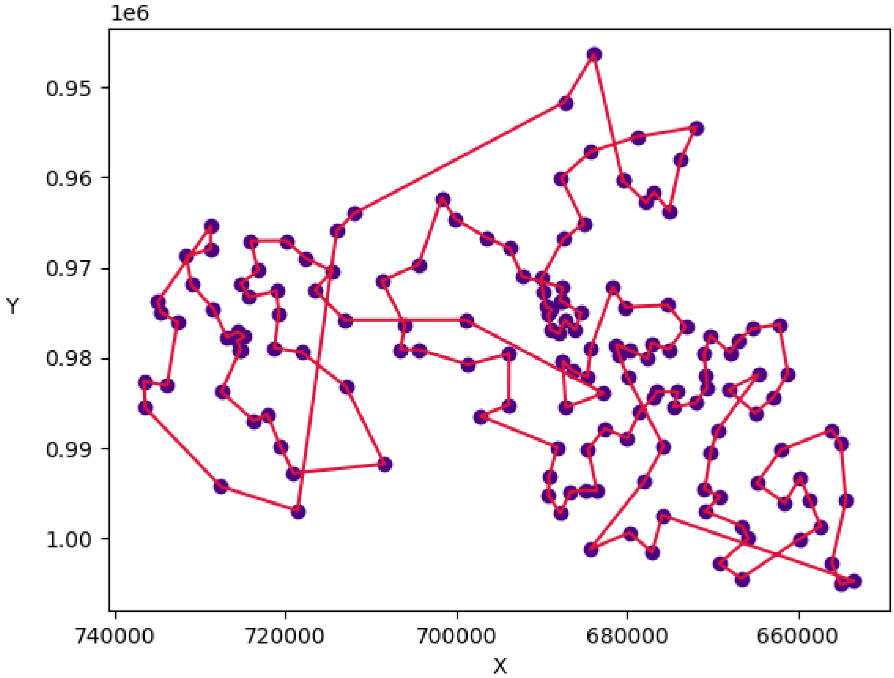
Tab. 3 – Porovnání nejlepších výsledků obou metod s výsledkem ArcGIS Pro



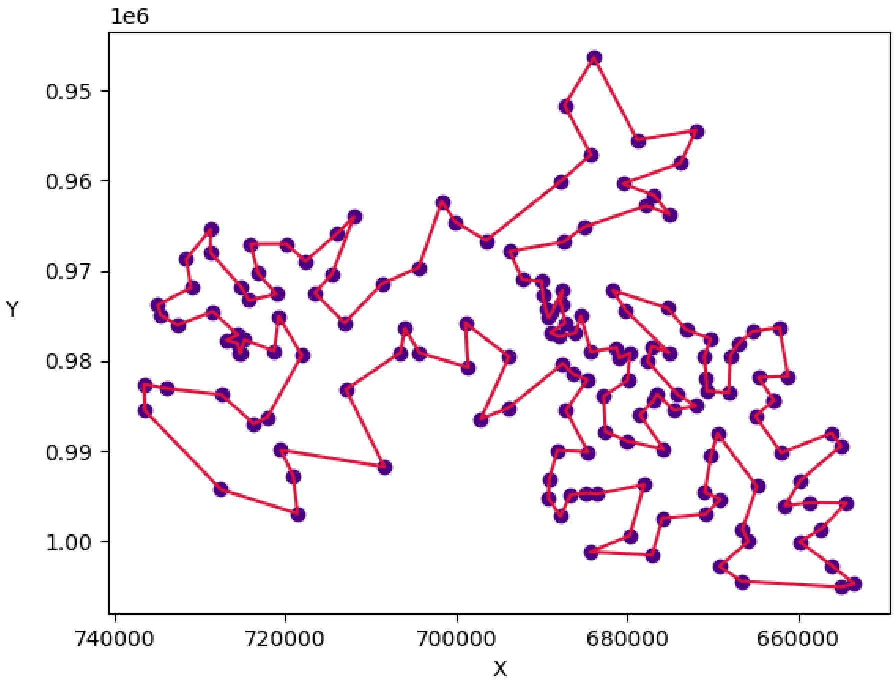
Obr. 2 – Výsledek metody Nearest Neighbor pro obce ČR



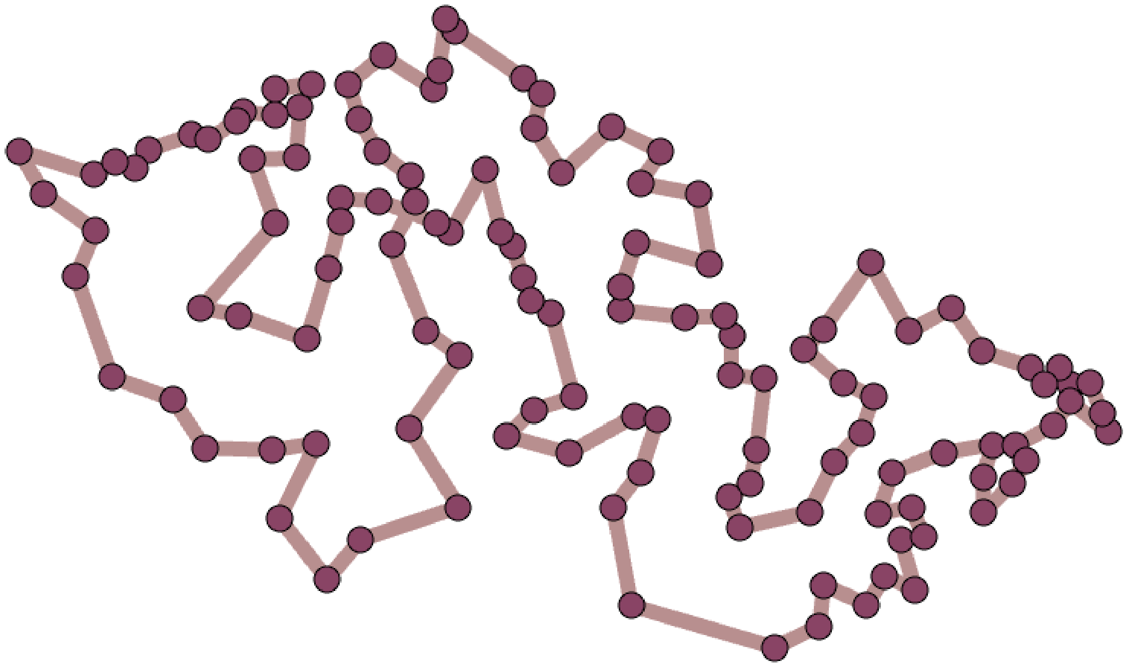
Obr. 3 – Výsledek metody Best Insertion pro obce ČR



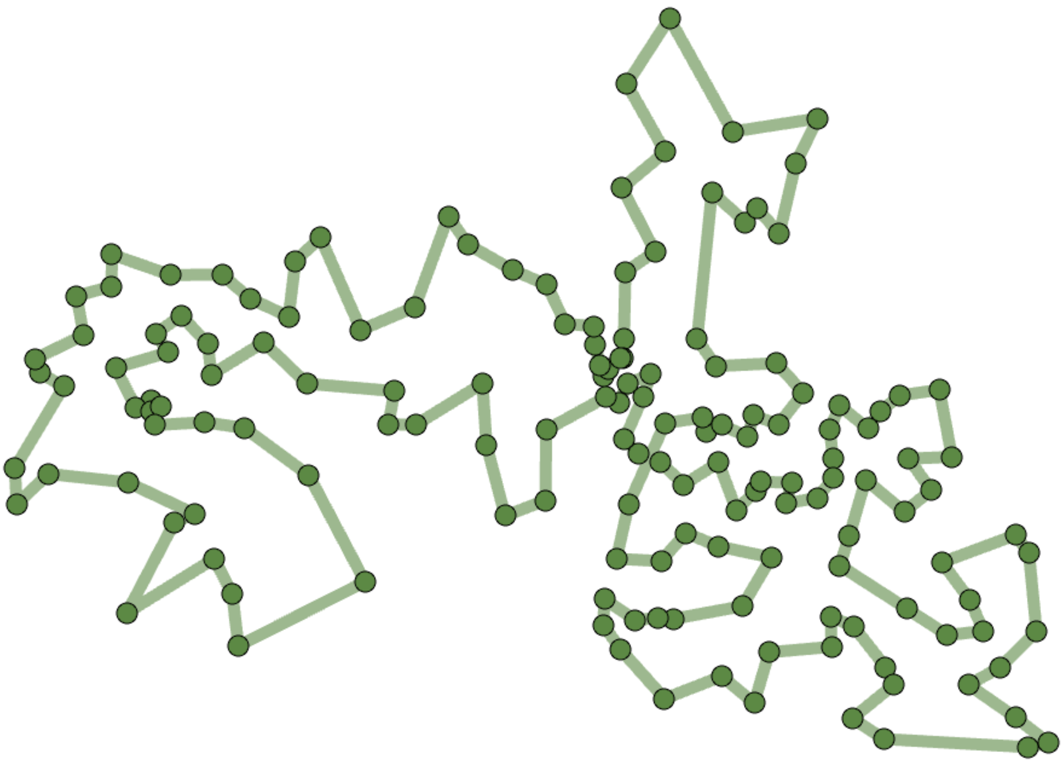
Obr. 4 – Výsledek metody Nearest Neighbor pro pošty v Libereckém kraji



Obr. 5 – Výsledek metody Best Insertion pro pošty v Libereckém kraji



Obr. 6 – Výsledek Network Analyst v SW ArcGIS Pro pro obce ČR



Obr. 7 – Výsledek Network Analyst v SW ArcGIS Pro pošty v Libereckém kraji

Závěr

V této úloze byly implementovány metody Nearest Neighbor a Best Insertion pro řešení problému obchodního cestujícího v programovacím jazyce Python, které byly otestovány na dvou vstupních datasetech tvořených 126 body obcí Česka s počtem obyvatel vyšším než 10 tisíc a 156 body pošt v Libereckém kraji. Nejlepší výsledky deseti opakování byly dále porovnány s výstupem nástroje *Network Analyst* v SW ArcGIS Pro.

Na základě výsledků délek vytvořených kružnic i vizuálního porovnání vykreslení těchto kružnic je patrné, že dle očekávání dopadla nejhůře metoda Nearest Neighbor. Délky kružnic vytvořené touto metodou jsou i v případě nejnižší hodnoty vždy výrazně vyšší než délky kružnic vytvořené metodou Best Insertion. Z vizualizací výsledků je také vidět, že kružnice vytvořené metodou NN se na několika místech kříží a tím narůstají i jejich délky. Metoda Best Insertion dosáhla lepších výsledků, ale jí vytvořená nejkratší kružnice je stále výrazně delší než výsledek ArcGIS Pro.

Možné vylepšení programu je například u metody Best Insertion, kde dochází v tomto případě k náhodnému výběru prvních třech bodů pro vytvoření počáteční kružnice. Místo toho by bylo možné kružnici inicializovat s využitím konvexní obálky.

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 (27.12.2022).

BAYER, T. (2022): Úvod do grafových algoritmů, prezentace k předmětu Geoinformatika, https://agony.natur.cuni.cz/~bayertom/images/courses/Geoinf/geoinf_grafy1.pdf (27.12.2022).

HOFMANN, K., PADBERG, M., RINALDI, G. (2001): Traveling salesman problem, http://seor.vse.gmu.edu/~khoffman/TSP_Hoffman_Padberg_Rinaldi.pdf (27.12.2022).

KOVÁŘ, P. (2012): Úvod do teorie grafů. Technická univerzita Ostrava, Západočeská univerzita v Plzni.

OSM (2022): Czech Republic, Open Street Map, Geofabrik Download Server, http://download.geofabrik.de/europe/czech-republic.html (3. 12. 2022).