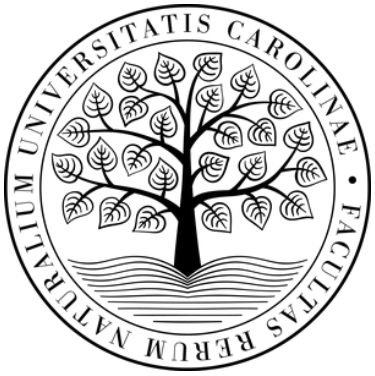
Univerzita Karlova

Přírodovědecká fakulta



Geoinformatika

Textová část první semestrální úlohy

Řešení problému obchodního cestujícího

Filip Zadražil

1.N-GKDPZ

2021/2022

1. Zadání úlohy

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`.

2. Popis a rozbor problému

Problém obchodního cestujícího, v angličtině Travelling salesman problem (=TSP), je název pro úplný problém (NP) v kombinatorické optimalizaci. Zjednodušeně je jeho podstatou nalezení nejkratšího existujícího propojení všech daných měst.

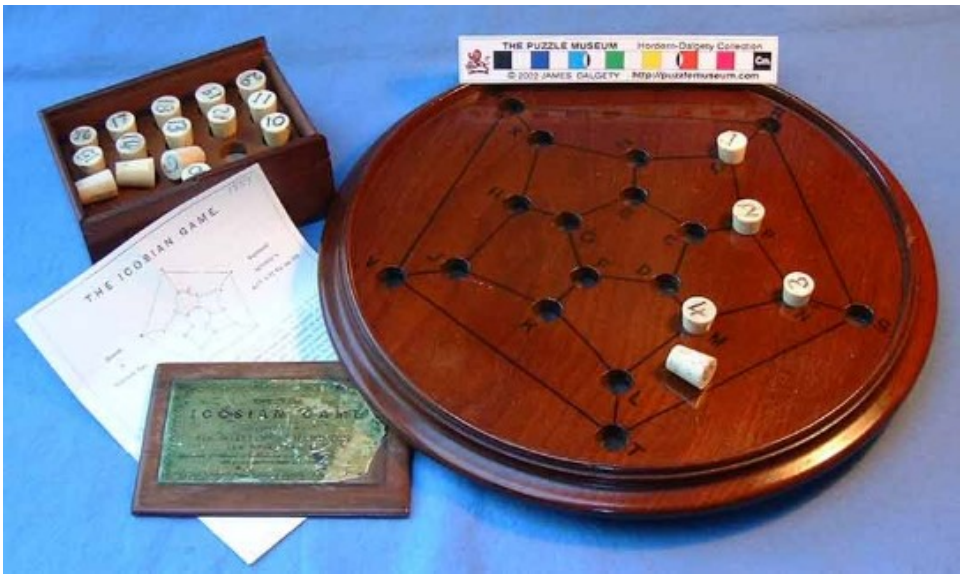
Jeho popularita je mimo jiné i důsledkem toho, že řešení tohoto problému nachází široké uplatnění v praxi. V oblasti geoinformatiky nás bude nejvíce zajímat aplikace v logistice a obecně při plánování tras. Příkladem v této oblasti může být např. kamionová doprava, zásilkové služby, roznos pošty atd. Uplatnění se však dá nalézt i mimo geografický prostor. Z mnoha případů lze uvést např. plánování pohybu dalekohledu či výroba mikročipů.

Vzhledem k tomu, že člověk cestuje mezi jednotlivými místy již odpradávna, není náhodou, že i TSP není záležitostí posledních desetiletí. První zmínka související s TSP byla publikována a matematicky rozebírána již v roce 1759 švýcarským matematikem Eulerem. Jednalo se o tzv. problém cesty šachového koně. Předmětem zkoumání bylo, jak skákat s šachovým koněm, aby navštívil všechna pole šachovnice a pouze jednou a vrátil se na původní pole. Vzhledem k tomu, že v 19. století, v období rychlé technologické evoluce, začal logicky růst i důraz na ekonomickou efektivitu. Šachového koně tak v této problematice nahradil člověk. Jedna z prvních dohledatelných zmínek souvisejících s tímto problémem je Příručka pro cestující prodavače z roku 1832, která se však nezabývá matematickou podstatou problému (Laporte 2006).

Těmi opravdu prvními, kteří matematicky zformulovali problém související s TSP byli irský matematik W. R. Hamilton a britský matematik Thomas Kirkman. Hamilton dokonce v roce 1857 vytvořil

icosiánskou hru založenou na nalezení (po něm pojmenované) Hamiltonovské kružnice. Jednalo se o dřevěnou desku s dvaceti políčky, na něž bylo nutné dosadit kameny označené čísly a určit tím jejich předepsané propojení (viz Obr. 1).

Obr. 1 – Hamiltonova icosiánská hra



zdroj: Golden (2021)

Dalším důležitým milníkem v historii vývoje TSP byla 30. léta 20. století. TSP byl zkoumán na univerzitách ve Vídni a Harvardu a bývá spojen zejména se jménem rakousko-amerického matematika Karla Mengera, který problém definoval a zároveň zhodnotil heuristiku nejbližšího souseda jako neoptimální. Poté se problémem výrazněji zabývali matematici Hassler Whitney („48 states problem“) a Merrill Flood („School bus routing problem“) na univerzitě v Princetonu a díky nim se opět prohloubil zájem o tuto problematiku (Lawler 1985). Dodnes používaný termín „Travelling salesman problem“ se vůbec poprvé objevil v knize z let 1931-32 od jednoho bývalého obchodního cestujícího. Pojednává se v ní o tom, jak být úspěšný a efektivní v tomto povolání. K širší veřejnosti se pojem dostal díky zprávě společnosti RAND Corporation z roku 1949, jejíž autorkou byla Julia Robinson a blíže pojednávala o Hamiltonově icosiánské hře (Larranaga et al. 1999).

Hlavní boom TSP pak nastal v 50. a 60. letech, kdy stejná společnost začala vyhlašovat soutěže a nabízet finanční odměny za vyřešení TSP mezi zadanými body (viz Obr. 2). V této době se o TSP začalo zajímat velké množství lidí nejen z vědeckých kruhů. Významnější posun v této problematice zaznamenali zejména vědci z RAND Corporation Dantzig, Fulkerson a Johnson, kteří ve svém článku popsali nejkratší cestu mezi 49 městy (48 měst z každého kontinentálního státu USA + Washington). Původně pracovali se 42 městy a vzdálenostmi po silnici. Poté bylo přidáno 7 měst, které leží na „hraně“ mezi Bostonem a Washingtonem (Lawler 1985; TSP 2021).

V 60. letech pak vznikl nový přístup, který se namísto hledání optimální cesty snažil o nalezení „dolní meze problému“. Šlo o nalezení takové délky, která je prokazatelně určena jako násobek optimální délky. Jednou z takovýchto metod bylo vytvořit minimální kostru grafu a následné zdvojnásobení délek všech jeho hran, což vytvořilo mez značící délku optimální cesty jako maximálně dvojnásobek váhy (délky) minimální kostry grafu. V roce 1976 publikovali Christofides a Serdyukov algoritmus, který dokáže nalézt cestu mezi body, jež je maximálně 1,5x delší než cesta optimální.

Obr. 2 – Plakát k veřejné soutěži o nalezení nejkratší cesty mezi 33 městy USA

Zdroj: TSP (2021)

Další významnou postavou problematiky TSP let 70. byl německý matematik Martin Grötschel. Ke zkoumání TSP využíval zejména algoritmy metody řezů (cutting planes) a metody mezí (branch and bound). Jeho kolegové Padberg a Rinaldi poté dokázali nalézt optimální cestu mezi 2392 body (rok 1987) (TSP 2021).

V roce 1991 vytvořil Gerhard Reinelt knihovnu TSPLIB, která obsahuje ukázkové instance pro TSP (a tomu podobné problémy) a slouží jako sbírka srovnávacích příkladů různých obtížností. Je využívána mnoha vědeckými skupinami k testování jejich výsledků.

V 90. letech 20. století započala práce na TSP výzkumná skupina sestávající z matematiků a informatiků, jakými jsou Applegate, Bixby, Chvátal a Cook. Ti společně vytvořili průlomový program *Concorde*, jehož úkolem je právě vyřešení TSP pro zadané množiny bodů. Tato skupina postupně pomocí tohoto programu vylepšovala rekordy pro nalezení optimální cesty pro co největší množství n bodů. Do výzkumu se poté zapojila i neméně významná trojice Helsgaun, Espinoza a Goycoolea a společně dokázali (především i díky stále se zkvalitňujícím dostupným technologiím) nalézt v roce 2017 optimální cestu pro 109399 bodů. Ta je pod označením „pla109399“ dostupná právě v Reineltově TSPLIB. Tato cesta využívá algoritmu LKH (jehož autorem je právě Helsgaun), která vychází z původní heuristiky Lin-Kernighan. Helsgaun pak se svým algoritmem dokonce drží rekord i v nalezení nejkratší cesty pro soubor 1 904 711 měst na planetě Zemi (TSP 2021).

Obecná formulace TSP říká, že máme *n* měst propojených silniční sítí o známé délce mezi jednotlivými městy a hledáme nejkratší cestu mezi všemi těmito městy za podmínek, že každé město bude navštíveno pouze jednou a počáteční a koncový bod cesty bude v totožném městě (Bayer 2021). Je potřeba také zmínit, že náročnost výpočtu optimální cesty mezi jednotlivými body se enormně zvyšuje s celkovým počtem bodů (Edelkamp, Schrödel 2012) .

TSP lze modelovat jako neorientovaný vážený graf, v němž jsou města uzly grafu, silnice mezi nimi jsou hranami grafu a vzdálenosti jsou váhou hrany. V této úloze je využíváno jednoduššího, symetrického modelu TSP, což znamená, že délka cesty z A do B je stejná jako z B do A. To platí pro všechny dvojice bodů. Pokud se však tyto délky liší, hovoříme o modelu asymetrickém (Pokorná 2008).

V této práci je užito dvou známých heuristik: Nearest Neighbor a Best Insertion. Obecný princip metody Nearest Neighbor spočívá v tom, že je vybrán libovolný uzel ze zkoumané množiny bodů. Pro tento uzel je nalezen nejbližší sousední uzel a cesta pokračuje právě do něj. V dalších uzlech, v nichž se ocitneme, je poté vybráno pokračování cesty pouze do nejbližšího z těch uzlů, jež dosud nejsou součástí Hamiltonovské kružnice (což je taková kružnice K_h, která obsahuje všechny uzly v grafu G):

$$\Delta w = \min_{u \notin Kh} w(u_i, u).$$

Když dojdeme do posledního nezpracovaného bodu, vedeme poslední hranu zpět do výchozího bodu, čímž kružnici uzavřeme (Bayer 2021).

Mnohem účinnější heuristikou je takzvaná Best Insertion. Její snahou se neustále udržovat Hamiltonovskou kružnici tvořenou i–1 uzly grafu G, kterou rozšiřuje v každém okamžiku přidáním vhodného uzlu tak, aby byl minimalizován nárust ohodnocení cesty

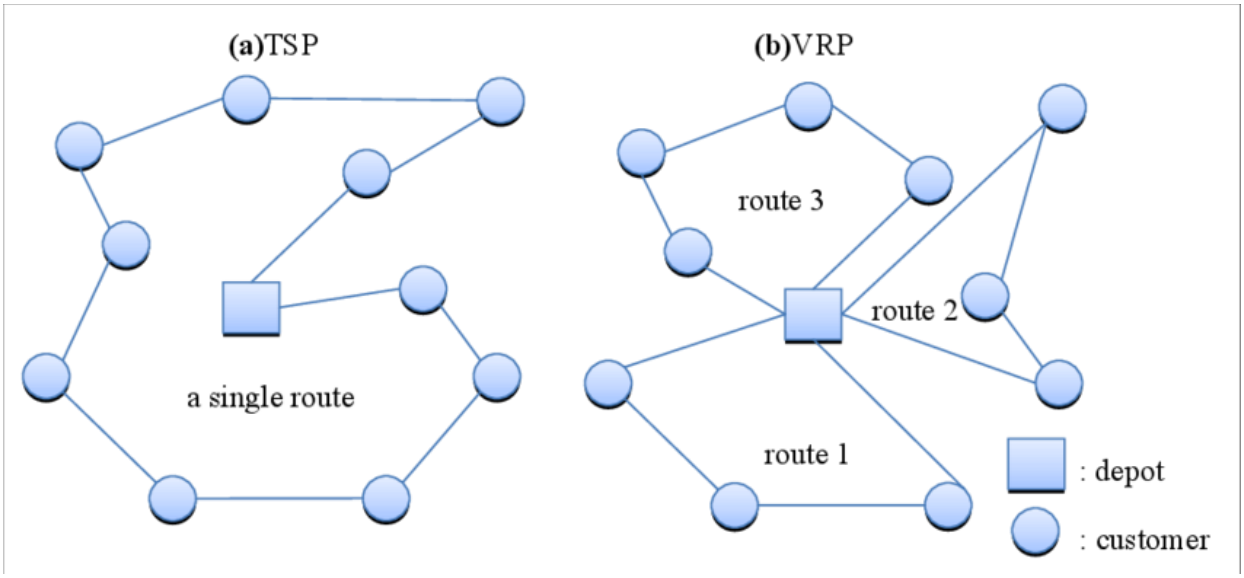
$$\Delta w = \min_{u \notin Kh} w(u_i, u) + w(u, u_i + 1) - w(u_i, u_i + 1).$$

Tato hodnota se následně přičte ke stávající hodnotě W a získáme tak aktualizovanou hodnotu W. Na rozdíl od metody NN ale nepřidává uzly na konec cesty, nýbrž do libovolného místa kružnice. Minimalizační kritérium vychází z trojúhelníkové nerovnosti. Výpočetní složitost je v zásadě podobná jako v případě NN. Vzhledem k náhodnému výběru uzlů *u* je metoda nedeterministická, což znaemená, že pro opakované spuštění generuje různé výsledky. Nalezená Hamiltonovská kružnice je však zpravidla kratší, neboť její segmenty se protínají mnohem méně častěji (Bayer 2021).

Mezi další známé heuristiky užívané při řešení problému TSP jsou například Greedy strategie, Christofidesův algoritmus, Lin-Kernighanova heuristika nebo heuristiky 2-opt a 3-opt, které fungují na základě lokálního vyhledávání a výměny *k* hran (Nilsson 2003). Problém obchodního cestujícího se však dá řešit i pomoci genetických algoritmů, simulovaného žíhání, metody včelí a mravenčí kolonie či již zmíněných přístupů založených na prohledávání grafu s ořezáváním (Pokorná 2008, Bayer 2021).

Úloha TSP má mnoho dalších modifikací. S těmi však tato úloha rovněž nepracuje. Jedná se např. o TSP s časovými okny. Při obchůzce zákazníků se bere jako váha hrany čas, za kterou obchodní cestující hranu překoná. V potaz se však bere i čas, který obchodní cestující u zákazníka stráví. Pomocí této modifikace jsme pak schopni vypočítat, v jakém čase dorazí obchodní cestující k *i*-tému zákazníkovi. Další modifikací je pak TSP s více obchodními cestujícími. Stejně jako u základního TSP, i zde platí, že se jednotliví obchodní cestující musí vrátit do výchozího místa a zároveň musí být každý zákazník navštíven pouze jednou. S touto modifikací souvisí problém podobný TSP s názvem Vehicle Routing Problem. Hlavní rozdíl mezi těmito úlohami můžeme odhalit na Obrázku 3. Další alternativou této modifikace je pak kombinace s možností více výchozích míst nebo kombinace s TSP s časovými okny (Pokorná 2008).

Obr. 3 – Rozdíl mezi úlohami TSP a VRP



Zdroj: Wan-Yu et al. (2014)

3. Řešení problému

Řešení zadaného problému bylo provedeno primárně metodami Nearest Neighbor (=NN) a Best Insertion (=BI). Pro obě metody byl na základě zadání a návodu vytvořen skript, pomocí nějž bylo možné metody aplikovat na vybrané datasety. Pro implementaci obou heuristik byl použit programovací jazyk Python. Jako vývojové prostředí byl zvolen software Visual Studio Code.

Vlastní zdrojový kód začíná naimportováním několika knihoven, jejichž funkce jsou v kódu využity. Na řádcích 9-21 následuje načtení příslušného shapefilu, z něhož je vytvořen data frame, jenž obsahuje název a souřadnice jednotlivých bodů vrtsvy, které jsou uloženy do seznamu seznamů. Na řádku 24 je definována funkce Nearest Neighbor, jejímž vstupním parametrem jsou souřadnice uzlů. Poté dochází k inicializaci zpracovaných a nezpracovaných uzlů a stanovení celkové váhy na hodnotu 0. Následuje náhodné zvolení startovního bodu a označení ho jako zpracovaný. Poté dochází na řádce 41 k aplikaci funkce *while()*, která prochází nezpracované body, dokud se v proměnné *u_nodes* žádné nezpracované body nenachází. Vnořenou funkcí je na řádku 46 *for* cyklus, který počítá vzdálenost všech dosud nezpracovaných bodů od toho uzlu, který je právě na konci cesty. Z těchto vypočítaných vzdáleností pak funkce *min()* nalezne nejmenší hodnotu, která je následně přičtena k celkové vzdálenosti. Závěrem cyklu je označení nalezeného bodu jako „zpracovaný“ (řádky 59-60), a poté dochází k dopočítání vzdálenosti mezi posledním zpracovaným bodem a startovním bodem. Poté už dochází jen k finálnímu vytisknutí výsledné váhy Hamiltonovské kružnice.

I metoda Best Insertion byla řešena velice obdobně. Jelikož je součástí stejného kódu, není potřeba otevírat nový soubor. Druhá použitá metoda na tu první přirozeně navazuje. Tato metoda je však trochu složitější. Po opětovné inicializaci zpracovaných a nezpracovaných uzlů totiž tentokrát dochází k náhodnému výběru hned 3 uzlů, a to pomocí prvního *for* cyklu na řádku 80. Po uložení startovního uzlu následuje druhý *for* cyklus (řádek 89), jenž vypočítá vzdálenost mezi prvními třemi uzly. Následuje řádkem 96 funkce *while()*, která stejně jako u NN stanovuje podmínku, že na konci celého procesu musí být v proměnná nezpracovaných uzlů prázdná. Poté postupně dojde vždy k vybrání náhodného bodu,

u nějž se pomocí již třetího *for* cyklu propočítá vzdálenost od jednotlivých hran (*Δw*) a tam, kde je hodnota nejnižší, do té hrany uzel přiřadí a spočítá celkovou délku kružnice. Na samém konci aplikace této metody dochází, stejně jako u NN k přidání startovacího uzlu na konec seznamu, čímž dojde k uzavření Hamiltonovské kružnice.

Na konci úlohy je důležité (a v případě TSP skoro nutné) výsledky vizualizovat. Tento úkon byl vyvolán speciálním příkazem ve tvaru *if __name__ == '__main__'*, který spustí níže vypsané příkazy (řádek 127). Dojde tedy ke spuštění funkcí NN a BI a uloží do proměnných z nich vrácené hodnoty *W* a *p_nodes.* Následně se vytvoří seznamy se souřadnicemi a číselným označením na řádcích 133-140. Tyto nové proměnné se pak už jen pomocí knihovny *matplotlib* nechají vykreslit do grafu (řádky 143-156).

4. Vstupy a výsledky

Pro otestování vytvořených algoritmů byly vybrány 2 datasety. Oba pochází z databáze ArcČR 500, verze 3.3. Prvním z datasetů je vrstva letišť na území ČR, kterých je 95 a je pojmenován *letiste_CR.shp* a druhým jsou obce v okresu Tábor s názvem *obce_tabor.shp*. Zde se nachází celkem 110 bodů.

Tab. 1 – Výsledky pro metodu Nearest Neighbor s tučně vyznačenými nejlepšími hodnotami

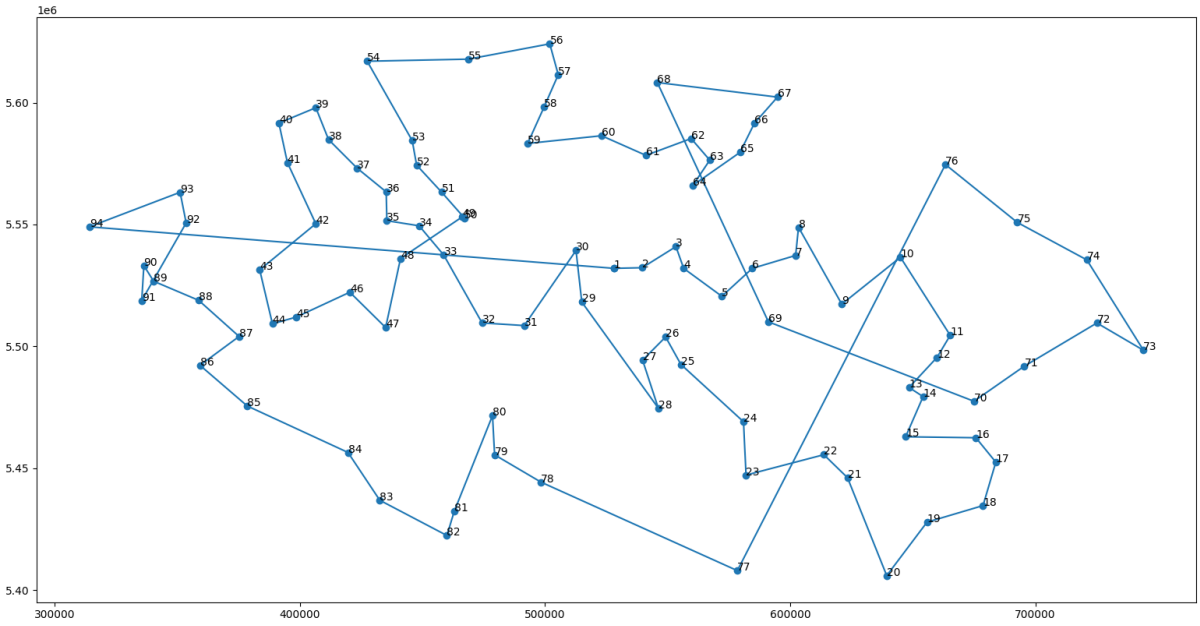
	Letiště ČR		Obce okresu Tábor	
	W [km]	k [%]	W [km]	k [%]
1	2803,947	121,391	396,643	116,130
2	2740,891	118,661	403,086	118,016
3	2806,734	121,512	424,557	124,303
4	2696,784	116,752	**391,539**	**114,636**
5	2666,749	115,451	418,939	122,658
6	2795,621	121,031	406,883	119,128
7	**2586,649**	**111,984**	406,717	119,079
8	2867,219	124,130	405,354	118,680
9	2637,423	114,182	418,034	122,393
10	2874,635	124,451	400,657	117,305

Tab. 2 – Výsledky pro metodu Best Insertion s tučně vyznačenými nejlepšími hodnotami

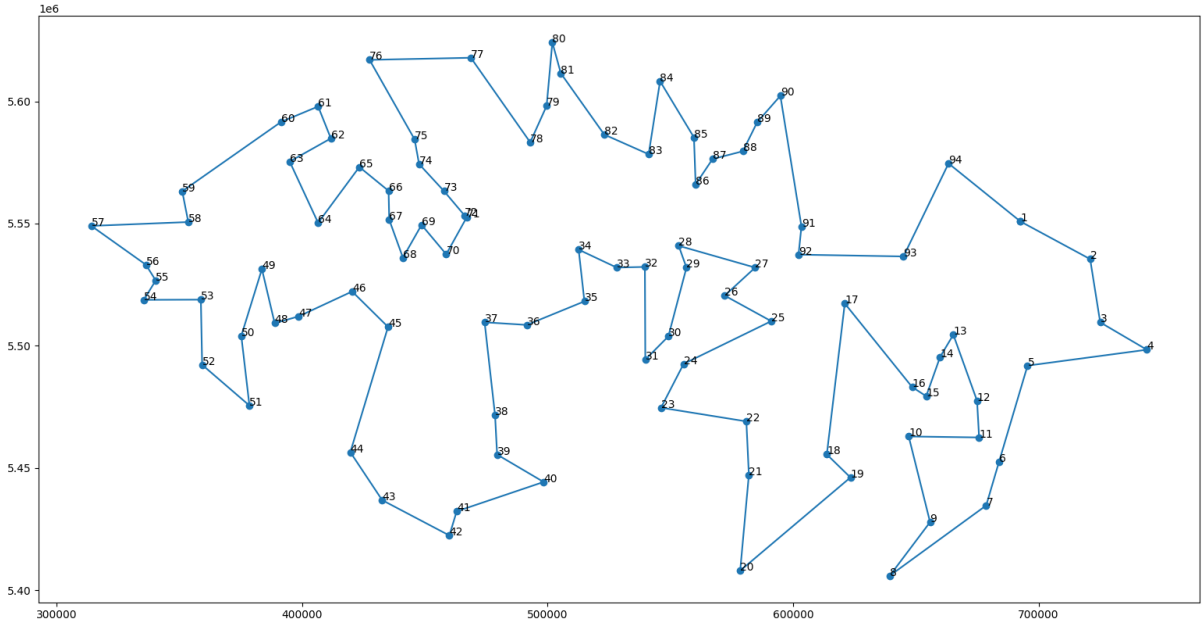
	Letiště ČR		Obce okresu Tábor	
	W [km]	k [%]	W [km]	k [%]
1	**2312,289**	**100,106**	368,186	107,798
2	2355,041	101,957	364,883	106,831
3	2369,396	102,578	369,852	108,286
4	2431,306	105,258	360,162	105,449
5	2405,302	104,133	374,442	109,630
6	2391,480	103,534	**348,072**	**101,909**
7	2365,586	102,413	372,372	109,024
8	2391,675	103,543	358,343	104,916
9	2313,909	100,176	362,294	106,073
10	2331,787	100,950	356,790	104,462

V Tabulkách 1 a 2 nalezneme výsledky pro oba datasety a obě heuristiky. V tomto případě se jedná o výsledky vzniklé na základě vlastního algoritmu spuštěném v programu *Visual Studio Code*. Z tabulek je pak zcela zřejmé, že výsledky pro Best Insertion dopadli mnohem lépe než výsledky pro Nearest Neighbor. Nejkratší vzdálenost Hamiltonovské kružnice pro letiště ČR měří 2312,289 km a pro obce okresu Tábor je to 348,072 km. Ukázky výsledků zkoumaných heuristik a datasetů, viz Obr. 3-6.

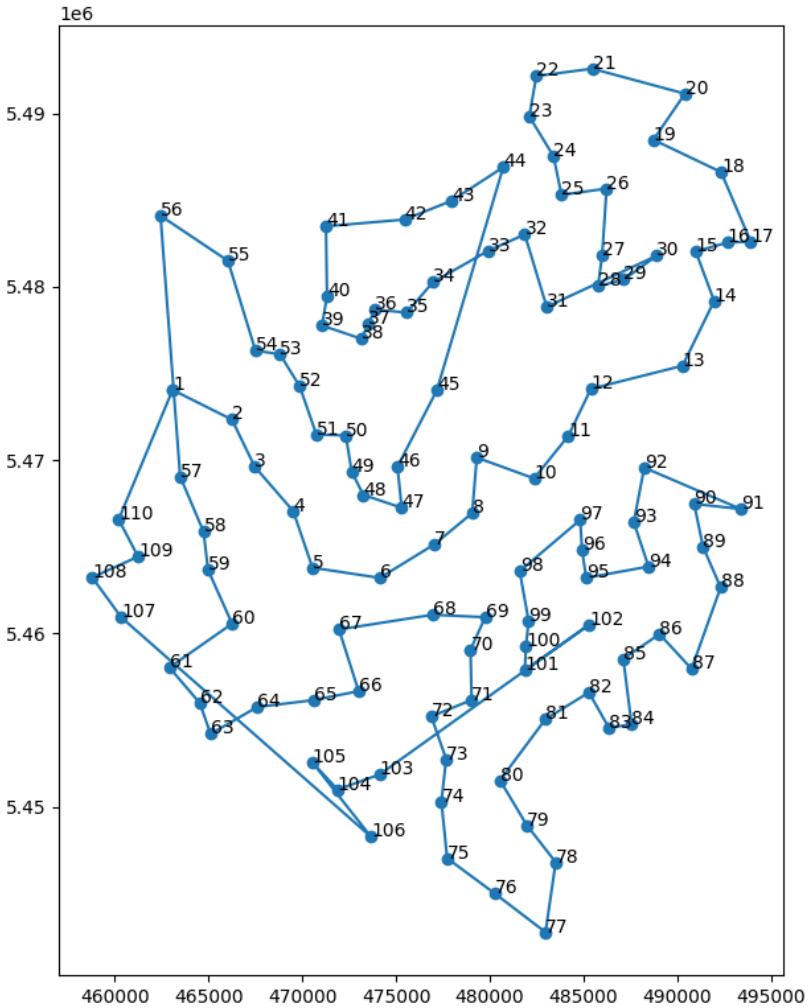
Obr. 3 – Ukázka výsledné cesty dle algoritmu Nearest Neighbor pro vrstvu letišť ČR



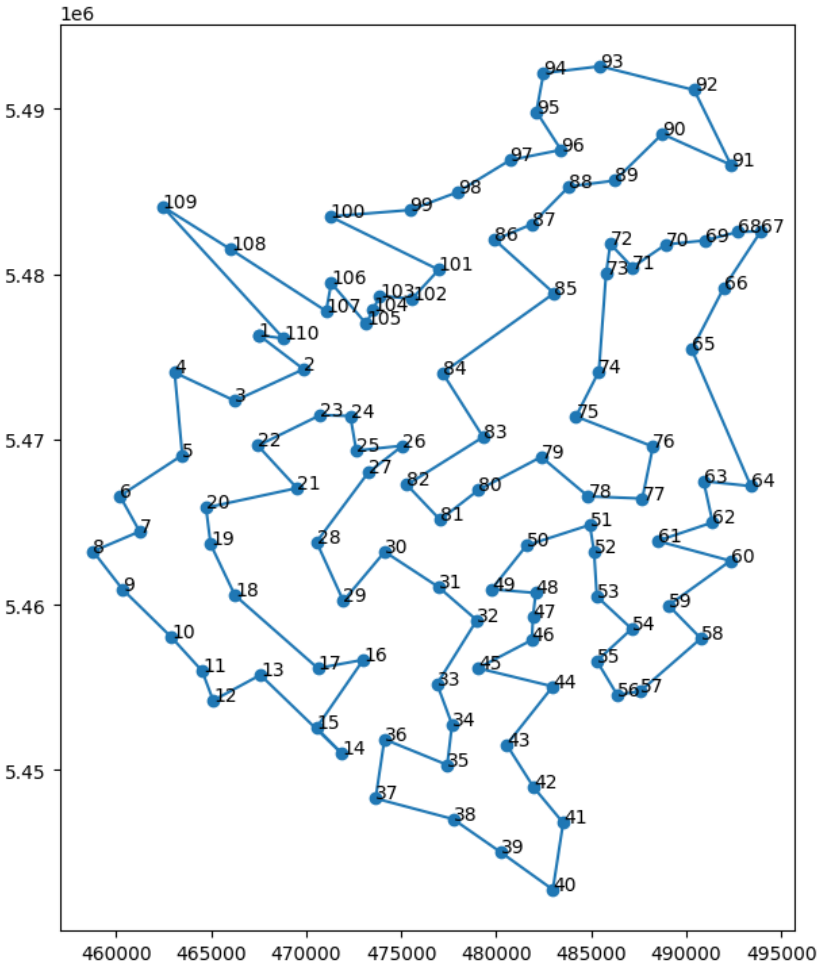
Obr. 4 - Ukázka výsledné cesty dle algoritmu Best Insertion pro vrstvu letišť ČR



Obr. 5 - Ukázka výsledné cesty dle algoritmu Nearest Neighbor pro vrstvu obcí okresu Tábor



Obr. 6 - Ukázka výsledné cesty dle algoritmu Best Insertion pro vrstvu obcí okresu Tábor



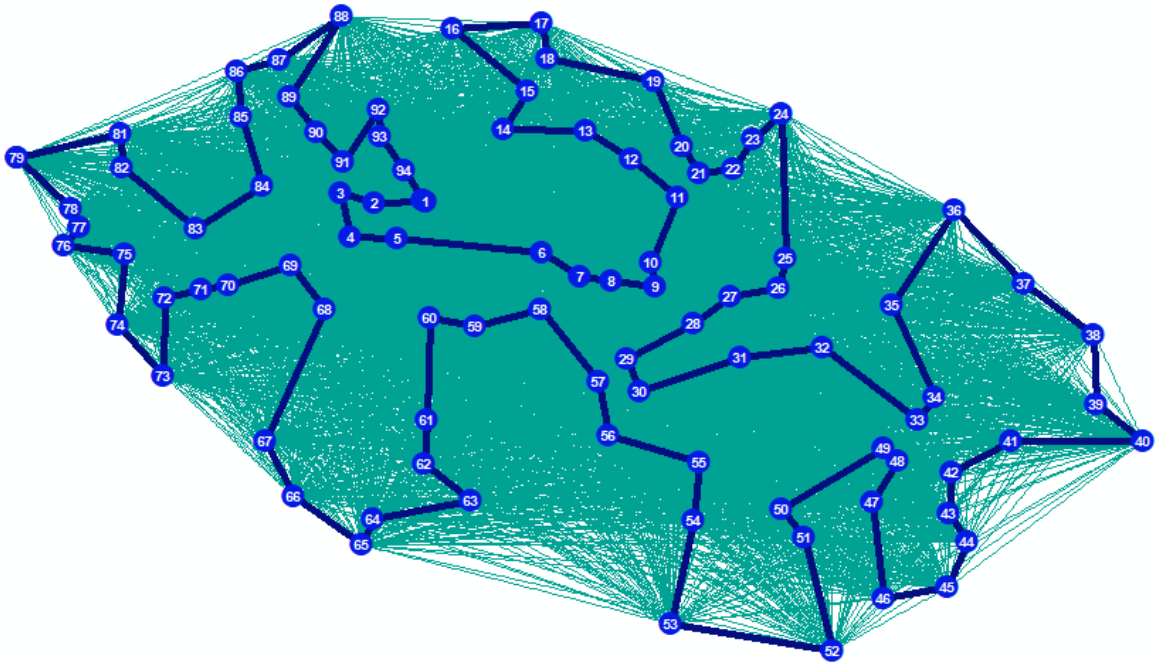
Pro náležité porovnání výsledků byla vytvořena nejkratší cesta mezi oběma bodovými vrstvami i v softwaru ArcMap 10.6.1. Pomocí nástroje *Network Analyst* a funkce *New Route* byla nalezena Hamiltonovská kružnice pro oba datasety. Jelikož ArcMap nedisponuje přímo vestavěnou funkcí pro řešení TSP, bylo nutné nejprve najít jeden z uzlů, který je na jednom z konců nejkratší hrany, ten duplikovat a nastavit ho jako startovní i koncový uzel. Teprve poté spustit analýzu. I přes tuto drobnou obtíž se však v obou případech GISovému softwaru podařilo vykreslit kratší cestu, než jakou na 10 pokusů vygeneroval vlastní algoritmus. I z toho důvodu byly pro výpočet hodnoty *k* stanoveny výsledné hodnoty z nástroje *Network Analyst* jako W_o. Samotná hodnota *k* se poté vypočítala jako podíl W/W_o. Vizualizace nejkratších nalezených cest a příslušné hodnoty dle nástroje Network Analyst je na Obrázcích 7-10. Hodnoty délky cest nalezneme v řádce *Total_Length* a to v jednotkách metrů. V pozadí uzlů v grafu je pak vrstva vzájemných spojnic mezi všemi body.

Tab. 3 – porovnání hodnot nejlepších výsledků pro jednotlivé metody/nástroje

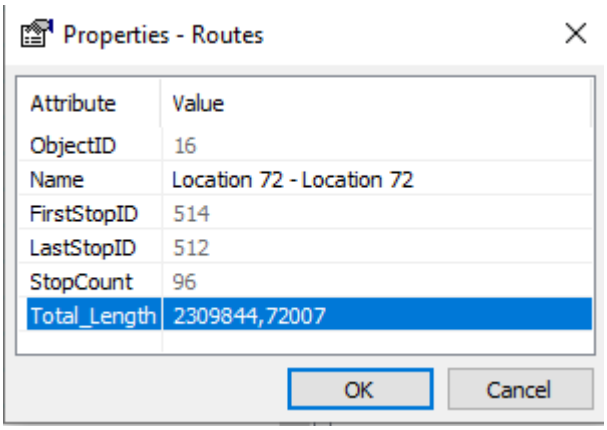
	Letiště ČR		Obce okresu Tábor	
metoda/nástroj	W [km]	k [%]	W [km]	k [%]
Nearest Neighbor	2586,649	111,984	391,539	114,636
Best Insertion	2312,289	100,106	348,072	101,909
Network Analyst (ArcMap)	2309,845	100,000	341,550	100,000

Porovnání výsledků nejlepších hodnot zobrazuje Tabulka 3. Z ní lze vypozorovat, že metoda Best Insertion je mnohonásobně účinnější než metoda Nearest Neighbor. Nejlepších výsledků však dosáhla cesta vypočítána nástrojem Network Analyst programu ArcGIS.

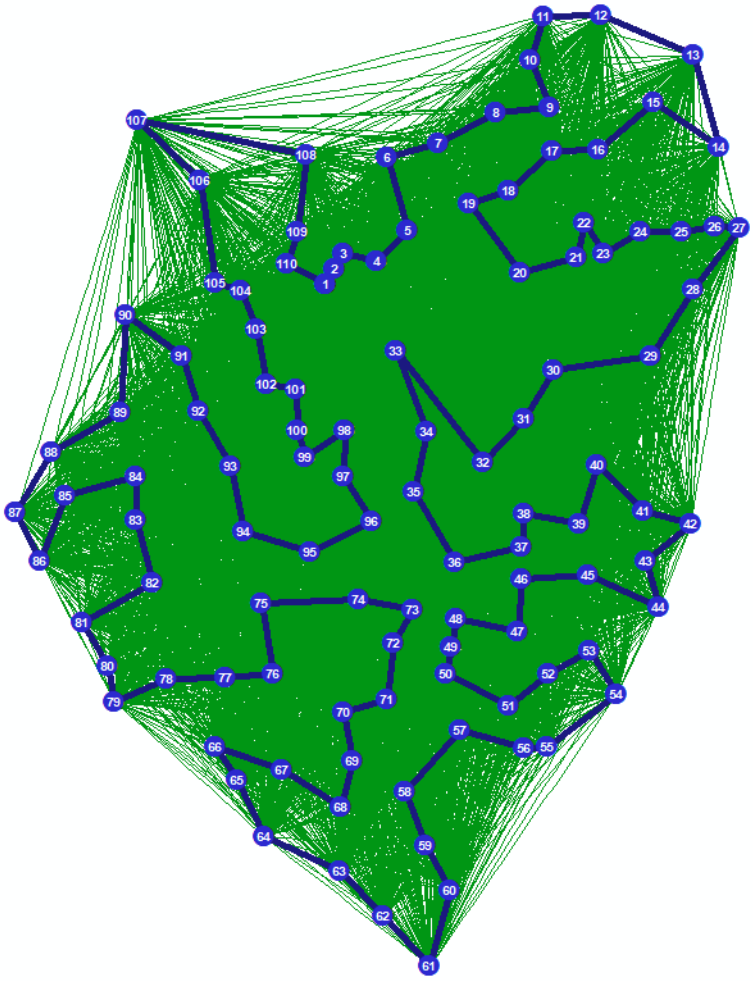
Obr. 7 – Nejkratší cesta dle nástroje Network Analyst pro vrstvu letišť ČR



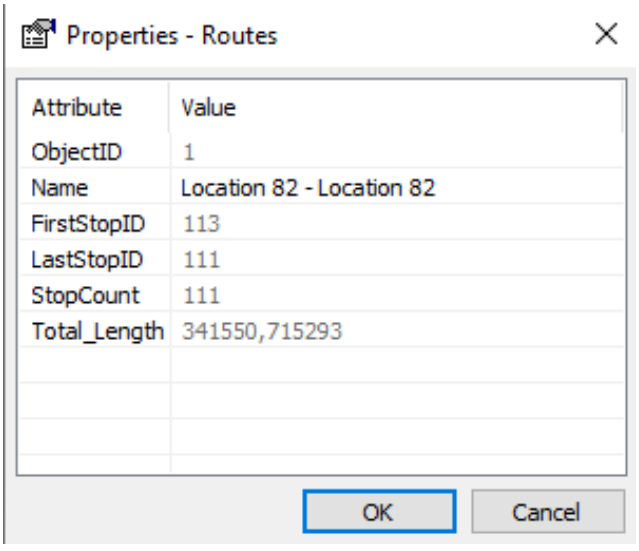
Obr. 8 – Výsledné hodnoty k nejlepší cestě dle Network Analyst pro vrstvu letišť ČR



Obr. 9 – Nejkratší cesta dle nástroje Network Analyst pro vrstvu obcí okresu Tábor



Obr. 10 – Výsledné hodnoty k nejlepší cestě dle Network Analyst pro vrstvu obcí okresu Tábor



5. Závěr

Tato práce zaměřená na řešení problému obchodního cestujícího v úvodní části podrobněji popisuje vývoj řešení této problematiky od samotného počátku. V dalších kapitolách se poté zaměřuje na samotné metody přístupu k řešení TSP. Na základě poskytnutého návodu a znalostí jazyka Python se podařilo vytvořit funkční kód, který aplikoval zadané metody (Nearest Neighbor a Best Insertion) na vybrané datasety. Zároveň byla pro účely porovnání vytvořena Hamiltonovská kružnice i v prostředí ArcGIS a bylo tak možné přímo porovnat konkrétní přístupy k řešení TSP a statisticky je zhodnotit.

Dle shrnutých výsledků lze označit metodu Best Insertion pro řešení problému obchodního cestujícího jako značně efektivnější oproti metodě Nearest Neighbor. Řečí čísel je metoda Best Insertion až o 11 % lepší než metoda Nearest Neighbor. Naopak oproti výsledku z Network Analyst je zaostává metoda Best Insertion o ne více jak 2 %. V případě letišť je to dokonce pouhá desetina procenta. Tento výsledek však byl předem očekávaný, neboť už ve 30. letech 20. století se Karl Menger zmiňoval o metodě nejbližšího souseda jako o neefektivní pro řešení TSP. Překvapením je pak kvalita nástroje Network Analyst, která dokázala vytvořit lepší výsledky než vytvořené algoritmy (po deseti opakování) pro oba datasety. Nutno však zmínit, že pokud by se nechal vytvořený kód spustit vícekrát, s velkou pravděpodobností by po několika (možná desítkách) dalších opakování zřejmě pokořil délku nejkratší cesty z Network Analyst.

Zkvalitnění práce lze určitě hledat např. ve vytvořeném kódu, který by se dal jistě napsat přehledněji, aby byl pro uživatele lépe čitelný. Příkladem je například použití knihoven dostupných v Pythonu, které by nám zároveň pomohli urychlit výpočet.

Další možná vylepšení skýtá metoda Best Insertion. Tu je totiž možno derandomizovat a vybrat v každém kroku uzel u, který by minimalizoval hodnotu Δw. Nevýhodou by však bylo přílišné zvýšení výpočetní náročnosti. Možným vylepšením by byla i aproximace iniciálního trojúhelníku konvexní obálkou.

Přínosem by bylo jistě i otestování vlastních výstupů s výsledky, které poskytují datasety pro TSP, které jsou volně dostupné v knihovně TSPLIB. Pro lepší porovnání by pak bylo jistě dobré vizualizovat jednotlivé výsledné cesty v totožném výstupu.

6. Použitá literatura

BAYER, T. (2021): Problém obchodního cestujícího, konstrukční heuristiky: Stručný návod na cvičení. Univerzita Karlova, Přírodovědecká fakulta, katedra geoinformatiky a kartografie. Dostupné z: https://web.natur.cuni.cz/~bayertom/images/courses/Geoinf/tsp_uloha.pdf (cit. 6. 1. 2022)

EDELKAMP, S., SCHRÖDEL, S. (2012): Heuristic search: Theory and application. Morgan Kaufmann/Elsevier.

GOLDEN, J. (2021): Icosian game. Dostupné z: https://www.geogebra.org/m/u3xggkcj (cit. 6. 1. 2022)

LAPORTE, G. (2006): A Short History of the Traveling Salesman Problem. Canada Research Chair in Distribution Management, 1-25.

LARRANAGA, P., KUIJPERS, C., M., H., MURGA, R., H., INZA, I., DIZDAREVIC, S. (1999): Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. Artificial Intelligence Review, 13, 129–170.

LAWLER, E., L. (1985): The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization. John Wiley & Sons, New York City.

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

POKORNÁ, P. (2008): Problém obchodního cestujícího pomocí metody Mravenčí kolonie. Bakalářská práce. Univerzita Pardubice, fakulta ekonomicko-správní.

TSP (2021): Travelling Salesman Problem. Webová stránka. Dostupné z: https://www.math.uwaterloo.ca/tsp/index.html (cit. 6. 1. 2022)

WAN-YU, L., CHUN-CHENG, L., CHING-REN, C., YOU-SONG, T. (2014): Minimizing the Carbon Footprint for the Time-Dependent Heterogeneous-Fleet Vehicle Routing Problem with Alternative Paths. Sustainability, 6, 7, 4658-4684.