

Best insertion – deterministická úprava

Tato strategie pro řešení problému obchodního cestujícího vychází z klasické Best Insertion metody. Společnou mají hlavní metodu. Je vždy vybrán uzel u z doposud vytvořené kružnice je vybráno místo, kam bude přidán, aby její váhu zvětšil o co nejméně podle vztahu

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

který vychází z trojúhelníkové nerovnosti. V běžné metodě by byl uzel u na toto místo rovnou přidán. V této úpravě je však jen zapamatován název uzlu u , jeho případná pozice, kdyby měl být přidán a o kolik by kružnici prodloužil. Tento postup je aplikován postupně na všechny uzly z množiny doposud nenavštívených uzlů a když je nalezen uzel, který dosáhl menšího protažení kružnice, je uložen jeho název a pozice, kam by měl být přidán. Po prohledání všech uzlů je uzel, který byl nalezen jako nejlepší přidán do kružnice, je přičtena jeho váha a je označen jako navštívený. Celý tento proces se opakuje, dokud zde je nějaký nenavštívený uzel.

Takto upravený kód je deterministický v případě, že inicializační kružnice je stejná.

Pseudokód:

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* (může být nahrazeno 3 pevně vybranými body, pro zajištění determinismu)

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:

Minimální vzdálenost M nastav na nekonečno

Projdi všechny doposud nenavštívené uzly u :

Minimální vzdálenost aktuálního uzlu nastav na nekonečno

Projdi doposud přidané uzly v do kružnice Q :

Spočítej novou vzdálenost mezi uzly v_i a v_{i+1} a odečti původní vzdálenost v_i a v_{i+1}

Pokud je tato vzdálenost menší než minimální vzdálenost uzlu u :

Nastav ji jako novou minimální vzdálenost uzlu u

Zapamatuj si jeho potenciální pozici

Pokud je minimální vzdálenost uzlu u menší než M :

Změň hodnotu M na minimální vzdálenost uzlu u

Zapamatuj si název uzlu u

Zapamatuj si jeho potenciální pozici jako *Pozice*

Nastav zapamatovaný uzel u jako navštívený

Vlož u do listu Q na *Pozici*

Přičti vzdálenost M k celkové vzdálenosti W

Uzavři kružnici přidáním prvního uzlu na konec Q

Vrať celkovou vzdálenost a Hamiltonovskou kružnici

Výsledky

Tato metoda byla porovnána s její neupravenou nedeterministickou verzí. Jako inicializační kružnice byly otestovány 3 možnosti. 1. byly vybrány 3 „náhodné“ body $[0, 1, 2]$. 2. byla inicializační kružnice nastavena 4 body s extrémními souřadnicemi x, y . Jedná se o body. Poslední inicializační kružnice byla definována počátečním bodem $u = 20$ a k němu byly nalezeni 1. 2 nejbližší sousedé.

Obě metody byly testovány pomocí stejných datasetů. `Coord_cernosice.csv`, který obsahuje souřadnice obcí spadajících do správního obvodu Černošic a Říčan, a dataset s názvem `coord_10000.csv`, který obsahuje souřadnice obcí s počtem obyvatel větším než 10 000.

Porovnávané skripty se nazývají `best_insertion.py` pro neupravenou metodu a `best_insertion_deterministic.py` pro upravenou metodu.

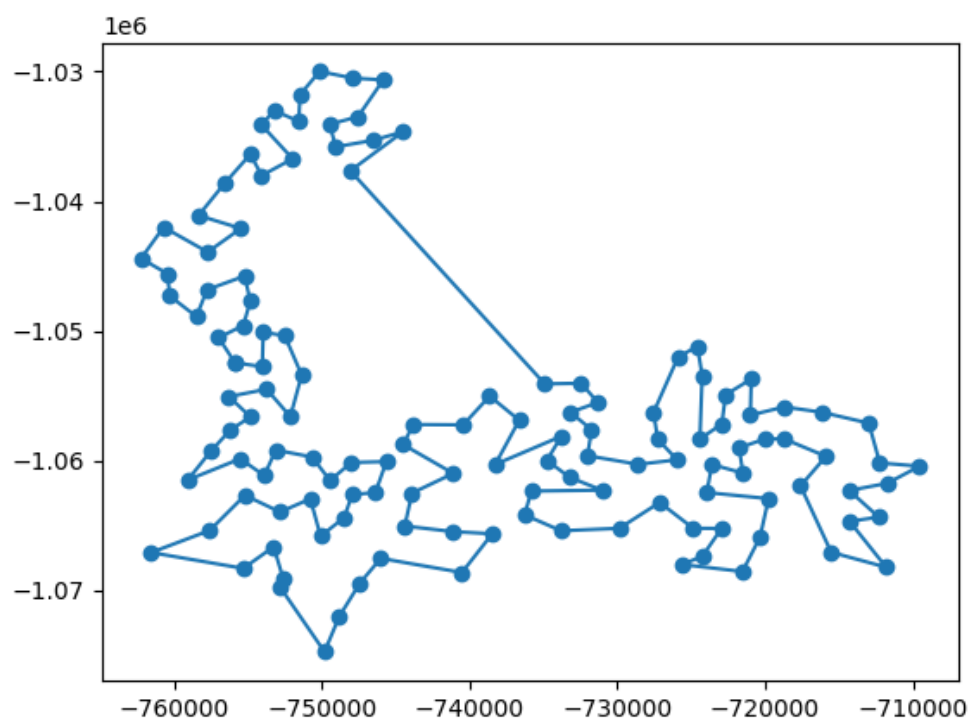
Výsledky nedeterministické metody byly zaznamenány 10x, protože pokaždé vycházejí trochu jinak. Výsledky všech typů inicializační kružnice jsou v tabulce č. 1. Při porovnání nejlepšího výsledku původní metody a výsledku upravené metody lze pozorovat, že upravená metoda zpravidla nedosahuje lepších výsledků. V závislosti na datasetech se tyto rozdíly mění. U datasetu `coord_cernosice.csv` je tento rozdíl pro 1. typ asi 6 %. U datasetu `coord_10000.csv` je tento rozdíl ještě patrnější a činí skoro 15 %. Pro druhý typ je rozdíl u datasetu `coord_cernosice.csv` necelých 2,5 % a u datasetu `coord_10000.csv` necelé 4 %. Pro třetí typ byl u `coord_cernosice.csv` výsledek lepší o 1,5 % a u datasetu `coord_10000.csv` byl výsledek o 14 % horší.

Výsledky 1 typu inicializační kružnice jsou vizualizovány na obrázcích 1 až 4. 2. typu na obrázcích 5, 6 a 3. typu na 7, 8.

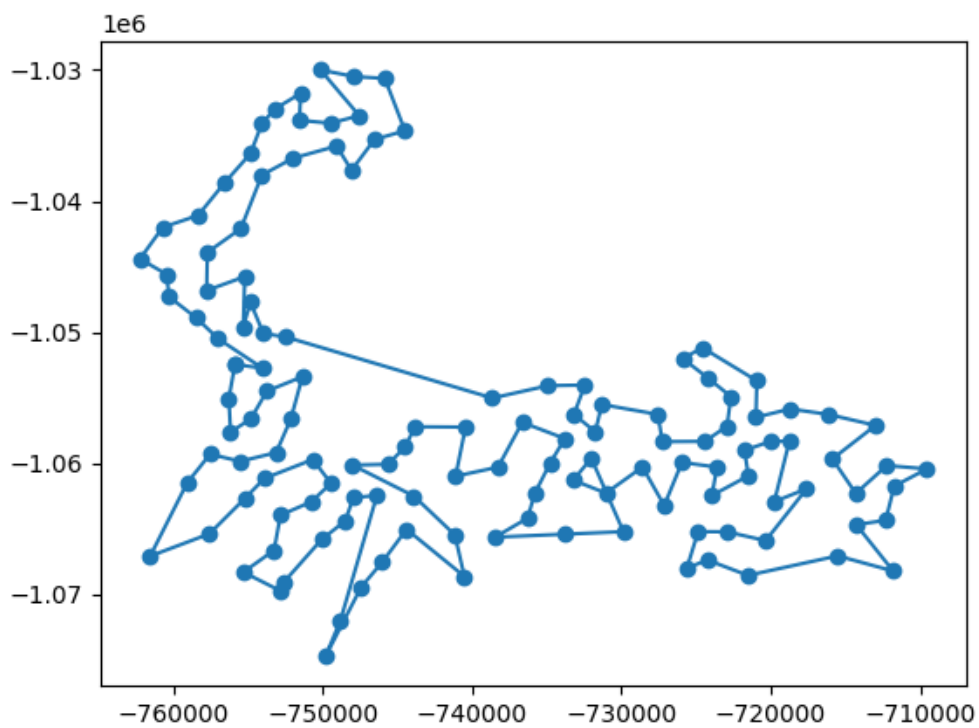
počátek = 0,1,2	coord_cernosice.csv [m]	coord_10000.csv [m]
1	378244,912	2763631,474
2	388184,608	2876059,844
3	376883,005	2790978,960
4	389680,391	2709744,650
5	384878,335	2736771,426
6	382995,811	2825940,871
7	378603,234	2776533,946
8	368087,892	2748125,681
9	395509,868	2710962,246
10	384091,715	2739865,969
Upravená BI typ 1	389859,589	3116093,162
Upravená BI typ 2	376597,059	2810236,875
Upravená BI typ 3	362744,293	3092201,409

Tabulka 1 porovnání výsledků obou metod

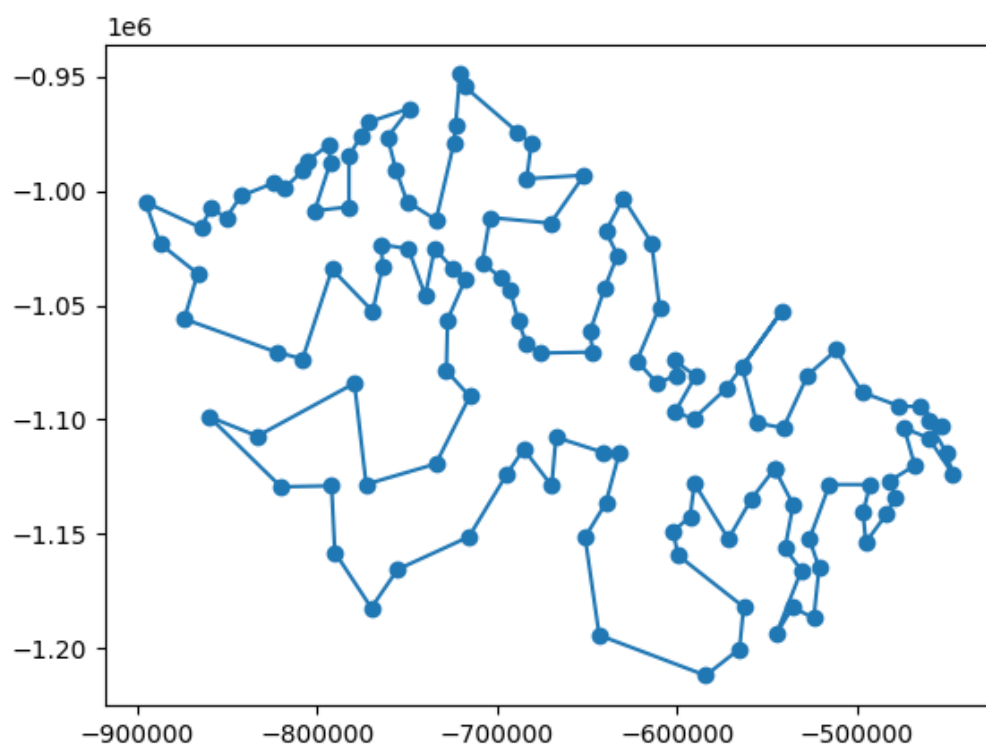
Velké rozdíly jsou závislé na datech. Deterministická úprava sice hledá nejmenší prodloužení kružnice, ale stále nebere v úvahu další kroky. Proto sice přidá vždy aktuálně nejlepší bod, ale v dalších krocích tento bod již nebude dostupný a celkový výsledek může vyjít hůře. Toto je dobře pozorovatelné na druhém datasetu a na obrázcích 3 a 4.



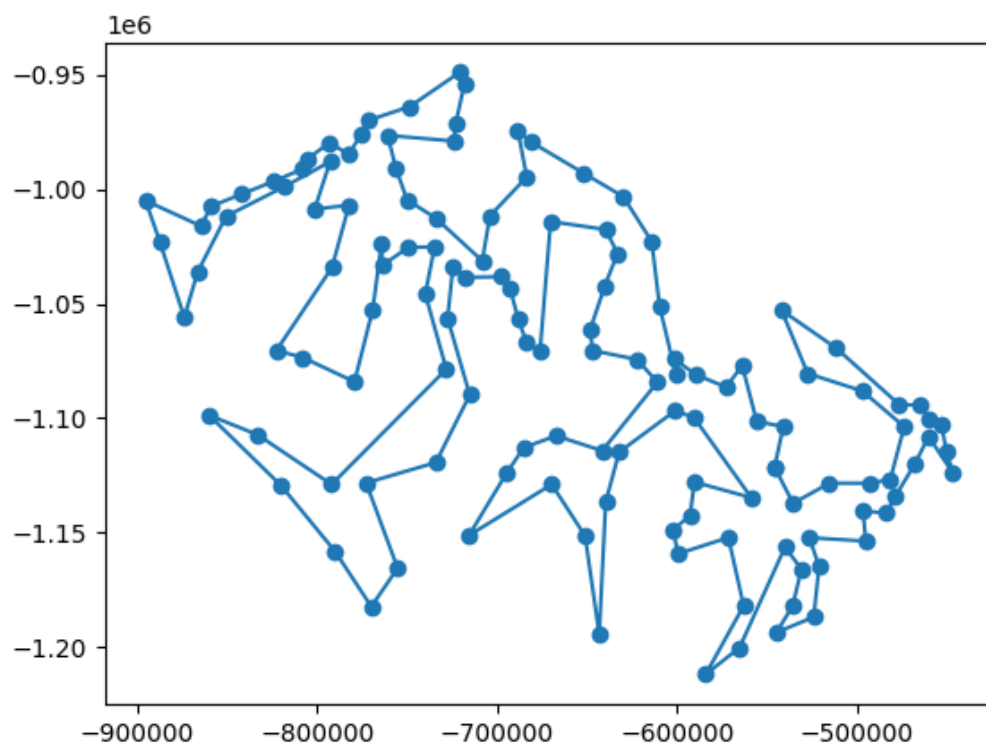
Obrázek 1 Dataset coord_cernosice.csv pomocí nedeterministické metody



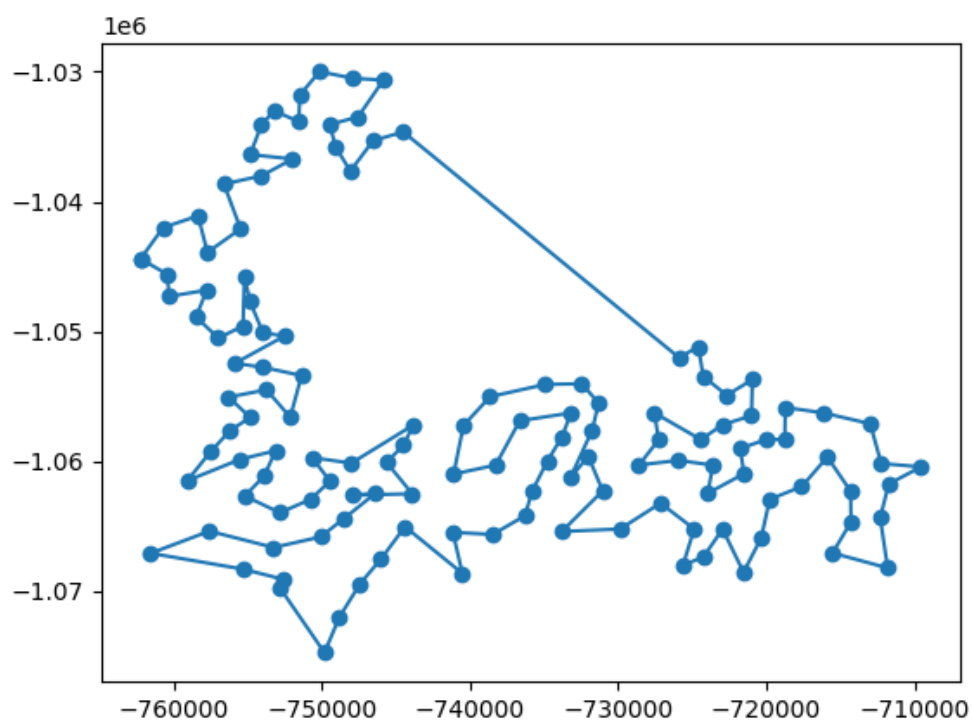
Obrázek 2 Dataset coord_cernosice.csv pomocí deterministické metody 1. typ



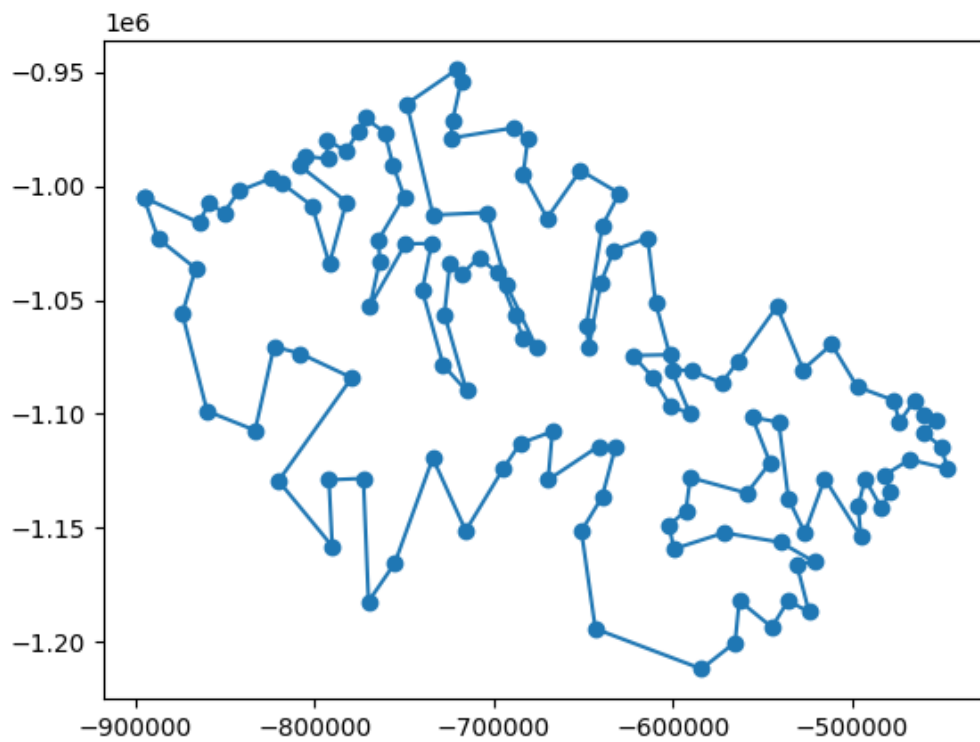
Obrázek 3 Dataset coord_10000.csv pomocí nedeterministické metody



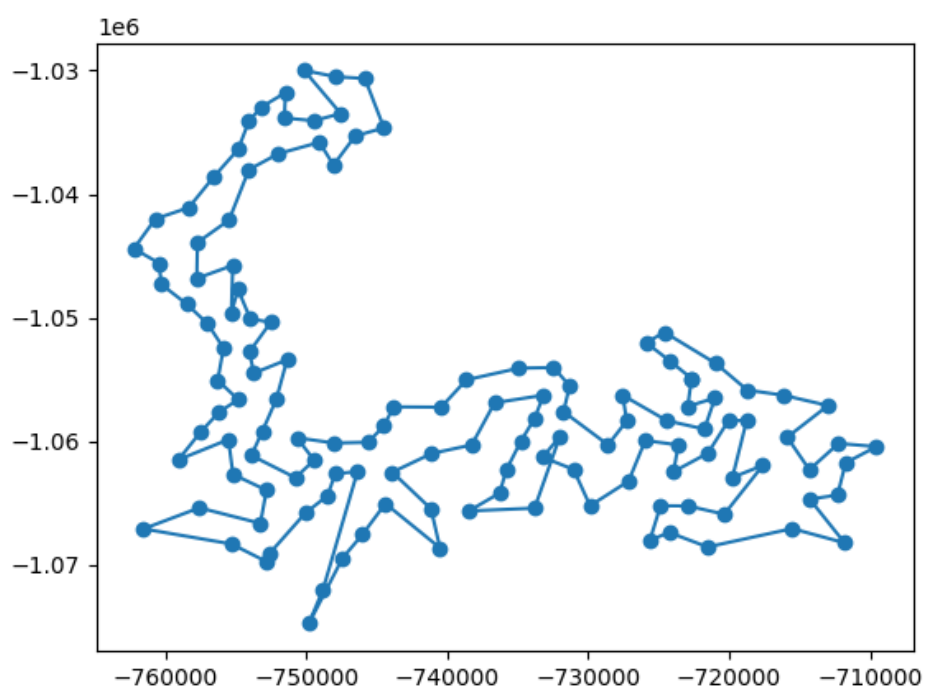
Obrázek 4 Dataset coord_10000.csv pomocí deterministické metody 1. typ



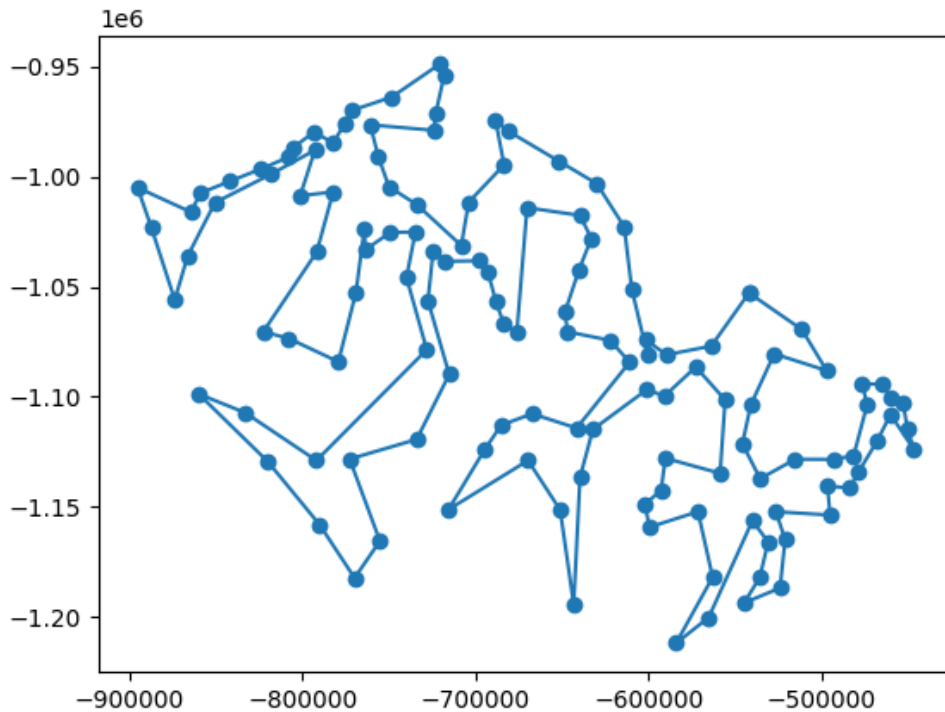
Obrázek 5 Dataset coord_10000.csv pomocí deterministické metody 2. typ



Obrázek 6 Dataset coord_10000.csv pomocí deterministické metody 2. typ



Obrázek 7 Dataset coord_10000.csv pomocí deterministické metody 3. typ



Obrázek 8 Dataset coord_10000.csv pomocí deterministické metody 3. typ