

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**  
FAKULTA STAVEBNÍ, OBOR GEODÉZIE, KARTOGRAFIE A GEOINFORMATIKA  
KATEDRA SPECIÁLNÍ GEODÉZIE

název předmětu

GEOINFORMATIKA

název úlohy

Úloha 4: Nejkratší cesta grafem

akademický rok	semestr	vypracoval(a)	Datum	klasifikace
2023/24	Zimní	Jan Koudelka, Vojtěch Müller	29.12.2023	

# Technická zpráva

## 1 Zadání

Implementujte Dijkstra algoritmus pro nalezení nejkratší cesty mezi dvěma uzly grafu. Vstupní data budou představována silniční sítí doplněnou vybranými sídly.

Otestujte různé varianty volby ohodnocení hran grafu tak, aby nalezená cesta měla:

- Nejkratší Euklidovskou vzdálenost,
- Nejmenší transportní čas.

Ve vybraném GIS konvertujte podkladová data do grafové reprezentace představované neorientovaným grafem. Pro druhou variantu optimální cesty navrhnete vhodnou metriku, která zohledňuje rozdílnou dobu jízdy na různých typech komunikací. Výsledky (dvě různé cesty pro každou variantu) umístěte do tabulky a vlastní cesty vizualizujte. Dosažené výsledky porovnejte s vybraným navigačním softwarem.

Jako doplňkové úlohy implementujte:

- Nalezení nejkratší cesty mezi všemi dvojicemi uzlů.
- Nalezení minimální kostry.
- Využití heuristiky Weighted Unin.
- Využití heuristiky Path Compression.

## 2 Popis

Hledání nejkratší cesty grafem vychází z teorie grafů. Jedná se zde o určení nejkratší cesty v jakékoli reálné situaci, kterou lze reprezentovat grafem. Nejběžněji se graf používá pro silniční síť, kde uzly reprezentují významné body a hrany reprezentují samotné silniční úseky. Řešení nejkratší cesty se pak používá při navigaci v kterémkoli softwaru. [1]

## 3 Postup

### 3.1 Vytvoření grafové reprezentace

K vytvoření grafu byl použit software ArcGIS Pro, kde byla vytvořena bodová vrstva s významnými sídly, křižovatkami a hranicemi obcí. Následně byla vytvořena liniová vrstva, která byla naplněna jednotlivými silnicemi, které spojují body. Do vrstvy hran byly následně přidány atributy pro ohodnocení jednotlivých hran. Pomocí jazyka Python byly následně naplněny grafy s jednotlivými ohodnoceními. Graf byl reprezentován seznamem souřadnic vrcholů a seznamem vrcholů s informacemi o sousedech a ohodnocení k jednotlivým sousedům.

### 3.1.1. Euklidovská vzdálenost

Byl přidán atribut délky pro jednotlivé úseky siliční sítě a tento atribut poté plní roly ohodnocení grafu.

### 3.1.2 Transportní čas

Byly přidány atributy pro výpočet metriky:

- Délka jednotlivých úseků. -  $d$
- Přímá délka mezi uzly. -  $D$
- Maximální povolená rychlost na úseku. -  $v$

Z maximální povolené rychlosti a délky úseku byl nejprve vypočítán čas, který je potřeba na projetí jednotlivých úseků, a to ze vzorce

$$t = \frac{d}{v}$$

Následně byl vypočítán koeficient křivolakosti ze vzorce

$$k = \frac{D}{d}$$

A poté byl vypočítán výsledný čas průjezdu úseku, který byl upraven o koeficient křivolakosti.

$$T = k \cdot t$$

Hodnota  $T$  byla poté použita jako ohodnocení grafu.

## 3.2 Dijkstra algoritmus [2]

Dijkstra algoritmus hledá nejkratší cestu mezi uzly grafu. Využívá princip, při kterém se postupně zpřesňuje odhad nejkratší délky. Od prvního uzlu procházíme graf prodlužováním postupně po jednom uzlu a počítáme ohodnocení, které je potřeba, abychom se k tomuto uzlu dostali. Najdeme tedy vždy předchůdce určitého uzlu, který zajišťuje nejkratší přístup.

K výpočtům se používá princip relaxace, kdy v uzlech procházených prioritní frontou testujeme, zda platí

$$d[v] > d[u] + w[u][v]$$

Hodnoty  $d$  představují aktuální odhad ohodnocení pro cestu do těchto uzlů a hodnota  $w$  je ohodnocení hrany mezi těmito uzly. Pokud platí psaný předpoklad, tak se aktualizuje odhad na

$$d[v] = d[u] + w[u][v]$$

a pro uzel  $v$  vzniká nový předchůdce

$$p[v] = u$$

Samotné hledání nejkratší cesty probíhá tak, že se nejprve nastaví odhad ohodnocení všech uzlů na nekonečno, poté se u prvního uzlu nastaví ohodnocení nula a prochází se nejprve sousední uzly. K nim

se zapíše hodnota odhad ohodnocení, pokud je splněna výše psaná podmínka. Takto se postupně prochází celý graf, z kterého bude výstupem list předchůdců.

Následně byla vytvořena funkce, která ze seznamu předchůdců určuje seznam uzlů, přes které prochází nejkratší cesta. Z těchto uzlů byly poté pomocí knihovny matplotlib vytvářeny výsledné grafy.

### 3.3 Nalezení nejkratší cesty mezi všemi dvojicemi uzlů

Pomocí dvou cyklů for byla zjištěna nejkratší cesta mezi všemi uzly.

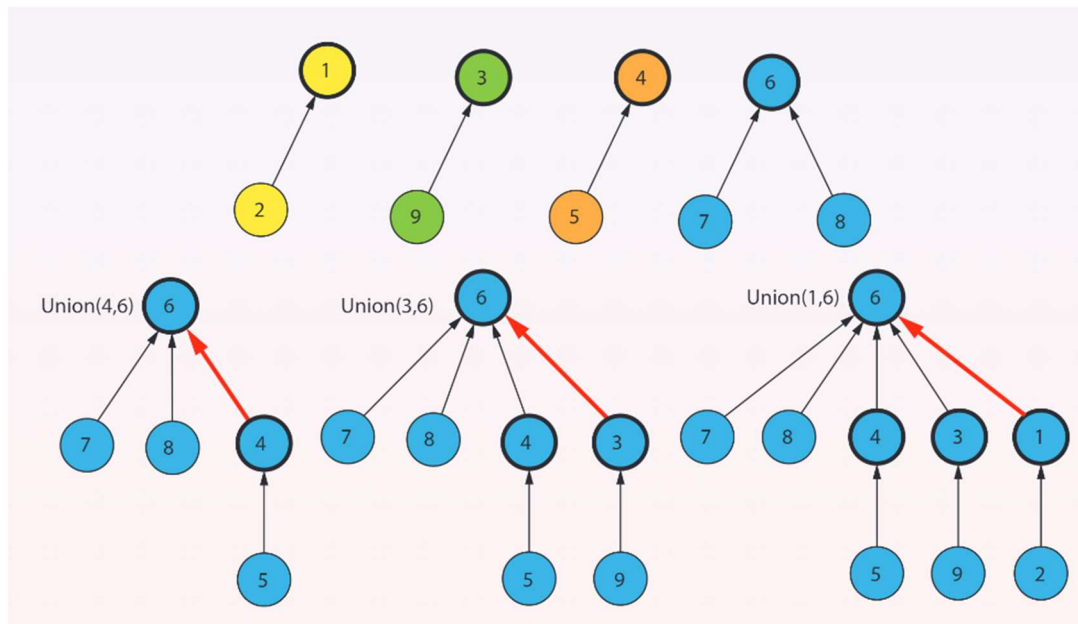
### 3.4 Nalezení minimální kostry [2]

K určení minimální kostry byl použit Borůvkův/Kruskalův algoritmus. Probíhá na principu, kdy se prochází jednotlivé hrany pomocí prioritní fronty a poté se hrany testuje se, zda hranu přidáme do stromu. Při testování hrany mohou nastat čtyři situace:

- Žádný uzel h neleží v jiném podstromu. Hrana vytvoří nový podstrom.
- Jeden uzel h je součástí některého z podstromů. Hranu připojíme k podstromu.
- Oba uzly h jsou v různých podstromech. Oba podstrome spojíme do jednoho.
- Oba uzly h jsou v jednom podstromu. Hranu zahodíme.

### 3.5 Využití heuristiky Weighted Union [2]

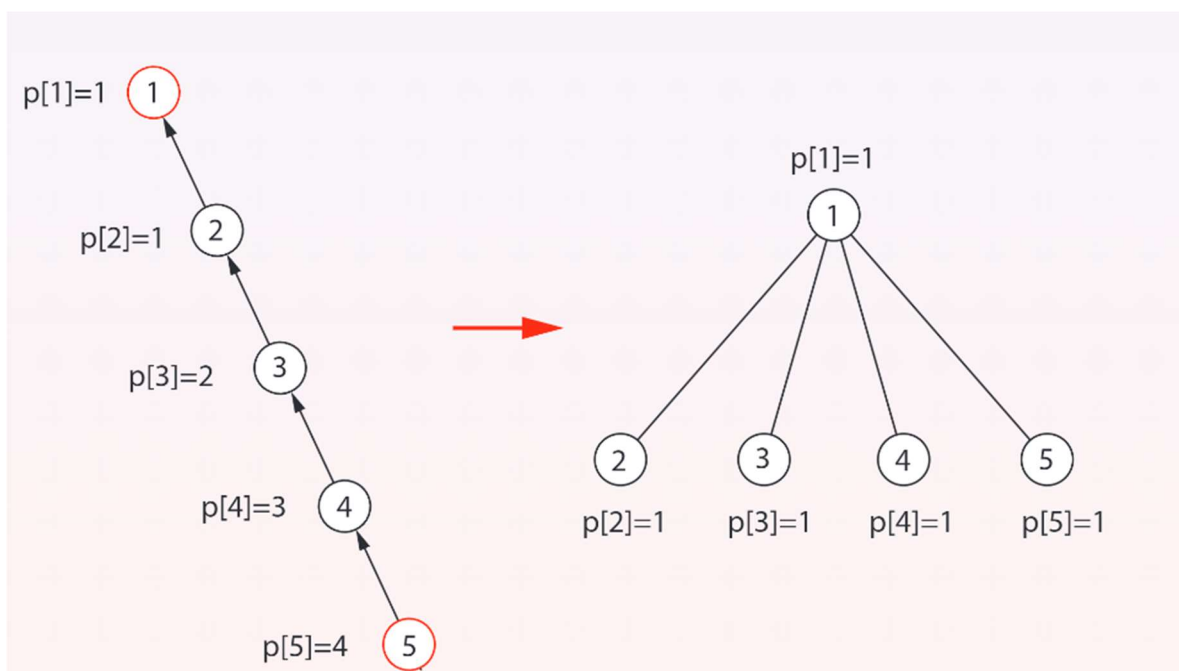
Tato funkce je využívána k efektivnímu spojování stromů, kdy je vždy lepší připojit kratší strom k delšímu. Princip je takový, že nalezneme kořen obou stromů, určíme, který strom je delší a ke kořenu delšího stromu připojíme kořen kratšího stromu.



### 3.6 Využití heuristiky Path Compression [2]

Tato funkce je využívána ke zkrácení stromu, ab operace s ním probíhaly rychleji. Strom lze zkrátit dvěma metodami. Jedním průchodem lze strom zkrátit o půlku. Dvouprůchodovým postupem lze celý strom připojit na jeden kořen. V této úloze byl použit dvouprůchodový systém.

Princip funkce probíhá v nalezení kořene a následném připojení všech uzlů na tento kořen.

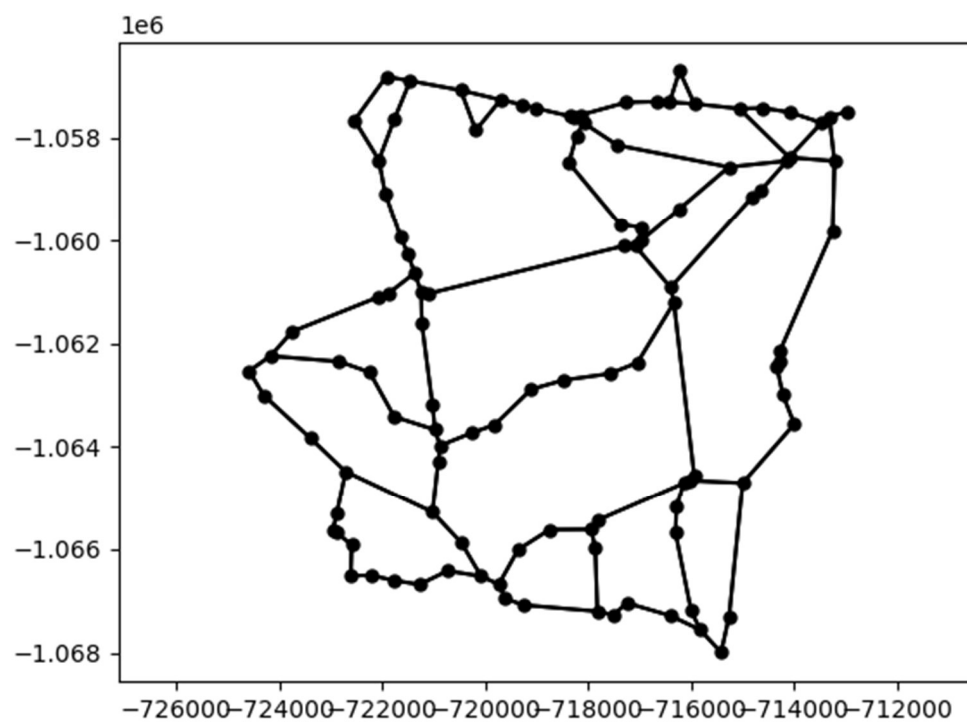


## 4 Výsledky

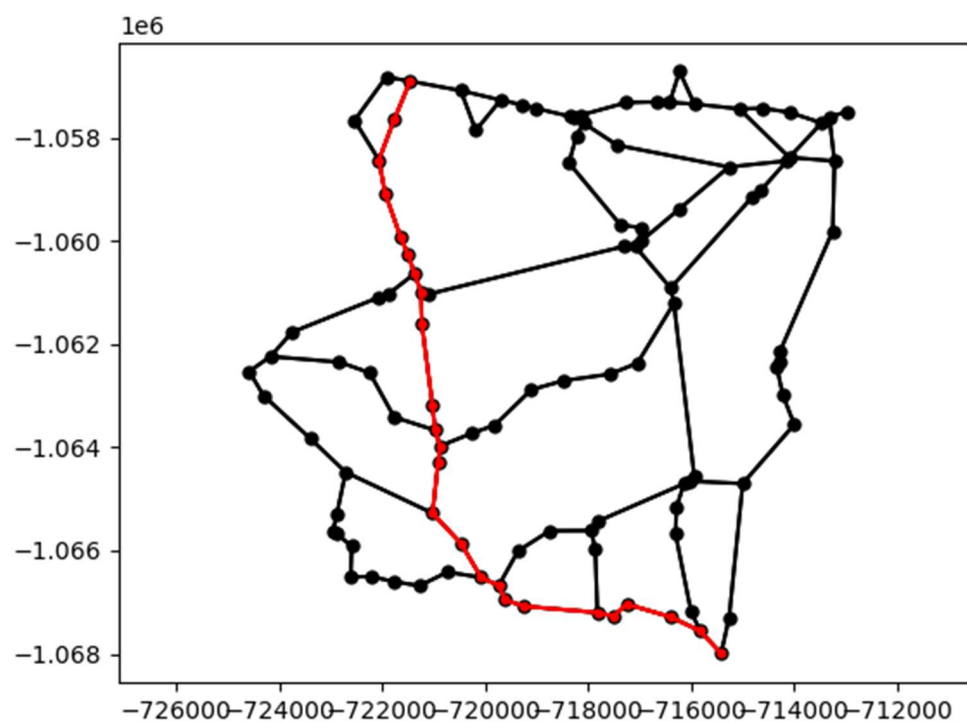
Prvním výstupem je grafová reprezentace silniční sítě. Ta se nachází v programu, který je přiložen v přílohách. Proměnná Graf3 obsahuje ohodnocení pomocí Euklidovské vzdálenosti a proměnná Graf2 obsahuje ohodnocení pomocí transportního času.

Následně byla určována cesta mezi dvěma uzly. Tyto uzly se nachází v obci Stříbrná Skalice a v obci Mukařov. Pro nejkratší Euklidovskou vzdálenost vychází ohodnocení na 16,46 km a pro nejkratší transportní čas vychází 13,66 minut.

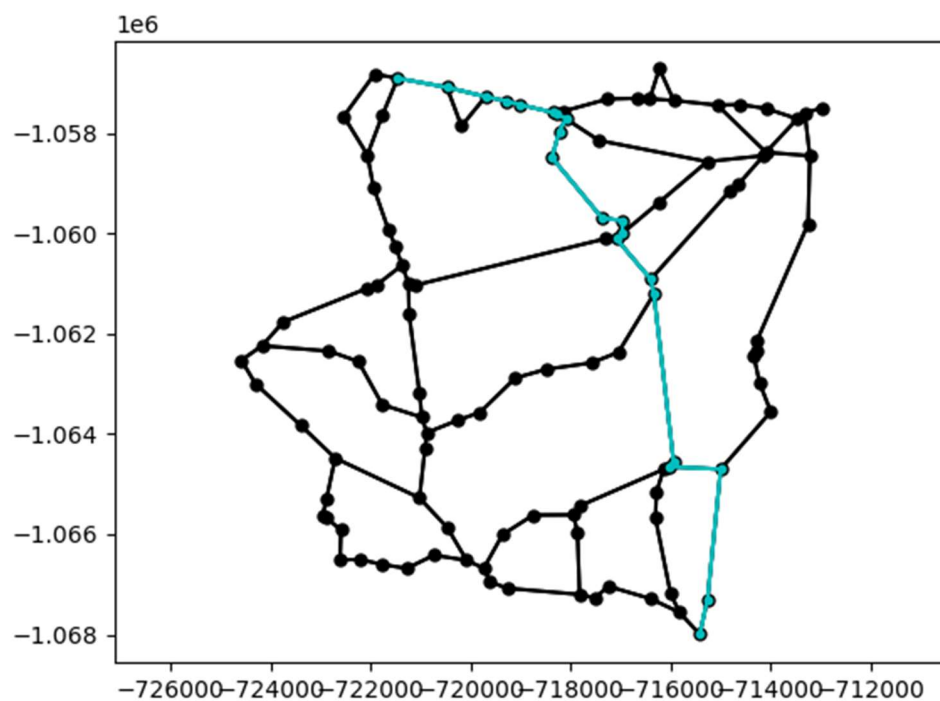
Obrázek 1: Vizualizace grafu.



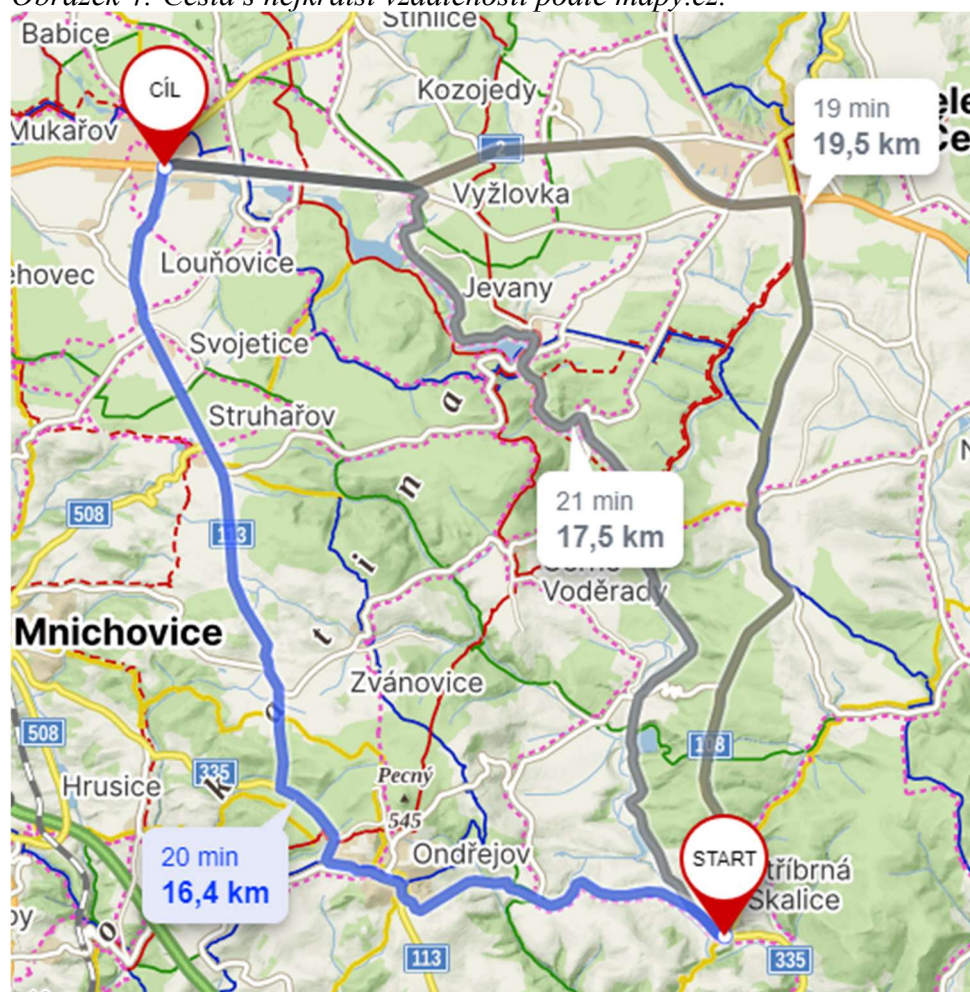
Obrázek 2: Cesta s nejkratší Euklidovskou vzdáleností.



Obrázek 3: Cesta s nejmenším transportním časem.

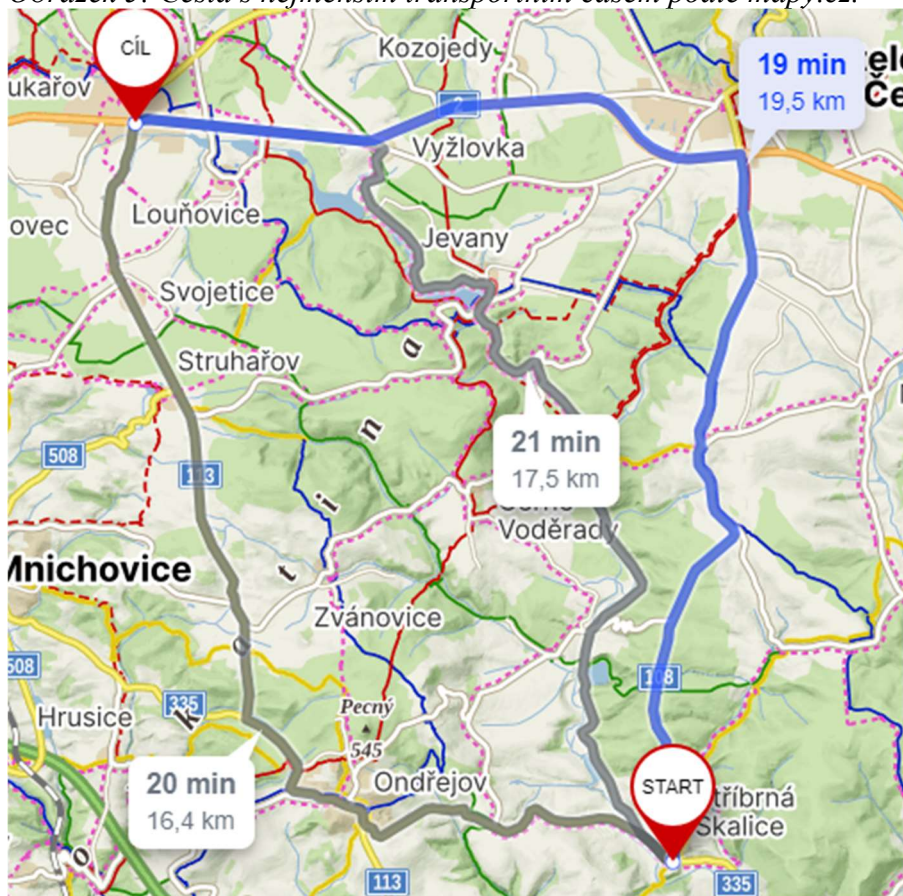


Obrázek 4: Cesta s nejkratší vzdáleností podle mapy.cz.



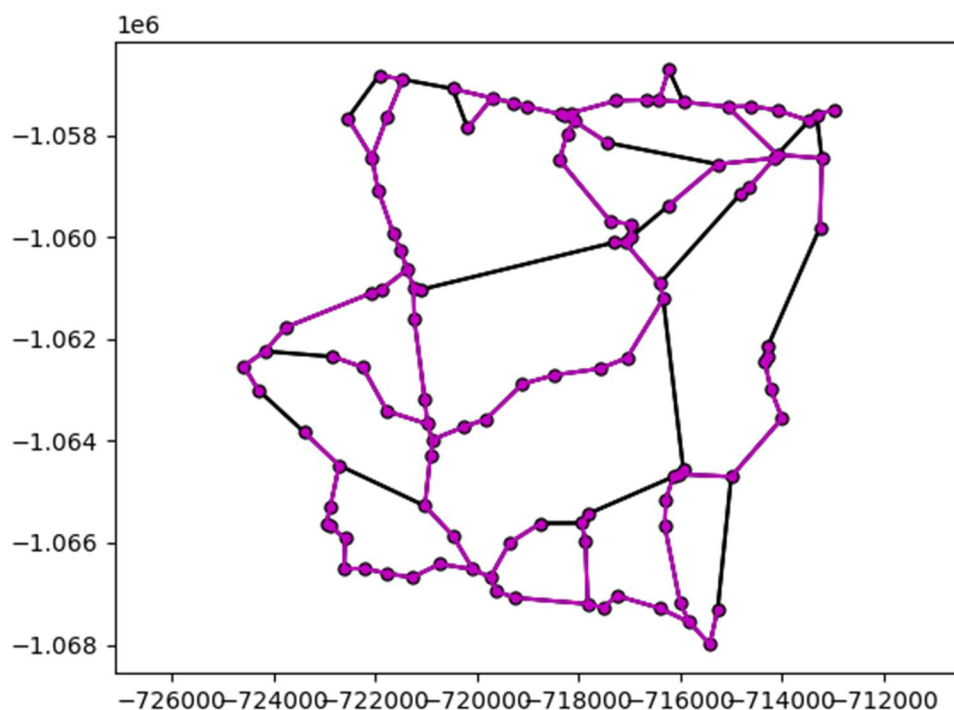


Obrázek 5: Cesta s nejmenším transportním časem podle mapy.cz.



Textový soubor s nejkratšími cestami mezi všemi dvojicemi uzlů se nachází v textovém souboru v přílohách. Nejkratší cesty byly vytvořeny na základě nejmenšího transportního času.

Obrázek 6: Minimální kostra grafu.





## 5 Závěr

Byl vytvořen program, který určuje nejkratší cestu grafem. První určení cesty proběhlo na grafu, kde bylo ohodnocení reprezentováno Euklidovskou vzdáleností. Výsledná cesta se shoduje se softwarem mapy.cz. Ohodnocení cesty bylo určeno na 16,46 a na mapy.cz na 16,4. Výstup je tedy přibližně shodný. Druhá cesta byla vytvořena z ohodnocení nejmenším transportním časem. Zde se cesta neshoduje s cestou v softwaru mapy.cz a navíc je čas výrazně kratší. Toto je způsobeno více faktory, ale všechny tyto faktory jsou nedokonalým ohodnocením a vytvořením grafu. Jedná se převážně o nedokonalé zpracování možností rychlostí. V úseku, kde se cesty rozdělí, se v cestě určené touto úlohou nachází serpentýny. V těchto místech je maximální povolená rychlost 90 km/h a díky koeficientu křivolakosti je tato rychlost výrazně zpomalena, ale stále je výrazně rychlejší, než se dá jet v realitě. To samé platí o úseku mezi vesnicemi Hradec, Jevany a Vyžlovka. Zde je navíc místy i nucené zpomalení, které nebylo do gradu započítáno. Další faktor, který zkracuje čas cesty jsou křižovatky, které v realitě nutí ke zpomalení, ale v grafu s ním nebylo počítáno. Časová reprezentace tedy neodpovídá skutečnosti.

Určení minimální kostry proběhlo na principu Borůvkova algoritmu. Jeho zobrazení lze vidět v obrázku 6 a lze pozorovat, že kostra skutečně obsahuje všechny uzly, ale ne všechny hrany.

## 6 Přílohy

- Skript pro spuštění funkce a s popisem funkce (*Koudelka\_Muller\_U4.py*)
- Textový soubor s nejkratšími cestami mezi všema uzly (*Nejkratsi\_cesta.txt*)

## 7 Zdroje

- [1] <https://github.com/k155cvut/ygei/blob/main/prednasky/geoinf8.pdf>
- [2] <https://github.com/k155cvut/ygei/blob/main/prednasky/geoinf9.pdf>