

## Úloha 3: Nejkratší cesta grafem

Předmět: Geoinformatika

Vypracoval: David Martínek

Datum: 4.1.2025

### 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 eukleidovskou vzdálenost
- nejmenší transportní čas

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

## 1) Teoretická část

a) **Dijkstra** v roce 1956 navrhl algoritmus pro nalezení nejkratší cesty v grafu s nezápornými vahami hran. Algoritmus využívá tzv. „greedy“ strategii, která postupně volí lokálně nejlepší možnosti s cílem dosáhnout optimálního celkového výsledku. V každé iteraci vybírá uzel s nejmenší známou vzdáleností, aktualizuje vzdálenosti jeho sousedů (tzv. relaxace hran) a pokračuje, dokud nejsou všechny dosažitelné uzly zpracovány. Pro efektivní vyhledávání nejkratších vzdáleností využívá prioritní frontu. Pro následnou rekonstrukci cesty je využíván seznam předchůdců (list of predecessors).

b) **Jarníkův** (Primův) algoritmus lze využít k nalezení minimální kostry ohodnoceného grafu. Poprvé jej formuloval Vojtěch Jarník na počátku 30. let 20. století, přičemž v 50. letech jej nezávisle znovuobjevili Robert C. Prim a Edsger W. Dijkstra. Algoritmus buduje minimální kostru postupně jako rostoucí strom – v každém kroku přidává nový vrchol propojený s již vybranými uzly hranou s nejnižším ohodnocením (greedy strategie). Využívá prioritní frontu k efektivnímu vyhledávání nejkratších hran vedoucích k dosud nezařazeným vrcholům a pokračuje, dokud nejsou všechny uzly součástí kostry. Na rozdíl od Borůvkova a Kruskalova algoritmu je efektivnější pro hustší grafy.

## 2) Použitá metrika

Zatímco v prvním případě hledání cesty grafem byla váha hran přímo rovna jejich eukleidovským vzdálenostem, v případě druhém bylo potřebné přijít s matematickým přepisem pro spravedlivé zohlednění reálných parametrů a jejich dopadu na hledání skutečné časově nejkratší vzdálenosti. Prvním takto zvoleným parametrem byla **maximální povolená rychlost**, kterou bylo možné uzel překonat. S předpokladem, že se vozidla v rámci testování Dijkstrova algoritmu pohybují právě rychlostí odpovídající jejich třídě, byl čas potřebný k překonání hrany stanovený dle následujícího vzorce:

$$travel\ time\ [s] = \frac{edge\ length\ [m]}{max\ speed\ [\frac{m}{s}]}$$

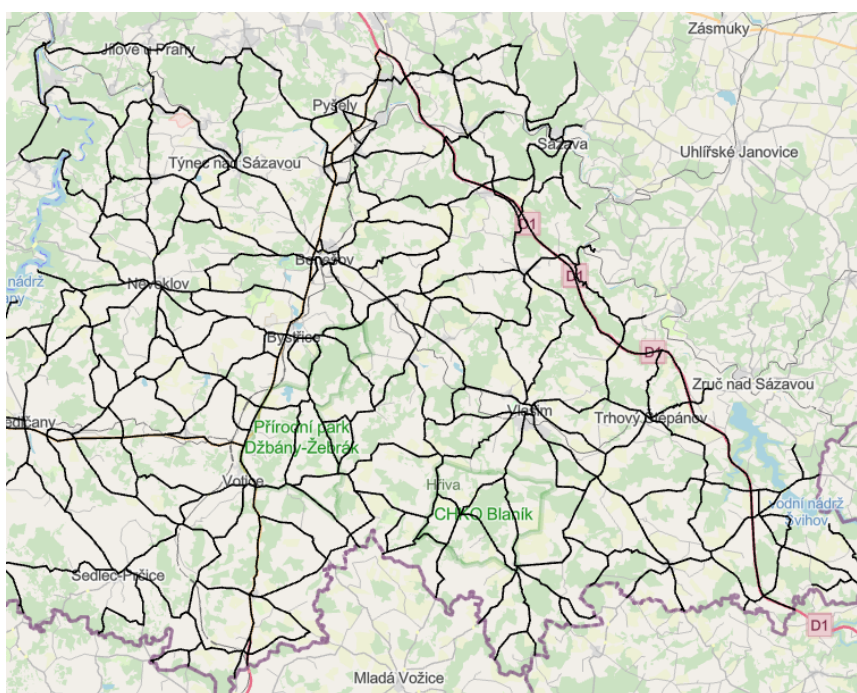
Druhým parametrem byla zvolena křivost (curvature), který zohledňoval časovou náročnost spojenou s objížděním překážek, převýšením atp. Na úseku mezi dvěma spojenými uzly jej šlo definovat takto:

$$curvature = \frac{euclidean\ distance}{total\ distance\ by\ road}$$

Tímto parametrem byl předchozí *travel time* vynásoben. V případě přímé cesty se cestovní čas nezměnil, zatímco v případě křivosti silnice byl cestovní čas spojený s překonáním hrany navýšen.

### 3) Vypracování

Úloha byla zpracována za využití softwaru ArcGIS Pro (3.2.0) a programovacího jazyka Python. Za testovací množinu silnic byla zvolena oříznutá data z jihozápadní části střečeského kraje z databáze ArcČR 500 (3.3). Pro potřeby algoritmu byl vygenerován textový soubor se čtyřmi atributy souřadnic bodů (počátku a konce silnice), třídou silnice a její délkou. Na základě silničních dat byly generovány v místech intersekcí uzly, za něž byly do textového souboru předány hodnoty souřadnic (obr. 1)

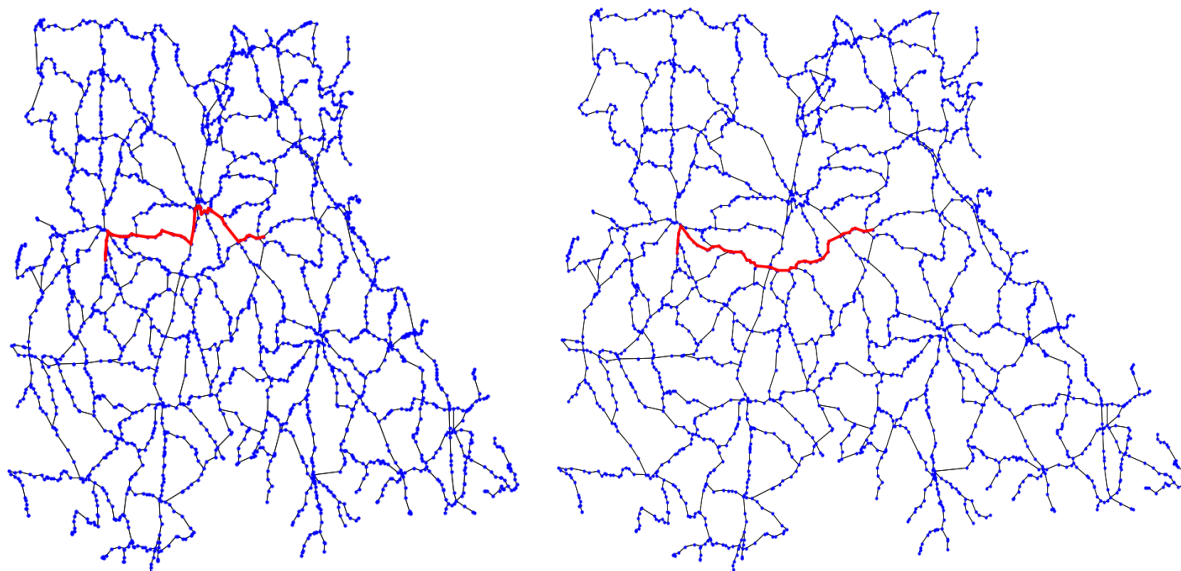


Obrázek 1 - Silniční síť

Připravená data byla konvertována pro potřeby práce s algoritmem do podoby seznamu (souřadnice uzlů) a slovníků hran. První slovník uzlům přiřazoval počáteční a koncové souřadnice hran a jejich délku. Druhý slovník namísto délky obsahoval cestovní čas, který byl spočten na základě délky silnice a maximální povolené rychlosti na dané třídě komunikace (která byla považována za cestovní rychlost). Tento čas byl opraven vynásobením koeficientem křivosti (*curvature*), který představuje podíl mezi délkou silnice a eukleidovskou vzdáleností mezi jejími koncovými body. Výsledek byl zbaven prázdných množin způsobených nekorektními vstupy pomocí nadefinované funkce *cleanGraph*. Pro případ záporně ohodnocených hran skript obsahuje funkci *fixNegativeValues*, která hodnotám všech hran přičte absolutní hodnotu minimální hodnoty + 1. Toto ošetření je následně negováno funkcí *cancelOffset* po proběhnutí algoritmů. Jakožto algoritmus pro generování minimální kostry byl zvolen Jarníkův algoritmus (*Prim*).

#### 4) Výsledek

Dijkstra algoritmus byl testován pro čtyři různé vstupy, přičemž výsledky byly porovnány s navigačním softwarem Mapy.cz. Prvně důkaz, že algoritmus při hledání nejkratší vzdálenosti a nejkratšího transportního času skutečně došel k jiným výsledkům (obrázek 2).



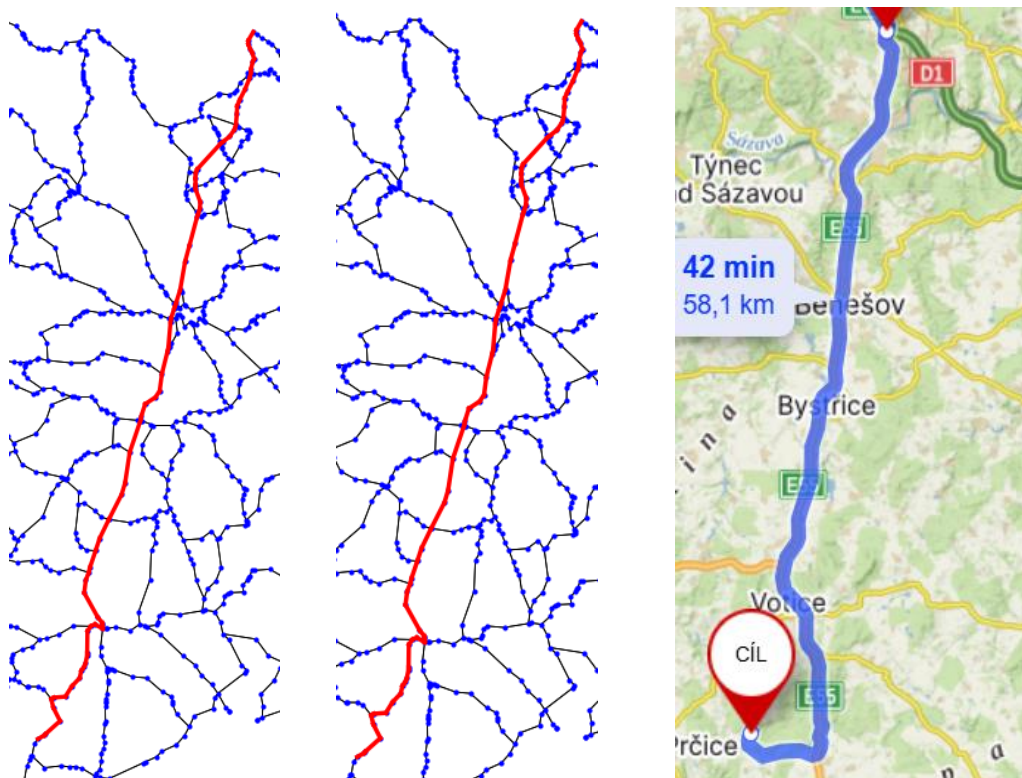
Obrázek 2 - Srovnání rozdílné volby trasy (vlevo: čas, vpravo: nejkratší vzdálenost)

Při porovnání byly dvojice bodů voleny náhodně. Výsledky poukazují na drobné nepřesnosti, kterou mohou mít na svědomí rozdíly v použitém datasetu, nepřesně nastavené váhy, chyby a generalizační procesy v algoritmu nebo pokročilost současných navigačních systémů a aktuální dopravní situace. Rozdíly mezi oběma systémy shrnuje tabulka 1.

Uzel		Dijkstra		Mapy.cz		Rozdíl	
Start	Cíl	délka [km]	čas [min]	délka [km]	čas [min]	délka [km]	čas [min]
Mirošovice	Ješetice	60	59.9	58.2	42	1.8	17.9
Jenkov	Chocerady	46.2	49.1	34.6	41	11.6	8.1
Rabyně	Čestín	50.2	47.9	37.1	44	13.1	3.9
K. Střimelnice	Kosova Hora	65	56.5	50.6	47	14.4	9.5

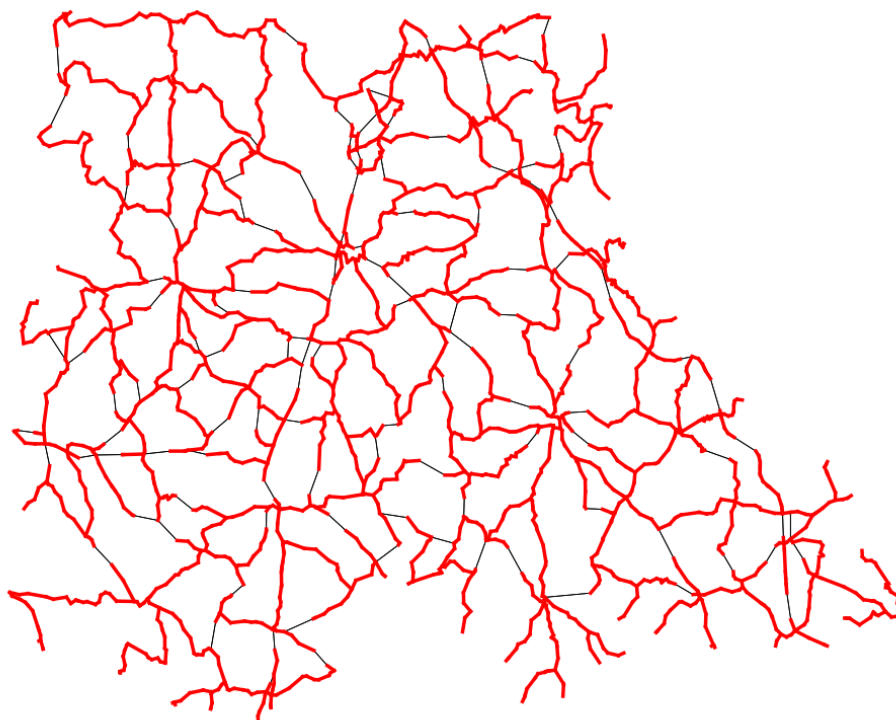
Tabulka 1 - Srovnání Dijkstra a Mapy.cz

Názornost srovnání vizualizuje obrázek 3.



Obrázek 3 - Ješetice - Mirošovice (vlevo: nejkratší, střed: nejrychlejší, vpravo: Mapy.cz

V rámci úlohy byla také vypracována funkce pro nalezení nejmenší kostry Jarníkovou metodou (obrázek 4). Délka kostry odpovídá přibližně 950 km.



Obrázek 4 - Minimální kostra

## 5) Závěr

Byl implementován Dijkstra algoritmus pro nalezení nejkratší cesty grafem a Jarníkův algoritmus pro nalezení minimální kostry. Zároveň byly testovány dvě varianty metrik – nejkratší eukleidovská vzdálenost a nejmenší transportní/cestovní čas, přičemž výsledky byly porovnány s navigačním softwarem Mapy.cz. Analýza ukázala, že Mapy.cz prezentují kratší trasy s menším transportním časem. Odchyly oproti navigačnímu softwaru lze přičíst odlišným datasetům, nastavení vah a aktuální dopravní situaci. Implementací Jarníkova algoritmu byla nalezena minimální kostra grafu, jež odpovídá přibližně 950 km.