

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.

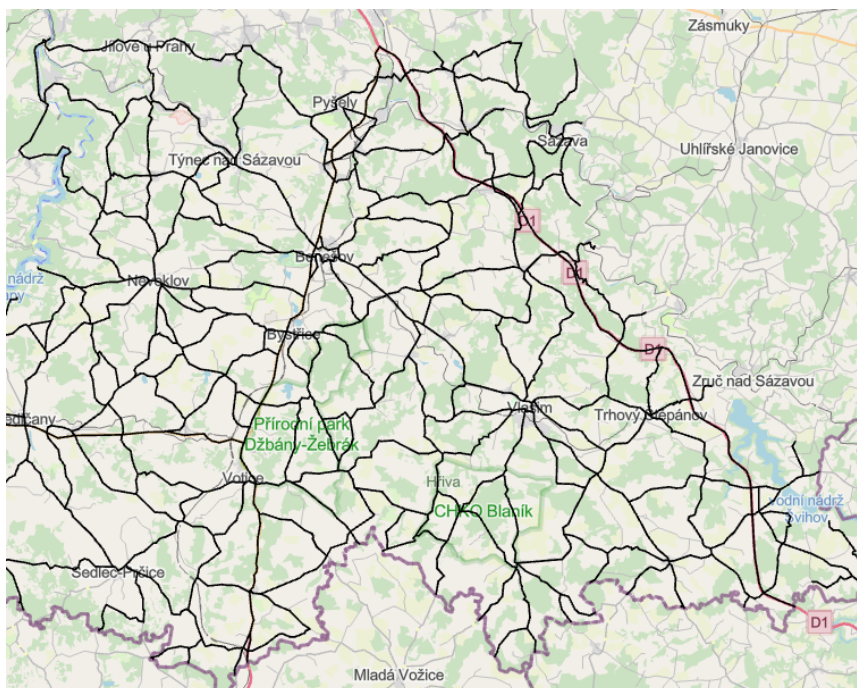
Otestujte různé varianty volby ohodnocení hran grafu tak, aby nalezená cesta měla:

- 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.

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 středočeského 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)

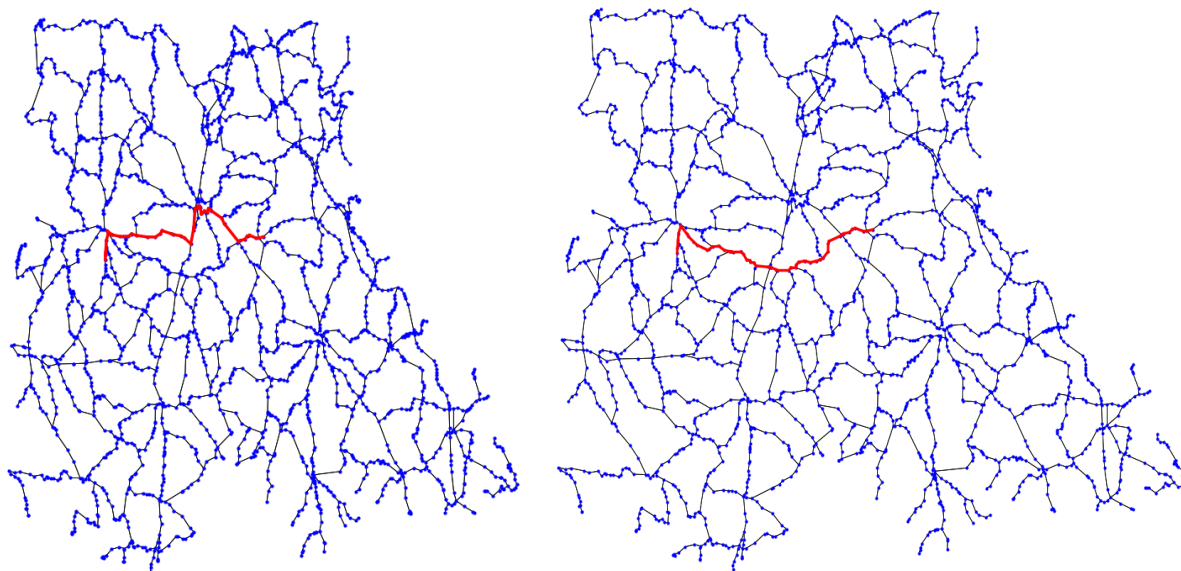


Obrázek 1 - Silniční síť

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*)

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).



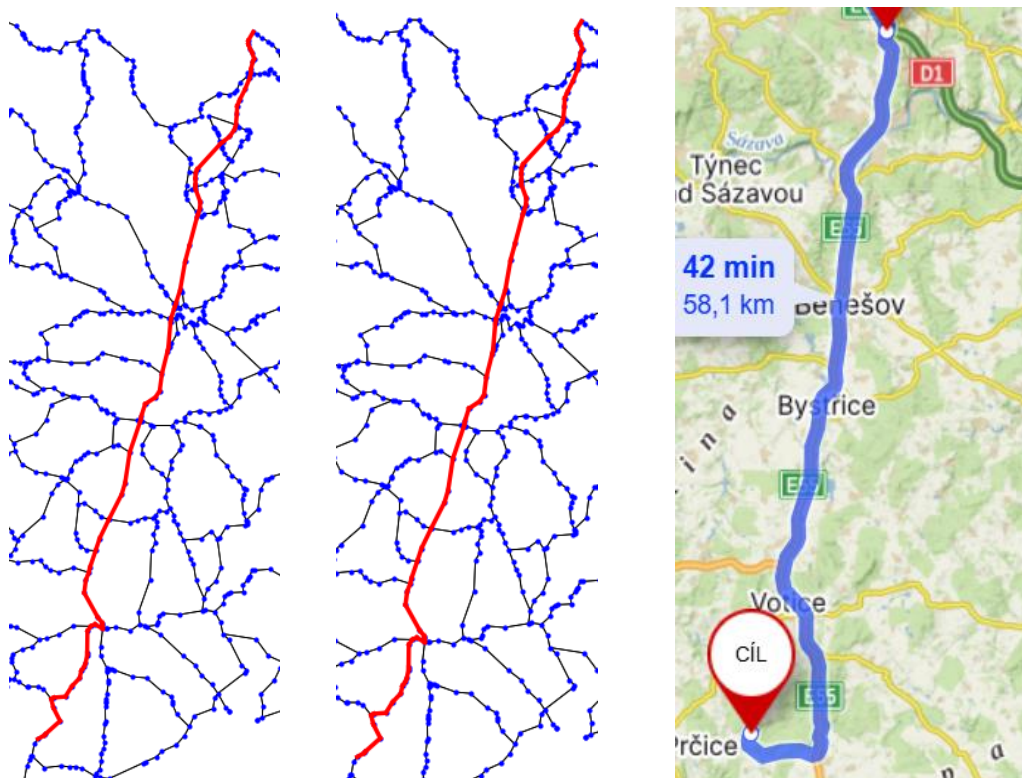
Obrázek 2 - Srovnání rozdílné volby trasy (vlevo: čas, vpravo: nejkratší vzdálenost)

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

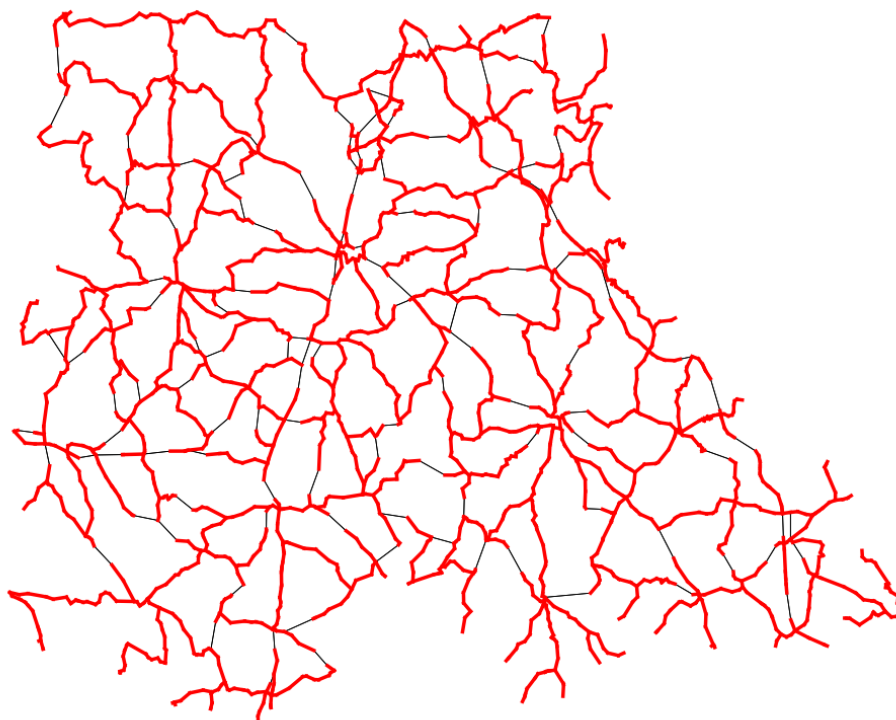
Tabulka 1 - Srovnání Dijkstra a Mapy.cz

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