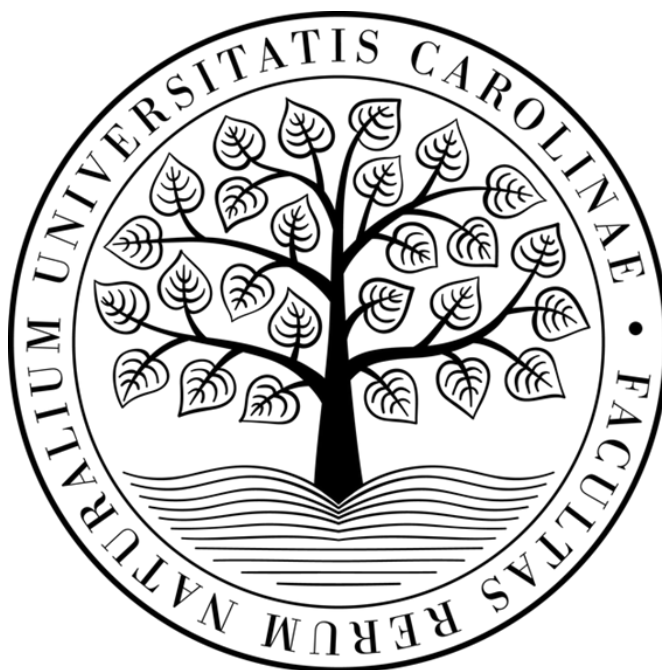


Univerzita Karlova

Přírodovědecká fakulta



GEOINFORMATIKA

Nejkratší cesta grafem

Martina Pavlová, Ludmila Vítková

1 N-GKDPZ

Praha, 2024

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 navrhnete 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.

Krok	Hodnocení
Dijkstra algoritmus	20b
<i>Řešení úlohy pro grafy se záporným ohodnocením</i>	+10b
<i>Nalezení nejkratších cest mezi všemi dvojicemi uzlů</i>	+10b
<i>Nalezení minimální kostry některou z metod</i>	+15b
<i>Využití heuristiky Weighted Union</i>	+5b
<i>Využití heuristiky Path Compression</i>	+5b
Max celkem	65b

Čas zpracování: do konce zápočtového týdne

Zpracování

Popis kódu:

Vstupní data:

Jako vstupní data byla vybrána data z OpenStreetMap (OSM), která obsahují informace například o ulicích, budovách a využití půdy. Tato data byla načtena pomocí modulu OSMnx, který může sloužit nejen k načítání, ale i analýze a vizualizaci silničních sítí. Stahování a načítání dat probíhá pomocí zadání adresy v jejímž okolí chceme data analyzovat. Lze vyhledávat celé státy, město anebo i jen městské části. V tomto případě došlo k načtení dat z městské čtvrti Kobylisy. Ihned po spuštění programu se objeví graf vybrané oblasti, ve kterém se nachází pouze silniční síť a uzly na ní.

Informace pro výpočet nejrychlejší trasy:

Pro určení nejrychlejší trasy bylo nutné nejdříve definovat maximální povolenou rychlost v úseku, podle které se následně může vypočítat čas, za který lze daný úsek projet. Maximální povolená rychlost byla nastavena na 130 km/h na dálnicích, 110 km/h na rychlostních silnicích, 90 km/h na ostatních silnicích mimo obce, 50 km/h v obcích a 30 km/h v obytných zónách a na neklasifikovaných komunikacích.

Dále byla definována funkce pro výpočet času průchodu mezi uzly. Tato funkce nejdříve vyhledá maximální povolenou rychlost na daném úseku. Pokud se daný typ komunikace nenachází v naší definovaném slovníku maximálních povolených rychlostí, přiřadí se k němu limit 30 km/h. V tomto případě nebyly vypisovány rychlosti například pro různé polní cesty nebo cesty v průmyslových areálech. Čas průchodu mezi uzly je vypočítán jako vzdálenost v kilometrech dělená maximální povolenou rychlostí a následně vynásobena 3600, aby došlo k převodu z hodin na vteřiny.

Dijkstra algoritmus:

Nalezení nejkratší a nejrychlejší trasy v grafu probíhalo pomocí implementace Dijkstra algoritmu, který inicializuje seznamy pro uchování předchůdců (P) a vzdáleností (D) pro každý uzel. S využitím prioritní fronty iterativně prochází uzly a aktualizuje vzdálenosti a předchůdce na základě aktuálně nejkratších cest. Tato funkce na konci vrací dva seznamy obsahující předchůdce a vzdálenosti a poskytující tak informace o nejkratších cestách ze zadaného počátečního uzlu ke všem ostatním uzlům v grafu.

Dalším krokem byla rekonstrukce nejkratší/nejrychlejší trasy v grafu. Ke svému fungování využívá informace ze seznamu předchůdců a z grafu. Cesta je konstruována iterativním sledováním předchůdců, které probíhá až do té doby, než je dosaženo počátečního bodu. Tato funkce vrací rekonstruovanou cestu, celkovou vzdálenost mezi dvěma uzly a také celkový čas strávený cestou mezi těmito dvěma vybranými uzly.

Poslední krokem v této části byla vizualizace nejkratší/nejrychlejší cesty, která proběhla pomocí knihovny NetworkX. Ke svému fungování využívá graf (G), souřadnice pro pozice uzlů (C) a nalezenou nejkratší/nejrychlejší trasu. Výsledný graf se vizualizuje pomocí knihovny matplotlib. Nejkratší/nejrychlejší trasa se od ostatních hran liší svým zeleným zvýrazněním.

Minimální kostra grafu:

Minimální kostra grafu byla vytvořena pomocí Borůvkova algoritmu s využitím heuristiky Weighted Union a Path Compression. Průběh začíná inicializací množin, kořenů a řádů pro každý vrchol. Funkce *Find* nachází pomocí iterativního hledání kořen množiny obsahující vrchol *u* a zároveň pro zrychlení následujících operací komprimuje cestu. Funkce *Weighted_union* nejprve najde kořeny množin obsahující vrcholy *u* a *v* pomocí funkce *Find*. Pokud jsou kořeny různé, provede sjednocení spojením kořenů. Cílem Borůvkova algoritmu je vytvoření stromu s minimální váhou tak, aby pokrýval všechny vrcholy grafu. Hlavní funkce *Boruvka* zpracovává seřazené hrany a postupně je přidává do minimální kostry grafu. Hlavním výstupem je pak seznam hran tvořící minimální kostru grafu.

Pro vizualizaci minimální kostry grafu byla použita knihovna NetworkX. Funkce ke svému průběhu používá vstupní graf (*G*) a pozice uzlů (*C*). Všechny hrany grafu jsou vykresleny světle šedou barvou a minimální kostra grafu je pak zvýrazněna červenou barvou.

Nalezení nejkratších cest mezi všemi dvojicemi uzlů:

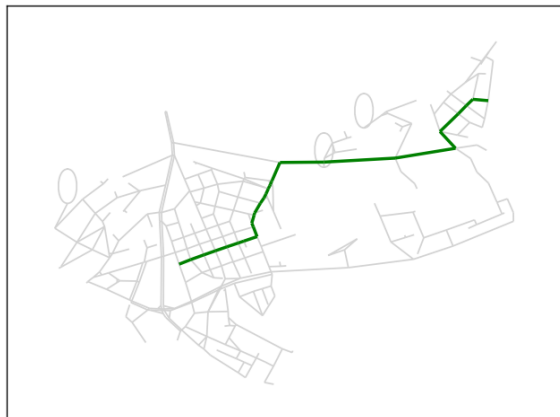
Nalezení nejkratších cest mezi všemi dvojicemi uzlů proběhlo za pomoci Floyd-Washallova algoritmu. Tato funkce inicializuje matici vzdáleností s počátečními hodnotami nastavenými na nekonečno. Matice je poté aktualizována zvažováním přímých vzdáleností hran a dynamickou úpravou cest tak, aby byly nalezeny nejkratší vzdálenosti mezi všemi dvojicemi uzlů. Výsledkem je matice vzdáleností obsahující nejkratší cesty mezi každou dvojicí uzlů v grafu.

Řešení úlohy pro grafy se záporným ohodnocením:

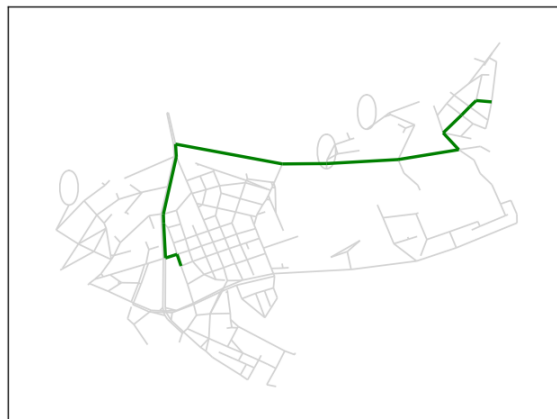
Pro řešení úlohy pro grafy se záporným ohodnocením byl použit Bellman-Fordův algoritmus. Algoritmus používá seznamy pro vzdálenost (*D*) a předchůdce (*P*). Základní operací algoritmu je opakovaná relaxace hran. Algoritmus v rámci relaxace iteruje každý uzel grafu. Seznamy vzdáleností a předchůdců jsou aktualizovány, pokud je nalezena kratší cesta. Následně kontroluje přítomnost negativního cyklu. Pokud je po dokončení relaxace nalezena kratší cesta, znamená to, že v grafu existuje záporné ohodnocení. Program na tuto skutečnost upozorní a dále již nejkratší cestu nepočítá. Pokud neexistuje negativní cyklus, funkce vrátí seznam předchůdců (*P*) a vzdáleností (*D*) reprezentující nejkratší cesty. Dále už pokračuje stejně jako v případě Dijkstra algoritmu při rekonstrukci a vizualizaci grafu nejkratší cesty.

Výsledky:

Výsledné grafy lze vidět na obrázcích 1 a 2, přičemž nejkratší nalezená trasa se nachází na obrázku 1 a nejrychlejší nalezená trasa na obrázku 2. Vzdálenost a čas těchto dvou tras lze vidět v tabulce 1.



Obrázek 1: Nejkratší nalezená trasa



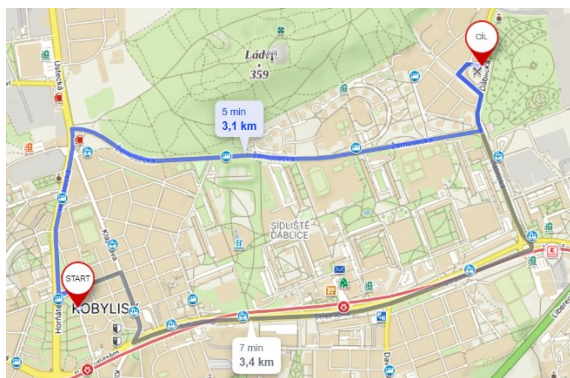
Obrázek 2: Nejrychlejší nalezená trasa

Trasa	Vzdálenost	Čas
Nejkratší	2,46 km	2,41 minut
Nejrychlejší	2,99 km	2,31 minut

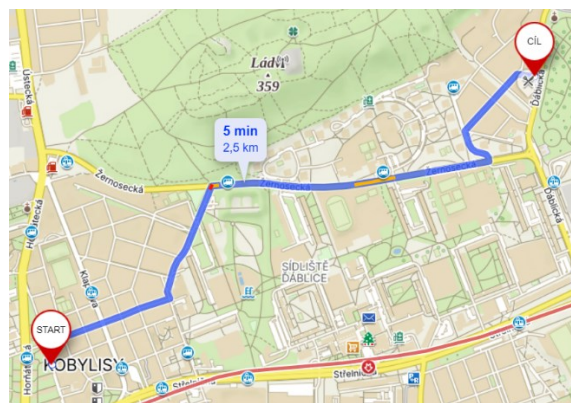
Tabulka 1: Charakteristiky výsledných cest

Porovnání s Mapy.cz:

Po zadání počátečního a koncového bodu byla nejprve zvolena možnost nejkratší trasy. Výsledek lze vidět na obrázku 3. Již na první pohled je ale vidět, že se nejedná o nejkratší možnou trasu. Na zkoušku byla zvolena možnost nejrychlejší trasy s aktuální dopravou, jejíž výsledek lze vidět na obrázku 4. Tato trasa je 2,5 km dlouhá, měla by trvat přibližně 5 minut a shoduje s námi nalezenou nejkratší trasou.

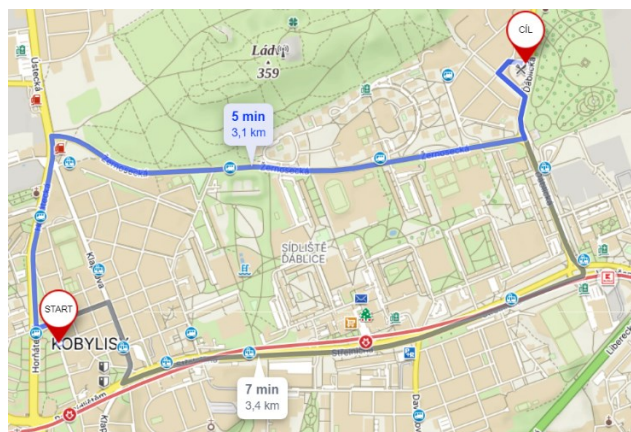


Obrázek 3: Nejkratší trasa podle Mapy.cz



Obrázek 4: Nejrychlejší trasa s provozem podle Mapy.cz

Nejrychlejší trasa podle Mapy.cz je 3,1 km dlouhá, měla by trvat také 5 minut a lze ji vidět na obrázku 5. Tato trasa se nejdříve napojuje na hlavní silnici a ve většině se shoduje s námi nalezenou nejrychlejší trasou. Jediný rozdíl je u jejího konce, kdy trasa podle Mapy.cz vede východnějším ulicí než ta námi nalezená. Důvodem je to, že v námi použitých datech tato východnější silnice již není.



Obrázek 5: Nejrychlejší trasa podle Mapy.cz

Minimální kostra grafu:

Na obrázku 6 lze vidět minimální kostru grafu vytvořenou s využitím nejkratší trasy. Hned na první pohled je vidět, že spojuje všechny uzly grafu. Hrany v této minimální kostře by měly mít minimální možnou celkovou váhu



Obrázek 6: Minimální kostra grafu

Nalezení nejkratších cest mezi všemi dvojicemi uzlů:

Program je schopný najít nejkratší cesty mezi všemi dvojicemi uzlů. Jelikož je ale ve vstupních datech přes 300 uzlů, vypisuje program pouze vzdálenost mezi dvěma určenými uzly. Jako tyto dva uzly byl zvolen počáteční a koncový bod, u kterých byla hledána nejkratší/nejrychlejší trasa. Nejkratší nalezená trasa má přibližně 2,5 km.

Zdroje:

OpenStreetMap Wiki (2023): Key:highway [online]. [cit. 20. 12. 2023]. Dostupné z: <https://wiki.openstreetmap.org/wiki/Key:highway>

TENKANEN, H., HEIKINHEIMO, V. (2020): Retrieving OpenStreetMap data [online]. [cit. 20. 12. 2023]. Dostupné z: https://autogis-site.readthedocs.io/en/2019/notebooks/L6/retrieve_osm_data.html?fbclid=IwAR3mYaA6V4Y23-samZ9tXyALsRXx3BkAk1lI9UILQu9NeknlzdLFMvsJZacv