Semestrální projekt

155UZPD - Úvod do zpracování prostorových dat

David Mráz, Pavel Kulmon

2015/2016

Obsah

- Zadání
- Data
- Úpravy
- Dotazy
- Závěr

Zadání

- Navrhněte a vytvořte tématické vrstvy (např. vodní toky, vodní plochy, lesy, silnice, železnice a pod.) na základě dat OpenStreetMap (viz schéma osm) a dalších otevřených zdrojů
- Aplikujte testy datové integrity a odstraňte případné nekonzistence v datech
- Vytvořte tutoriál - tj. sadu atributových a prostorových dotazů nad databází pgis_uzpd

- Zvolené tématické vrstvy
  - Cyklotrasy
  - Pěší trasy
  - Veřejná WC
  - Veřejné budovy
  - Městské části
  - Objekty MPP

Cíle

- Importovat vrstvy IPR do PostGIS databáze
- Prozkoumat některé možnosti PGRoutingu
- Vytvořit sadu (užitečných) dotazů cílených na využití PGRoutingu

Data

- Převzata z IPR
- Formát Esri Shapesile
- Převod do pgsql
- U nebodových vrstev vytvořeny Multigeometrie
- Převod Multigeometrií na Singlegeometrie
- IS_Valid, IS_Empty,

- Cyklotrasy – ST_LineString
- Pěší trasy – ST_LineString
- Městské části – ST_Polygon
- Objekty MPP – ST_Point
- Veřejná WC – ST_Point
- Veřejné budovy - ST_Point

Úpravy

- Funkce pgroutingu pgr_dijkstra – sloupce source integer, target integer
- Zaplnění sloupců pomocí funkce pgr_createTopology
- Sloupec nákladů – cost (hodnota odvozena z délky úseku)
- Funkce pgroutingu pgr_astar – sloupce x1, y1, x2, y2
- Zaplnění sloupců pomocí souřadnic počátečních a koncových bodů daného LineStringu
- Explicitní cast hodnot počátečního a koncového vrcholu na integer (při volání funkce)

Úpravy

- Pro funkci pgroutingu pgr_kDijkstraCost (Path) vybereme nejprve čísla vertexů zájmových bodů
- Vstupem funkce je ovšem integer[], musíme tedy převést výsledek dotazu do ARRAY

Dotazy 1

- /* 1.a Nalezněte nejkratší cestu od záchodku v Praze Petrovicích k záchodku na Praze 17 pomocí Dikstrova algoritmu */
- ```
  CREATE TABLE pesi_dijkstra AS
  SELECT seq, id1 AS node, id2 AS edge, cost
  FROM pgr_dijkstra('SELECT gid AS id, source, target, cost FROM pesi_single',(
  SELECT pesi_start_vert.id FROM pesi_start_vert),(
  SELECT pesi_end_vert.id FROM pesi_end_vert),false,false);
  ```
- Možnost vložit číslo cílového vertexu pomocí poddotazu přímo ve volání funkce
- Nalezne cestu mezi source a target pouze tehdy, pokud taková cesta existuje (tj. v grafu sítě nejsou oddělené podgrafy atd.)
- /* 1.b Jaká je délka této nejkratší cesty? */
- ```
  SELECT sum(cost) FROM pesi_dijkstra;
  ```

Dotazy 2

- /* 1.c Sestavte geometrii této nejkratší cesty (LineString vrstvu obsahující pouze nalezenou cestu) */
- ```
  CREATE TABLE pesi_dijkstra_geom AS
  SELECT A.* FROM pesi_single AS A
  JOIN pesi_dijkstra AS B
  ON A.gid = B.edge;
  ```
- /* 1.d Přes které městské části vede nalezená nejkratší cesta? */
- ```
  CREATE TABLE pesi_dijkstra_casti AS
  SELECT DISTINCT(A.gid),A.nazev,A.geom
  FROM mestskecast AS A JOIN pesi_dijkstra_geom AS B
  ON ( ST_Intersects(A.geom,B.geom) OR ST_Within(B.geom,A.geom));
  ```

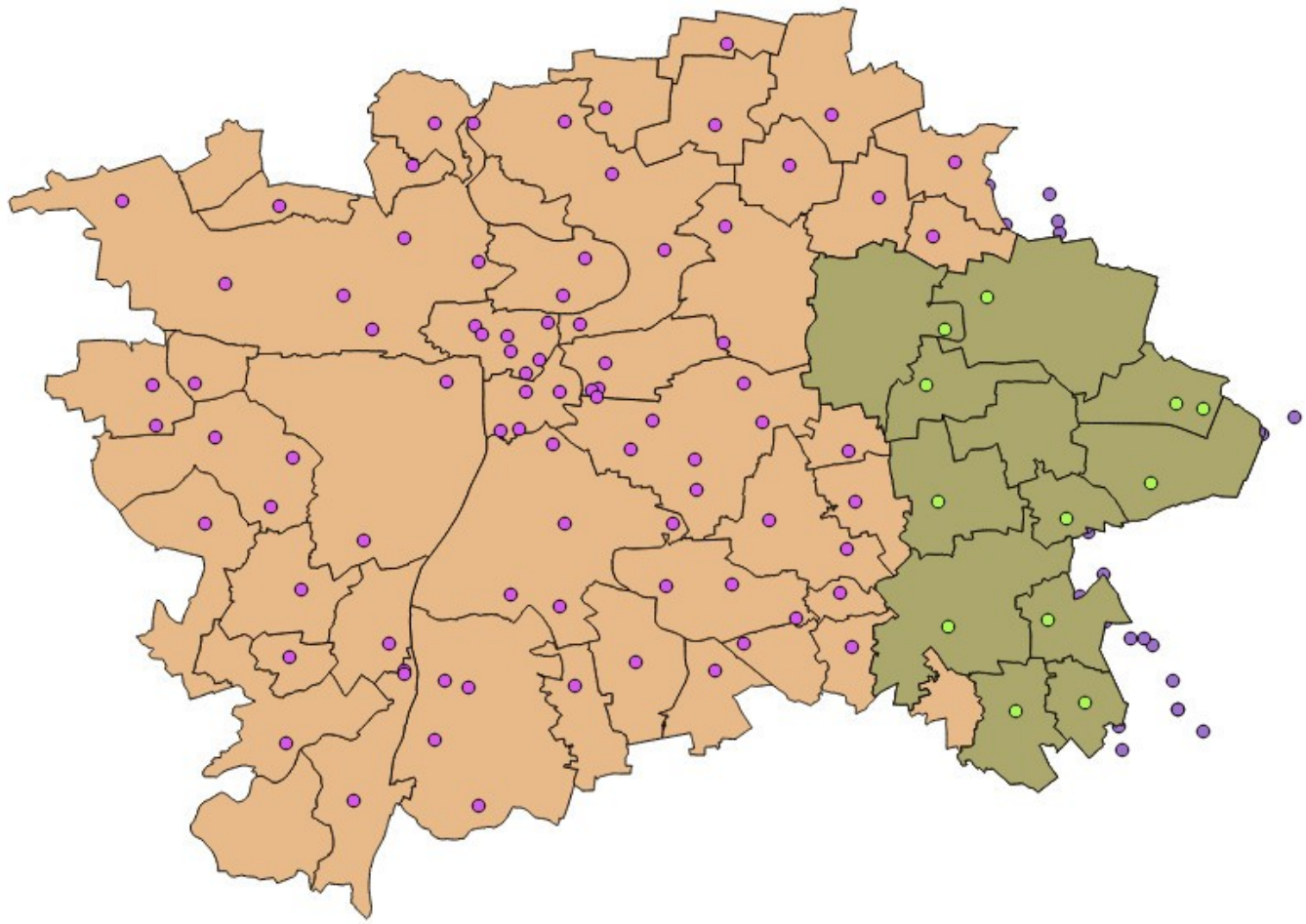
Doatzy 3

- /* 2.a Nalezněte tu samou nejkratší cestu s použitím algoritmu A* */
- ```
  CREATE TABLE pesi_astar AS
  SELECT seq, id1 AS node, id2 AS edge, cost
  FROM pgr_astar('SELECT gid AS id, source, target, cost, x1, y1, x2, y2 FROM
  pesi_single',(
  SELECT pesi_start_vert.id FROM pesi_start_vert)::integer,(
  SELECT pesi_end_vert.id FROM pesi_end_vert)::integer,false,false);
  ```
- Funkce voláme s posledními dvěma parametry false, protože máme neorientovaný graf s reversní cenou
- /* 2.b Jaká je délka takto nalezené cesty? */
- ```
  SELECT sum(cost) FROM pesi_astar;
  ```
- /* 2.c Je nalezená trasa pomocí algoritmu A* stejná jako trasa nalezená pomocí Dijkstrova algorimtu? */

Dotazy 4

- ```
  SELECT A.gid FROM pesi_dijkstra_geom AS A
  JOIN pesi_astar_geom AS B
  ON ST_Equals(A.geom,B.geom)
  WHERE ST_Equals(A.geom,B.geom) = false;
  ```
- Výsledek dotazu je prázdný, což zanamená že tam neexistují nestejné geometrie (trasy jsou tedy stejné)
- /* 3.a Studenti pořádají závody. Každý student si vybere jeden z pražských záchodků jako své startovní místo. Studentů je o jednoho méně než záchodků a tak ten který zůstane se stane cílovým místem. Který ze studentů má největší šanci na výhru v závodě, vzhledem ke vzdálenosti startovního a cílového místa, pakliže se studenti mají za úkol pohybovat pouze po pěších trasách? (určete pomocí Dijkstrova algoritmu ve verzi Many to One - pgr_kDijkstra) */
- Nejprve vybereme náhodně jeden záchodek z vrstvy verejnawc a přiřadíme mu číslo nejbližšího vertexu z topologie vrstvy pesi_single
  ```

Dotazy 5

- ```
  CREATE TABLE target_toal_vert AS
  SELECT DISTINCT ON(A.gid) A.gid AS wcgid, B.id AS vertgid
  FROM (
  SELECT * FROM verejnawc
  ORDER BY random() LIMIT 1 ) AS A,
  pesi_single_vertices_pgr AS B
  ORDER BY A.gid, ST_Distance(A.geom,B.the_geom);
  ```
- Ke všem záchodkům vrstvy verejnawc najdeme nejbližší vertex z topologie vrstvy pesi_single
- ```
  CREATE TABLE wc_nearest AS
  SELECT DISTINCT ON(A.gid) A.gid AS wcgid ,B.id AS vertgid
  FROM verejnawc AS A, pesi_single_vertices_pgr AS B, target_toal_vert AS C
  WHERE A.gid <> C.wcgid
  ORDER BY A.gid, ST_Distance(A.geom,B.the_geom);
  ```
- Vstupem do funkce pgr_kDijkstraCost je ovšem integer[], proto musíme vertexy převést do pole integerů

Dotazy 6

- CREATE TABLE array_wc_nearest AS
  SELECT ARRAY(
  SELECT wc_nearest.vertgid FROM wc_nearest) AS vertArray;
- A konečně vyhodnotíme nejkratší cesty
- CREATE TABLE pesi_dijkstra_many_cost AS
  SELECT *
  FROM pgr_kDijkstraCost('SELECT gid AS id, source, target, cost FROM pesi_single',(
  SELECT target_toal_vert.vertgid FROM target_toal_vert)::integer,(
  SELECT array_wc_nearest.vertArray FROM
  array_wc_nearest)::integer[],false,false);
- Dotaz na nejkratší z cest (bod od kterého je to nejblíže) by vrátil -1, případně 0
- V grafu můžou existovat buď body mezi kterými není cesta, nebo (závislé na naší úloze) mohou být body které mají nejbližší vertex shodný s cílovým, předpokládáme že tyto body nechceme, proto

Dotazy 7

- ```
  CREATE TABLE nearest_start AS
  SELECT * FROM pesi_dijkstra_many_cost
  WHERE cost <> -1 AND cost <> 0
  ORDER BY cost ASC LIMIT 1;
  ```

- Sestavení cesty kterou se z tohoto bodu má student vydat

- ```
  CREATE TABLE nearest_start_path AS
  SELECT id1 AS path, ST_AsText(ST_LineMerge(ST_Union(b.geom))) AS geom
  FROM pgr_kDIjkstraPath('SELECT gid AS id, source, target, cost FROM
  pesi_single',(
  SELECT target_toal_vert.vertgid FROM target_toal_vert)::integer,(
  SELECT array_wc_nearest.vertArray FROM
  array_wc_nearest)::integer[],false,false) AS a,
  pesi_single AS b, nearest_start AS c
  WHERE a.id3 = b.gid AND a.seq = c.seq
  GROUP BY id1
  ORDER BY id1;
  ```

Dotazy 8

- /* 4.a Na Praze 15 jsou právě dve policejní stanice. Policisté potřebují mezi stanicemi převážet důležité dokumenty a v zájmu zdraví policistů se pohybují po cyklostezkác */

- ```
  ALTER TABLE cyklo_single ADD COLUMN rychlost integer;
  UPDATE cyklo_single SET rychlost = 3 WHERE dopr_stav NOT IN
  (6,9,16,15,19,20);
  UPDATE cyklo_single SET rychlost = 2 WHERE dopr_stav IN (6,9,16);
  UPDATE cyklo_single SET rychlost = 1 WHERE dopr_stav IN (15,19,20);
  ALTER TABLE cyklo_single ADD COLUMN cost2 double precision;
  ```

- Vybereme policejní stanice na Praze 15

- ```
  CREATE TABLE mpp_p15 AS
  SELECT * FROM obj_mpp
  WHERE ST_Contains((
  SELECT geom FROM mestskecast
  WHERE nazev LIKE 'Praha 15'),obj_mpp.geom) = true;
  ```

Dotazy 9

- Nejbližší vertexy sítě k těmto stanicím
- ```
  CREATE TABLE mpp_nearest AS
  SELECT DISTINCT ON(A.gid) A.gid AS mppgid, B.id AS vertgid
  FROM mpp_15 AS A, cyklo_single_vertices_pgr AS B
  ORDER BY A.gid, ST_Distance(A.geom,B.the_geom);
  ```
- Nalezení nejkratší cesty
- ```
  CREATE TABLE mpp_dijkstra2 AS
  SELECT seq, id1 AS node, id2 AS edge, cost
  FROM pgr_dijkstra('SELECT gid AS id, source, target, cost2 AS cost FROM
  cyklo_single',(
  SELECT vertgid FROM mpp_nearest LIMIT 1),(
  SELECT vertgid FROM mpp_nearest LIMIT 1 OFFSET 1),false,false);
  ```

Dotazy 10

- /* 5.a Šerif Wiggum z Prahy 21 se rozhodl že pojede navštívit své kolegy v ostatních městských částech. Ví ovšem, že nikdy na kole neujel více než 12 km, a tak ho zajímá ke kterým policejním stanicím má šanci dojet. Úlohu řešte použitím funkce driving_distance. */

- ```
  CREATE TABLE mpp_p21 AS
  SELECT * FROM obj_mpp
  WHERE ST_Contains((
  SELECT geom FROM mestskecast
  WHERE nazev LIKE 'Praha 21'),obj_mpp.geom) = true;
  ```

- /* Nejbližší vertexy grafu pro každou policejní stanici */

- ```
  CREATE TABLE mpp_nearest_all AS
  SELECT DISTINCT ON(A.gid) A.gid AS mppgid ,B.id AS vertgid
  FROM obj_mpp AS A, cyklo_single_vertices_pgr AS B
  ORDER BY A.gid, ST_Distance(A.geom,B.the_geom);
  ```

- /* Nalezení čísel vertexů které jsou v dané dojezdové vzdálenosti */

Dotazy 11

- CREATE TABLE mpp_driving_res AS
  SELECT * FROM pgr_drivingDistance('
  SELECT gid AS id, source, target, cost FROM cyklo_single',(
  SELECT vertgid FROM mpp_nearest_all AS A, mpp_p21 AS B
  WHERE A.mppgid = B.gid), 13000, false, false);
- /* Vytvoření vrstvy geometrie s vertexy v dojezdové vzdálenosti */
- CREATE TABLE mpp_driving_geom AS
  SELECT A.* FROM cyklo_single_vertices_pgr AS A
  JOIN mpp_driving_res AS B
  ON B.id1 = A.id;
- /* Vytvoření vrstvy se stanicemi ke kterým se dostane */
- CREATE TABLE mpp_driving_stat AS SELECT A.* FROM obj_mpp AS A
  JOIN mpp_nearest_all AS B ON A.gid = B.mppgid
  JOIN mpp_driving_res AS C ON B.vertgid = C.id1;

Dotazy 12

- /* Které městské části při tom může navštívit? */
- ```
  CREATE TABLE mpp_driving_part AS
  SELECT DISTINCT(A.nazev), A.geom
  FROM mestskecast AS A, mpp_driving_stat AS B
  WHERE ST_Contains(A.geom,B.geom);
  ```
  ```

Výsledek



Závěr

- Vrstvy IPR byly úspěšně importovány do PostGIS databáze
- Imporotvané vrstvy byly (po úpravě) použity pro síťové analýzy s využitím PGRoutingu
- Byly vytvořeny a vyřešeny dotazy využivající funkce PGRoutingu pgr_dijkstra, pgr_astar, pgr_kDijkstraPath, pgr_kDijkstraCost, pgr_drvingDistance
- Atributové ani prostorové dotazy nebyly speciálně vytvářeny, protože se jimi zabývaly jiné skupiny a velké množství takových dotazů bylo řešeno jako vedlejší produkt síťových analýz

Děkujeme za pozornost

David Mráz

Pavel Kulmon