

České vysoké učení technické v Praze

Fakulta stavební

Obor Geoinformatika

Katedra geomatiky



Úvod do zpracování prostorových dat semestrální projekt

skupina d16

Ondřej Svoboda

zimní semestr 2016/2017

Obsah

1	Zadání.....	3
2	Úvod.....	4
2.1	Cíl.....	4
2.2	Použitá data.....	4
2.3	Použitý software.....	4
3	Tématické vrstvy.....	5
3.1	Silnice.....	5
3.2	Vesnice.....	9
3.3	Řeky.....	9
3.4	Letiště.....	10
4	Dotazy.....	11
4.1	Příklad 1.....	11
4.2	Příklad 2.....	13
4.3	Příklad 3.....	14
4.4	Příklad 4.....	14
4.5	Příklad 5.....	15
4.6	Příklad 6.....	15
4.7	Příklad 7.....	16
5	Závěr.....	17

1 Zadání

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

2 Úvod

2.1 Cíl

Cílem tohoto projektu je vytvořit tématické vrstvy nad daty OpenStreetMap a RÚIAN se zaměřením na síťové analýzy a k nim sadu ukázkových dotazů.

2.2 Použitá data

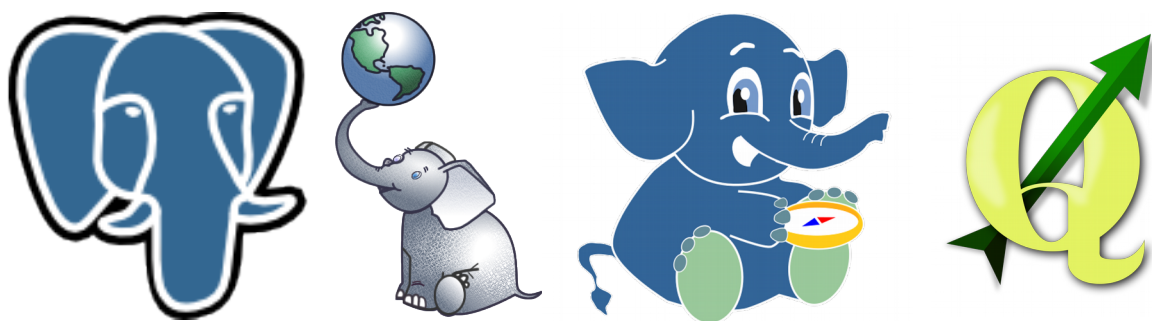
Všechna použitá data se nachází ve schématech *osm* a *ruian* v databázi *pgis_uzpd* na fakultním serveru *geo102*.



2.3 Použitý software

Databáze *pgis_uzpd* používá objektově-relační databázový systém *PostgreSQL 9.1.23* s rozšířením *PostGIS 2.1.3*, které přidává podporu pro ukládání a správu prostorových dat. Dalším rozšířením databáze *pgis_uzpd*, které bylo použito v tomto projektu, je *pgRouting*. Jedná se o nadstavbu přidávající funkce pro síťové analýzy.

Projekt byl zpracován pomocí programu *QGIS 2.18.3 Las Palmas* a jeho zásuvného modulu *DB Manager*, který umožňuje prohlížení dat, provádění SQL příkazů a přidání dat do mapového okna jako samostatné vrstvy.



3 Tématické vrstvy

Byly vytvořeny tyto tématické vrstvy:

- silnice
- vesnice
- řeky
- letiště

Jako společný souřadnicový systém byl vybrán systém *S-JTSK / Krovak East North -- SJTSK*.

3.1 Silnice

Nejdůležitější vrstvou pro následné síťové analýzy je vrstva silnic. Tato vrstva byla vytvořena z tabulky *czech_line* ve schématu *osm*. Geometrie byla převedena do souřadnicového systému *S-JTSK / Krovak East North -- SJTSK*.

```
create table roads as
select      highway as categor,
            oneway,
            st_transform(geom, 5514) as geom
from czech_line
where highway in
    (      'motorway', 'trunk', 'primary', 'secondary', 'tertiary',
            'motorway_link', 'trunk_link', 'primary_link',
            'secondary_link', 'tertiary_link');
select updateGeometrySRID('roads', 'geom', 5514);
```

Následujícím dotazem bylo zjištěno, že vytvořená vrstva obsahuje pouze LineStringy.

```
select      distinct st_geometryType(geom)
from        roads;
```

Poté byl přidán primární klíč a prostorový index.

```
alter table roads add column gid serial;
alter table roads add primary key (gid);
create index on roads using gist (geom);
```

Pro plánované síťové analýzy a extenzi pgRouting byly přidány potřebné sloupce a nad nimi vytvořeny indexy.

```
alter table roads add column source integer;
alter table roads add column target integer;
create index on roads(source);
create index on roads(target);
```

Pro výpočet nejkratší cesty byl přidán sloupec s délkou.

```
alter table roads add column length double precision;  
update roads set length = st_length(geom);
```

Pro variantu výpočtu nejrychlejší cesty bylo záměrem nastavit rychlost podle toho, zda se úsek silnice nachází v obci či nikoli. Polygony s takovou informací ale nebyly nalezeny, proto byla k úsekům silnice na základě jejich kategorie přiřazena odhadovaná průměrná rychlost.

```
alter table roads add column speed integer;  
update roads  
set speed = case categor  
                when 'motorway' then 110  
                when 'motorway_link' then 85  
                when 'trunk' then 105  
                when 'trunk_link' then 80  
                when 'primary' then 65  
                when 'primary_link' then 50  
                when 'secondary' then 50  
                when 'secondary_link' then 40  
                when 'tertiary' then 40  
                when 'tertiary_link' then 30  
            end;
```

Na základě rychlosti pak byl přidán sloupec s časem potřebným k překonání každého úseku silnice.

```
alter table roads add column time double precision;  
update roads set time = length/(speed/3.6);
```

Data OSM obsahují informace o jednosměrných komunikacích. Podle oficiální [dokumentace](#) je doporučený postup přidat sloupce pro cestu oběma směry a „zakázanému směru“ přiřadit vysokou hodnotu.

```
alter table roads add column this_way double precision;  
update roads  
set this_way = case oneway  
                when '-1' then 1000000000  
                when 'reverse' then 1000000000  
                else 0  
            end;  
  
alter table roads add column oppo_way double precision;  
update roads  
set oppo_way = case oneway  
                when 'yes' then 1000000000  
                when 'true' then 1000000000  
                when '1' then 1000000000  
                else 0  
            end;
```

Pro nadstavbu pgRouting je nutné vytvořit topologii.

```
select pgr_createTopology('roads', 0.001, 'geom', 'gid');
```

Při prvním experimentu se síťovými analýzami bylo zjištěno, že data nejsou topologicky čistá. Extenze *pgRouting* pracuje tak, že na silnici, tedy linii, je možné se dostat pouze po jednom z jejích koncových bodů a totéž platí i o jejím opuštění. Linie ovšem v bodech křížení či dotyku nebyly vždy rozděleny. V případě, že se jedná o nadjezdy, podjezdy, nebo více úrovně křižovatky, to je žádoucí, ale stejný problém byl zjištěn i u většiny kruhových okruhů. Důsledkem bylo to, že nalezená nejkratší nebo nejrychlejší cesta byla plná nelogičností. Vyhýbala se kruhovým objezdům a v mnoha případech objížděla evidentně výhodnější trasu.

Původním záměrem bylo zajistit topologickou čistotu pouze s funkcemi *PostGISu*. Za tímto účelem byla vrstva pomocí nástroje *pgsql2shp* vyexportována do formátu ESRI Shapefile a následně za použití *shp2pgsql* nahrána do lokální databáze. Důvodem k tomuto převodu dat bylo to, že funkce *ST_Split* od verze *PostGISu 2.2.0* podporuje rozdělení *LineStringů* pomocí *MultiPointů* a databáze *pgis_uzpd* bohužel disponuje verzí 2.1.3. Přesto ani po usilovném snažení nebylo dosaženo požadovaných výsledků.

Naštěstí pro právě tyto případy nabízí extenze *pgRouting* funkci *pgr_nodeNetwork*.

Nejprve byla vrstva silnic analyzována pomocí funkce *pgr_analyzeGraph*.

```
select pgr_analyzeGraph('roads', 0.001, 'geom', 'gid', 'source', 'target');
```

S tímto výsledkem.

```
NOTICE:          ANALYSIS RESULTS FOR SELECTED EDGES:
NOTICE:          Isolated segments: 2805
NOTICE:          Dead ends: 17793
NOTICE: Potential gaps found near dead ends: 14168
NOTICE:          Intersections detected: 19483
NOTICE:          Ring geometries: 822
```

Následně byla provedena oprava pomocí funkce *pgr_nodeNetwork*.

```
select pgr_nodeNetwork('roads', 0.001, 'gid', 'geom', 'noded');
```

Pro nově vytvořenou vrstvu *roads_noded* byla vytvořena topologie a pro kontrolu byla opět zavolána funkce *pgr_analyzeGraph*. Překvapivě byly opět nalezeny linie, které se křížily.

```
NOTICE:          ANALYSIS RESULTS FOR SELECTED EDGES:
NOTICE:          Isolated segments: 41
NOTICE:          Dead ends: 3660
NOTICE: Potential gaps found near dead ends: 38
NOTICE:          Intersections detected: 19
NOTICE:          Ring geometries: 1
```

Proto byla ještě jednou provedena oprava pomocí funkce *pgr_nodeNetwork* a znovu vytvořena topologie. V podruhé opravené vrstvě již křížící se linie nebyly nalezeny.

```
NOTICE:          ANALYSIS RESULTS FOR SELECTED EDGES:
NOTICE:          Isolated segments: 41
NOTICE:          Dead ends: 3645
NOTICE: Potential gaps found near dead ends: 23
```

NOTICE: Intersections detected: 0

NOTICE: Ring geometries: 1

Do finální opravené vrstvy s názvem *roads_noded_2* se nezkopírovaly všechny sloupce z původní vrstvy *roads*. Proto byla spojením všech tří vrstev vytvořena finální tabulka *roads_f*.

```
create table roads_f as
select      rn2.id as gid, r.categor as categor, r.oneway as oneway,
            rn2.geom as geom
from        roads_noded_2 rn2
left join   roads_noded rn on rn2.old_id = rn.id
left join   roads r on rn.old_id = r.gid;
```

Do této vrstvy byly opět přidány potřebné sloupce pro extenzi *pgRouting*.

Kontrola geometrie proběhla v pořádku.

Sloupec	Název sloupce	Datový typ	Not Null
1	gid	bigint	NO
2	categor	text	YES
3	oneway	text	YES
4	geom	geometry	YES
5	source	integer	YES
6	target	integer	YES
7	length	double precision	YES
8	speed	integer	YES
9	time	double precision	YES
10	this_way	double precision	YES
11	oppo_way	double precision	YES

3.2 Vesnice

Pro síťové analýzy je nutné mít také koncový a počáteční bod. Za tyto vody byly vybrány vesnice *Onen Svět* a *Peklo* z vrstvy *czech_point* ve schématu *osm*. Vrstva byla převedena do souřadnicového systému *S-JTSK / Krovak East North – SJTSK*.

```
create table villages as
select      osm_id as id, name, st_transform(geom, 5514) as geom
from        czech_point where osm_id in (359094467, 1599445934);

select updateGeometrySRID('villages', 'geom', 5514);
alter table villages add primary key (gid);
create index on villages using gist (geom);
```

Sloupec	Název sloupce	Datový typ	Not Null
1	gid	bigint	NO
2	name	text	YES
3	geom	geometry	YES

3.3 Řeky

Za účelem prostorových analýz nad nalezenými cestami byla z vrstvy *czech_line* ve schématu *osm* vytvořena vrstva řek. V potaz byly brány pouze velké řeky se jmény. Vrstva byla převedena do souřadnicového systému *S-JTSK / Krovak East North – SJTSK*.

```
create table rivers as
select      name, st_transform(st_collect(geom), 5514) as geom
from        czech_line
where       waterway = 'river'
            and
            name is not null
group by    name;

select updateGeometrySRID('rivers', 'geom', 5514);
alter table rivers add column gid serial;
alter table rivers add primary key (gid);
create index on rivers using gist (geom);
```

Sloupec	Název sloupce	Datový typ	Not Null
1	gid	bigint	NO
2	name	text	YES
3	geom	geometry	YES

3.4 Letiště

Dále byla vytvořena vrstva letišť jako kombinace polygonů z vrstvy *czech_polygon* převedených na body a bodů z vrstvy *czech_point*. Obě vrstvy pochází ze schématu *osm*. V potaz byly brány pouze velká letiště se jmény. Vrstva byla převedena do souřadnicového systému *S-JTSK / Krovak East North – SJTSK*.

```
create table airports as (
select      name, st_transform(st_centroid(st_collect(geom)), 5514) as geom
from        (      select      name, st_centroid(geom) as geom
                  from        czech_polygon
                  where        aeroway = 'aerodrome'
                  and
                  name is not null

                  union
                  select      name, geom
                  from        czech_point
                  where        aeroway = 'aerodrome'
                  and
                  name is not null) a
group by    name);

select updateGeometrySRID('airports', 'geom', 5514);
alter table airports add column gid serial;
alter table airports add primary key (gid);
create index on airports using gist (geom);
```

Sloupec	Název sloupce	Datový typ	Not Null
1	gid	bigint	NO
2	name	text	YES
3	geom	geometry	YES

4 Dotazy

4.1 Příklad 1

Student Gabriel pochází z malé vesnice jménem *Onen Svět* v Plzeňském kraji. Momentálně studuje obor Profesionální pilot na fakultě dopravní na ČVUT v Praze a právě tam se seznámil se studentkou Dorotou. Ta ovšem nepochází z Prahy, ale z další malé vesky jménem *Peklo* nacházející se v Královéhradeckém kraji. Co osud nechtěl, ti dva se dali dohromady a nyní bojují s velkou vzdáleností mezi jejich rodnými domy. Gabriel si od rodičů půjčil auto a nyní zjišťuje, jaká cesta je pro něj nejvýhodnější. Jaké jsou rozdíly mezi nejrychlejší a nejkratší cestou z *Onoho Světa* do *Pekla*? Řešte pomocí algoritmu Dijkstra.

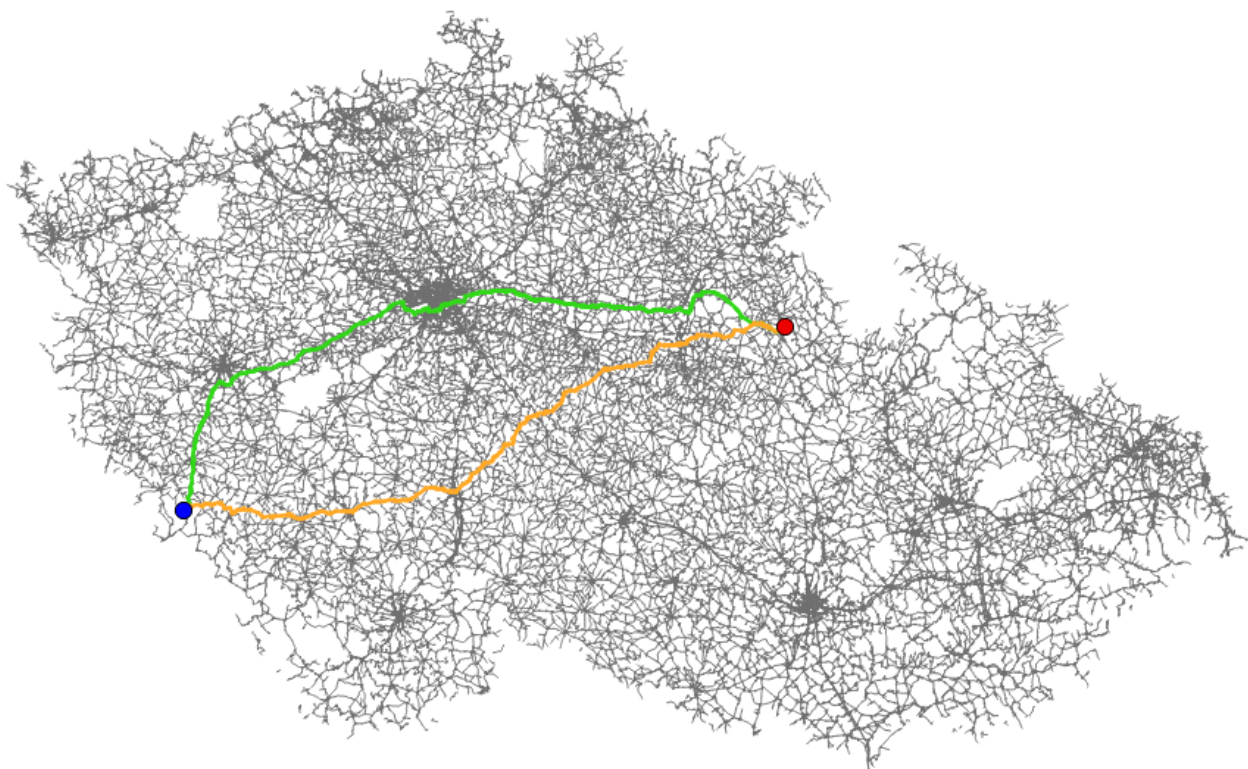
Vzhledem k podrobnosti dat byly jako počáteční a koncový uzel cesty zvoleny nejbližší koncové body linií.

Nejrychlejší cesta pomocí algoritmu Dijkstra:

```
create table path_d_time as
select      seq, path_seq, node, edge, cost, agg_cost, b.length, b.geom as geom
from        pgr_dijkstra(
            'select gid as id, source, target, (time+this_way) as cost,
              (time+oppo_way) as reverse_cost from roads_f',
            (
              select      id
              from        roads_f_vertices_pgr r,
              (select geom from villages where name = 'Onen Svět') os
              order by    st_distance(r.the_geom, os.geom) limit 1),
            (
              select      id
              from        roads_f_vertices_pgr as r,
              (select geom from villages where name = 'Peklo') p
              order by    st_distance(r.the_geom, p.geom) limit 1)) a
left join    roads_f b on a.edge = b.gid;
```

Nejkratší cesta pomocí algoritmu Dijkstra:

```
create table path_d_len as
select      seq, path_seq, node, edge, cost, agg_cost, b.time, b.geom as geom
from        pgr_dijkstra(
            'select gid as id, source, target, (length+this_way) as cost,
              (length+oppo_way) as reverse_cost from roads_f',
            (
              select      id
              from        roads_f_vertices_pgr r,
              (select geom from villages where name = 'Onen Svět') os
              order by    st_distance(r.the_geom, os.geom) limit 1),
            (
              select      id
              from        roads_f_vertices_pgr r,
              (select geom from villages where name = 'Peklo') p
              order by    st_distance(r.the_geom, p.geom) limit 1)) a
left join    roads_f b on a.edge = b.gid
```



Na obrázku je vesnice *Onen Svět* zobrazena modrou tečkou, vesnice *Peklo* pak červenou. Nejkratší nalezená cesta je zobrazena oranžovou linií, nejrychlejší zelenou.

Níže uvedená tabulka ukazuje rozdíly v délce a čase mezi nejrychlejší a nejkratší cestou.

Cesta	Délka [km]	Čas
nejrychlejší	307.8	3 h 31 min
nejkratší	294.8	5 h 40 min

Pro zajímavost byla zjištěna délka a čas nejrychlejší trasy pomocí Google Maps a Mapy.cz

Cesta	Délka [km]	Čas
projekt (pgRouting)	307.8	3 h 31 min
Google Maps	309	3 h 50 min
Mapy.cz	310	3 h 37 min

Trasa nalezená pomocí Google Maps i Mapy.cz se v zásadě shodovala s trasou nalezenou pomocí extenze *pgRouting* a tomu odpovídá i velká podobnost mezi porovnávanými délkami. Poněkud větší rozdíly se ukázaly v čase, konkrétně Google Maps odhadují, že ujet stejnou trasu bude trvat déle. To je s největší pravděpodobností způsobeno tím, že Google Maps berou v potaz aktuální hustotu provozu.

4.2 Příklad 2

Student Gabriel si i přes větší časovou náročnost vybral variantu s nejkratší cestou. Důvodem je to, že jím zapůjčený automobil nemá dálniční známku a od kratší cesty si slibuje i menší spotřebu pohonných hmot. Jak se liší nejkratší cesta z *Onoho Světa* do *Pekla* vypočtená pomocí algoritmů Dijkstra a A*?

Cesta pomocí algoritmu Dijkstra byla vypočtena v příkladu 1.

Algoritmus A* vyžaduje na vstupu souřadnice uzlů.

```
alter table roads_f add column x1 double precision;
update roads_f set x1 = st_x(st_startpoint(geom));

alter table roads_f add column y1 double precision;
update roads_f set y1 = st_y(st_startpoint(geom));

alter table roads_f add column x2 double precision;
update roads_f set x2 = st_x(st_endpoint(geom));

alter table roads_f add column y2 double precision;
update roads_f set y2 = st_y(st_endpoint(geom));
```

Výpočet nejkratší trasy pomocí algoritmu A*:

```
create table path_a_len as
select      seq, id1 as node, id2 as edge, cost, b.time, b.geom as geom
from        pgr_astar(
            'select gid as id, source, target, (length+this_way) as cost,
              x1, y1, x2, y2, (length+oppo_way) as reverse_cost
              from roads_f',
            cast(( select id from roads_f_vertices_pgr r,
                        (select geom from villages where name = 'Onen Svět') os
                        order by st_distance(r.the_geom, os.geom) limit 1)
              as int),
            cast(( select id from roads_f_vertices_pgr r,
                        (select geom from villages where name = 'Peklo') p
                        order by st_distance(r.the_geom, p.geom) limit 1)
              as int),
            true, true) a
left join    roads_f b on a.id2 = b.gid;
```

Níže uvedená tabulka ukazuje rozdíly v délce a čase mezi algoritmy Dijkstra a A*.

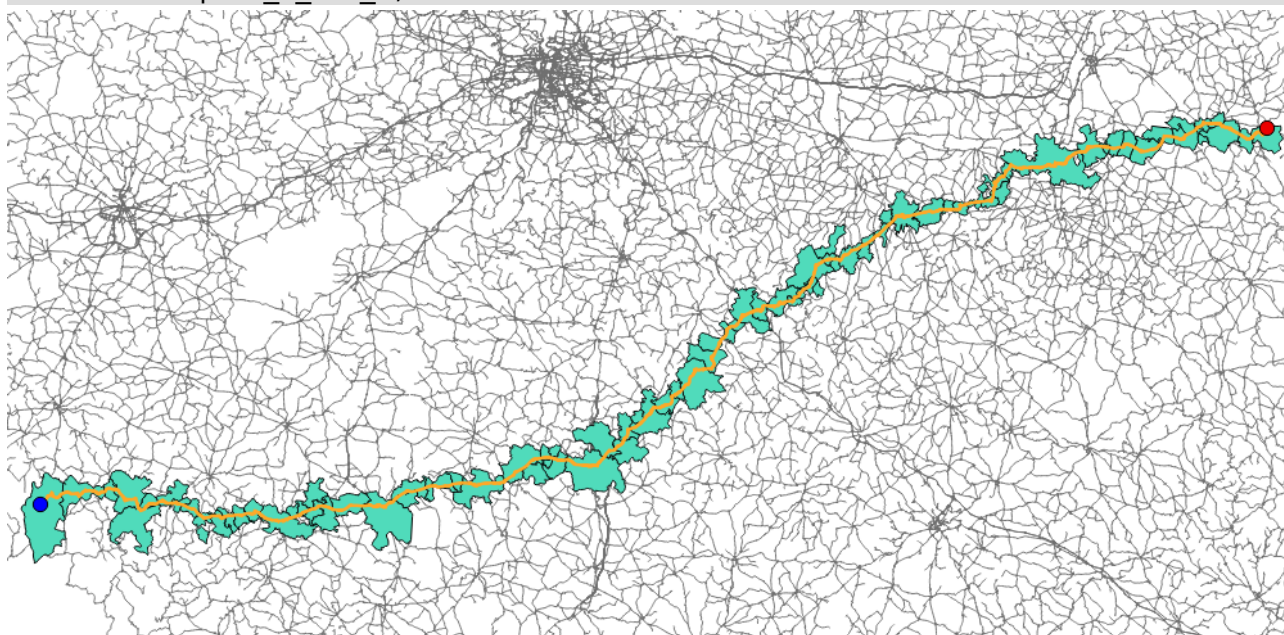
Algoritmus	Délka [km]	Čas
Dijkstra	294.8	5 h 40 min
A*	294.8	5 h 40 min

Délka i čas se shodují. Bylo zjištěno, že oba algoritmy vypočítaly trasu po stejných hranách.

4.3 Příklad 3

Kolika obcemi při své nejkratší cestě z *Onoho Světa* do *Pekla* student Gabriel projede?

```
with path_d_len_o as (
  select      distinct o.kod, o.geom
  from        path_d_len p
  join        obce o on    p.geom && o.geom
                        and
                        st_intersects(p.geom, o.geom))
select      count(*)
from        path_d_len_o;
```



Gabriel při své nejkratší cestě z *Onoho Světa* do *Pekla* projede 84 obcemi.

4.4 Příklad 4

Na území které obce ujde Gabriel při své nejkratší cestě z *Onoho Světa* do *Pekla* největší vzdálenost?

```
with path_d_len_o as (
  select      distinct o.nazev, o.geom
  from        path_d_len p
  join        obce o on    p.geom && o.geom
                        and
                        st_intersects(p.geom, o.geom))
select      o.nazev,
            (st_length(st_intersection(o.geom, u.geom)))/1000 as length
from        path_d_len_o o
join (
  select      st_collect(geom) as geom
  from        path_d_len) u on o.geom && u.geom
order by    length desc limit 1;
```

Největší vzdálenost najde Gabriel na území obce Pardubice.

4.5 Příklad 5

Přejede Gabriel na své nejkratší cestě z *Onoho Světa* do *Pekla* nějakou velkou řeku víckrát než jednou?

```
select      ri.name, count(*)
from        path_d_len p
join        rivers ri on      p.geom && ri.geom
                                and
                                st_crosses(p.geom, ri.geom)

group by    ri.name
having      count(*) > 1;
```

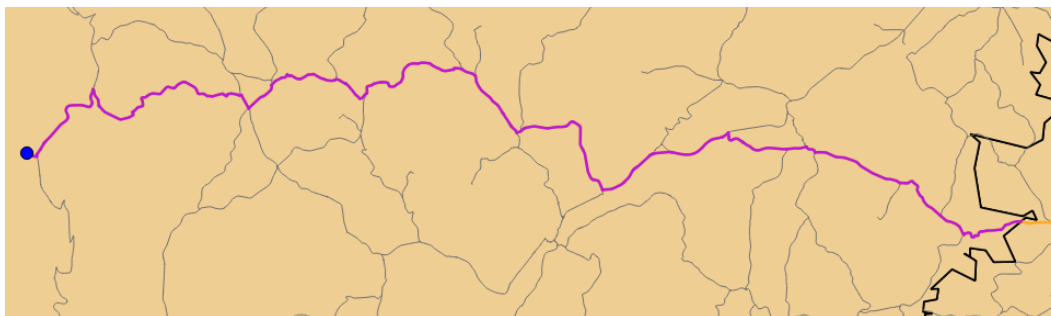
Název	Počet překročení
Otava	3
Zdobnice	2

4.6 Příklad 6

Jak dlouho bude Gabriel na své nejkratší cestě z *Onoho Světa* do *Pekla* v momentě, kdy bude opouštět rodný kraj?

```
with path_d_len_i as (
    select      p.seq, p.cost as length, r.speed,
                st_intersection(p.geom, k.geom), k.kod as code
    from        path_d_len p
    left join ( select      kod, geom
                  from      kraje
                  where      nazev = 'Západočeský') k
                on          p.geom && k.geom
                  and
                  st_intersects(p.geom, k.geom)
    left join    roads_f r on p.edge = r.gid)
select          sum(length/(speed/3.6))/60 as time
from            path_d_len_i
where           seq < (    select      min(seq)
                          from          path_d_len_i
                          where         code is null);
```

V momentě, kdy bude Gabriel opouštět Západočeský kraj, bude na cestě 1 hodinu a 1 minutu.



4.7 Příklad 7

Gabriel je studentem oboru Profesionální pilot a jeho studium je mu také koníčkem. Pozoruje letadla, kdy může, a proto by byl rád, kdyby jeho cesta vedla v blízkosti letišť. Měl by tak větší pravděpodobnost zahlédnout nějaký stroj v oblacích. Kolik letišť se nachází do 2 km od nejkratší cesty z *Onoho Světa* do *Pekla*? Je tento počet větší či menší v porovnání s nejrychlejší cestou?

```
with path_d_len_b as (
    select      st_buffer(geom, 2000) as geom
    from (      select      st_collect(geom) as geom
                from        path_d_len) u),
    path_d_time_b as (
    select      st_buffer(geom, 2000) as geom
    from (      select      st_collect(geom) as geom
                from        path_d_time) u)
select        'shortest' as path_type, count(*) as airports
from          airports a
join          path_d_len_b b on a.geom && b.geom
              and
              st_within(a.geom, b.geom)
union
select        'fastest' as path_type, count(*) as airports
from          airports a
join          path_d_time_b b on      a.geom && b.geom
              and
              st_within(a.geom, b.geom);
```

Do 2 km od nejkratší cesty z *Onoho Světa* do *Pekla* se nachází 8 letišť, pro nejrychlejší cestu to jsou pouze 3 letiště. Nyní jsou veškeré Gabrielovy pochybnosti o tom, jakou cestu zvolit, pryč.

5 Závěr

V rámci tohoto projektu byly vytvořeny čtyři tématické vrstvy – silnice, vesnice, řeky a letiště. Zdrojem dat byly vrstvy ve schématech *osm* a *ruian*. Všechny vytvořené vrstvy byly transformovány do souřadnicového systému *S-JTSK / Krovak East North* – *SJTSK*. Zvýšená pozornost byla věnována čistotě topologie vrstvy silnic, protože to má velký vliv na výsledky síťových analýz. K opravě topologických vad byla použita funkce *pgr_nodeNetwork*, která je součástí extenze *pgRouting*, ovšem poněkud překvapivě neopravila všechny problémy napoprvé a bylo ji nutné zavolat dvakrát.

Nad vytvořenými vrstvami bylo formulováno 7 příkladů s ukázkou možných řešení.

Velmi oceňuji jednoduchost práce s extenzí *PostGIS*. Funkce se jmenují tak, jak bych očekával a oficiální dokumentace obsahuje všechny potřebné informace. V kombinaci s programem *QGIS* a jeho zásuvným modulem *DB Manager* je navíc data možné ihned vizualizovat.