Shortest Path Search with pgRouting
What is pgRouting?
An Extension for PostgreSQL / PostGIS, ...
An Open Source project, ...
A Library providing, ...
Shortest Path
DD and TSP
@justjkk
@jay_mahadeokar
Multi-Modal
SQL Function

SELECT * FROM shortest_path('SELECT gid as id,
   source::integer,
   target::integer,
   length::float8 as cost
   FROM ways',
  605, 359, false, false);
<table>
<thead>
<tr>
<th>vertex_id</th>
<th>edge_id</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>605</td>
<td>599</td>
<td>0.19925085940845</td>
</tr>
<tr>
<td>604</td>
<td>598</td>
<td>0.100258103875674</td>
</tr>
<tr>
<td>603</td>
<td>597</td>
<td>0.201123583387407</td>
</tr>
<tr>
<td>602</td>
<td>596</td>
<td>0.204847680679676</td>
</tr>
<tr>
<td>601</td>
<td>595</td>
<td>0.158334540345002</td>
</tr>
<tr>
<td>4293</td>
<td>9602</td>
<td>0.0183273901669081</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Most users need pgRouting for Road Networks.

How do they look like?
Like this ...
... or like this ...
... or sometimes like this.
What makes them real?

Traffic lights  Signs  Road marking
How can pgRouting help here?
gid, source, target, cost, reverse_cost, x1, y1, x2, y2, rule, to_cost
Traffic lights slow down

... so costs must increase.
\[
\begin{align*}
\text{cost}(A) &= \text{cost}(A) + 30\text{sec} \\
\text{cost}(B) &= \text{cost}(B) + 30\text{sec} \\
\text{cost}(C) &= \text{cost}(C) + 30\text{sec} \\
\text{cost}(D) &= \text{cost}(D) + 30\text{sec}
\end{align*}
\]
Signs inform about restrictions and rules
ONE WAY
\text{cost}(A) = \text{length}(A)\]
\[\text{reverse}\_\text{cost}(A) = \infty\]
Sometimes the costs have different meaning.
cost(A) = length(A) / 2
reverse_cost(A) = length(A) * 2

cost(B) = length(B) * 3
reverse_cost(B) = length(B) / 3
Turn restrictions obviously restrict turns.
\begin{align*}
\text{cost}(A) &= \text{cost}(A) + \text{to}_\text{cost}(A, D) = \infty \\
\text{cost}(B) &= \text{cost}(B) + \text{to}_\text{cost}(B, D) \\
\text{cost}(C) &= \text{cost}(C) + \text{to}_\text{cost}(C, D) \\
\text{cost}(D) &= \text{cost}(D) + \text{to}_\text{cost}(D, D)
\end{align*}

\text{rule}(A) = \text{"D"}
Road type can be used for cost calculation.
Not only road **types** but also **conditions**
Jughandle intersection
And you know what is great about pgRouting?
All costs are *dynamic*!

... which is the opposite to pre-calculated
If the road is closed ...
... or there is an accident, ...
... there is a sign with restrictions limited to a certain time, ...
... bad weather conditions ...
... or any other obstacle, ...
You don't need to rebuild and reload your network.
... and wait forever.
You only need to adjust the cost for this particular road,

and the next search will go another way.
Cost can be virtually anything
pgRouting can be used for different kinds of networks
Canals and Rivers
Hiking trails
... or any other kind of networks.
pgRouting Demo

http://websi.openvrp.com

http://map.veloland.ch

http://www.ridethecity.com

http://www.pgrouting.org/gallery.html
Plans until FOSS4G 2012

- Integrate new functions
  - All-Pair-Shortest-Path
  - Time-Dependent-Shortest-Path
  - Multi-Modal
  - Two-way A-Star, ...
- Drop CGAL dependency
- PostgreSQL 9.x & PostGIS 2.0 support
- Source cleanup
pgRouting Project

pgRouting extends the PostGIS / PostgreSQL geospatial database to provide geospatial routing functionality.

Advantages of the database routing approach are:

- Data and attributes can be modified by many clients, like QuantumGIS and uDig through JDBC, ODBC, or directly using PL/pgSQL. The clients can either be PCs or mobile devices.
- Data changes can be reflected instantaneously through the routing engine. There is no need for precalculation.
- The “cost” parameter can be dynamically calculated through SQL and its value can come from multiple fields or tables.

Core Features

pgRouting provides:

- Shortest Path Dijkstra: routing algorithm without heuristics
- Shortest Path A-Star: routing for large datasets (with heuristics)
- Shortest Path Dijkstra (with heuristics)
- Traveling Salesperson Problem (TSP)
- Driving Distance calculation (Isolines)

Links

- pgRouting Workshop
- PostgreSQL
- PostGIS

Download

Current release: 1.05

Make a Donation

Donate [EUR]

Support pgRouting with some donation! Read more about how your donation will help the project.
Photos from *sxc.hu* and *flickr* under *Creative Commons* Licence.