Shortest path search in real road networks with pgRouting

Anton Patrushev (Orkney, Inc.)
Introduction

pgRouting is an extension of PostgreSQL and PostGIS. A predecessor of pgRouting - pgDijkstra, written by Sylvain Pasche from Camptocamp, was extended by Orkney (Japan) and renamed to pgRouting.

pgRouting includes:
* Shortest path search (Dijkstra, A*, Shooting*)
* Traveling Salesperson Problem (TSP) solution
* Driving distance polygon calculation
How it works

C++ core $\rightarrow$ C wrapper $\rightarrow$ PL/PgSQL function $\rightarrow$ SQL query (data)
Shortest path algorithms

• **Dijkstra**
  - Well known and fair shortest path algorithm
  - Always finds mathematically shortest path
  - Good for sparse networks

• **A**
  - Well known heuristic shortest path algorithm
  - Needs vertex geometry information
  - Searches through less number of vertexes
  - Good for dense networks

• **Shooting**
Shooting*

• Edge-based heuristic shortest path algorithm
• Supports maneuver restrictions and traffic lights
• Properly processes parallel edges
• Good for dense networks

Predecessors:
• Modified A* - Wolfgang Schmid, 2000, Stephan Winter, 2002
• C* - Ingrid Flinsenberg, 2004
## Comparison with other algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Formula</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dijkstra</td>
<td>cost = cost(vertex1, vertex2)</td>
<td>Each vertex can be visited only once</td>
</tr>
<tr>
<td>A*</td>
<td>cost = cost(vertex1, vertex2) + H(vertex1, vertex2)</td>
<td>Each vertex can be visited only once</td>
</tr>
<tr>
<td>Shooting*</td>
<td>cost = cost(edge1) + cost(edge1, edge2) + H(edge1)</td>
<td>Each edge can be visited only once</td>
</tr>
</tbody>
</table>

**Parallel edges handling**

- **Dijkstra/A***: Needs handling for parallel edges when using actual road networks.
- **Shooting***: More efficient handling of parallel edges compared to other algorithms.

---

Shortest path search in real road networks with pgRouting
Real road networks

A) Signs (restrict maneuvers)
B) Traffic lights (delays)
C) Road marking (restrict maneuvers)
A) Right turn costs too much
B) Driving through vertex b passage costs extra
C) Driving from a to f costs too much
## Routing data structure

<table>
<thead>
<tr>
<th>Dijkstra Edge</th>
<th>A* Edge</th>
<th>Shooting Edge</th>
</tr>
</thead>
</table>
| - id
- cost
- reverse_cost
- Source vertex
  - id
  - x
  - y
- Target vertex
  - id
  - x
  - y |
| - id
- cost
- reverse_cost
- Source vertex
  - id
  - x
  - y
- Target vertex
  - id
  - x
  - y |
| - id
- cost
- reverse_cost
- Adjacent edge
  - rule
  - to_cost
- Source vertex
  - id
  - x
  - y
- Target vertex
  - id
  - x
  - y |
Thank you!

pgrouting.postlbs.org
orkney.co.jp