Estimating scalability issues while finding an optimal assignment for carpooling

Luk Knapen

Universiteit Hasselt IMOB

July 17, 2013
Basic Idea - Concepts

Application Area: Evaluation of Carpooling Advisor Software

- Commuting, periodic trip execution
- Provide global *matcher service*
- Matcher shall advise so that carpooling negotiation success is maximal
- Feedback from negotiation is used to train the advisor

Hypotheses

- Sufficient business model so that people provide feedback
  - about negotiation result
  - mutual qualifications (reputation)
- Single driver constraint: route for each passenger is embedded in the driver’s route
- No carpool parkings considered
Basic Idea - Concepts

Agent based model used to support *matcher software evaluation*

- critical mass of users required
- performance and effectiveness
- investigate transient effects at startup
Basic Idea - Concepts

Phase 1: Agent Based Model
Phase 2: Real World

(1) Register

(2) Advise

(3) Agreement

(4) Feedback

Matching Service

Node is periodicTripEx
Edge has weight $w$
Select best matching to generate advise

Logit
Determining carpool negotiation probability

Negotiation success is determined using similarities

- Profile similarity: (Person,Person) attribute
- Safety reputation: Person attribute
- Path similarity: (Trip,Trip) attribute
- Cohesion: Pool attribute
- Time Interval similarity: (Person,Person,Trip) attribute
- Timeliness reputation: (Person,Trip) attribute
Determining carpool negotiation probability

Functions Used

- sRep()
- tRep()
- prdTripEx
- profSim()
- pathSim()
- tis()
- R
  - [0,1]
- pool
- R
  - [-0.5,0.5]
- cohesion()
- agreement
Profile similarity

- Consists of factors relevant for homophily
  - Socio-economic class: discrete, ordered
  - Age: discrete, ordered
  - Gender: discrete, categorical

- Motivation: limit number of inputs to logit estimator

\[
s_{cat}(v_0, v_1) = \begin{cases} 
1 & \text{if } v_0 = v_1 \\
0 & \text{else}
\end{cases} \quad (1)
\]

\[
s_{ord}(v_0, v_1) = 1 - \frac{|v_0 - v_1|}{\text{range}} \quad (2)
\]

\[
\text{profSim} = \sum_i s_i \cdot w_i \quad (3)
\]
Determining carpool negotiation probability

Path similarity

- TAZ (Traffic Analysis Zone) based
- Not symmetric
- Compares *carpooled* and *solo* paths for driver

\[
\text{pathSim} = \frac{d(O_A, D_A)}{d(O_A, O_B) + d(O_B, D_B) + d(D_B, D_A)}
\]
Determining carpool negotiation probability

Time Interval Compatibility

- \( t_{d,\text{early}_C} \) and \( t_{d,\text{late}_C} \)
- \( t_{d,\text{early}_B} \) and \( t_{d,\text{late}_B} \)
- \( t_{d,\text{early}_A} \) and \( t_{d,\text{late}_A} \)
Determining carpool negotiation probability

**Time Interval Similarity** based on common periods to start/finish trip

\[
d(L) = \text{dur}(W(i, L) \cap W(j, L)) \quad (5)
\]

\[
t\text{is}(T_A, T_B, L) = 1 - e^{-\alpha \cdot d(L)} \quad (6)
\]

\[
t\text{is}(T_A, T_B) = \min(t\text{is}(T_A, T_B, \text{Orig}(B), t\text{is}(T_A, T_B, \text{Dest}(B)))) \quad (7)
\]
Determining carpool negotiation probability

Estimating negotiation success probability

Diagram showing relationships between variables such as `tis` (time interval similarity), `profSim`, `pathSim`, `prdTripEx_0`, `indiv_0`, `sRep_0`, `cohesion`, `tRep`, and `sRep`, with notation like `weight = prob` and nodes such as `logit`, `agreement_i`, `cohesion_0`, `agreement_j`, `cohesion_1`, `prdTripEx_1`, `indiv_1`, and `sRep_1`.
Periodic Trip to Vehicle Assignment

(B) w_CD
(C)
(D)
(E)
(F)
(G)

(A) w_AB
(B)
(C)
(D)
(E)
(F)
(G)

PTE

Vehicle
Periodic Trip to Vehicle Assignment

Linear problem statement for assignment

\[ \forall i, j \in [0, N - 1] : x_{i,j} \leq 1 \tag{8} \]

\[ \forall i \in [0, N - 1] : \sum_{j \in [0, N - 1]} x_{i,j} = 1 \tag{9} \]

\[ \forall j \in [0, N - 1] : \sum_{i \in [0, N - 1]} x_{i,j} \leq \text{cap}(v_j) \tag{10} \]

\[ \forall i, j \in [0, N - 1], i \neq j : x_{i,j} - x_{i,i} \leq 0 \tag{11} \]

- \( i \) identifies trip
- \( j \) identifies vehicle
- \( N \) number of persons

\( x_{i,j} \) are boolean

Eqn(1) Each trip assigned to one vehicle
Eqn(2) Vehicle capacity is limited
Eqn(4) Car owner shall drive
Problem size can become a problem

- Problem size is large (Flanders: 6e6 people, 2.5e6 commuters)
- 20% randomly (uniform) selected persons are assumed to show interest for carpooling
- Dynamic
  - Links are deleted and added by (de)registration
  - Link weights evolve over time (by changing reputation and cohesion)
- Matrix is sparse
- Integer programming problem
- Try to find solution starting from previous one
The assignment problem is a *star cover problem*
## Graph Characteristics

<table>
<thead>
<tr>
<th>Probab. Threshold</th>
<th>nVertices</th>
<th>nEdges</th>
<th>$\frac{nEdges}{nVertices}$</th>
<th>completeness</th>
<th>nComps</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.700</td>
<td>55297</td>
<td>2210379</td>
<td>39.97</td>
<td>7.229E-4</td>
<td>21</td>
</tr>
<tr>
<td>0.725</td>
<td>53882</td>
<td>1513769</td>
<td>28.09</td>
<td>5.214E-4</td>
<td>36</td>
</tr>
<tr>
<td>0.750</td>
<td>51517</td>
<td>995710</td>
<td>19.33</td>
<td>3.752E-4</td>
<td>45</td>
</tr>
<tr>
<td>0.775</td>
<td>47872</td>
<td>618861</td>
<td>12.93</td>
<td>2.700E-4</td>
<td>67</td>
</tr>
<tr>
<td>0.800</td>
<td>42519</td>
<td>356954</td>
<td>8.40</td>
<td>1.974E-4</td>
<td>114</td>
</tr>
<tr>
<td>0.825</td>
<td>35366</td>
<td>191860</td>
<td>5.42</td>
<td>1.534E-4</td>
<td>184</td>
</tr>
<tr>
<td>0.850</td>
<td>26464</td>
<td>95227</td>
<td>3.60</td>
<td>1.360E-4</td>
<td>258</td>
</tr>
<tr>
<td>0.875</td>
<td>16387</td>
<td>39104</td>
<td>2.39</td>
<td>1.456E-4</td>
<td>342</td>
</tr>
<tr>
<td>0.900</td>
<td>6603</td>
<td>10624</td>
<td>1.61</td>
<td>2.437E-4</td>
<td>403</td>
</tr>
<tr>
<td>0.925</td>
<td>1339</td>
<td>1615</td>
<td>1.21</td>
<td>9.014E-4</td>
<td>194</td>
</tr>
<tr>
<td>0.950</td>
<td>24</td>
<td>13</td>
<td>0.54</td>
<td>0.024</td>
<td>11</td>
</tr>
<tr>
<td>0.975</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.024</td>
<td>11</td>
</tr>
</tbody>
</table>
Graph Characteristics

Component size frequency ... misleading: see next slide
### Eight largest components in each graph

<table>
<thead>
<tr>
<th>Prob</th>
<th>cs1</th>
<th>cs2</th>
<th>cs3</th>
<th>cs4</th>
<th>cs5</th>
<th>cs6</th>
<th>cs7</th>
<th>cs8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.700</td>
<td>32096</td>
<td>23155</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.725</td>
<td>31335</td>
<td>22468</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.750</td>
<td>29969</td>
<td>21439</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>0.775</td>
<td>27888</td>
<td>19827</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0.800</td>
<td>24688</td>
<td>17484</td>
<td>16</td>
<td>14</td>
<td>12</td>
<td>9</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>0.825</td>
<td>20388</td>
<td>14311</td>
<td>40</td>
<td>18</td>
<td>17</td>
<td>13</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0.850</td>
<td>7154</td>
<td>5620</td>
<td>5288</td>
<td>3255</td>
<td>2459</td>
<td>1205</td>
<td>380</td>
<td>129</td>
</tr>
<tr>
<td>0.875</td>
<td>3262</td>
<td>2303</td>
<td>2053</td>
<td>1007</td>
<td>980</td>
<td>941</td>
<td>935</td>
<td>915</td>
</tr>
<tr>
<td>0.900</td>
<td>1303</td>
<td>601</td>
<td>523</td>
<td>265</td>
<td>230</td>
<td>219</td>
<td>217</td>
<td>175</td>
</tr>
<tr>
<td>0.925</td>
<td>127</td>
<td>120</td>
<td>108</td>
<td>62</td>
<td>57</td>
<td>55</td>
<td>45</td>
<td>38</td>
</tr>
<tr>
<td>0.950</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.975</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Graph Characteristics: Trivial Stars

<table>
<thead>
<tr>
<th>Prob Threshold</th>
<th>nComp</th>
<th>nTrivial stars</th>
<th>Size 2</th>
<th>Size 3</th>
<th>Size 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.700</td>
<td>21</td>
<td>14</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.725</td>
<td>36</td>
<td>25</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.750</td>
<td>45</td>
<td>26</td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.775</td>
<td>67</td>
<td>47</td>
<td>47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.800</td>
<td>114</td>
<td>66</td>
<td>66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.825</td>
<td>184</td>
<td>99</td>
<td>97</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0.850</td>
<td>258</td>
<td>126</td>
<td>123</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.875</td>
<td>342</td>
<td>154</td>
<td>153</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.900</td>
<td>403</td>
<td>162</td>
<td>158</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0.925</td>
<td>194</td>
<td>97</td>
<td>94</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0.950</td>
<td>11</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.975</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Graph Characteristics: Node Degrees

Vertex InDegree Frequency Distribution

Number of vertices vs. ln(InDegree) with different probability thresholds.
Graph Characteristics: Node Degrees

Vertex OutDegree Frequency Distribution

Number of vertices vs. ln(OutDegree) for different probability values.

- Prob = 0.760
- Prob = 0.770
- Prob = 0.780
- Prob = 0.790
- Prob = 0.800
- Prob = 0.810
- Prob = 0.820
- Prob = 0.830
- Prob = 0.840
- Prob = 0.850
- Prob = 0.860
- Prob = 0.870
- Prob = 0.880

Luk Knapen (IMOB)
Graph Characteristics: Connected Components

Number of Connected Components

Number of Components

Threshold probability

0

Graph Characteristics: Connected Components

Number of Connected Components

Number of Components

Threshold probability

0
Graph Characteristics: Connected Components
Conclusions

- Problem size causes non-trivial issues
- Partitioning is easy
- From threshold 0.85 on, *grid computing* is useful
- Allocate tasks to processors in decreasing size order (load balancing)