Using Local Search for Efficient Subgraph Isomorphism

Aaron Cederquist

Weighted Graph Data Mining

- Directed, labeled, weighted graphs
- Find Frequent patterns in graph database
- Applications
- Complexity Challenge

Graph in database

Candidate pattern
**Weighted Subgraph Isomorphism**

- Central problem
  - limits candidate graph size
  - beyond NP-Complete

- Constraint Satisfaction Problem construct
- Local Search techniques used in heuristics

---

**Experiment setup**

- **Seeded Pattern**
  - 3 node labels, 2 edge labels (18 different types of labeled edges)
  - 20 nodes, 328 edges
  - Embedded 200 times in 2,000 node graph with 2.8m edges

- 4 local-search heuristic algorithms used
Results

- 2nd best only 7.36% of value of optimal
- Exhaustive search: 2.2m iterations, 809m checks, 0.3μs/check (235 seconds)
- Results for roulette heuristic:

<table>
<thead>
<tr>
<th># samples</th>
<th>Max iterations per sample</th>
<th>% correct/time (1 run)</th>
<th>Time to reach 90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200k</td>
<td>13% / 8.7s</td>
<td>149s</td>
</tr>
<tr>
<td>10</td>
<td>20k</td>
<td>46% / 35s</td>
<td>141s</td>
</tr>
<tr>
<td>100</td>
<td>2k</td>
<td>56% / 37s</td>
<td>110s</td>
</tr>
<tr>
<td>1000</td>
<td>200</td>
<td>28% / 45s</td>
<td>360s</td>
</tr>
<tr>
<td>Exhaustive</td>
<td>2.2m</td>
<td>100% / 235s</td>
<td>235s</td>
</tr>
</tbody>
</table>

Conclusion

- Finding optimal more important than preventing a false negative
- Balance between depth of sample and number of samples
- Weighted random heuristic worked best