Finding Maximal Cliques Using MATLAB

Dr Richard Mitchell
Cybernetics Intelligence Research Group
Department of Cybernetics
The University of Reading, UK
R.J.Mitchell@reading.ac.uk

Work done during Sabbatical at National Grid
Abstract

- Maximal Clique Problem
  - well known NP complete problem
- Early good algorithm: Bron Kerbosch
  - Backtracking + Sets to stop repeats
  - Written in obscure Algol 60
  - Uses nodes with most connections
- What if use nodes with least connections
- Both implemented in MATLAB
- Show hybrid of two may be best
Maximal Clique Problem

Have nodes, some are connected

Fully connected nodes are cliques

Can group 1,2,7; 1,2,3; 1,6,7 and 2,5

Seems easy, but is NP complete problem
Applications of Cliques

- In Mean Tracking Cluster Algorithm
  - Windows move through data space
  - Those which overlap are merged
  - Find all pairs to merge
  - May be mutual pairs to merge
- Merging in async seq logic states
  - Combine all states mutually mergable
- Graph theory
  - Graph has nodes connected by edges
  - Find all fully connected sets of nodes
Bron Kerbosch

- Backtracking plus Candidates & Nots Sets and Compsub – has nodes in next clique
- Extend(Nots, Candidates, Compsub)
  - Find node with most connections
  - for number of non connections
    Select node
    Nots = Nots connected to node
    Candidates = Candidates connected to node
    If any Candidates
      Extend (Nots, Candidates, Compsub+node)
    Else Compsub is next clique
    Move Node from Candidates to Nots
New Algorithm

- Backtrack but select node with fewest connections
- Process node then remove
- Why? NP gets worse with N
  - So work on small problems
  - Main problem becomes smaller
- Works ok to certain extent
- To stop repeat searches, have ‘notts’
- But this is list of nots rather than set
On Implementation

- Standard backtrack on 785 pairs
- Took 26,320 seconds in MATLAB
- RJM algorithm first attempt: 2.62 secs!
- Bron Kerbosch MATLAB similar
- Careful coding in MATLAB
  - use built in MATRIX functions
  - (less code to interpret)
  - Work on columns not rows
  - Algorithm took 0.2 secs
Testing

- ought to do algorithm O() analysis
- Easier, get MATLAB to do run and time it
- Test on graphs with
  - 10, 15, 20, ... 50 nodes
  - Each with 10%, 30%, 50%, 75%, 90% or 95% interconnected

10 nodes @30%: 14 edges, 11 cliques
40 nodes@75%: 546 edges, 1816 cliques
50 nodes@90%: 1103 edges, 119778 c’s
# Times for Bron Kerbosch

<table>
<thead>
<tr>
<th></th>
<th>10%</th>
<th>30%</th>
<th>50%</th>
<th>75%</th>
<th>90%</th>
<th>95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0</td>
<td>0.0020</td>
<td>0.0060</td>
<td>0.0040</td>
<td>0.0040</td>
<td>0.0040</td>
</tr>
<tr>
<td>15</td>
<td>0.0020</td>
<td>0.0060</td>
<td>0.0100</td>
<td>0.0140</td>
<td>0.0040</td>
<td>0.0080</td>
</tr>
<tr>
<td>20</td>
<td>0.0060</td>
<td>0.0100</td>
<td>0.0240</td>
<td>0.0300</td>
<td>0.0320</td>
<td>0.0160</td>
</tr>
<tr>
<td>25</td>
<td>0.0060</td>
<td>0.0240</td>
<td>0.0380</td>
<td>0.1040</td>
<td>0.1200</td>
<td>0.0520</td>
</tr>
<tr>
<td>30</td>
<td>0.0100</td>
<td>0.0340</td>
<td>0.0740</td>
<td>0.2060</td>
<td>0.5270</td>
<td>0.3140</td>
</tr>
<tr>
<td>35</td>
<td>0.0180</td>
<td>0.0440</td>
<td>0.1240</td>
<td>0.4050</td>
<td>0.9790</td>
<td>0.5330</td>
</tr>
<tr>
<td>40</td>
<td>0.0220</td>
<td>0.0620</td>
<td>0.1940</td>
<td>0.7990</td>
<td>3.4930</td>
<td>3.1450</td>
</tr>
<tr>
<td>45</td>
<td>0.0280</td>
<td>0.0920</td>
<td>0.3020</td>
<td>1.5100</td>
<td>12.354</td>
<td>7.1180</td>
</tr>
<tr>
<td>50</td>
<td>0.0380</td>
<td>0.1100</td>
<td>0.4150</td>
<td>2.1290</td>
<td>55.450</td>
<td>30.820</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>30%</td>
<td>50%</td>
<td>75%</td>
<td>90%</td>
<td>95%</td>
</tr>
<tr>
<td>---</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0.0040</td>
<td>0.0040</td>
<td>0.0040</td>
<td>0.0020</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0.0020</td>
<td>0.0080</td>
<td>0.0120</td>
<td>0.0160</td>
<td>0.0040</td>
<td>0.0060</td>
</tr>
<tr>
<td>20</td>
<td>0.0040</td>
<td>0.0180</td>
<td>0.0400</td>
<td>0.0440</td>
<td>0.0360</td>
<td>0.0100</td>
</tr>
<tr>
<td>25</td>
<td>0.0080</td>
<td>0.0360</td>
<td>0.0660</td>
<td>0.1940</td>
<td>0.1660</td>
<td>0.0500</td>
</tr>
<tr>
<td>30</td>
<td>0.0180</td>
<td>0.0600</td>
<td>0.1380</td>
<td>0.3560</td>
<td>0.9810</td>
<td>0.4290</td>
</tr>
<tr>
<td>35</td>
<td>0.0260</td>
<td>0.0900</td>
<td>0.2400</td>
<td>0.7670</td>
<td>2.0070</td>
<td>0.9230</td>
</tr>
<tr>
<td>40</td>
<td>0.0420</td>
<td>0.1260</td>
<td>0.3950</td>
<td>1.6760</td>
<td>7.8010</td>
<td>5.7140</td>
</tr>
<tr>
<td>45</td>
<td>0.0660</td>
<td>0.1820</td>
<td>0.5890</td>
<td>3.2450</td>
<td>30.1410</td>
<td>15.0640</td>
</tr>
<tr>
<td>50</td>
<td>0.0840</td>
<td>0.2340</td>
<td>0.9010</td>
<td>4.5290</td>
<td>157.807</td>
<td>88.5450</td>
</tr>
</tbody>
</table>
Conclusion and Further Work

- Usually Bron Kerbosch better
- For highly connected, RJM can be better
- RJM spends time searching ‘not list’
- Hybrid algorithm perhaps worthwhile

- Consider better ‘not list’
- Do formal algorithm order analysis.