Vertex Cover of Bipartite Graph

Marco Gallotta

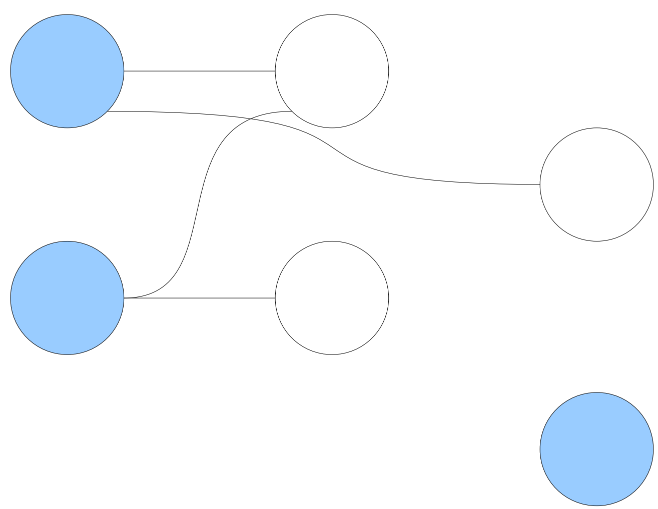
SACO Training Camp 2
2-4 May 2008

Problem: Declone

- You have a 2D grid with clones at lattice points
- Can destroy all clones in a row or column in a single shot
- Want to determine the minimum number of shots required to destroy all clones

Vertex Cover

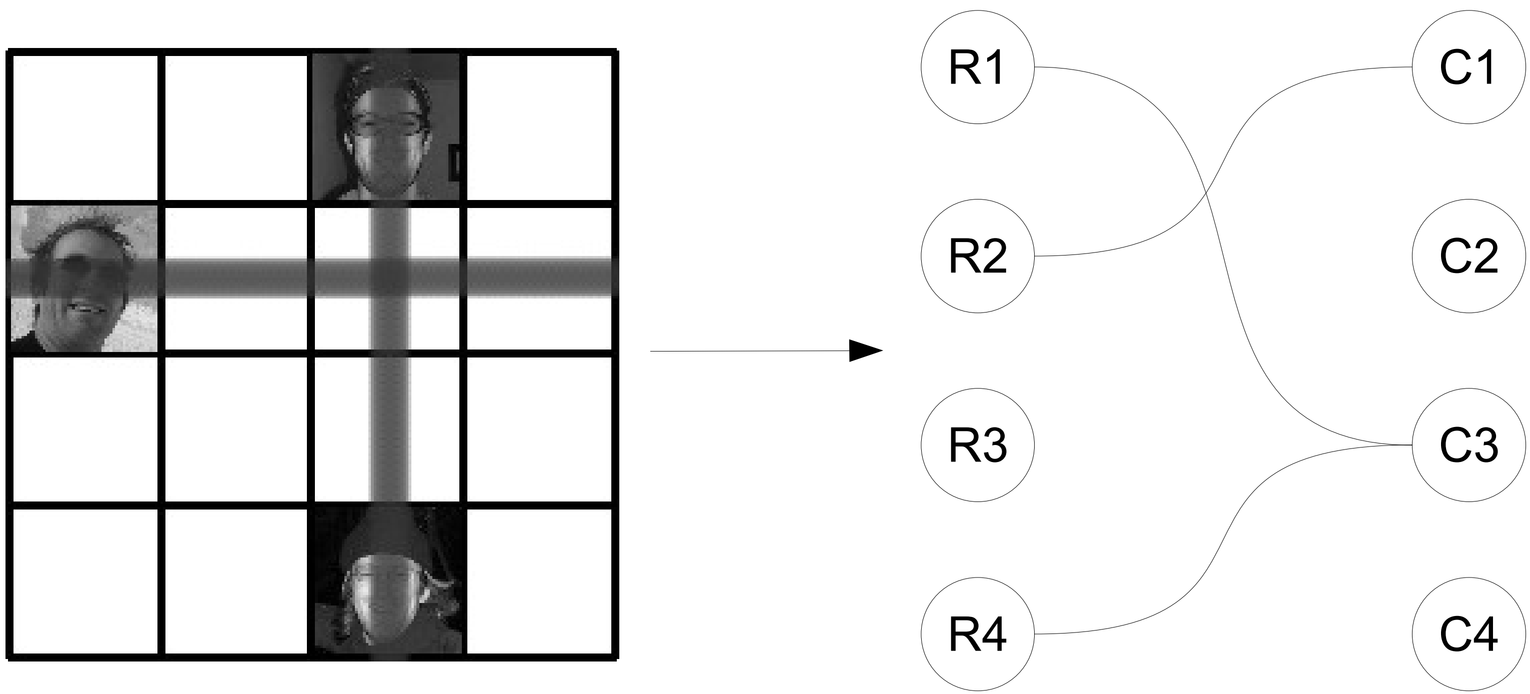
- Vertex Cover: Subset of vertices *S* such that each edge has at least one endpoint in *S*
- Vertex Cover Problem: Minimise the vertex cover



- Vertex Cover

Relationship to Declone

- Declone is equivalent to the vertex cover problem

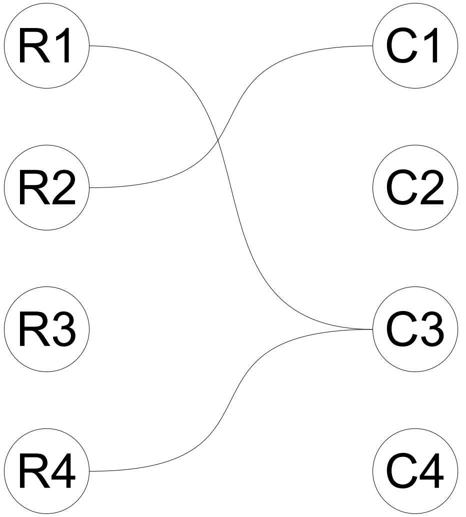


Problem!

- The minimum vertex cover is NP-complete! :(
- But N goes all the way up to 250,000?!

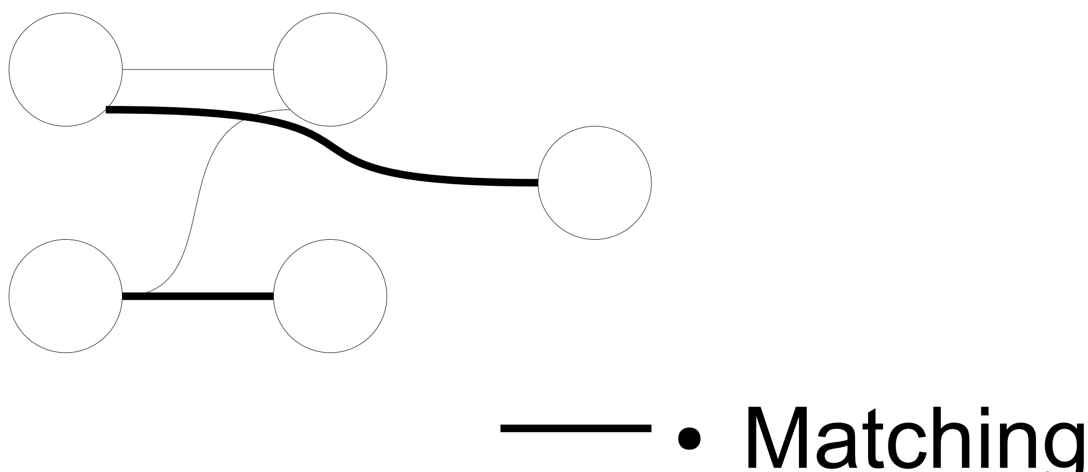
Observation

- The graph is bipartite – remember what that means?
- König's theorem: In a bipartite graph, the number of edges in maximum matching is equal to the number of vertices in a minimum vertex cover



Maximum Matching

- A matching is a set of edges *S* with no two edges in *S* sharing a common vertex
- A maximum matching maximises the size of *S*



Roundup

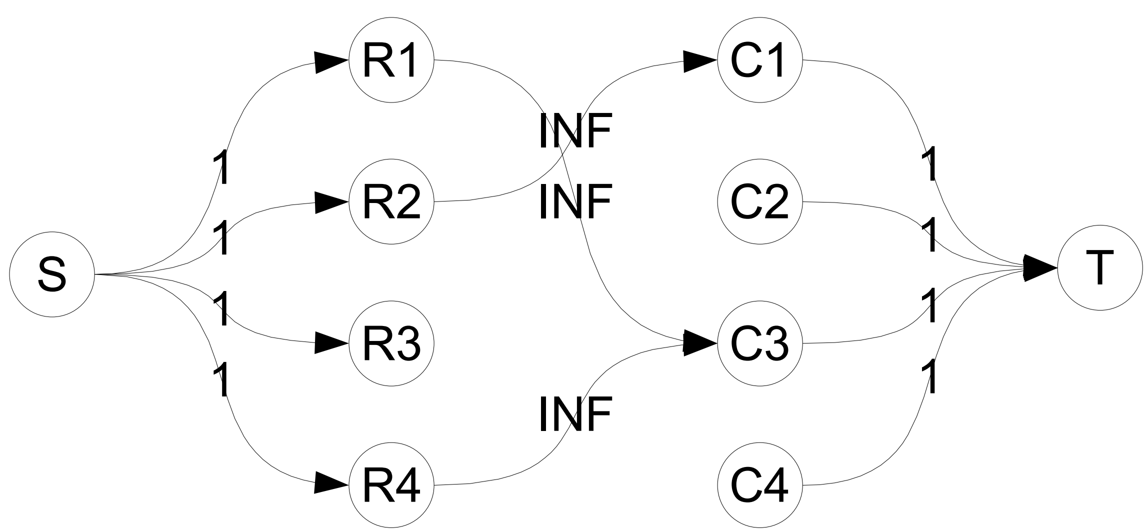
- Our graph is bipartite
- Therefore (only in bipartite graphs) our vertex cover problem is equivalent to finding a maximum matching
- So how do we find a maximum bipartite matching?

Maximum Bipartite Matching

Network flow!

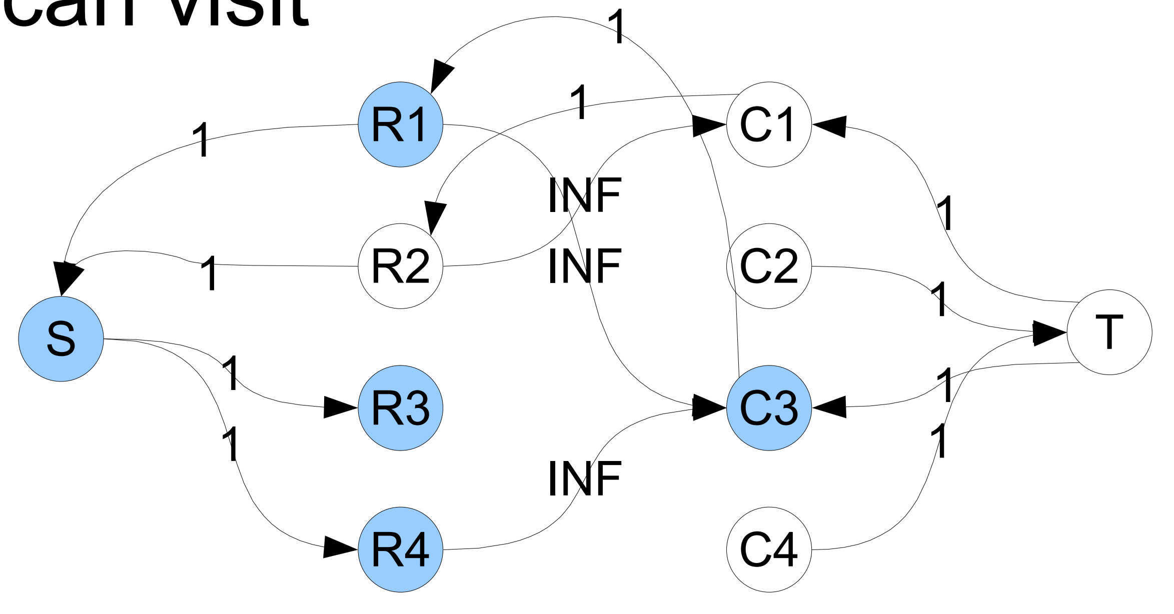
Huh, How?

- Add a super source *S* that has an edge of weight 1 to all row vertices
- Add a super sink *T* that has an edge of weight 1 to all column vertices
- Set the weights of all other edges to infinity

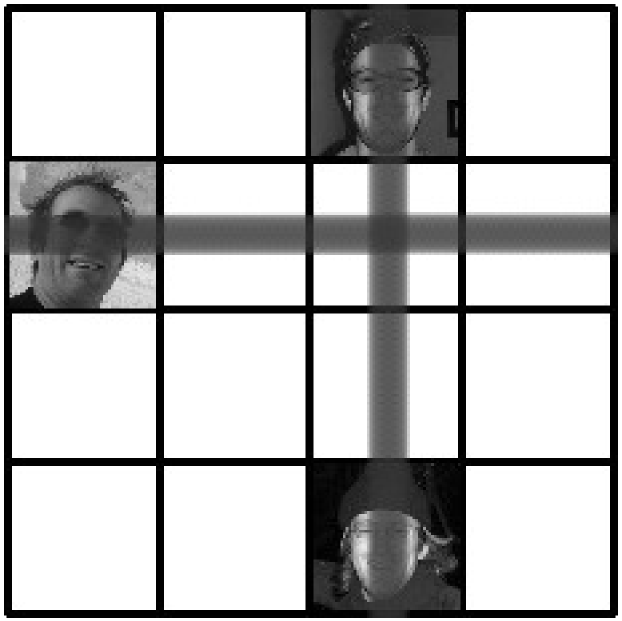


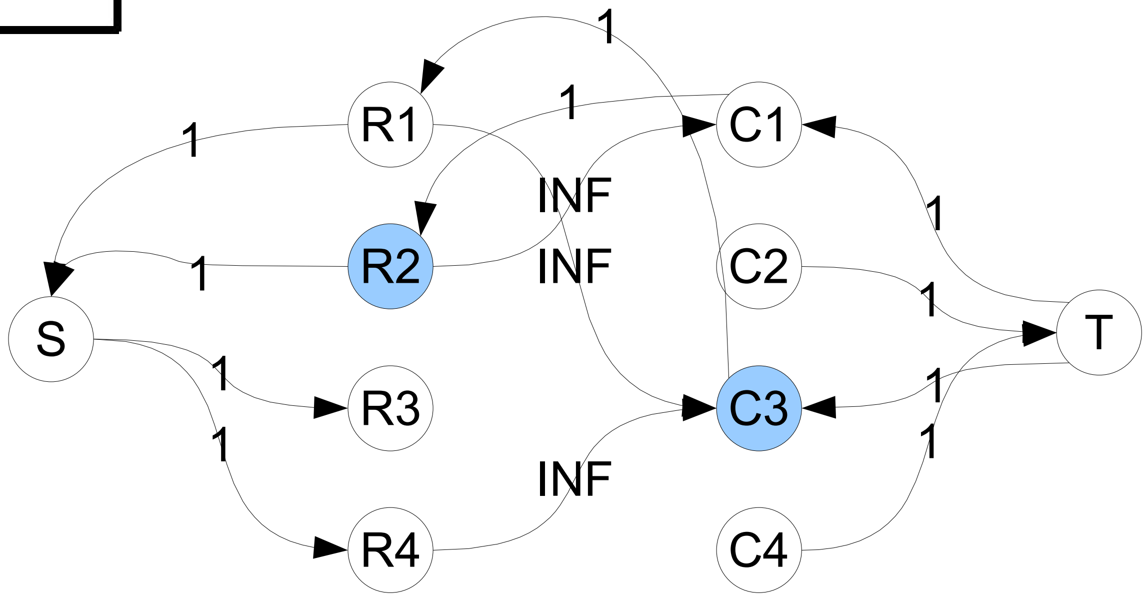
Final Leg

- Perform Ford-Fulkerson on the above graph
- DFS the residual graph and mark off those you visit (blue in graph below)
- The answer: all rows you cannot visit and all columns you can visit

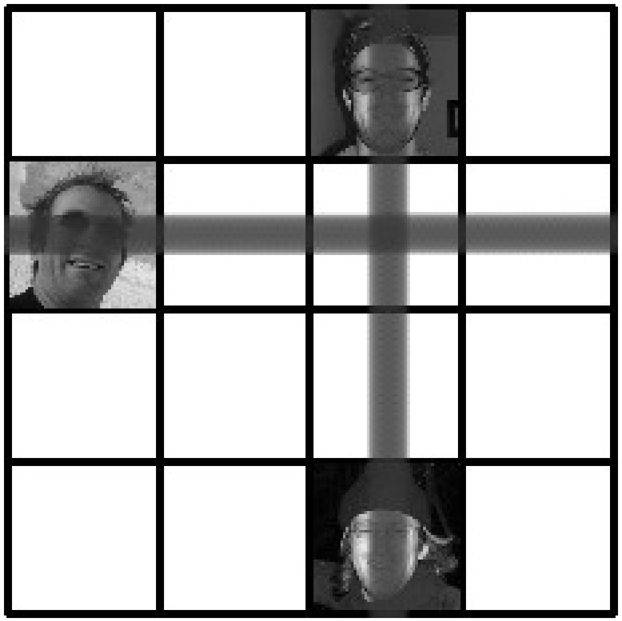


TADA!





TADA!



- Why does it work?