COSC 341
Theory of Computing
Lecture 23
All about matching

Stephen Cranefield
stephen.cranefield@otago.ac.nz

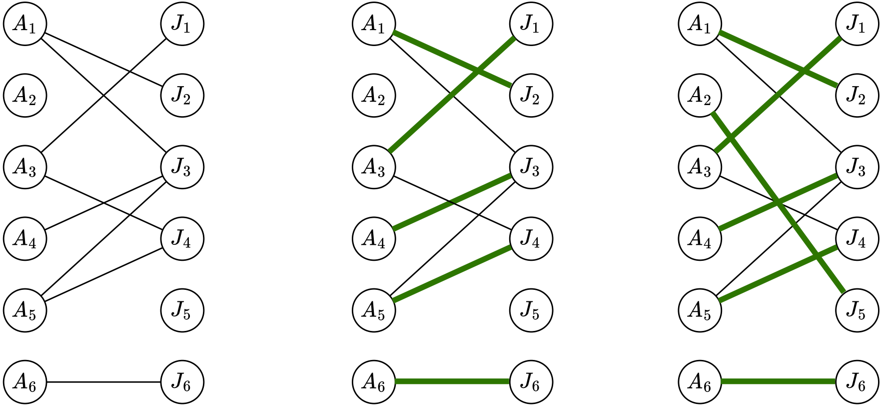
Lecture slides (mostly) by Michael Albert

Keywords: Matchings

Matchings in graphs

- A bipartite graph is a graph, G, whose vertices can be partitioned into two sets A and B such that every edge has one endpoint in A and the other in B.
- A matching, M, in a bipartite graph is a set of edges, no two of which share a common endpoint.
- A maximum matching in G is a matching whose size is as large as it possibly can be.
- A perfect matching is a matching in which every vertex of G is the endpoint of some edge in the matching.
- A matching saturates A if every vertex in A is the endpoint of an edge in the matching.

Example matchings



Applicants **Jobs**

A maximum matching

An added edge allows a perfect matching

Example based on https://www.geeksforgeeks.org/maximum-bipartite-matching/

Questions arising

Let G be a bipartite graph with parts A and B:

- Does G have a perfect (saturating) matching?
- Find a perfect (saturating) matching for G.
- Does G have a matching containing at least k edges? (k a parameter)
- Find a matching in G having at least k edges.
- What is the size of a maximum matching for G?
- Find a maximum matching for G.

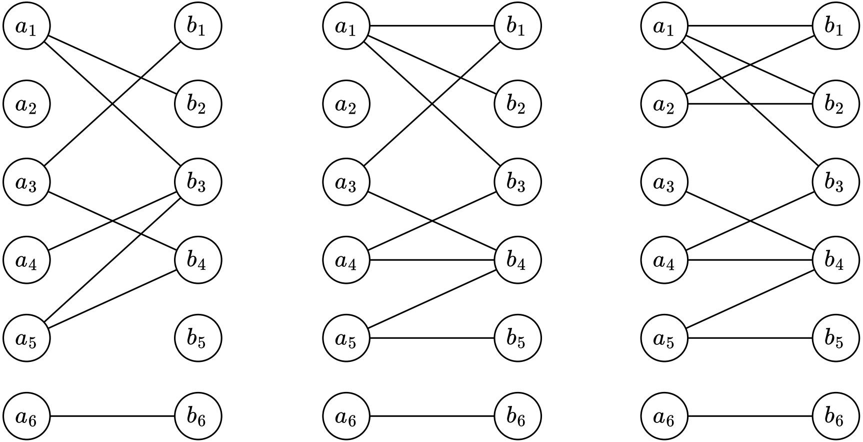
Hall's theorem

In a graph G, for a set X of vertices, let $N(X)$ denote those vertices that are neighbours of some vertex in X.

Theorem

A bipartite graph G with parts A and B has a matching that saturates A if and only if $|N(X)| \geqslant |X|$ for every $X \subseteq A$.

Applying Hall's theorem



What does Hall's Theorem say about the possibility of finding maximum matchings for the graphs above?

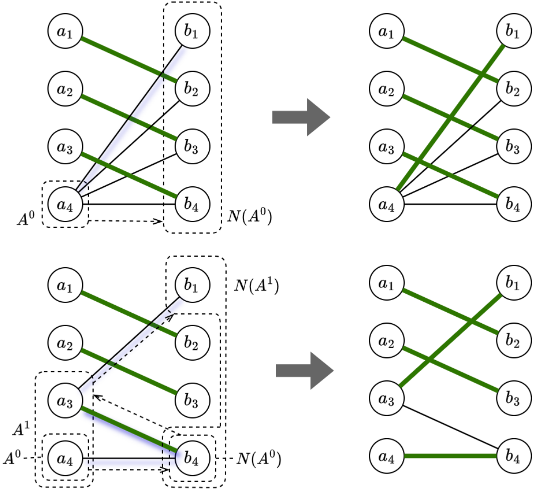
Proof of Hall's theorem

- Direction $\Rightarrow$: Suppose we have a saturating matching. Then each vertex $a \in A$ has a different distinct neighbour in B via the matching, and possibly others, so $|N(X)| \geqslant |X|$.
- Direction $\Leftarrow$: Let M be a maximum matching (there must be one). We need to use Hall's condition to show that M saturates A. Suppose M does not saturate A. Then there must be some unmatched $a \in A$. Denote the set $\{a\}$ by A_0. Consider the set of its neighbours, $N(A^0)$.
  - If any member b of $N(A^0)$ is unmatched, we have a single-edge "augmenting path" and can add (a, b) to the matching, contradicting the fact that M is a maxiumum matching.
  - Therefore all members of N must be matched. Consider the set A^1 containing a and all the vertices in A that are matched by M to some vertex in $N(A^0)$. This has size $|N(A^0)| + 1$. Hall's condition says that the set of neighbours of A^1 must have at least that many elements. Therefore, $N(A^1)$ contains at least one element b of B that is not in $N(A^0)$.
    - If b is unmatched, we must have a multi-edge "augmenting path" from a to b via a vertex in $N(A^0)$ and then one in A^1. This has an edge not in M, then one in M and then one not in M. We can add the first and third to M and remove the second from M, resulting in a larger matching than our supposedly maximum one: contradiction!
    - If b is not matched, repeat the steps above to form A^2, $N(A^2)$, A^3, ... until we find an unmatched node in B and therefore an augmenting path from a, giving us a way to increase the size of M, which is a contradiction.

Illustrations of the proof of Hall's theorem

Augmenting path of length 1 (see shadowed edge)

Augmenting path of length 3 (see shadowed edges)



Above: different bipartite graphs and the application of Hall's theorem (top: in one step; bottom: in two steps). Augmenting paths are highlighted with blue shadows, and the matchings are updated by swapping the status (in or out of the matching) of the edges in those paths.

Ideas arising from the proof

An augmenting path for a matching M is a path P that starts and finishes at an unmatched vertex and whose edges alternate between not belonging to M and belonging to M.

The edges traversed in P alternate between edges that are in M (every odd edge in the traversal sequence) and those that are in M (every even edge).

We can use the edges in an augmenting path to create a larger matching. Simply remove from M any edge in P that is in M and add any edge in P that is not in P. There is one more edge of the latter type in P, so the matching increases in size by 1.

Lemma

If a matching M is not maximum then it has an augmenting path P.
In fact, if a maximum matching has t more edges than M then there are t vertex-disjoint augmenting paths for M.

The Hopcroft-Karp algorithm

An algorithm that computes a maximum matching for a bipartite graph.
See explanation on YouTube.

Require: A bipartite graph G with parts A and B
Ensure: A maximum matching M of G

$M \leftarrow \{\}$
repeat
- From the free vertices in A do a breadth-first search alternating edges out of and in the matching until you reach a free vertex in B or none are found.
if one or more free vertices in B are reached **then**
- Construct by depth-first search a maximal set of augmenting paths for M
- Update M by switching the augmenting paths
end if
until no augmenting path is found
return M

The Hopcroft-Karp algorithm explained

- In the first round we just construct (greedily) some maximal matching.
- Thereafter, we find a maximal collection of disjoint shortest augmenting paths and switch to extend the matching.
- “Obviously” (see Notes 20) the shortest augmenting paths get longer in each round.
- After $\sqrt{|V|}$ rounds they'll have length at least $\sqrt{|V|}$.
- The matching we have at that point can differ from the maximum matching by at most $\sqrt{|V|}$ edges since that's the largest number of disjoint augmenting paths we could possibly have.
- So $O(\sqrt{|V|})$ iterations through the loop will occur and (with care) the total cost is $O(|E|\sqrt{|V|})$, so finding a maximum matching in a bipartite graph is easilly polynomial in time.
- What happens if we generalise the problem from edges of size 2 to edges of size three?

Hypergraphs

- A hypergraph is a set of vertices and *hyperedges* where a hyperedge is just a subset of the vertices.
- If each hyperedge contains k elements then the hypergraph is called k-uniform.
- A 2-uniform hypergraph is just a graph.
- A *matching* in a hypergraph is a disjoint set of hyperedges. It is *perfect* if its union is the set of all vertices.

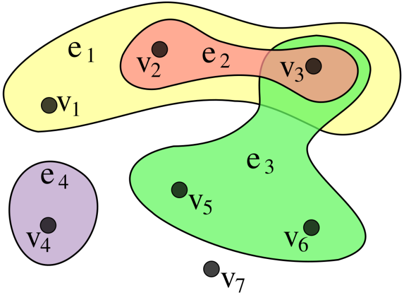
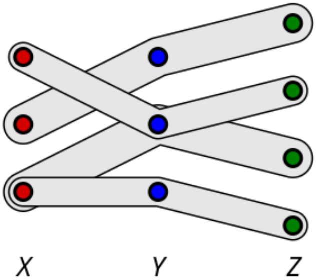


Image credit: Hypergraph.svg: Kilom691derivative work: Pgdx, , via Wikimedia Commons

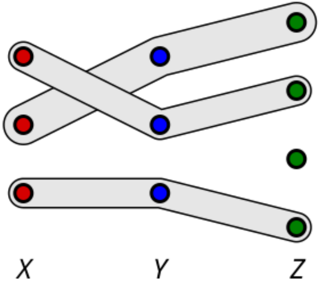
The 3D-matching problem

3D-MATCHING

Does a 3-uniform hypergraph have a perfect matching?



A 3-uniform hypergraph



A matching that is not perfect

This problem is NP-complete

See reduction from 3-SAT in Notes 20 and slides by Carl Kingsford, UMCP, which show a clause widget