Maximum (cardinality) matchings in bipartite graphs

So far we have seen simple matching algorithms. In particular a simple algorithm for finding a maximum matching in a tree.

Trees are a particular type of bipartite graphs. So the natural question is: what is the complexity of the maximum matching problem in bipartite graphs?

Historically some of the most important theorems for bipartite graphs were proved directly and only later were these bipartite results obtained as corollaries of more general non-bipartite theorems.

Moreover bipartite graphs deserve special treatment because the majority of real world applications of matching theory deal with bipartite graph models.

Big, Big Problem

It is very, VERY hard to prove that there is an algorithm that solves this problem.

In a couple of lectures we will discuss an algorithm that does it. Proving that it works ... is VERY far from trivial.

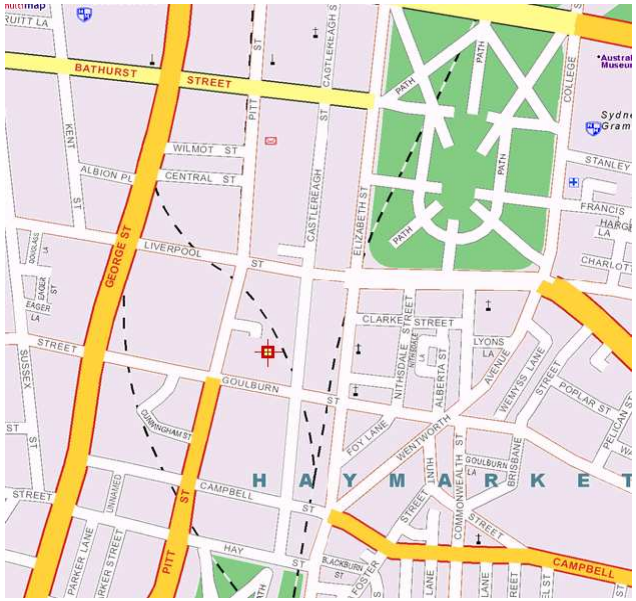
Plan

- Try to get some information on the size of the largest matchings in bipartite graphs mathematically, by looking at other graph properties
- Then look at algorithms and use our knowledge of the properties mentioned above to prove that our algorithms work.

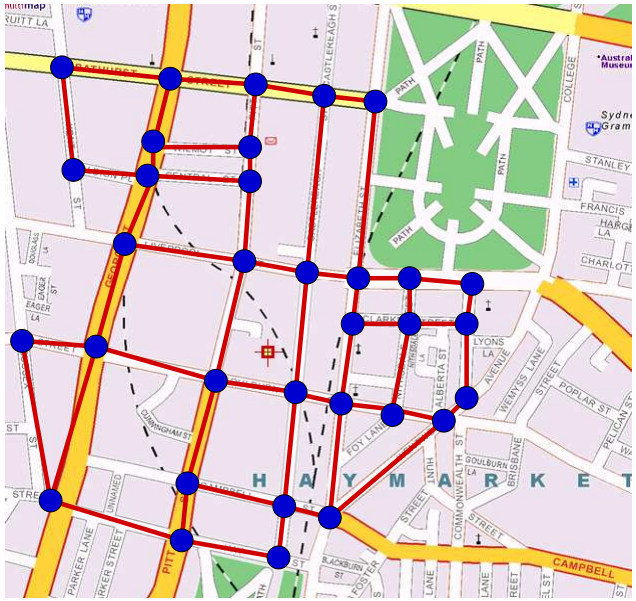
We start from results that hold for ANY graph.

Let’s throw few more structures in the big pot

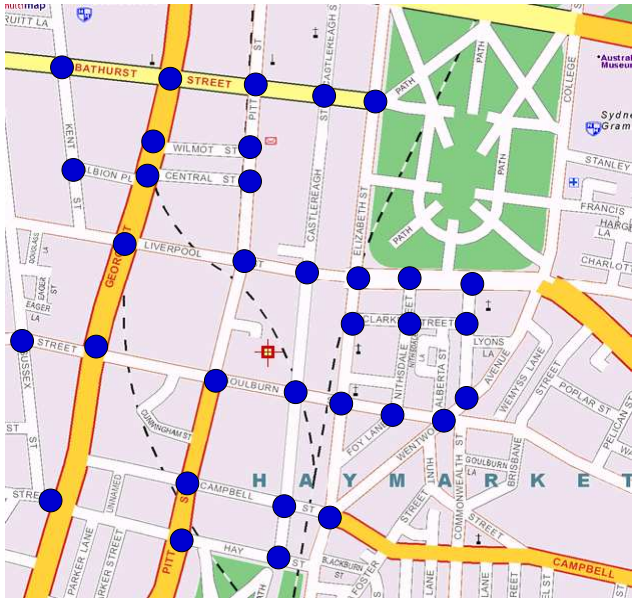
1. Vertex covers: sets of vertices meeting all edges.
2. Independent sets: sets of vertices meeting NO edge.
3. Edge covers: sets of EDGES hitting all vertices.


BATHURST STREET
GEORGE ST
PITT ST
CAMPBELL
H A Y M A R K E T

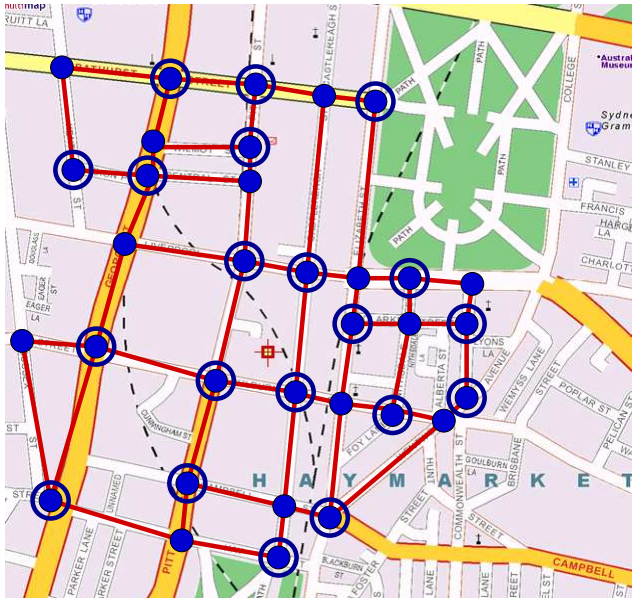
4-1


BATHURST STREET
GEORGE ST
PITT ST
CAMPBELL
H A Y M A R K E T

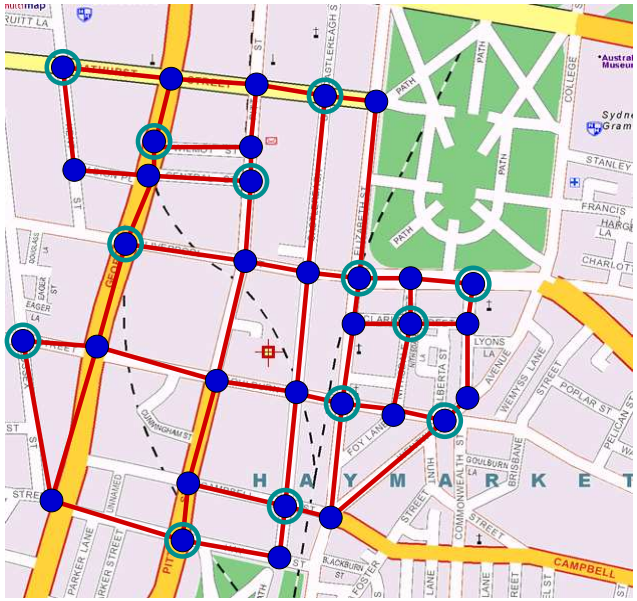
4-3


BATHURST STREET
GEORGE ST
PITT ST
CAMPBELL
H A Y M A R K E T

4-2


BATHURST STREET
GEORGE ST
PITT ST
CAMPBELL
H A Y M A R K E T

4-4



4-5

Useful functions

Let G be any graph. Then

$\nu(G)$ denote the matching number of G (i.e. the cardinality of the largest matchings in G)

$\tau(G)$ the vertex covering number (i.e. the cardinality of the smallest sets of vertices incident to all edges in G)

$\alpha(G)$ the independence number (i.e. the cardinality of the largest sets U of vertices such that $G[U]$ is empty) and

$\rho(G)$ the edge covering number (i.e. the smallest number of edges that are incident to any vertex in G).

First relationship

Let $G = (V, E)$ be a graph. If U is a vertex cover in V then $V \setminus U$ is an independent set.

Obvious: if U meets all edges of G, then any pair of vertices that is NOT in U can meet none!

Consequence: If U is a smallest vertex cover, then $V \setminus U$ is a largest independent set. Hence $\tau(G) = |V| - \alpha(G)$, for any graph G.

Algorithmic consequence: if we have a program that computes in polynomial time a smallest vertex cover, the same program could be easily hacked to return, essentially within the same time complexity, a largest independent set as well.

Second relationship

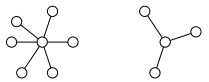
Things are not as simple for matchings. Let's try anyway.

Matchings are related to edge covers:

$\nu(G) = |V| - \rho(G)$, for any graph G with no isolated vertex.

Let C be a collection of edges forming a minimum edge cover.

$G[V(C)]$, the graph *induced* by the set $V(C)$, consists of a number of "stars" (little graphs like the ones drawn on the right) because any other "larger" graph would not be minimal.



Let n be the total number of vertices $G[V(C)]$. If κ is the number of connected components of the graph, n_i is the number of vertices in the ith component (for $i \in \{1, \ldots, \kappa\}$), and m_i the number of edges in the ith component, then

$$n = \sum_{i=1}^{\kappa} n_i.$$

but also

$$n = \sum_{i=1}^{\kappa} (m_i + 1) = \sum_{i=1}^{\kappa} m_i + \kappa = \rho + \kappa.$$

Therefore κ, the number of connected components in $G[V(C)]$ is $n - \rho(G)$

Picking one edge from each of them we get a matching M of size $n - \rho(G)$. Hence $\nu(G) \geq n - \rho(G)$.

Conversely let M be a maximum matching in G.

For every vertex v NOT incident to an edge of M there must be an edge connecting v with some edge in M.

We can thus define a set of lines covering every vertex of G containing $\nu(G) + (n - 2\nu(G))$ lines. Hence $\rho(G) \leq n - \nu(G)$.

Algorithmic consequences

What if we tried to find large matchings by first finding small edge covers?

1. A minimal edge cover is minimum if and only if it contains a maximum matching.
2. A maximal matching is maximum if and only if it is contained in a minimum line cover.

However an efficient algorithm for finding a minimum edge cover would NOT give us a very efficient way of finding a maximum matching.

König's equality

Let's now turn to bipartite graphs.

If $G = (V, E)$ is a bipartite graph, then $\nu(G) = \tau(G)$.

Important result as it will be key to the proof that the eventual matching algorithm works.

König's equality is saying that the smallest vertex covers and the largest matchings in a bipartite graph have the same number of elements.

What does it mean? Why does it work?

Let U be one of a vertex cover in G and let M be a matching in G. By definition all edges in the graph must be incident to some vertex in U. In particular if $e = \{v, w\}$ is and edge in M, then either v or w must be in U (otherwise e would not be covered). Similar argument applies to each edge in M. Hence U must contain at least one vertex from each edge in M. Hence U must contain at least as many vertices as there are edges in M. This, in symbols, is written $|U| \geq |M|$.

Now choose U to be a vertex cover of size $\tau(G)$ (a minimum vertex cover) and M to be a matching of size $\nu(G)$ (a maximum matching, that is). The argument above implies $\tau(G) \geq \nu(G)$.

In other words,

starting from a matching M, one can define a vertex cover with no less than $|M|$ vertices.

What does it mean? Why does it work?

To complete the proof of König's equality we need to verify that $\nu(G) \geq \tau(G)$.

This time we will prove that

> starting from a cover U, one can define a matching with no less than $|U|$ edges.

Let U be a minimum vertex cover in G. Each vertex in U will cover a number of edges of G. Let's peel off such edges until we get to a subgraph G' of G that

(C1) U is still a smallest vertex cover of G' (in symbols $\tau(G') = \tau(G)$); and

(C2) G' is the smallest such graph (hence the removal of a single edge from G' leaves a graph that has a vertex cover W that has one less vertex than U).

G' is a matching in G (!!)

(Notice that this implies $\tau(G) = \tau(G') = |E(G')| \leq \nu(G)$, which is what we are after).

Proof of the main claim

Assume that G' is not a matching: there are two edges x and y incident to a vertex a in G'. We will prove that already G' has a vertex cover smaller than U ... which is impossible because of (C1) and (C2).

1. Consider $G' - x$ and $G' - y$. There is a cover S_x in $G' - x$ with $|S_x| = \tau(G') - 1$ and neither end-points of x belong to S_x. Similarly there is a cover S_y in $G' - y$ with $|S_y| = \tau(G') - 1$ and neither end-point of y are in S_y. Hence $|S_x| = |S_y|$.
2. Take the subgraph G'' of G' induced by a (the common endpoint of x and y) and $S_x \oplus S_y = (S_x \setminus S_y) \cup (S_y \setminus S_x)$.
3. Let $t = |S_x \cap S_y|$. Then, trivially, $|V(G'')| = 2(\tau(G') - 1 - t) + 1$.

4. Furthermore, since G'' is bipartite, there is a set T (the smaller of the two colour classes of G'') which covers G'' and has size at most $\frac{|V(G'')|}{2} - 1 = \tau(G') - 1 - t$.
5. But $T \cup (S_x \cap S_y)$ covers G' ... and $|T \cup (S_x \cap S_y)| \leq \tau(G') - 1 - t + t = \tau(G') - 1$. Contradiction.