Lecture notes on non-bipartite matching

Given a graph $G = (V, E)$, we are interested in finding and charaterizing the size of a maximum matching. Since we do not assume that the graph is bipartite, we know that the maximum size of a matching does not necessarily equal the minimum size of a vertex cover, as it is the case for bipartite graphs (König's theorem). Indeed, for a triangle, any matching consists of at most one edge, while we need two vertices to cover all edges.

To get an upper bound on the size of any matching M, consider any set U of vertices. If we delete the vertices in U (and all edges adjacent to it), we delete at most $|U|$ edges of the matching M. Moreover, in the remaining graph $G \setminus U$, we can trivially upper bound the size of the remaining matching by $\sum_{i=1}^{k} \lfloor \frac{|K_i|}{2} \rfloor$, where K_i, $i = 1, \cdots, k$, are the vertex sets of the connected components of $G \setminus U$. Therefore, we get that

$$|M| \leq |U| + \sum_{i=1}^{k} \left\lfloor \frac{|K_i|}{2} \right\rfloor. \tag{1}$$

If we let $c_0(G \setminus U)$ denote the number of odd components of $G \setminus U$, we can rewrite (1) as:

$$|M| \leq |U| + \frac{|V| - |U|}{2} - \frac{c_0(G \setminus U)}{2},$$

or

$$|M| \leq \frac{1}{2} \left(|V| + |U| - c_0(G \setminus U) \right). \tag{2}$$

We will show that we can always find a matching M and a set U for which we have equality; this gives us the following minmax relation, called the Tutte-Berge min-max formula:

Theorem 1 *For any graph $G = (V, E)$, we have*

$$\max_{M} |M| = \min_{U \subseteq V} \frac{1}{2} \left(|V| + |U| - c_0(G \setminus U) \right),$$

where $c_0(G \setminus U)$ is the number of connected components of odd size of $G \setminus U$.

To prove this theorem, we will first show an algorithm to find a maximum matching. This algorithm is due to Edmonds [1965], and is a pure gem. As in the case of bipartite matchings (see lecture notes on bipartite matchings), we will be using augmenting paths. Indeed, Theorem 2 of the bipartite matching notes still hold in the non-bipartite setting; a matching M is maximum if and only if there is no augmenting path with respect to it. The difficulty here is to find the augmenting path or decide that no such path exists. We could try to start from the set X of exposed (unmatched) vertices for M, and whenever we are at a vertex u and see an edge $(u, v) \notin M$ followed by an edge (v, w) in M, we could put a directed edge from u to w and move to w. If we get to a vertex that's adjacent to an exposed vertex (i.e. in X), it seems we have found an augmenting path. This is not necessarily the case, as we may have a found a *flower*, see Figure 1. This flower does not contain an augmenting path. More formally, a *flower* consists of an even alternating path P from an exposed vertex u to a vertex v, called the *stem*, and an odd cycle containing v in which the edges alternate between in and out of the matching except for the two edges incident to v; this odd cycle is called a *blossom*.

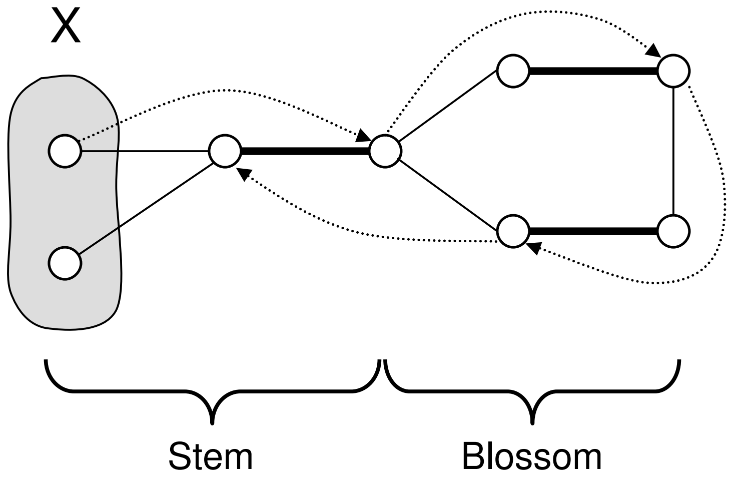

Figure 1: A flower. The thick edges are those of the matching.

The algorithm will either find an augmenting path or a flower or show that no such items exist; in this latter case, the matching is maximum and the algorithm stops. If it finds an augmenting path then the matching is augmented and the algorithm continues with this new matching. If a flower is found, we will first take the stem P and replace M by $M \triangle P = (M \setminus P) \cup (P \setminus M)$ to get a new matching of the same size as M for which we have a flower with an empty stem. In other words, we have a matching M and a blossom B in which one of its vertices v is exposed. We will now create a new graph G/B in which we shrink B into a single vertex b. Notice that we have also a matching M/B in this new graph, and that the sizes of M and M/B differ by exactly $\frac{|B|-1}{2}$ (as we deleted so many edges). We use the following crucial theorem.

Theorem 2 *M is a maximum size matching in G if and only if M/B is a maximum size matching in G/B.*

Proof: ($\Longrightarrow$) Suppose N is a matching in G/B larger than M/B. Pulling N back to a set of edges in G, it is incident to at most one vertex of B. Expand this to a matching N^+

in G by adjoining $\frac{1}{2}(|B|-1)$ edges to match every other vertex in B. Then $|N^+|$ exceeds $|M|$ by the same amount that $|N|$ exceeds $|M/B|$.

($\Longleftarrow$) By contradiction. If M is not of maximum size in G then it has an augmenting path P between exposed vertices u and v. As B has only one exposed vertex, we can assume that $u \notin B$. Let w be the first vertex of P which belongs to B, and let Q be the part of P from u to w. Notice that, after shrinking B, Q remains an augmenting path for M/B. This means that M/B is not maximum either, and we have reached a contradiction. $\triangle$

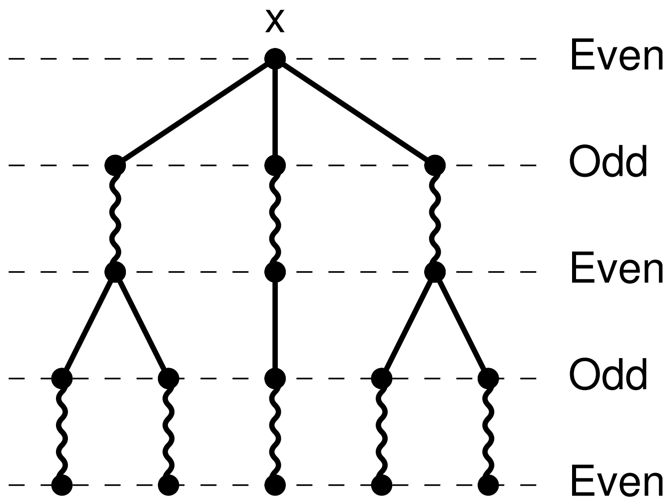

Figure 2: An alternating tree. The squiggly edges are the matching edges.

To find either an alternating path or a flower, we proceed as follows. We label all exposed vertices to be *Even*, and keep all the other vertices unlabelled at this point. As we proceed, we will be labelling more vertices to be *Even* as well as labelling some vertices to be *Odd*. We process the *Even* vertices one at a time, say we are currently processing u, and consider the edges adjacent to u. There are several possibilities:

1. If there is an edge (u, v) with v unlabelled, we label v as *Odd*. As v cannot be exposed (as otherwise it would have been already *Even*), we label its "mate" w (i.e. (v, w) is an edge of the matching) as *Even*. We have extended the alternating tree we are building (see Figure 2).

2. If there is an edge (u, v) with v labelled *Even* and v belongs to another alternating tree than u does, we have found an alternating path (just traverse the 2 alternating trees from u and v up to their roots) and augment the matching along it, and start again from this new, larger matching.

3. If there is an edge (u, v) with v labelled *Even* and v belongs to the same alternating tree as u does, we have found a flower and we take the symmetric difference with its

stem and shrink the blossom. We recursively find a maximum matching in this graph G/B (and this may result in further shrinkings) and when the algorithm terminates, we use Theorem 2 to expand it to a maximum matching in the original graph.

Correctness. Now suppose that none of these possibilities apply any more for any of the *Even* vertices. Then we claim that we have found a maximum matching M' in the current graph $G' = (V', E')$ (which was obtained from G by performing several shrinkings). To show this, consider $U = Odd$ and consider the upper bound (2) for G'. As there are no edges between *Even* vertices (otherwise 2. or 3. above would apply) and no edges between an *Even* vertex and an unlabelled vertex (otherwise 1. would apply), we have that each *Even* vertex is an (odd) connected component by itself in $G' \setminus Odd$. Thus $c_0(G' \setminus Odd) = |Even|$. Also, we have that $|M'| = |Odd| + \frac{1}{2}(|V'| - |Odd| - |Even|)$, the second term coming from the fact that all unlabelled vertices are matched. Thus,

$$\frac{1}{2}(|V'| + |Odd| - c_0(G' \setminus Odd)) = \frac{1}{2}(|V'| + |Odd| - |Even|) = |M'|,$$

and this shows that our matching M' is maximum for G'. Applying repeatedly Theorem 2, we get that the algorithm constructs a maximum matching in G.

Running Time. The algorithm will perform at most n augmentations (of the matching) where $n = |V|$. Between two augmentations, it will shrink a blossom at most $n/2$ times, as each shrinking reduces the number of vertices by at least 2. The time it takes to construct the alternating tree is at most $O(m)$ where $m = |E|$, and so the total time is $O(n^2 m)$.

Correctness of Tutte-Berge Formula. We can now prove Theorem 1. As we have argued the Tutte-Berge formula holds for the shrunk graph $G' = (V', E')$ at the end of the algorithm. Now, let's see what happens when we unshrink one blossom B (to get the graph $\hat{G} = (\hat{V}, \hat{E})$. The matching M increases in size by $\frac{|B|-1}{2}$. On the other hand, when the blossom was shrunk, it became an *Even* vertex and therefore our set U ($= Odd$) is unchanged as we unshrink the blossom. The set of odd connected components is unaffected as well (i.e. $c_0(G' \setminus U) = c_0(\hat{G} \setminus U)$), but the number of vertices increases by $|B| - 1$ (i.e. $|\hat{V}| = |V'| + |B| - 1$). Thus, since the Tutte-Berge formula is true for G', it is also true for $\hat{G}$. By expanding one blossom at a time, we get that the Berge-Tutte formula holds for the original graph G. This proves Theorem 1.

The Tutte-Berge formula says that a graph has a perfect matching if and only if for every set U the number of odd connected components of $G \setminus U$ is at most $|U|$.