CS 369P: Polyhedral techniques in combinatorial optimization
Instructor: Jan Vondrák
Lecture date: September 30, 2010 **Scribe: David Tobin**

1 Matching in Non-Bipartite Graphs

There are several differences between matchings in bipartite graphs and matchings in non-bipartite graphs. For one, König's Theorem does not hold for non-bipartite graphs. For a simple example, consider a cycle with 3 vertices. The maximum matching is 1 edge, but the minimum vertex cover has 2 vertices.

Additionally, incidence matrices are not totally unimodular in non-bipartite graphs. The incidence matrix for a triangle is

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

which has determinant 2.

Finally, $P_{match}(G) = \left\{\mathbf{x} \in \mathbb{R}_+^E : \forall v \in V : x(\delta(v)) = 1\right\}$ is not equal to the convex hull of the matchings of G. As an example, let G be a triangle. The assignment of $\frac{1}{2}$ to each edge satisfies the vertex constraints but is not a convex combination of matchings.

1.1 Characterization of optimality

Definition 1 *For a matching M on a graph G, an M-augmenting path is a path $(e_1, e_2, \ldots, e_k)$ of odd length from v_1 to v_2 such that v_1 and v_2 are not covered by M, $e_1, e_k \notin M$, and the edges e_i alternate membership in M.*

Theorem 2 *For a matching M on a graph G, M is a maximal matching in $G \Leftrightarrow$ there is no M-augmenting path in G.*

Proof of $\Rightarrow$. Assume for the sake of contradiction that M is maximal and there is some M-augmenting path P. Let $M' = M \Delta P = (M \setminus P) \cup (P \setminus M)$. But then $|M'| = |M| + 1$, so M is not maximal, which is a contradiction. □

Proof of $\Leftarrow$. Assume that a larger matching M' exists; we will show that there exists and M-augmenting path. Consider the graph G' which has the same vertices as G and the edges $M \Delta M'$. The degree of each vertex in G' is at most two, since each vertex is adjacent to at most one edge in M and at most one edge in M'. Thus the edges of G' form a collection of disjoint cycles and paths. Along a cycle or path the edges must alternate between edges of M and M' because M and M' are both matchings. This forces all cycles to be of even length.

By assumption $|M'| > |M|$. So there must be a path P where M' contributes more edges than M. Since the edges of P must alternate between edges of $(M' \setminus M)$ and edges of $(M \setminus M')$, P is an M-augmenting path. □

This proof yields an "algorithm" for finding maximal matchings in non-bipartite graphs. If there are any vertices not covered by the current matching ("exposed" vertices), search for an M-augmenting path from each exposed vertex v. If an augmenting path P is found, augment M by updating M to $M \Delta P$. If no M-augmenting paths can be found, M is optimal.

This raises the question of whether or not the algorithm can be be performed in polynomial time. In particular, we would like an efficient method for finding augmenting paths or certifying that no augmenting paths exist.

1.2 M-alternating trees

At any step in the algorithm, let M be the current matching and let X be the set of exposed vertices (vertices not covered by M).

Definition 3 *An M-alternating tree is a tree in G with a root vertex $r \in X$ such that along every path $P = e_1 e_2 \dots e_j$ from r to a leaf v the edges alternate between being in M and not being in M ($e_i \in M \Leftrightarrow i$ is even).*

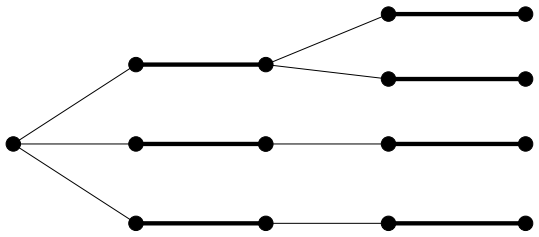

Figure 1: An M-alternating tree

Lemma 4 *If an M-alternating tree contains a vertex $v \in X$ distinct from the root, then there exists an M-augmenting path.*

Proof: If the tree contains another vertex $v \in X$, there is a unique path in the tree from r to v. The edges along this path alternate by the definition of an M-alternating tree, and the edge incident to v is a non-matching edge by the definition of the set X. Thus the path from r to v is an M-augmenting path. □

Definition 5 *For an M-alternating tree T with root $r \in X$, Odd is the set of vertices in T at an odd distance from r and Even is the set of vertices in T at an even distance from r (including r).*

Definition 6 *A maximal M-alternating tree is an M-alternating tree such that no Even vertex in the tree has an edge to a vertex not in the tree (i.e. no additional vertices can be added to the M-alternating tree).*

Now consider an algorithm to divide up a graph into a collection of maximal M-alternating trees:

1. Choose a vertex $r \in X$ as the root of a new M-alternating tree and initialize the set Even to $\{r\}$.

2. Add all vertices adjacent to vertices in Even, and not visited so far, to the set Odd. If any of the vertices in Odd are elements of X, then we have an M-augmenting path. Add the augmenting path to M and then start the algorithm over.

3. For each new Odd vertex, add the matching edge to the tree and add the corresponding vertex to the set Even.

4. Repeat steps 2 and 3 until no more vertices can be added to the M-alternating tree.

5. Remove the vertices of the M-alternating tree rooted at r and go back to step 1 to start a new M-alternating tree.

This algorithm runs until every vertex remaining in X is the root of an M-alternating tree.

Lemma 7 *If the algorithm finds is a collection of maximal M-alternating trees such that*

(i) The set of roots of the trees is X.

(ii) There are no edges between Even vertices.

then M is a maximum matching.

Proof: Consider the sets of Even and Odd vertices for a collection of maximal M-alternating trees. By assumption, there are no edges between two vertices in Even. There are no edges from Even to vertices outside of (Even $\cup$ Odd); otherwise those edges and their other endpoints would have been added as Odd vertices during the process of creating the M-alternating trees. Thus any edge in G with one endpoint in Even must have the other endpoint in Odd.

In each M-alternating tree, the Even and Odd edges can be paired up by the matching edges in the tree, leaving only the root unpaired. So for each M-alternating tree, the number of Even vertices is 1 more than the number of Odd vertices. Across all of the trees,

$$|\text{Even}| \quad = \quad |\text{Odd}| + (\#\text{ of M-alternating trees}) = |\text{Odd}| + |X|$$

However, as we argued, Even vertices can be matched only to Odd vertices. So, in any matching at least $|X|$ vertices must be unmatched. The current matching has $|X|$ unmatched vertices, so the current matching M must be optimal. □

Corollary 8 *If G is bipartite and the algorithm finds a collection of maximal M-alternating trees, then M is a maximal matching.*

Proof: By Lemma 7, we only need to show that there are no Even-Even edges when the algorithm terminates. If there were an Even-Even edge within a tree, it would form an odd-length cycle because Even vertices are always an even distance from the root of the tree. Bipartite graphs cannot contain odd-length cycles, so such an edge cannot exist. The other case to consider is an Even-Even edge between different trees. But one of the trees must have been constructed before the other, and the first tree would have used the edge to continue growing if it had existed. □

1.3 Even-Even edges in non-bipartite graphs

Whenever we have an Even-Even edge in an M-alternating tree, it forms an odd-length cycle in the M-alternating tree. We can call the relevant subgraph of the tree a *flower*. The odd-length cycle formed by the Even-Even edge is called the *blossom*, and the path from the root to the closest vertex in the blossom is called the *stem*.

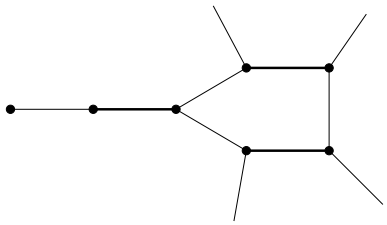

Figure 2: A flower in G

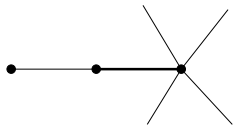

Figure 3: $G \setminus B$ after the blossom has been contracted

If we encounter a blossom in an M-alternating tree, we can contract the blossom into a vertex and continue growing the tree. Let G/B denote G contract B, which is the graph formed by replacing B with a single new vertex and adding and edge to the new vertex from every vertex that used to connect to a vertex in B.

Lemma 9 *For a graph G, a matching M, and a blossom B, M is a maximum matching in $G \Leftrightarrow$ M/B is a maximum matching in G/B.*

Proof of $\Rightarrow$. We can equivalently show that if there exists an (M/B)-augmenting path in (G/B), then there also exists an M-augmenting path in G. Consider a particular (M/B)-augmenting path in (M/B). If this path does not pass through B, then the claim holds trivially, so assume that the path does pass through B. To get an augmenting path in G, we can follow the augmenting path in (G/B) from the root until it reaches the cycle B. The continuation of the path in (G/B) after it leaves B either starts with an edge in M or an edge not in M. The path in G can go one way around B or the other to reach the continuation of the path; since B is an odd cycle we can always choose the correct direction so that the edges on the augmenting path in G alternate correctly. □

Proof of $\Leftarrow$. We can equivalently prove that if M is not a maximal matching in G, then (M/B) is not a maximal matching in (G/B). Let S be the stem corresponding to the blossom B. If M is not a maximal matching, consider $M' = M \Delta S$. Taking the symmetric difference with S switches the exposed vertex from one end of S to the other, so the exposed vertex in M' is part of B. Because S has the same number of edges in M and not in M, $|M| = |M'|$. Thus M' is also not maximal, and there must be an M'-augmenting path Q. There are two cases for the augmenting path Q:

(i) If Q does not touch B, then we are done because the path is also an (M'/B)-augmenting path in G/B.

(ii) If Q touches B, then consider a path in G that follows Q from the endpoint not in B to the first vertex at which the path meets B. (It cannot be the case that both endpoints are in B, because we have only one exposed vertex in B.) We claim that with respect to M'/B in G/B, the vertex B is exposed, so that path is an (M'/B)-augmenting path. This claim holds because every vertex in B had a matching edge in M' to another vertex in B, with the exception of the node that is exposed in M'. The vertices with matching edges inside of B cannot have M' edges going out of B, and the exposed vertex does not have any incident edge in M'. Hence the contracted blossom is exposed with respect to M'/B.

In both cases there is an (M'/B)-augmenting path. This means that M'/B is not maximal, so M/B (of the same size) is also not a maximal matching. □

1.4 Edmonds' Algorithm

Adding the tools to handle Even-Even edges to the original algorithm yields the following algorithm:

1. Construct a collection of maximal M-alternating trees as previously described. Each time an M-augmenting path is found, increase the size of M and repeat this step.

2. If there is an Even-Even edge, contract the blossom formed by that Even-Even edge and continue growing the tree. If an augmenting path is found in $G/B_1/B_2/B_3\ldots$, then we can also find an augmenting path in G by expanding the blossoms (as described in the proof of Lemma 9). Use this augmenting path to increase the size of M and then go back to step 1.

3. If there are no augmenting paths and no Even-Even edges, stop.

Claim 10 *If the algorithm terminates, then the matching M is maximal.*

Proof: When the algorithm terminates the current matching $M/B_1/B_2\ldots$ is maximal in $G/B_1/B_2\ldots$ by Lemma 7. By Lemma 9, M is therefore maximal in G. □

Claim 11 *The algorithm terminates in polynomial time.*

Proof: It is easy to see that each basic operation needed to maintain the alternating trees can be implemented efficiently. The crucial observation is that only $O(n)$ operations can be performed before we find an M-augmenting path, or we terminate. This is because each operation either extends a tree and hence decreases the number of vertices outside of all trees, or we contract a blossom which decreases the total number of vertices. We can repeat this only $O(n)$ times before we find an M-augmenting path or a collection of maximal M-alternating trees. The number of times we can augment M is also $O(n)$, so the total number of basic operations is $O(n^2)$. (We do not claim a specific running time as this would depend on the particular data structures that we use.) □

1.5 The Tutte-Berge Formula

In a bipartite graph, we can certify the maximality of a matching with a vertex cover. But König's Theorem does not hold for non-bipartite graphs, so we wish to find a different certificate of optimality for non-bipartite matching. Consider the final state of Edmonds' Algorithm when run on a non-bipartite graph. We show that the set Odd is a certificate of optimality.

Lemma 12 *Contracting a blossom B produces an Even vertex and does not change the Odd/Even membership of any other vertices.*

Proof: The parity of a vertex is determined by the path length from the root to that vertex. Consider the vertex at the beginning of the blossom. The vertex has two edges in B; those edges cannot be edges in M because there are two of them. Thus the vertex at the beginning of the blossom is Even, so when the blossom is contracted it becomes an Even vertex as well. When a blossom is contracted, all outgoing edges from the blossom that are part of the tree are edges coming off of Even nodes. So when the blossom is contracted, vertices connected to the root through the blossom all experience an even change in distance from the root. □

Lemma 13 *A vertex in $G/B_1/B_2\ldots$ always represents a connected subgraph on an odd number of vertices in G.*

Proof: By the lemma above, vertices in Odd do not contain any contracted subgraphs, so the claim is trivially true. An Even vertex either represents itself of a blossom. Whenever a blossom B is contracted, it contains an odd number of vertices. Some of those vertices might also represent blossoms, but by induction those represent an odd number of vertices. The total number of vertices represented by B is then an odd sum of odd numbers, which must be odd. □

Claim 14 *If X and Odd are the sets of exposed and odd-distance vertices when the algorithm terminates, then the graph $G \setminus Odd$ has at least $|Odd| + |X|$ connected components with an odd number of vertices.*

Proof: Consider the graph $G \setminus \text{Odd}$. When the algorithm terminates, all edges from each Even vertex go to Odd. Therefore, after unshrinking, each Even vertex becomes a component which is not connected to anything else in $G \setminus \text{Odd}$. By the lemma above, it is a component of odd size. As we already argued, we have $|\text{Even}| = |\text{Odd}| + |X|$. □

Now we come to the min-max relation for non-bipartite matchings, which is known as the Tutte-Berge formula. First, we formulate it in terms of exposed vertices.

Definition 15 *For a graph H, odd(H) is the number of connected components of odd size in H.*

Lemma 16 $\min$ *# of exposed vertices in any matching* $= \max_{U \subseteq V} (odd(G \setminus U) - |U|)$

Proof: First we prove the "weak duality" aspect of the claim: $\forall U \subseteq V$, at least $\text{odd}(G \setminus U) - |U|$ vertices must be exposed in any matching. Because each odd component has an odd number of vertices, the only way we can match all vertices of an odd component is that at least one vertex must be matched outside of the component. The only vertices that could be matched to the odd

components are the vertices of U, and each vertex in U can participate in at most one matching edge. Hence, at least $\text{odd}(G \setminus U) - |U|$ vertices must remain unmatched.

To show that equality is achieved by some set U, consider the set $U = \text{Odd}$ at the end of Edmonds' algorithm. We proved that $G \setminus U$ has $|U| + |X|$ components of odd size. Therefore, indeed the number of exposed vertices is equal to $\text{odd}(G \setminus U) - |U|$. This shows that the set $U = \text{Odd}$ is a certificate of optimality for the matching. □

Notice that for any matching M, $|M| = \frac{1}{2}(|V| - |X|)$. Thus we can manipulate the expressions to get a more common form of the minimax relation.

Theorem 17 *(Tutte-Berge formula)*

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

Proof:

$$\begin{aligned}
\max |M| &= \max \tfrac{1}{2}\left(|V| - |X|\right) \\
&= \tfrac{1}{2}\left(|V| - \min |X|\right) \\
&= \tfrac{1}{2}\left(|V| - \max_{U \subseteq V} \left(\text{odd}(G \setminus U) - |U|\right)\right) \\
&= \min_{U \subseteq V} \frac{1}{2}\left(|V| + |U| - \text{odd}(G \setminus U)\right)
\end{aligned}$$

□

Corollary 18 *(Tutte's theorem) G has a perfect matching $\Leftrightarrow \forall U \subseteq V; odd(G \setminus U) \leq |U|$*

Proof: Directly from the Tutte-Berge formula, $\max |M| = \frac{1}{2}|V|$ iff $\min_{U \subseteq V}(|U| - \text{odd}(G \setminus U)) = 0$.
□