CMPUT 675: Topics in Combinatorics and Optimization **Fall 2016**

Lecture 12 (Oct 3): Maximum Matchings in General Graphs

Lecturer: Zachary Friggstad *Scribe: Zachary Friggstad*

12.1 Alternating Paths Proof

We begin by completing the proof of the following theorem. See the previous lecture for notation.

Recall that $\nu(G)$ denotes the size of a maximum matching in G.

Theorem 1 *Let M be a matching, $B = (V_B; E_B)$ an M-blossom with base v_k and stem $v_1, \ldots, v_k$ consisting of the edge set P. Then $\nu(G) = |M|$ iff $\nu(G/V_B) = |M| - \frac{|V_B|-1}{2} = |M - E_B|$*

Proof.[second half]

Suppose, now, that M is not a maximum matching in $\mathcal{G}$. The set $N := (M - P) \cup (P - M)$ is also a matching in $\mathcal{G}$ with $|M| = |N|$ that leaves the base of blossom $\mathcal{B}$ exposed. Let $N' = N - E_B$ and note that N' is also a matching in $\mathcal{G}/V_B$ (the graph obtained by contracting the blossom).

Because N is not a maximum matching, there is an N-alternating path Q in $\mathcal{G}$. We show in two cases that we can construct an N'-alternating path in $\mathcal{G}/V_B$.

- **Case**: Q shares no nodes with V_b.
  Then Q is also an N'-alternating path in $\mathcal{G}/V_B$.
- **Case**: Q shares a node with V_b.
  Write the nodes of Q as $v_1, v_2, \ldots, v_k$. As both v_1 and v_k are N-exposed, either $v_1 \notin V_B$ or $v_k \notin V_B$ (because N only leaves one exposed node in the blossom). Suppose $v_1 \notin V_B$.

  Let ℓ be the least index such that $v_\ell \in V_B$. As no edge of N includes precisely one node in V_B, then $v_{\ell-1}v_\ell \notin N$ so ℓ is even. Consider the path $v_1, v_2, \ldots, v_{\ell-1}, \overline{v}$ in $\mathcal{G}/V_B$ where $\overline{v}$ represents the node obtained by contracting V_B. This is an N'-alternating path in $\mathcal{G}/V_B$.

In either case, we see N' is not a maximum matching in $\mathcal{G}/V_B$. Finally, as $|M'| = |N'|$ we see that M' is also not a maximum matching in $\mathcal{G}/V_B$.

■

12.2 M-Alternating Forests

The concept of an M-alternating forest helps us find either alternating paths or blossoms with stems. As we shall see, if such a forest contains neither, then we can use this as a certificate that M is a maximum matching.

Definition 1 *Let M be a matching in a graph $\mathcal{G} = (V; E)$. An M-**alternating forest** is a subgraph $(U; F)$ of $\mathcal{G}$ where each component is a rooted tree with the following properties. Here, we define the **height** $\mathrm{ht}(v)$ for some $v \in U$ to be its distance to the root of its component (with roots r having $\mathrm{ht}(r) = 0$).*

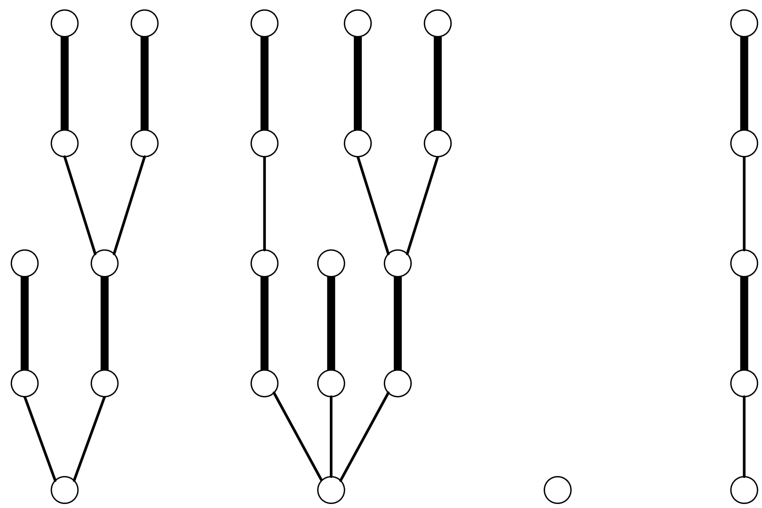

Figure 12.1: An M-alternating forest. Note all nodes are required to lie in the forest, but any node not in the forest is matched to another node not in the forest.

1. *The roots of the trees are exactly the M-exposed nodes.*
2. *Every leaf $v \in U$ has even height.*
3. *Every node $v \in U$ with odd height has its only child being the node u it is matched to. That is, $vu \in M$ is the only edge between v and a node at height $\mathrm{ht}(v) + 1$.*

$(U; F)$ is called a **maximal M-alternating forest** *if for any other M-alternating forest $(U'; F')$ with $U \subseteq U'$ we have $U = U'$.*

Figure 12.1 depicts an M-alternating forest. Note the first property also means the number of trees in an M-alternating forest is exactly $|V| - 2|M|$.

The following three results show the usefulness of looking at M-alternating forests. For brevity, let $\rho(v)$ denote the root of the component of the M-alternating forest $(U; F)$ for any $v \in U$ (it will be clear from the context which forest is being discussed).

For the remaining lemmas in this section, we fix $(U; F)$ be an M-alternating forest.

Lemma 2 *Suppose $u, v \in U$ have even height and $uv \in E$. If $\rho(u) \neq \rho(v)$, the path $\rho(u) - u$ in F, followed by edge uv, followed by the path $v - \rho(v)$ in F is an M-alternating path.*

Proof. Immediate from construction. ■

Lemma 3 *Suppose $u, v \in U$ have even height and $uv \in E$. If $\rho(u) = \rho(v)$, let w be the least common ancestor of u and v. Let V_B be all nodes lying between u and v in the tree. Let E_B be all corresponding edges in this path plus the edge uv.*

Then $B = (V_B, E_B)$ is an M-blossom and the path in F from $\rho(u)$ to w is a stem for B.

Proof. The set (V_B, E_B) is an odd-length cycle. It is easy to verify any odd-length cycle is factor-critical. By construction, $|M \cap E_B| = \frac{|V_B|-1}{2}$. So B is in fact a blossom.

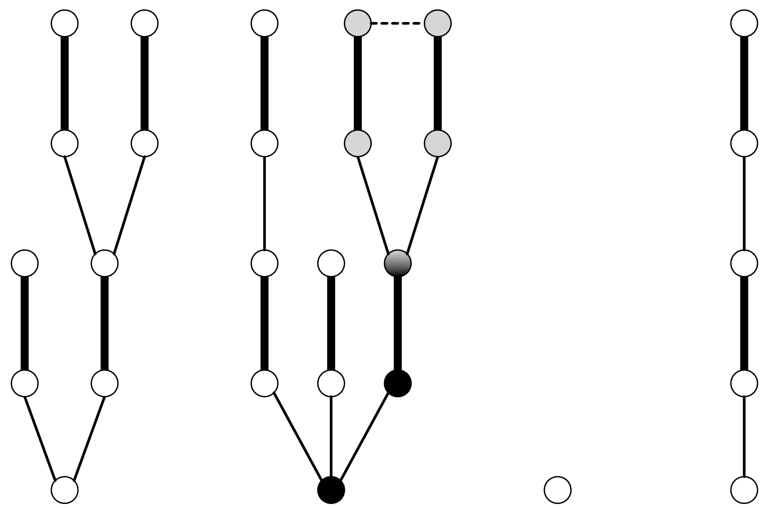

Figure 12.2: The dashed edge identifies a blossom. The blossom is depicted with grey nodes, and the stem with black nodes (the base of the blossom is shaded both grey and black).

That the $\rho(u) - w$ path in F is a stem for B is immediately verified (note w must also have even height because every odd-height node has only one child in F). ■

A blossom found this way is depicted in Figure 12.2

Lemma 4 *Suppose there is an edge $uv \in E$ with $u \in U$ having even height and $v \notin U$. Then $vw \in M$ for some other $w \notin U$ and $(U \cup \{v, w\}, F \cup \{uv, vw\})$ is an M-alternating forest.*

Proof. node v must be matched because all exposed nodes are in U (they are the roots of the components). Then node w cannot be in U, otherwise the $w - \rho(w)$ path must have odd length as it would start and end with an edge not in M. So w would have odd height, but the definition of M-alternating forests would then mean $v \in U$. Finally, that adding v, w to U and uv, vw to F leaves an M-alternating forest is then easy to verity. ■

An alternating path found this way is depicted in Figure 12.3

Lemma 5 *Now suppose $(U; F)$ is a* maximal *M-alternating forest. If no edge between even-height nodes exists as in the previous two lemmas, then M is a maximum matching.*

Proof. There also cannot be an edge uv as in Lemma 4 because $(U; F)$ is maximal. So, every $u \in U$ with even height has all of its neighbours in U and having odd height.

Let $X \subseteq U$ be the set of odd-height nodes and $Y \subseteq U$ the set of even-height nodes. The above observation means each $u \in Y$ forms a singleton (isolated) component in $\mathcal{G} - X$. In other words, eacy $u \in Y$ has its only neighbours being in X.

Note that M leaves $|Y| - |X|$ nodes exposed (the nodes in X are in one-to-one correspondence with the matched nodes in Y). But each matching must match each node of Y into a node of X, so no matching can leave fewer nodes exposed. Therefore, M is a maximum matching in $\mathcal{G}$. ■

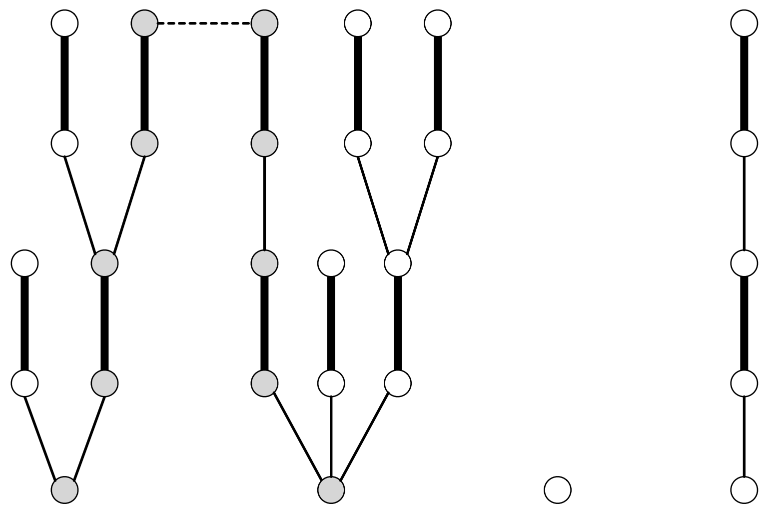

Figure 12.3: The dashed edge identifies an alternating path. The vertices on this path are depicted in grey.

12.3 The Algorithm

The way this algorithm is used is described in Algorithm 2

Finally, we iterate this by starting with $M \leftarrow \emptyset$ and repeatedly calling `augment`$(\mathcal{G}, M)$ to get a larger matching until we get a declaration that M is a maximum matching in $\mathcal{G}$.

Each run of Algorithm 1 can be implemented in $O(m)$ time. Simply use a breadth-first search starting with the M'-exposed nodes as tree roots and expanding all even-height vertices to grow the forest.

Apart from recursive calls, the work done in a single call of `augment` is $O(n)$ (since each alternating path, each matching, and and blossom has size $O(n)$). There are also $O(n)$ recursive calls as each shrinking of a blossom reduces the number of nodes by at least 2.

Overall, it takes $O(m \cdot n^2)$ time to find an M-alternating path, so the overall algorithm takes $O(m \cdot n^3)$ time. This can be sped up a lot by clever observations (in the textbook) that do not explicitly contract a blossom and restart. Between the main discussion in the textbook and the exercises, an $O(n \cdot m \cdot \log n)$ algorithm is devised. The running time can be improved to $O(\sqrt{n} \cdot m)$ using observations about alternating many paths simultaneously in linear time[1].

[1] see, e.g., V. V. Vazirani, *A simplification of the MV matching algorithm and its proof*, `https://arxiv.org/abs/1210.4594`, 2012. This is based on earlier work by Micali and Vazirani

---

Algorithm 1 Blossom, Alternate, or Quit

Input: A graph $\mathcal{G}' = (V', E')$ and a matching M'.
Output: An M-alternating path P, an M-blossom $\mathcal{B}$ with a stem P, or a declaration that M is a maximum matching.

$U \leftarrow M'$-exposed nodes
$F \leftarrow \emptyset$
Let $\mathcal{T}$ denote the forest (U, F) throughout the algorithm.
while there is some $uv \in E$ with u having even height in $\mathcal{T}$ and $v \notin U$ **do**
  $w \leftarrow$ the vertex with $vw \in M'$ $\{$we showed $w \notin U\}$
  $U \leftarrow U \cup \{v, w\}$
  $F \leftarrow F \cup \{uv, vw\}$
end while $\{$Now (U, F) is a maximal M'-alternating forest$\}$
if $uv \in E$ for some u, v having even height in $\mathcal{T}$ **then**
  if u, v are in the same component **then**
    return the corresponding blossom (the $u - v$ path in $\mathcal{T}$ plus edge uv)
  else
    return the corresponding alternating path
  end if
else
  return a declaration that M' is a maximum matching in $\mathcal{G}'$
end if

Algorithm 2 `augment`$(\mathcal{G}', M')$

Input: A graph $\mathcal{G}' = (V', E')$ and a matching M'
Output: A larger matching M'' or a declaration that M' is a maximum matching

run Algorithm 1 with $\mathcal{G}', M'$
if a blossom $\mathcal{B} = (V_B, E_B)$ is returned **then**
  if `augment`$(\mathcal{G}'/V_B, M' - E_B)$ finds a larger matching $\widehat{M}$ **then**
    $M'' \leftarrow \widehat{M}$ plus a matching of size $\frac{|V_B|-1}{2}$ in $\mathcal{B}$ (avoiding the only $\widehat{M}$ matched vertex in V_B, if any)
    return M''
  else
    return a declaration that M' is a maximum matching in $\mathcal{G}'$
  end if
else if an M'-alternating path P is returned **then**
  return $(M' - P) \cup (P - M')$
else
  return a declaration that M' is a maximum matching in $\mathcal{G}'$
end if
