Matchings in Graphs

Lecturer: *Meena Mahajan* Meeting: 3-4
Scribe: *Rajesh Chitnis* 21-28th January 2010

1 Some preliminary definitions

Definition 1 *Let $\rho(G)$ denote the size of maximum matching in G. A vertex $v \in V(G)$ is called* ***critical*** *if $\rho(G) > \rho(G - v)$*

Definition 2 *We define three special subsets of the vertex set of G.*
$D(G) = \{\ u :\ u \text{ is not critical } \}$
$A(G) = \{\ u :\ u \text{ is critical and has a non-critical neighbour } \}$
$C(G) = \{\ u :\ u \text{ is critical and all neighbours of } u \text{ are critical } \}$
The set $A(G)$ is called as the ***Tutte Set*** *of G. Clearly, the sets $D(G), A(G)$ and $C(G)$ partition the vertex set of G.*

Definition 3 *Let $\mathcal{L} = \{M_1, \ldots, M_d\}$ be a family of equisized matchings in G. We again define three subsets of the vertex set of G using the family $\mathcal{L}$.*
$D(\mathcal{L}) = \{\ u :\ \exists\ i \in [d] \text{ such that } u \text{ is free in } M_i\ \}$
$A(\mathcal{L}) = \{\ u :\ u \notin D(\mathcal{L}) \text{ and } u \text{ has a neighbour in } D(\mathcal{L})\}$
$C(\mathcal{L}) = V(G) \setminus \Big(D(\mathcal{L}) \cup A(\mathcal{L})\Big)$

2 Some Remarks

Remark 4 *We cannot let $\mathcal{L}$ be the set of all maximum sized matchings in G since $\mid \mathcal{L} \mid$ may be exponential which would inhibit us searching for a polynomial time algorithm*

Remark 5 *Surprisingly we will show that $\exists$ a polynomial sized family of maximum matchings in G such that*

$$D(\mathcal{L}) = D(G)$$
$$A(\mathcal{L}) = A(G)$$
$$C(\mathcal{L}) = C(G)$$

3 Structural Algorithm to find Maximum Matching

The following algorithm returns a maximum cardinality matching along with a witness set.

$M = \emptyset$; $\mathcal{L} = \{M\}$
Step One - Find $M \in \mathcal{L}$ that is not $\mathcal{L}$-good
$M' = \text{NextMatch}(M, \mathcal{L})$
if $\mid M' \mid = \mid M \mid + 1$ **then**
$\mathcal{L} = \{M'\}$; Update D, A, C and go to Step One
else
$D(\mathcal{L}) \nsubseteq D(\mathcal{L} \cup \{M\})$
i.e. M' leaves a vertex x in $A(\mathcal{L}) \cup C(\mathcal{L})$ free ;
$\mid M' \mid = \mid M \mid$; $\mathcal{L} = \mathcal{L} \cup \{M'\}$; Update D, A, C and go to Step One
end if

Output any $M \in \mathcal{L}$
The **NextMatch** sub-routine is described at a later stage.

4 Preliminaries regarding above algorithm

Definition 6 *A matching $M \in \mathcal{L}$ is said to be $\mathcal{L}-$good if*

1. *M does not match any vertex of $A(\mathcal{L})$ to any vertex of $A(\mathcal{L}) \cup C(\mathcal{L})$*
2. *M is near-perfect on all components of $G[D(\mathcal{L})]$*

Note that above definiton implicitly implies that all components of $G[D(\mathcal{L})]$ are odd.

Lemma 7 *Let $\mathcal{L}$ be a family of equisized matchings and let $M \in \mathcal{L}$. If M is $\mathcal{L}$-good, then*

1. *M is a maximum matching*
2. *$A(\mathcal{L})$ is a witness set*

Proof: All components of $C(\mathcal{L})$ are even as they have perfect matchings. On the other hand all components of $D(\mathcal{L})$ are odd since they have near-perfect matchings. Hence $o(G - A(\mathcal{L}))$ is equal to number p of connected components of $D(\mathcal{L})$. Now, for every vertex in $A(\mathcal{L})$ there is a matching edge connecting it to a vertex in a unique component of $D(\mathcal{L})$. The number of free vertices is equal to $(p- \mid A(\mathcal{L}) \mid) = \Big(o(G - A(\mathcal{L})) - \mid A(\mathcal{L}) \mid \Big) = def(A(\mathcal{L}))$. Thus, $A(\mathcal{L})$ is a witness set and M is a maximum matching ■

Corollary 8 *Let $\mathcal{L}$ be a family of equisized matchings. If one matching from $\mathcal{L}$ is $\mathcal{L}$-good then every element of $\mathcal{L}$ is $\mathcal{L}$-good.*

Proof: Let $M \in \mathcal{L}$ be $\mathcal{L}$-good. Then by Lemma 7, M is maximum and $A(\mathcal{L})$ is a witness set. Thus $\mid M \mid = \frac{|V|+|A|-o(G-A)}{2} = \frac{|V|+|A|-o(D)}{2}$ as $C(\mathcal{L})$ has perfect matchings and so each component is even. Let $M' \in \mathcal{L}$. Then $\frac{|V|+|A|-o(D)}{2} = \mid M \mid = \mid M' \mid$. In the matching M' let there be l_1, l_2 edges from $A(\mathcal{L})$ to $D(\mathcal{L}), C(\mathcal{L})$ repectively. Noting that G has no edges between $D(\mathcal{L})$ and $C(\mathcal{L})$, we have $\mid M' \mid \leq \frac{|D|-o(D)}{2} + l_1 + \frac{|A|-l_1-l_2}{2} + l_2 + \frac{|C|-l_2}{2} = \frac{|V|+l_1-o(D)}{2}$. Recalling that $\mid M' \mid = \mid M \mid = \frac{|V|+|A|-o(D)}{2}$, we have $l_1 \geq \mid A \mid$. But l_1 was number of edges from $A(\mathcal{L})$ to $D(\mathcal{L})$ in M' and hence $l_1 = \mid A(\mathcal{L}) \mid$. So M' has no edges from $A(\mathcal{L})$ to $A(\mathcal{L}) \cup .C(\mathcal{L})$. Also looking carefully at the bound on $\mid M' \mid$ we see that M' must pick maximum possible edges from each component of $D(\mathcal{L})$ which would mean a near-perfect matching for each component of $D(\mathcal{L})$. Thus M' is also $\mathcal{L}$-good. ■

5 Gallai-Edmond Structure Theorem

The vertex set of G can be partitioned into three sets $D(G), A(G)$ and $C(G)$ such that

1. $A(G)$ is a witness set
2. $G[C(G)]$ has a perfect matching
3. Any maximum matching in G
   - Is perfect on $G[C(G)]$
   - Is near-perfect on each component of $G[D(G)]$
   - Matches vertices in $A(G)$ to distinct components in $G[D(G)]$
4. Each component of $G[D]$ is hypomatchable or factor-critical i.e. if we remove any vertex from a component of $G[D]$, then that component has a perfect matching.

6 The NextMatch sub-routine

NextMatch$(M, \mathcal{L})$ is a sub-routine applied when $M \in \mathcal{L}$ and M is $\mathcal{L}$-bad

1. $\exists\ x \in A$ `such that` $y = M(x) \notin D$. `Note that` $x \in A$ `implies` $\exists$ `some neighbour` z `of` x `such that` $z \in D$

(a) If z is free in M, then y is free in $M' = M + xz - xy$

(b) Suppose z is not free in M.
But $z \in D$ and so $\exists\ N \in \mathcal{L}$ s.t. z is free in N. Let ρ be a $M - N$ alternating path starting from z. Note that the first edge of ρ is a M-edge as z is free in N. If ρ ends in a M-edge, then augment (N, ρ). So suppose that ρ ends in a N-edge. If ρ avoids the edge xy then it also avoids the vertices x and y (as otherwise it would have to use the edge xy). We switch M on $\rho + xz + xy$ to release y. Only case left is that ρ uses edge xy. Suppose ρ uses xy through y first. Then do $M - xy$ and switch on the "tail" to release y. Similarly for x.

2. $\exists$ `component` T `of` $G[D]$ `such that` M `is not near-perfect on` T

(a) Suppose $M|_T$ is perfect on T.
Then for $x \in T$, $\exists\ N \in \mathcal{L}$ which leaves x free. So $N|_T$ is not perfect and hence leaves some x, y free. Go to Case 2(c)

(b) Suppose $M|_T$ leaves some x, y free but x, y are not free in M.
Now $x \in D$ implies $\exists\ N \in \mathcal{L}$ which leaves x free. Let ρ be a $M - N$ alternating path starting at x. Clearly the first edge of ρ is a M-edge as x is free in N. If ρ ends in a M-edge, then augment (N, ρ). So suppose that ρ ends in a N-edge at say some z. If $z \notin D$, then switch N to release z. So suppose $z \in D$. If ρ avoids y, then switch M on ρ and go to Case 2(c). So, only thing to consider now is that if ρ hits y. Let the last vertex before y on the $M - N$ path be u. If $x - u - y$ then switch M on the sub-path $\rho \setminus (x - -u)$ to release u. Else if $x - y - u$ then switch on the sub-path $\rho \setminus (x - -y)$ and go to Case 2(c).

(c) Suppose $M|_T$ leaves some x, y free but atleast one of them (say x) is free in M.
If $xy \in E(G)$, then if y is free in M do $M + xy$ else do $\Big(M - (y, M(y)) + xy\Big)$ which will release $M(y)$. So let ρ be the shortest xy path in T. Let z be the neighbour of x on ρ. Let $N \in \mathcal{L}$ leave z free. Let η be a $M - N$ alternating path starting at z. If $\eta = \emptyset$, then do $M + xz$. So assume η starts with a M-edge. If η ends with a M-edge, then augment (N, η). So consider that η ends with a N-edge. Since x is free in M, if η visits x then it ends at x. If η does not end at x, then switch M on η and add edge xz. So assume η ends at x. If η avoids y, switch M on η to get M'. Now z, y are free in T wrt M' and z is free in G. Repeat Case 2(c) with z, y instead of x, y and note that $d_T(z, y) < d_T(x, y)$. Else η goes through y. Then switch M on the subpath $x \rightarrow u$ of $(\eta + xz)$ thus releasing u