CS 598CSC: Combinatorial Optimization	**Lecture date: Feb 23, 2010**
Instructor: Chandra Chekuri	**Scribe: Rajhans Samdani**

1 Min Cost Perfect Matching

In this lecture, we will describe a strongly polynomial time algorithm for the minimum cost perfect matching problem in a **general** graph. Using a simple reduction discussed in lecture 9, one can also obtain an algorithm for the maximum weight matching problem. We also note that when discussing perfect matching, without loss of generality, we can assume that all weights/costs are non-negative (why?).

The algorithm we describe is essentially due to Edmonds. The algorithm is *primal-dual* based on the following LP formulation and its dual.
Primal:

$$\begin{array}{rcll} \min \sum_{e\in E} w(e)x(e) & & & \text{subject to:} \\ x(\delta(v)) & = & 1 & \forall v \in V \\ x(\delta(U)) & \geq & 1 & \forall U \subset V, |U| \geq 3, |U| odd \\ x(e) & \geq & 0 & \forall e \in E \end{array} \tag{1}$$

Dual:

$$\begin{array}{rcll} \max \sum_{U\subseteq V,|U|\geq 3,|U| odd} \pi(U) & & & \text{subject to:} \\ \sum_{U\subseteq V,|U|\geq 3,|U| odd, \delta(U)\ni e} \pi(U) & = & w(e) & \forall e \in E \\ \pi(U) & \geq & 0 & \forall U \subseteq V, |U| \geq 3, |U| odd \end{array}$$

We note that non-negativity constraints on the dual variables are only for odd sets U that are not singletons (because the equations for the singleton sets are equalities). In certain descriptions of the algorithm and details, the dual variables for the singleton are distinguished from the odd sets of size ≥ 3, however we don't do that here.

Like other primal dual algorithms, we maintain a feasible dual solution π and an integral infeasible primal solution x and iteratively reduce the infeasiblity of x. Here x corresponds to a matching and we wish to drive it towards a maximum matching. In particular, we will also maintain the primal complementary slackness, that is,

$$x(e) > 0 \Rightarrow \sum_{U:e\in\delta(U)} \pi(U) = w(e)$$

(a primal variable being positive implies the corresponding dual constraint is tight). Thus, at the end, if we have a perfect matching in the primal, it is feasible and certifies its optimality.

The main question is how to update the dual and the primal. Also, we observe that the dual has an exponential number of variables and hence any polynomial time algorithm can only maintain an implicity representation of a subset of the variables.

1.1 Notation

The algorithm maintains a *laminar family* of odd subsets of V denoted by Ω. Ω always includes the singletons $\{v\}, v \in V$. It maintains the invariant that $\pi(U) = 0$ if $U \notin \Omega$, hence Ω is the implicit representation of the "interesting" dual variables. Note that $|\Omega| \leq 2|V|$.

Given G, Ω and $\pi : \Omega \rightarrow \mathbb{R}$ where π is dual feasible, we say an edge is π-*tight* (or *tight* when π is implicit) if $\sum_{U \in \Omega : e \in \delta(U)} \pi(U) = w(e)$.

Let E_π be the set of tight edges (with Ω, π implicit) and G_π be the graph induced by them. We obtain a new graph G' in which we contract each maximal set in Ω into a (pseudo) vertex. For a node $v \in G'$, let $S_v \in \Omega$ be the set of nodes of G contracted to v.

For each $U \in \Omega$, $|U| \geq 3$, consider the graph $G_\pi[U]$ and let H_U be the graph obtained from $G_\pi[U]$ by contracting each maximal *proper* subset $S \subset U$ where $S \in \Omega$. The algorithm also maintains the invariant that H_U has a Hamiltonian cycle B_U.

1.2 Recap on Edmonds Gallai Decomposition

We restae the Edmonds Gallai decomposition and make some observations which help us in proposing and analysing an algorithm for min cost perfect matching. We also use the notation from this section in subsequent sections.

Theorem 1 (Edmonds-Gallai) *Given a graph $G = (V, E)$, let*

$$\begin{aligned} D(G) &:= \{v \in V \mid \text{there exists a maximum matching that misses } v\} \\ A(G) &:= \{v \in V \mid v \text{ is a neighbor of } D(G) \text{ but } v \notin D(G)\} \\ C(G) &:= V \setminus (D(G) \cup A(G)). \end{aligned}$$

Then, the following hold.

1. *The set $U = A(G)$ is a Tutte-Berge witness set for G.*
2. *$C(G)$ is the union of the even components of $G - A(G)$.*
3. *$D(G)$ is the union of the odd components of $G - A(G)$.*
4. *Each component in $G - A(G)$ is factor-critical.*

It is also easy to see the following remark.

Remark 2 *Let M be a maximum matching in G. Then either there exist an $M-$blossom or $D(G)$ consists of singleton nodes.*

1.3 Algorithm

Following is the algorithm for min cost perfect matching using primal dual method.
Initialize: $\Omega = \{\{v\} | v \in V\}, \pi(U) = 0 \; \forall U$ with odd $|U|, M = \phi$, and $G' = G_\pi$.
while(M is not a perfect matching in G') **do**

1. $X \leftarrow M-$exposed nodes in G'.
2. Find $X - X, M-$alternating walk P in G'.

3. If P is an $M-$augmenting path then do

$$M \leftarrow M \Delta E(P)$$

continue.

4. If P has an $M-$blossom B, then do *shrinking* as:

$$U = \cup_{v \in B} S_v, \Omega \leftarrow \Omega \cup U, \pi(U) = 0, G' \leftarrow G'/B, M \leftarrow M/B$$

continue.

5. Else P is empty $\Rightarrow M$ is a maximum matching in G'. Compute $D(G'), A(G')$, and $C(G')$ as in *Edmonds-Gallai decomposition*. Let ϵ be the largest value such that $\pi(S_v) = \pi(S_v) + \epsilon, \forall v \in D(G')$ and $\pi(S_v) = \pi(S_v) - \epsilon, \forall v \in A(G')$ maintains feasibility of π in G.

   If ϵ is unbounded then G has no perfect matching; **STOP**.

   If $\pi(S_v) = 0$ for some $v \in A(G')$ and $|S_v| \geq 3$, then *deshrink* as

   - Remove S_v from Ω.
   - Update G_π and G'.
   - Extend M by a perfect matching in $B_{S_v} - \{v\}$.

end while
Extend M in G' to a perfect matching in G_π and output it.

Example: Consider the execution of this algorithm on the following graph:

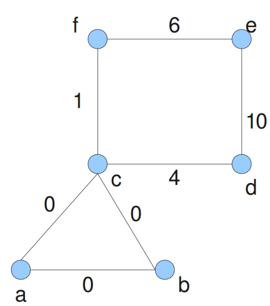

Figure 1: Original graph G

The execution is shown in figures 2 to 9. Red edges are the current edges in the matching; black edges are tight.

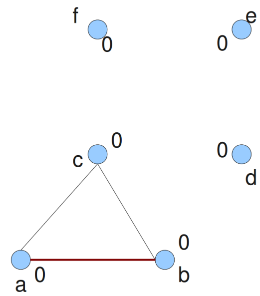

Figure 2: G_π after iteration 1.

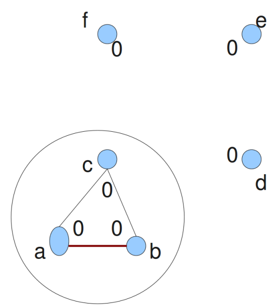

Figure 3: G_π after iteration 2. Shrinking.

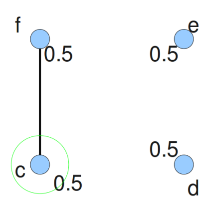

Figure 4: G_π after iteration 3. Edge tight.

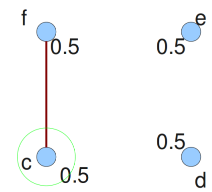

Figure 5: G_π after iteration 4. Augment.

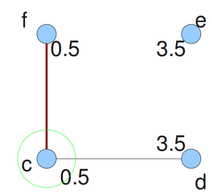

Figure 6: G_π after iteration 5. Edge tight.

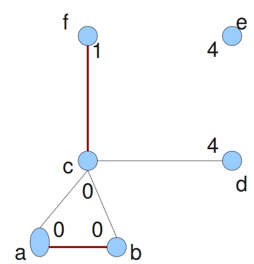

Figure 7: G_π after iteration 6. Deshrink.

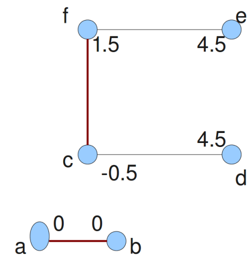

Figure 8: G_π after iteration 7. Edge tight. Some edges become slack and hence disappear from G_π.

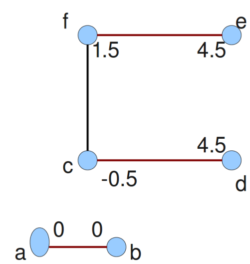

Figure 9: G_π after iteration 8. Augment. Maximum hence STOP.

1.4 Proof

Lemma 3 *The algorithm maintains the following invariants over the iterations*

- π *is dual feasible*
- Ω *is laminar*
- *for each* $U \in \Omega, H_U$ *has a hamiltonian cycle* B_U.

Proof Sketch. We need to check that each iteration maintains the given properties. So we do an analysis of all the considered cases and see that in each case, this property is preserved.
If M is augmented then Ω and π don't change.
If we shrink a blossom B in finding P then we add $U = \cup_{u \in B} S_v$ to Ω. This preserves laminarity since nodes in G' correspond to the maximal sets in Ω. Since we set $\pi(U) = 0$ no dual violation happens. Moreover, B is an odd cycle and hence H_U indeed contains a hamiltonian cycle for the new set U added to Ω.

For the final case we observe that we are not adding any sets to Ω and ϵ is chosen to ensure dual feasiblity. Deshrinking preserves laminarity. $\square$

Claim 4 *If M is a matching in G' then there is a matching N in G_π where number of $N-$exposed nodes is same as $M-$exposed nodes.*

Proof: We can recursively expand the nodes in G' and extend M using the fact that H_U has a Hamiltonian cycle for each $U \in \Omega$. $\square$

Corollary 5 *If M is a perfect matching in G' then it can be extended to perfect matching N in G_π.*

Claim 6 *If the algorithm terminates with a perfect matching then it is an optimal matching.*

Proof: Let π be the feasible dual solution at the end of the algorithm. If M is a perfect matching in G_π then, $\{x(e) = 1$ if $e \in M$ and $x(e) = 0$ otherwise$\}$, is a feasible primal solution and x and π satisfy complementary slackness conditions thus implying that both the solutions are optimal. $\square$

The above claims show that if the algorithm terminates then it outputs an optimum solution. Now we establish that the algorithm indeed terminates.

Lemma 7 (Main Lemma) *The algorithm terminates in $O(|V|^2)$ iterations.*

Each iteration can be implemented in $O(m)$ time with minimal data structures, Thus we have the following theorem due to Edmonds.

Theorem 8 (Edmonds) *There is a an $O(n^2 m)$ time algorithm for the min cost perfect matching problem.*

As a corollary we also obtain

Corollary 9 *The polytope $Q(G)$ described by the inequalities below:*

$$\begin{array}{rcll} x(\delta(v)) & = & 1 & \forall v \in V \\ x(\delta(U)) & \geq & 1 & \forall U \subset V, |U| \geq 3, |U| odd \\ x(e) & \geq & 0 & \forall e \in E \end{array} \tag{2}$$

is the convex hull of the perfect matchings in G.

Proof: The algorithm shows that for any given weights $w : E \rightarrow \mathcal{R}^+$, the linear program $\min\{w \cdot x \mid x \in Q(G)\}$ has an integral optimum solution whenever $Q(G) \neq \emptyset$. Since $w \geq 0$ can be assumed w.l.o.g. so $Q(G)$ is an integral polyhedron. $\square$

Now we finish the proof of the key lemma on termination. First we observe the following.

Proposition 10 *In any iteration, the number of $M-$exposed nodes in G' does not increase which implies that the induced matching in G cannot decrease in size.*

Proof: It is easy to see that steps (1) - (4) of the algorithm don't increase the number of $M-$exposed nodes in G'. The only non-trivial case is step (5) in which the dual value is changed. Now in this step, recall the Edmonds Gallai decomposition and notice that $A(G')$ is matched only to $D(G')$. The dual update in this step leaves any edge uv between $D(G')$ and $A(G')$ tight and so all the $M-$edges remain tight in this step. $\square$

Claim 11 (Main claim) *If a set U is added to Ω in some iteration ("shrinking") then it is removed from Ω ("deshrinking") only after the matching size has increased.*

Proof: When U is added to Ω, it corresponds to the blossom of an $M-$flower where M is the current matching in G'. Let v be the node in G' corresponding to U after it is shrunk. If X is the set of $M-$exposed nodes then there is an $X-v$, $M-$alternating, even length path. If there is no matching augmentation then v continues to have an $X-v$, $M-$alternating, even length path or v is swallowed by a larger set U' that is shrunk. In the former case, v cannot be in $A(G')$ and hence cannot be "deshrunk". In the latter case, U' is not deshrunk before a matching augmentation and hence U, which is inside U', cannot be deshrunk before a matching augmentation. $\square$

Claim 12 *Suppose iteration i has a matching augmentation and iteration $j > i$ is the next matching augmentation. Then between i and j, there are at most $|V|$ shrinkings and at most $|V|$ deshrinkings.*

Proof: Let Ω be the laminar family of shrunk sets at the end of iteration i. By the previous claim, before iteration i, we can only deshrink sets in Ω. Hence # of deshrinkings is $\leq |\Omega| - |V|$ since we cannot deshrink singletons. Thus the number of deshrinkings is $\leq |V|$.

Similarly number of of shrinkings is at most $|\Omega'| - |V|$ where Ω' is the laminar family just before iteration j. Again this gives an upper bound of $|V|$. $\square$

Now lets have a look at what else can happen in an iteration other than augmentation, shrinking, and deshrinking. In step (5), an edge $uv \in E \backslash E_\pi$ can become tight and join E_π. One of the following two cases must happen since dual values are increased for nodes in $D(G')$, decreased for nodes in $A(G')$, and unchanged for $C(G')$:

1. $u, v \in D(G')$.
2. $u \in D(G'), v \in C(G')$.

The following two claims take care of these cases.

Claim 13 *If edge uv becomes tight in an iteration and $u, v \in D(G')$ then the next iteration is either a shrinking iteration or an augmentation iteration.*

Proof: Let X be the set of $M-$exposed nodes in G'. If $u, v \in D(G')$ then $G' + uv$ creates an $X - X$ $M-$alternating walk of odd length because $D(G')$ consists of all the vertices reachable by a walk of even length from X. This implies that in the next iteration we have a non-empty walk leading to an augmentation or shrinking. $\square$

Claim 14 *If $u \in D(G')$ and $v \in C(G')$ then in $G'' = G' + uv, v$ is reachable from X by an $M-$alternating path.*

Now we can prove the key lemma. We have a total of $\frac{|V|}{2}$ augmentation iterations. Between consecutive augmentation iterations, there are at most $2|V|$ shrinking and deshrinking iterations. Each other iteration is an edge becoming tight. Case 1 iterations can be charged to shrinking iterations. Total number of case 2 iterations is at most $|V|$ since each such iteration increases by 1, the number of nodes reachable from X with an $M-$alternating path. No other iteration decreases the number of nodes reachable before a matching augmentation. Thus the number of iterations between augmentation is $O(|V|)$. Hence total number of iterations is $O(|V|^2)$.