Advanced Algorithms - Recitation	December 11, 2005
Lecture 7	
Lecturer: Shahar Dobzinski	*Scribe: Ziv Wities*

1 Complementary Slackness

Given a Primal system, we will also look at its Dual system:

Primal P		Dual D	
min $\sum_{j=1}^{n} c_j x_j$, so that:		max $\sum_{i=1}^{m} b_i y_i$, so that:	
$\forall i\ (1 \leq i \leq m)$	$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$	$\forall j\ (1 \leq j \leq n)$	$\sum_{i=1}^{m} a_{ij} y_i \leq c_j$
$\forall j\ (1 \leq j \leq n)$	$x_j \geq 0$	$\forall i\ (1 \leq i \leq m)$	$y_i \geq 0$

Theorem 1 *(Complementary Slackness Principal)*
If x, y are solutions to P, D, respectively,
and $\alpha, \beta \geq 1$ fulfill the following conditions:

1. $\forall j : x_j = 0$ *or* $\frac{c_j}{\alpha} \leq \sum_i a_{ji} y_i \leq c_j$
2. $\forall i : y_i = 0$ *or* $b_i \leq \sum_j a_{ji} x_j \leq \beta b_i$

Then: $\sum_{j=1}^{n} c_j x_j \leq \alpha\beta \sum_{i=1}^{m} b_i y_i$.

Let us look at the LP system for the SET-COVER problem, where U is the set of elements, $S \subseteq P(U)$, and for each set $s \in S$ we mark $c(s)$ to be the cost of set s.

- $\min \sum_{s \in S} c(s) x_s$ so that:
- $\forall e \in U : \sum_{s|e \in s} x_s \geq 1$
- $\forall s \in S : x_s \geq 0$

Let us look at its dual system:

- $\max \sum_{e \in U} y_e$ so that:
- $\forall s \in S : \sum_{e|e \in s} y_e \leq c(s)$
- $\forall e \in U : x_e \geq 0$

We shall see an f-approximation algorithm for SET-COVER.

2 SET-COVER Algorithm

Primal condition: $\alpha = 1$.
Hence $\forall s \in S$, $x_s \neq 0 \Rightarrow \sum_{e|e \in s} y_e = c(s)$
Dual condition: $\beta = f$.
Hence $\forall e \in U$, $y_e \neq 0 \Rightarrow \sum_{s|e \in s} x_s \leq f$

As long as there is an element e which is not yet covered:

- Increase y_e until the set s is tight.
- Add s to the cover ALG.
- Remove from U all the elements covered by s.

Claim 2 *Primal solution ALG is a legal solution.*

Proof: Because for each uncovered $e \in U$, the algorithm adds a set s which covers it to the cover.

Claim 3 *The Dual solution is also legal.*

Proof: Because we always increase a y_e only until the first set $s \in S$ is tight, never beyond $c(s)$, and so the condition $\sum_{e|e \in s} y_e \leq c(s)$ is preserved.

Claim 4 *The Primal condition is met.*

Proof: Because for any s in the cover (i.e. $x_s \neq 0$) we always increased y_e until the inequality was tight, hence turning it into an equation as necessary.

Claim 5 *The Dual condition is also met.*

Proof: Obviously the number of sets any element e is in must be smaller or equal to the frequency f, by the definition of frequency.

Conclusion: From the Complementary Slackness Theorem, we have proved f-approximation.

3 MIN-MULTICUT

We are given a graph G=(V,E), and a group $K = \{(s_1, t_1), ..., (s_k, t_k)\}$, where $s_i, t_i \in V$.
We define $k = |K|$.
For every $e \in E$, we define a weight c_e.
Goal: To find a cut of minimal weight so that each pair of vertices $(s_i, t_i) \in K$ is separated by the cut.
This is a very difficult problem, even under the assumption (which we shall make) that G is a tree.

We will define variables. For each $e \in E$, $0 \leq d_e \leq$ represents whether or not e is in the cut.
For each i, we will mark as P_i the path between s_i and t_i. (Note: P_i is uniquely defined, as G is a tree.)
We will reach primal and dual systems as follows:

Primal P	Dual D
min $\sum_{e\in E} c_e d_e$, so that: $\forall i : (1 \leq i \leq k) \quad \sum_{e\in P_i} d_e \geq 1$ $\forall j : \quad d_e \geq 0$	max $\sum_{i=1}^{k} f_i$, so that: $\forall e : \quad \sum_{i\|e\in P_i} f_i \leq c_e$ $\forall i : \quad f_i \geq 0$

We note that the dual system is precisely the LP program for finding maximal flow in a tree graph.

4 MIN-MULTICUT Algorithm

Primal condition: $\alpha = 1$.
Hence $\forall e \in E$, $d_e \neq 0 \Rightarrow \sum_{i|e\in P_i} f_i = c_e$
In other words, each edge in the multicut is at maximal flow.
Dual condition: $\beta = 2$.
Hence $\forall i$, $f_i \neq 0 \Rightarrow \sum_{e\in P_i} d_e \leq 2$
In other words, in a path with nonzero flow, at most 2 edges may be added to the cut.

Initilization: We begin with a group $D = \emptyset$, and we assume $\forall i : f_i = 0$.
We choose a vertex v, and mark it as our root. We define the depth of the root to be 0.
For any vertex u, we define u's depth to be u's distance from the root v.
We will define the *lca* (least common ancestor) of two vertices s_i, t_i to be the vertex with the smallest depth value on the path P_i, and we will mark this vertex $lca(s_i, t_i)$.

Flow: For every vertex v in the graph, in order of nonascending depth (i.e. deepest vertices first):
If there exist an i so that $v = lca(s_i, t_i)$, then we greedily increase the flow from s_i to t_i (i.e., raise the flow as much as the path's capacity will allow).
We now add to D all edges which have reached their maximal flow during this iteration.
We will mark the final result as $D = \{e_1, ...e_l\}$.

Backwards Deletion: For every $1 \leq j \leq l$:
If $D \backslash \{e_j\}$ is a multicut, remove e_j from D.

Claim 6 *The primal solution is a legal one.*

Proof: On every path P_i, there is a sated edge which is now in the cut, else we'd have kept increasing the flow on that path.

Claim 7 *The dual solution is a legal one.*

Proof: We never exceeded the capacity of any edges.

Claim 8 *The primal condition is met.*

Proof: We chose only the sated edges (i.e. the ones meeting this condition) for the cut.

Claim 9 *The dual condition is met.*

Proof: next tirgul.

Conclusion: By the Complementary Slackness Theorem, we have proved 2-approximation.