6.893 Approximability of Optimization Problems	September 22, 1999
Lecture 5	
Lecturer: Michel Goemans	*Scribe: Matthew Frank*

1 The Primal-Dual Method

Let's look at an example we've seen before:

Minimum Weight Vertex Cover

Given:	$G = (V, E)$ and,
	weights $w_1, \ldots, w_n$ on V
Task:	Find $\min_{S \subseteq V} \sum_{i \in S} w_i$
subject to:	$\forall (i,j) \in E, i \in S$ or $j \in S$.

In Lecture 4 we saw an algorithm [Hochbaum] that converted Vertex Cover to an Integer Programming (IP) problem, relaxed the IP to a linear programming (LP) problem, and then used rounding to convert the LP solution back to integers:

LP formulation:

Variables:	$x_1 \ldots x_n$
	$x_i = 1 \Rightarrow \text{vertex}_i \in S$
Objective:	$\min \sum_{i=1}^{n} w_i x_i$
Constraints:	$x_i + x_j \geq 1 \qquad \forall (i,j) \in E$
	$0 \leq x_i \leq 1$

To round, we take the optimal LP solution, x^*, and choose as our solution to the vertex cover

$$\text{VC} = \{i : x_i^* \geq \frac{1}{2}\}$$

But this approach has two problems. First, solving the LP is expensive, so we'd like to find something cheaper. Second, for some algorithms rounding may not work as a method for converting the LP optimum back into an integer solution that has provably good approximation bounds. The Primal-Dual method is a technique that can be used across a broad class of problems to quickly find an integer solution with provable approximation bounds.

2 Duality of Linear Programming

For any LP there is a Dual LP (see *e.g.* introductory books by Chvátal 1983 or Bertsimas and Tsitsiklis 1997).

Primal			Dual	
	$\min_x c^T x$	$\longleftrightarrow$		$\max_y b^T y$
subject to:	$Ax \geq b$		subject to:	$A^T y \leq c$
	$x \geq 0$			$y \geq 0$

As an example, for the Vertex Cover problem, the Dual formulation is:

$$
\begin{array}{ll}
\text{Variables:} & y_{ij} \text{ corresponding to } (i,j) \in E \\
\text{Objective:} & \max \sum_{(i,j)\in E} y_{ij} \\
\text{Constraints:} & \sum_{j:(i,j)\in E} y_{ij} \le w_i \quad \forall i \\
& y_{ij} \ge 0
\end{array}
$$

Theorem 1 (Weak Duality) *Given feasible solutions x and y to the primal and dual problems, then $c^T x \geq b^T y$. In particular,*

$$\min_x c^T x \geq \max_y b^T y$$

.

Proof Given any feasible solution x to the Primal and y to the Dual, then $x \geq 0$, $Ax - b \geq 0$ and $y \geq 0$ so

$$y^T(Ax - b) = y^T Ax - y^T b \geq 0$$

Likewise, $y \geq 0$, $c - A^T y \geq 0$ and $x \geq 0$ so

$$c^T x - y^T Ax \geq 0$$

Finally, we add these two results to get:

$$c^T x - y^T b \geq 0$$

■

More specifically, for the Vertex Cover problem,

$$\sum_{i=1}^n w_i x_i \quad \geq \quad \sum_{i=1}^n \left(\sum_{j:(i,j)\in E} y_{ij} \right) x_i \quad = \quad \sum_{(i,j)\in E} (x_i + x_j) y_{ij} \quad \underset{(x_i + x_j \geq 1)}{\geq} \quad \sum_{(i,j)\in E} y_{ij}$$

In fact a much stronger result is true:

Theorem 2 (Strong Duality) *If a linear programming problem has an optimal solution x^*, then its dual has an optimal solution y^*, and $c^T x^* = b^T y^*$.*

But the proof is more difficult than for Weak Duality.

3 Primal-Dual Algorithm for Vertex Cover [Bar-Yehuda, Even]

Instead of solving the LP or its Dual optimally we will do something a little different.
Construct

1. A vertex cover (*i.e.* an integeral solution to the LP).
2. A feasible solution to the Dual. (But not necessarily an optimal solution).

such that

$$\underbrace{\sum_i^n w_i x_i}_{\text{weight of VC}} \quad \leq 2 \sum_{(i,j)\in E} y_{ij} \ \leq \ 2\text{LP}_{(\text{Dual})} \ = \ 2\text{LP} \ \leq 2\text{OPT}$$

The Algorithm

Initially let $y = 0, x = 0$
This satisfies $y_{ij} \geq 0$ and $\sum y_{ij} \leq w_i$
but not $x_i + x_j \geq 1$ so it's not a vertex cover
While x is not a vertex cover
choose $(i,j) \in E$ s.t. $x_i + x_j = 0$.
increase y_{ij} until either $\sum_{k:(i,k)\in E} y_{ik} = w_i$ or
$\sum_{k:(k,j)\in E} y_{kj} = w_j$
i.e. increase y_{ij} until one constraint becomes tight
$x_i = 1$ if constraint is tight for i
else $x_j = 1$

Claim 3 *x is a vertex cover. (Because we keep looping until it is).*

Claim 4 *y is dual feasible. (Correct by construction).*

Claim 5 $\sum_i^n w_i x_i \leq 2OPT$

Proof At each step we kept the constraints tight, so $\forall i\ w_i = \sum_{j:(i,j)\in E} y_{ij}$.

$$\Rightarrow \sum_{i=1}^n w_i x_i = \sum_{i=1}^n \sum_{j:(i,j)\in E} x_i y_{ij}$$

But $\forall i\ x_i \in \{0,1\}$ so

$$\sum_{i=1}^n \sum_{j:(i,j)\in E} x_i y_{ij} = \sum_{\{i:x_i=1\}} \sum_{j:(i,j)\in E} y_{ij} \leq \sum_{(i,j)\in E} 2y_{ij} \leq 2\max_y \sum_{(i,j)\in E} y_{ij} = 2\min_x \sum_{i=1}^n w_i x_i \leq 2\text{OPT}$$

■

THE BIG IDEA: We used the LP formulation to construct a Dual problem and then cleverly constructed a feasible solution to the Dual. The properties of the dual (*e.g.* complementary slackness), enabled us to prove bounds on the approximation algorithm.

4 The Generalized Steiner Tree Problem

This is a harder problem than Vertex Cover.

We are given a graph $G = (V,E)$ with edge weights and $s_i, t_i \in V$ such that s_i and t_i need to communicate. We'd like to find a minimum cost subgraph such that all the $< s_i, t_i >$ pairs are connected.

Problem Formulation:

Given: $G = (V,E)$ w.l.o.g. fully connected,
weights $w_e \geq 0,\ \forall e \in E$
$< s_1, t_1 >, < s_2, t_2 >, \ldots, < s_k, t_k >$
paris of communicating edges
Find: A min-cost subgraph $H \subseteq G$ such that
$\forall i\ s_i, t_i$ are connected in H

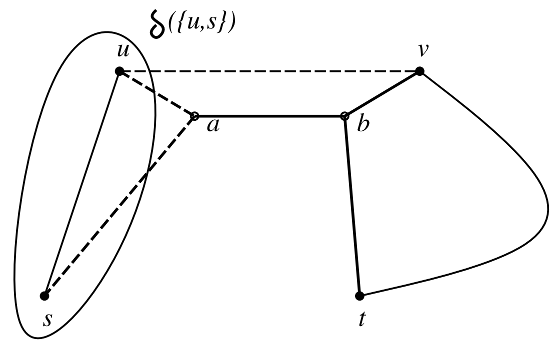

Figure 1: An example of the Steiner Tree problem with communicating pairs $< u, v >$ and $< s, t >$. The edges in the cut-set $\delta(\{u, s\})$ are shown as dashed lines.

This problem is NP Hard. A 2-approximable algorithm was first given by Agrawal, Klein and Ravi in '91 and then by Goemans and Williamson in '92.

We will proceed in the same manner as for Vertex Cover: formulate the problem as an integer program, find the dual, and then find a clever method to construct a feasible Dual solution. Then we will use the properties of the Dual to prove bounds on the approximation.

We will simplify the formulation with a couple of definitions:

Definition 6 *$\lambda(S) = 1$ if $S \subseteq V$ and $\overline{S}$ form a cut that seperate one of the $< s_i, t_i >$ pairs. More formally:*

$$\lambda : 2^V \to \{0, 1\}$$
$$\lambda(S) = 1 \Leftrightarrow \exists i : |S \cap \{s_i, t_i\}| = 1$$

Definition 7 *Given $G = (V, E)$ and $S \subseteq V$, then $\delta(S) = \{(i, j) \in E : i \in S, j \notin S\}$. That is, $\delta(S)$ is the set of edges that cross the cut boundary.*

Integer Programming Formulation

$$\begin{array}{ll} \text{Variables:} & x_e = \begin{cases} 1 & \text{if } e \in H \\ 0 & \text{if } e \notin H \end{cases} \quad \forall e \in E \\ \text{Objective:} & \min_x \sum_{e \in E} w_e x_e \\ \text{Constraints:} & \sum_{e \in \delta(S)} x_e \geq 1 \;\; \forall S : \lambda(S) = 1 \\ & x_e = \{0, 1\} \end{array}$$

We will take the LP relaxation $x_e \geq 0$, so $\text{OPT}_{\text{Steiner}} \geq \min_{x \geq 0} \sum_{e \in E} w_e x_e$

Dual Formulation

$$\begin{array}{ll} \text{Objective:} & \max_y \sum_{S \subseteq V : \lambda(S)=1} y_s \\ \text{Constraints:} & \sum_{S : e \in \delta(S)} y_s \leq w_e \quad \forall e \in E \\ & y_s \geq 0 \end{array}$$

Note, that Duality gives us that $\min_{x \geq 0} \sum_{e \in E} w_e x_e \geq \max_y \sum_{S \subseteq V : \lambda(S)=1} y_s$ so

$$\mathrm{OPT}_{\mathrm{Steiner}} \geq \max_y \sum_{S \subseteq V : \lambda(S)=1} y_s$$

Our goal is to simultaneously construct a feasible solution, H, to the primal problem and a feasible solution, y, to the dual so that $\sum_{e \in H} w_e \leq 2 \sum_{S \subseteq V} y_s$.

The Algorithm

Let F be the current solution. (Eventually F will be the solution H)

$F \leftarrow \emptyset$ (no edges)

index $k \leftarrow 0$

$y \leftarrow 0$ (Note that there are exponential number of ys, but only $2n$ of them will ever be nonzero)

$\mathcal{C}$ be the connected components of F

($\mathcal{C} \leftarrow \{\{V\}\}$ intially)

$\forall i \in V \quad d(i) \leftarrow 0$ ($d(i)$ will represent $\sum_{S : i \in S} y_s$)

While F is not feasible ($i.e.$ $\exists C \in \mathcal{C} : \lambda(C) = 1$)

(Informally, $\forall C \in \mathcal{C} : \lambda(C) = 1$ we are going to keep increasing y_s until some edge constraint becomes tight, and then add that edge).

Find an edge $(i,j) \in E$ with $i \in C_p \in \mathcal{C}$ and $j \in C_q \in \mathcal{C}$ and $p \neq q$

which minimizes $\epsilon = \dfrac{w_e - (d(i) + d(j))}{\lambda(C_p) + \lambda(C_q)}$

Note that $d(i) + d(j) = \sum_{S : (i,j) \in \delta(S)} y_s$

$k \leftarrow k + 1$

$e_k \leftarrow (i,j)$

$F \leftarrow F \cup e_k$

$\forall i \in C \in \mathcal{C} : \lambda(C) = 1$ increase $d(i)$ by ϵ

$\mathcal{C} \leftarrow \mathcal{C} - \{C_p, C_q\} + \{C_p \cup C_q\}$

Last Step: remove irrelevant edges

For $j \in k \ldots 1$

if $F \setminus \{e_j\}$ is still feasible

then $F \leftarrow F \setminus \{e_j\}$

Output: Let $H \leftarrow F$ and output (H, y)

Explanation of the Algorithm

The basic idea is that we start with each vertex in its own connected component. Then we grow little balls around the components which contain disconnected communicators. When two balls merge, we merge their components into a single connected component, add the edge between them to the solution, and then continue.

Figure 2 shows a run of the algorithm. The iterations go as follows:

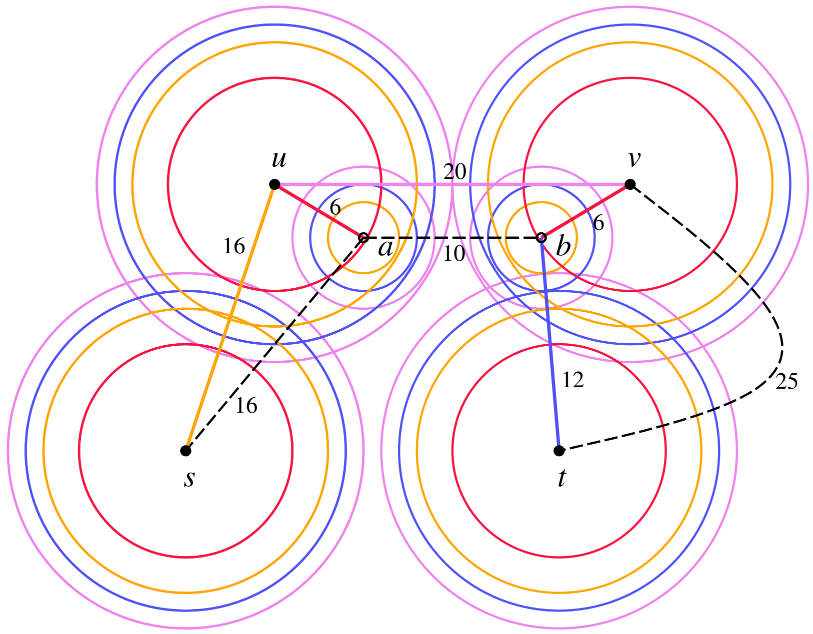

Figure 2: How the algorithm runs on our example problem. The edges are added in the order $(u,a),(v,b),(u,s),(b,t),(u,v)$ the final step removes the edge (u,a). The algorithm delivers a solution with weight 54. The minimal solution is $\{(u,a),(v,b),(s,a),(b,t),(a,b)\}$ with a weight of 50.

iteration	ϵ	e_k	new $\mathcal{C}$
initial			$\{\{s\},\{t\},\{u\},\{v\}\}$
1	6	(u,a)	$\{\{s\},\{t\},\{a,u\},\{v\}\}$
2	0	(v,b)	$\{\{s\},\{t\},\{a,u\},\{b,v\}\}$
3	2	(u,s)	$\{\{t\},\{a,s,u\},\{b,v\}\}$
4	1	(b,t)	$\{\{a,s,u\},\{b,t,v\}\}$
5	1	(u,v)	$\{\{a,b,s,t,u,v\}\}$

Claim 8 *Given a Steiner Tree Problem ST, the above algorithm gives a feasible solution H and a feasible LP dual solution y such that:*

$$\min_{x\geq 0}\sum_{e\in E} w_e x_e = \max_y \sum_{S\subseteq V} y_s \leq OPT_{ST} \leq \sum_{e\in H} w_e \leq 2\sum_{S\subseteq V} y_s \leq 2\max_y \sum_{S\subseteq V} y_s \leq 2OPT_{ST}$$

(i.e. *This is a 2-approximable algorithm for the Generalized Steiner Tree Problem.)*

Proof of feasibility: In each step of the **while** loop we only consider edges for addition if they are not already in a connected component. Therefore the resulting set of edges, H, is acyclic. So, between any pair of nodes $< s_i, t_i >$ that need to communicate there is a unique path. So the **last step** won't remove any of the edges on that path. Therefore, H is primal feasible.

y is dual feasible because once a constraint becomes tight we remove that edge from further consideration. ■

We still need to show that $\sum_{e\in H} w_e \leq 2\sum_{S\subseteq V} y_s$. We will do this in two steps.

Definition 9 $deg_H(S) = |H\cap\delta(S)|$. *That is,* $deg_H(S)$ *is the number of edges crossing the cut* $(S,\overline{S})$.

Lemma 10 *Given the output, (H, y), of the algorithm then* $\sum_{e \in H} w_e = \sum_{S \subseteq V} deg_H(S) y_s$.

Proof During every step of the algorithm, when we added an edge e we ensured that the corresponding constraint got tight.

$$\Rightarrow w_e = \sum_{S: e \in \delta(S)} y_s$$
$$\Rightarrow \sum_{e \in H} w_e = \sum_{e \in H} \Big(\sum_{S: e \in \delta(S)} y_s \Big) = \sum_{S \subseteq V} \sum_{e \in \delta(S) \cap H} y_s = \sum_{S \subseteq V} \deg_H(S) y_s$$

■

Lemma 11

$$\sum_{S \subseteq V} deg_H(S) y_s \leq 2 \sum_{S \subseteq V} y_s$$

Sketch of Proof We're going to do an induction over the while loop of the algorithm. At every step y changes, but H is fixed. We're going to show that $\sum_{S \subseteq V} \deg_H(S) y_s \leq 2 \sum_{S \subseteq V} y_s$ is invariant throughout the running of the algorithm.

At each step of the algorithm we show that the increase in the left hand side of the inequality is less than the increase in the right hand side. That is:

$$\epsilon \sum_{S \in \mathcal{C}: \lambda(S)=1} \deg_H(S) \ \leq \ 2\epsilon |\{C \in \mathcal{C} : \lambda(C) = 1\}|$$

The ϵs cancel so we are left with

$$\frac{\sum_{S \in \mathcal{C}: \lambda(S)=1} \deg_H(S)}{|\{C \in \mathcal{C} : \lambda(C) = 1\}|} \leq 2$$

That is: the average degree of the active sets is at most 2. The intuition for why this is so, is that we are essentially building a tree. The following lemma, shows that every leaf (node of degree 1) of the tree is an active set. ■

Lemma 12 *During the running of the algorithm, if $\lambda(C) = 0$ for some $C \in \mathcal{C}$ then $deg_H(S) \neq 1$.*

Proof of Lemma 12 Suppose $\deg_H(S) = 1$. Then there is a unique edge $e \in H$ and $e \in \delta(C)$. Since H is acyclic there must be a pair of vertices $< s_i, t_i >$ that need to communicate using a path that includes e. But this path crosses the cut $(C, \overline{C})$, exactly once, so exactly one of s_i, t_i must lie in C. Therefore $\lambda(C) \neq 0$. ■

5 Summary

The basic Primal-Dual method consists of constructing an IP for a problem, relaxing it to an LP, and finding the Dual of the LP. Then we use the dual prices to construct a greedy algorithm that has provable approximation bounds.