ORIE 6300 Mathematical Programming I September 4, 2008

Lecture 3

Lecturer: David P. Williamson *Scribe: Shanshan Zhang*

In the last lecture, we studied the standard form LP: $\min(c^T x, Ax = b, x \geqslant 0)$. We also considered two methods, by reduction and by bounding, of taking the dual.

In this lecture, we will study some combinatorial applications of duality. Let's start with the *covering problem*. In a *covering problem*, you are given sets, $S_1, \dots, S_m$, all subsets of $\{1, \dots, n\}$ for some n. The covering problem is to choose $X \subseteq \{1, \dots, n\}$ that hits the sets $S_1, \dots, S_m (X \cap S_j \neq \emptyset, \forall j)$ such that minimizes $|X|$ is minimized. (It is also called the hitting set problem.) We can formulate this covering problem as an **integer** program, i.e., a linear program where the variables are required to be integer. We will have variables x_i associated with element $i = 1, \dots, n$. We think of setting $x_i = 1$ if $i \in X$, otherwise $x_i = 0$. Thus, we get the following integer program:

$$\begin{array}{rcll} \min \sum_{i=1}^{n} x_i & & & \\ x_i & \geq & 0 \text{ for each } i = 1, \dots, n & \qquad \text{(integer covering problem)} \\ \sum_{i \in S_j} x_i & \geq & 1 \text{ for each } j = 1, \dots, m & \\ x_i \in \{0, 1\} & & & \end{array}$$

If we delete the integer constraints in the above problem, we get the *fractional covering problem.* Note that in an optimum solution we will have $x_i \leq 1$ for each i (as otherwise replacing x_i by $\min(1, x_i)$ gives a better solution with smaller objective function value).

Instead of the integer covering problem we will consider the *fractional covering problem.*

Note that a fractional covering problem is an LP with a 0-1 matrix A of the form $\min(\mathbf{1}x : Ax \geq \mathbf{1})$, where $\mathbf{1}$ denotes the vector with all coordinates 1. Any LP of this form is a fractional covering problem: the elements correspond to the columns of A. The rows define the sets: set S_j contains all elements where row j has a 1.

We take the linear programming dual of this problem using the method of taking duals of linear programs of general form that was discussed at the end of last lecture. The variables of the dual correspond to the rows of the primal matrix. In our case variables correspond to the sets.

$$\begin{array}{rcll} \max \sum_{j=1}^{m} y_j & & & \\ y_j & \geq & 0 \text{ for each } j = 1, \dots, m & \qquad \text{(fractional packing problem)} \\ \sum_{j : i \in S_j} y_j & \leq & 1 \text{ for each } i = 1, \dots, n & \end{array}$$

To understand the meaning of this linear program, we will first consider the integer solutions to this dual LP. Note that no variables can be above 1 due to the constraints. So the integer version of this linear program selects a maximum number of sets (i.e., the sets with S_j that have $y_j = 1$), subject to constraints. The constraints require that the sets selected must be disjoint: for each element i the number of sets selected that contains i is at most 1. This problem is traditionally called the integer set packing problem (we want to pack as many disjoint sets as possible). The linear program is then named the fractional set packing problem.

Lemma 1 *The fractional set packing and fractional set covering problems are duals of each other.*

Next, we will derive a famous theorem from networks, the max flow-min cut theorem, from LP duality. The maximum flow problem is defined by a directed graph $G = (V, A)$, and two distinguished nodes, s (the *source*) and t (the *sink*). The graph has directed arcs, (u, v), which go from a vertex u to another vertex v. Note that the arcs are directed, i.e., an arc (u, v) going from u to v is different from an arc (v, u) going from v to u. The goal of the problem is to send as much flow as possible from the source to the sink such that each arc carries at most one unit of flow, and for every node of the graph other that the source and sink, the total amount of flow entering the node is equal to the amount leaving the node.

We formulate the maximum flow problem as a linear program where the variables correspond to paths from s to t. For each such path, P, we will have a variable x_P. Note that this is an unusual formulation, as there can be exponentially many paths in a graph; our LP can have very many variables. For now, do not worry about this. We will only need the duality theorem, which is true no matter how many variables we have.

For the linear programming formulation, we will use P to denote paths from s to t; we let A denote the set of all edges, and let $\mathcal{P}$ denote the set of all paths from s to t. The variable x_P states how much flow is sent from the source to the sink on path P. The constraints express that for each arc $(u, v) \in A$, at most one unit of flow can be sent through arc (u, v):

$$\begin{aligned} \max \sum_{P \in \mathcal{P}} x_P & \\ x_P &\geq 0 \text{ for each path } P \in \mathcal{P} \\ \sum_{P:(u,v) \in P} x_P &\leq 1 \text{ for each arc } (u, v) \in A \end{aligned}$$

Note that the flow problem is exactly a fractional set packing problem, where the elements are edges of the graph, and the sets are the paths from s to t. In the integer packing problem, we want to find as many disjoint paths from s to t as possible.

From the above general discussion of packing and covering problems, we know that the dual of this packing problem is a covering problem with the paths as sets and the edges as elements:

$$\begin{aligned} \min \sum_{(u,v) \in E} z_{uv} & \\ z_{uv} &\geq 0 \text{ for each arc } (u, v) \in A \\ \sum_{(u,v) \in P} z_{uv} &\geq 1 \text{ for each path } P \in \mathcal{P} \end{aligned}$$

For the maximum flow problem defined above, we define an *s-t cut* as a set S of nodes that contains s and does not contain t. An arc (u, v) is *in the cut* if it leaves S, i.e., u is in S and v is not. Note that every s-t cut gives an integer solution to the dual of the maximum flow problem by setting $z_{uv} = 1$ if e leaves set S, and 0 otherwise. All paths from s to t must leave set S at some point, hence they must contain at least one arc (u, v) with $z_{uv} = 1$. (Note that a path can leave S more than once, assuming it entered S again in between the two). Hence, this dual variable assignment is dual-feasible.

This implies that each cut defines an integer solution to the dual LP, and the value of this solution is the number of edges leaving the cut. For a cut S, let $n(S)$ denote the number of edges leaving the cut. The minimum *s*-*t* cut problem is to find the cut S with $n(S)$ as small as possible. We saw that cuts are integer solutions to this LP, so the LP minimum, min $\sum_{(u,v)\in A} z_{uv}$, is at most the size of the minimum cut.

Lemma 2 *The maximum flow value is at most the minimum cut value.*

Proof: This is easy to see directly, but also follows from weak duality: all cut values are values of dual feasible solutions, and so are upper bounds on the maximum flow value. Hence, the maximum flow value is upper bounded by the minimum cut value. □

We will prove that the minimum *s*-*t* cut is equal to optimal dual solution. We will prove the equality by showing that for any optimal solution to the dual problem, there is a cut S such that $n(S)$ is less than or equal to the value of the solution. By strong duality, we know that the maximum flow is equal to optimal dual solution. Therefore, we prove that the maximum flow is equal to the minimum *s*-*t* cut.

To do this we need some observations and definitions. Let z^* be an optimal dual solution in what follows. Let $cost(s,w)$ for a node v mean the minimum, over all s to w paths P, of the sum of the optimal dual variable values for the edges on that path:

$$cost(s,w) = min_{s\rightarrow w \text{ path P}} \sum_{(u,v)\in P} z^*_{uv}.$$

We will consider the following sets $S_\rho = \{v : cost(s,v) \leq \rho\}$. The following observations will be useful.

- The constraints in the linear program require that $cost(s,t) \geq 1$.
- From this, we get that, for each $\rho < 1$, we have that $t \not\in S_\rho$.
- For each $\rho \geq 0$, we have that $s \in S_\rho$. This is true essentially by definition. The empty path from s to s has no edges, so the sum of z values along the edges is an empty sum, and hence has value 0.

So far, we see that S_ρ defines an *s*-*t* cut for each $0 \leq \rho < 1$. In addition, we will need the following inequality, which is often referred to as the triangle inequality:

Lemma 3 *For each arc $(u,v) \in A$, we have that $cost(s,v) \leq cost(s,u) + z^*_{uv}$,*

Proof: The inequality follows from the fact the path from s to w consisting of the minimum-cost path from s to v followed by arc (u,v) has cost exactly $cost(s,v) + z^*_{uv}$. The $cost(s,w)$ is the minimum cost of a path from s to w and so is no greater than this quantity. □

We want to show that there exists a value of ρ such that the corresponding cut S_ρ is sufficiently small. What we need to prove the theorem is to show exhibit a cut of value at most $\sum_{(u,v)\in A} z^*_{uv}$. We will do this by selecting one of the cuts S_ρ at random, by selecting ρ uniformly at random from the interval $[0,1)$. The value of this cut is a random variable, and we will show that its expected value is at most $\sum_{(u,v)\in A} z^*_{uv}$, and hence there must exist at least one such cut that achieves this bound.

We will continue the discussion next class.