Mathematical Programming **Lecture 3**
OR 630 Fall 2006 **August 31, 2006**
adapted from notes written by Éva Tardos

Recall the following definitions from last class. In a *covering problem*, you are given sets, $S_1, \ldots, S_m$, all subsets of $\{1, \ldots, n\}$ for some n. The covering problem is to select as few elements as possible so that at least one element is selected in each set. 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 $i = 1, \ldots, n$, and the constraints will be

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

Note that in an optimum solutions we will have $x_i \leq 1$ for each i (as otherwise replacing x_i by $\min(1, x_i)$ gives a better solution. So an integer solution has values 0 and 1. Think of the elements i with $x_i = 1$ as selected, and those with $x_i = 0$ as not selected. (This is also sometimes called the hitting set problem.)

Instead of the integer covering problem we will consider here the *fractional covering problem*, i.e., the LP given above.

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, and 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 to take 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.

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

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. For now, the maximum flow problem will be formulated as follows. (Note that this is not the traditional formulation, but is equivalent to it. Later in the semester we will show that they are equivalent.)

The maximum flow problem is defined by a directed graph G, and two distinguished nodes, s and t. The graph has directed edges, $e = vw$, which connect from a vertex v to another vertex w. Note that the edges are directed, i.e., an edge $e = vw$ going from v to w is different from an edge $e = wv$ from w to v.

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 and use e to denote edges of the graph; we let E denote the set of all edges, and let $\mathcal{P}$ denote the set of all paths from s to t. The constraints express that for each edge e, at most one path can be selected using edge e:

$$\begin{aligned} \max \sum_{P \in \mathcal{P}} x_P & \\ x_P &\geq 0 \text{ for each path } P \in \mathcal{P} \\ \sum_{P : e \in P} x_P &\leq 1 \text{ for each edge } e \in E \end{aligned}$$

Note that the flow problem is exactly a 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:

$$\begin{aligned} \min \sum_{e \in E} z_e & \\ z_e &\geq 0 \text{ for each edge } e \in E \\ \sum_{e \in P} z_e &\geq 1 \text{ for each path } P \in \mathcal{P} \end{aligned}$$

For the maximum flow problem defined above, we define a *cut* as a set S of nodes that contains s and does not contain t. The edges in the cut are those edges that leave S, i.e., edges $e = vw$ where v is in S and w is not. Note that every cut gives an integer solution to the dual of the maximum flow problem by setting $z_e = 1$ 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 edge with $z_e = 1$. (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 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, $\min \sum_{e \in E} z_e$, 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 use LP duality to prove that the maximum flow is equal to the minimum cut. To do this we need some observations and definitions. Let z be the optimal dual vector in what follows.
Let $cost(s, v)$ for a node v mean the minimum, over all s to v paths P, of the sum of the optimal dual variable values for the edges on that path:

$$cost(s, v) = min_{s \to v \text{ path P}} \sum_{e \in P} z_e.$$

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 $t \notin S_\rho$.
- 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 a 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 edge $e = vw$, we have that $cost(s, w) \leq cost(s, v) + z_e$,*

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 edge e has cost exactly $cost(s, v) + z_e$. The $cost(s, w)$ is the minimum cost of a path from s to w (and is actually smaller than the right-hand side if there are shorter paths to w than this one.) ■

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 $\sum_{e \in E} z_e$. 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 $\sum_{e \in E} z_e$, and hence there must exist one such cut that achieves this bound.

We want to compute the expected number of edges leaving the cut S_ρ. To compute this expectation, first consider the probability that a given directed edge $e = vw$ leaves the randomly selected cut S_ρ. Edge $e = vw$ leaves S_ρ if and only if v is in S_ρ and w is not in S_ρ. This happens if and only if $cost(s, v) \leq \rho < cost(s, w)$. If $cost(s, v) \geq cost(s, w)$, then the edge $e = vw$ does not leave any of the sets S_ρ. If $cost(s, v) < cost(s, w)$, then the probability that edge e leaves the randomly selected S_ρ is exactly $cost(s, w) - cost(s, v)$. Note that, by the triangle inequality, we get that $cost(s, w) - cost(s, v) \leq z_e$; hence the probability that edge e leaves the selected set is at most z_e. Now, we compute the expected number of edges leaving the set. We can do this by introducting an indicator variable $I(e, \rho)$, which is equal to 1 if e leaves S_ρ, and is 0 otherwise. Then, we have that

$$n(S_\rho) = \sum_{e \in E} I(e, \rho).$$

By the linearity of expectation (that is, the expectation of a sum is the sum of the expectations), the expected value of $n(S_\rho)$ is equal to the sum, over all edges $e \in E$, of the expectation of $I(e, \rho)$. Since $I(e, \rho)$ is a 0-1 random variable, its expectation is equal to the probability that this variables is equal to 1; that is, the probability that edge e leaves the cut S_ρ, which is exactly what we bounded above.

$$\mathbf{E}(n(S_\rho)) \;\; = \;\; \sum_{e \in E} \mathbf{Pr}(e \text{ leaves set } S_\rho) \leq \sum_e z_e.$$

Now, we just put the pieces together as we expected, to prove the max flow-min cut theorem. So far, we know that the max flow is at most the min cut. From LP duality, we also know that the max flow value is the same as the value of the optimum LP dual value: $\sum_e z_e$. Then, to show that max flow = min cut, we need to find a cut, S, such that $n(S) \leq \sum_e z_e$.

By the standard definition of the expectation, if the expected number of edges leaving the randomly selected cut S_ρ is at most $\sum_e z_e$ then at least one of the cuts that contribute to this expectation must have value at most the expectation, and hence at most $\sum_e z_e$. This is the cut we needed, and so the proof is complete.

Recall that cuts induce integer solutions to the dual LP; so we now proved that the dual LP always has integer optimum solutions. In fact, the primal linear program, the maximum flow problem, also has integer optimum solutions. We will see this later when we talk more about graphs and flows. I do not know of a direct LP based proof of the integrality of the primal.

Our next topic is to understand the relation between a geometric and algebraic interpretation of linear programming. We will also use geometry to give a math proof of the strong duality theorem. Section 17 of Chvátal gives a short introduction to this topic.

We will consider the feasible region of a set of inequalities. Given a set of inequalities we define the feasible region as $P = \{x : Ax \leq b\}$. Today we will consider two equivalent definition of a vertex of this region. One definition is geometric, one is algebraic, and we will show that they are equivalent.

Lemma 4 *For a point $\hat{x} \in \Re^n$, we define $A^=$ to consist of the rows of A where $a_i\hat{x} = b_i$, and let $b^=$ denote the corresponding coordinates of b. The following two conditions are equivalent:*
(a) For each direction $y \in \Re^n$, if $\hat{x} + y$ and $\hat{x} - y$ are both in P, then $y = 0$.
(b) The matrix $A^=$ has rank n.
A point $\hat{x}$ satisfying these conditions is called a vertex of P.

We will prove this lemma next class.