

## Lecture 6: The Dual LP and Approximation Algorithms

Every linear program has a *dual* linear program. For this discussion, let's consider the following minimization LP.  $A$  is an  $n \times m$  matrix.

$$\begin{aligned} \min \quad & c \cdot x \\ \text{subject to} \quad & Ax \geq b \\ & x \geq 0 \end{aligned} \tag{1}$$

Then the dual to the above program is the following. In the following,  $A'$  is the transpose of  $A$ .

$$\begin{aligned} \max \quad & b \cdot y \\ \text{subject to} \quad & A'y \leq c \\ & y \geq 0 \end{aligned} \tag{2}$$

The original program is called the *primal* linear program. In the dual, there is a variable  $y_i$  corresponding to every (non-trivial) constraint in the primal. Furthermore, every variable in the primal leads to a constraint in the dual. The following fact is called *weak duality*.

**Fact 1** (Weak Duality). *Let  $(x, y)$  be two feasible solutions to (1) and (2), respectively. Then  $b \cdot y \leq c \cdot x$ .*

*Proof.* The proof is the following where the inequalities follow from feasibility

$$b \cdot y = \sum_{i=1}^n y_i b_i \leq \sum_{i=1}^n y_i \sum_{j=1}^m A_{ij} x_j = \sum_{j=1}^m x_j \sum_{i=1}^n A_{ij} y_i \leq \sum_{j=1}^m x_j c_j = c \cdot x \quad \square$$

Thus, *any* feasible solution to the dual LP implies a lower bound on the primal optimum, which in turn is a lower bound on the optimum. The following theorem, called *strong duality*, tells us that the best such possible lower bound is in fact the value of the primal LP. That is, the dual optimum equals the primal optimum. This fact is one of the most important theorems of linear optimization. We'll not cover the proof in this course, but it's not difficult and is given in the linear programming reference mentioned in the course webpage.

**Theorem 1.** *The optimum value of (1) equals the optimum value of (2).*

The following corollary is going to be useful. Let  $(x^*, y^*)$  be *any* pair of optimal solutions to the primal and the dual. The following are the complementary slackness conditions.

**Fact 2.** (1)  $x^* \cdot (A'y^* - c) = 0$ ; that is, for all  $1 \leq j \leq m$ , either  $x_j^* = 0$  or  $A_j \cdot y^* = c_j$ .  
(2)  $y^* \cdot (Ax^* - b) = 0$ ; that is, for all  $1 \leq i \leq n$ , either  $y_i^* = 0$  or  $a_i \cdot x^* = b_i$ .

*Proof.* Follows from proof of Fact 1. Since there's equality between  $b \cdot y^*$  and  $c \cdot x^*$ , all the inequalities are equality, which translates to the fact above.  $\square$

For the design of approximation algorithms, the following “relaxed” complementary slackness will be useful.

**Fact 3.** *Let  $x$  and  $y$  be feasible solutions to (1) and (2) satisfying the following two conditions: (1) For all  $1 \leq j \leq m$ , either  $x_j = 0$  or  $A_j \cdot y \geq c_j/\alpha$ , and (2) For all  $1 \leq i \leq n$ , either  $y_i = 0$  or  $a_i \cdot x \leq \beta b_i$ . Then,  $c \cdot x \leq (\alpha\beta)b \cdot y$ .*

*Proof.* (Exercise) □

## Complementary Slackness

Complementary slackness tells us that when primal LP sets some variable to non-zero, then it’s for some “good reason”. Consider the set cover LP and it’s dual:

$$\begin{array}{l} \min \quad \sum_{j=1}^m c(S_j)x_j : \quad x_j \geq 0 \quad (3) \\ \quad \sum_{j:i \in S_j} x_j \geq 1, \quad \forall i \in U \end{array} \quad \left| \quad \begin{array}{l} \max \quad \sum_{i \in U} y_i : \quad y \geq 0 \quad (4) \\ \quad \sum_{i \in S_j} y_i \leq c(S_j), \quad \forall j = 1 \dots m \end{array} \right.$$

Complementary slackness tells us that  $x_j > 0$  implies  $\sum_{i \in S_j} y_i = c(S_j)$ . Consider the following set cover algorithm: pick a set  $S_j$  if  $x_j > 0$ . Not if  $x_j \geq 1/f$  or anything, just if  $x_j > 0$ .

**Claim 1.** *The above algorithm returns an  $f$ -approximate solution.*

*Proof.* The cost of our solution is equal to, by complementary slackness,  $\sum_{S_j \text{ picked}} \sum_{i \in S_j} y_i$  which is at most  $f \cdot \sum_{i \in U} y_i \leq f \cdot LP$ . □

Let’s take another example which will be useful next week. Recall the facility location LP (5). Consider its dual (8). Consider the facility-client pairs  $(i, j)$  with  $x_{ij} > 0$ . Recall we filtered this solution to get  $\bar{x}$  where we had  $\bar{x}_{ij} > 0$  implying  $c(i, j) \leq \rho C_j$ . Complementary slackness already gives us the following property:

**Claim 2.** *If  $x_{ij} > 0$ , then  $c(i, j) \leq \alpha_j$ .*

So we don’t need the filtering step at all in some sense. Order the clients in order of increasing  $\alpha_j$ ’s, and do the clustering step. This itself gives a 4 approximation, and I leave the details as an exercise.

## Primal-Dual Algorithms

The primal-dual schema is the following for minimization problems. The algorithm starts with a feasible dual solution (normally the all zero solution suffices). It then raises duals variables till some dual constraint gets tight. Complementary slackness implies the corresponding primal variable can be made positive – and the algorithm sets that primal variable to 1 (if the domain is  $\{0, 1\}$ ). Thus, the dual constraints show us which primal variables to “pick”. The process continues till the dual variables can’t be increased no more at which point this stage of the algorithm terminates. The dual value obtained so far is a lower bound on the optimum solution. The non-trivial part is to prove that the primal variables picked are bounded within a factor from this dual value. This part often differs from problem to problem although there are often underlying similarities.

## The Facility Location Problem

### LP relaxation.

$$\min \quad \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in C} c(i, j) x_{ij} \quad (x_{ij}, y_i \geq 0) \quad (5)$$

$$\sum_{i \in F} x_{ij} \geq 1 \quad \forall j \in C \quad (6)$$

$$y_i \geq x_{ij} \quad \forall i \in F, \forall j \in C \quad (7)$$

### Dual.

$$\max \quad \sum_{j \in C} \alpha_j \quad (\alpha_j, \beta_{ij} \geq 0) \quad (8)$$

$$\sum_{j \in C} \beta_{ij} \leq f_i \quad \forall i \in F \quad (9)$$

$$\alpha_j - \beta_{ij} \leq c(i, j) \quad \forall i \in F, \forall j \in C \quad (10)$$

As is the PD schema, we will like to find an algorithm which solves the UFL problem along with (in fact guided by) finding a (hopefully feasible) solution  $\alpha, \beta$  to (8), and then analyzing the cost of the solution by comparing it with the dual solution. Intuitively, we would like to open facility  $i$  if and only if  $f_i = \sum_j \beta_{ij}$ , and connect client  $j$  to facility  $i$  iff  $\alpha_j = c(i, j) + \beta_{ij}$ . Furthermore, we would like  $\beta_{ij} > 0$  to imply  $j$  connects to  $i$ . As we'll see, this is what we cannot ascertain for all clients but only for a subset. We argue about the connection cost of remaining clients using metricity of the cost function.

Initialize  $\alpha_j$ 's and  $\beta_{ij}$ 's to 0. We will maintain a set of active clients  $A$  initialized to  $C$ . These clients are ones who haven't been assigned an open facility and we will increase  $\alpha_j$  only of active clients. A facility is *tentatively open* iff  $\sum_j \beta_{ij} = f_i$ . We call a pair  $(i, j)$  *tight* if  $\alpha_j = c(i, j) + \beta_{ij}$ .

The algorithm raises the  $\alpha_j$  for every active client  $j \in A$ . It also raises  $\beta_{ij}$  for every tight pair where  $j \in A$ . This is done till one of the following happens

1. A pair  $(i, j)$  becomes tight. If  $i$  is tentatively open, remove  $j$  from the set of active clients  $A$ .
2. A new facility  $i$  becomes tentatively open. All clients  $j$  with  $(i, j)$  being a tight pair are removed from  $A$ .

Let  $X$  be the set of facilities tentatively opened by the above algorithm. Note that each client  $j$  is part of a tight pair  $(i, j)$  with  $i \in X$ . Construct the graph  $G(X, E)$  where there is an edge  $(i, i') \in E$  iff there is a client  $j$  with  $\beta_{ij} > 0$  and  $\beta_{i'j} > 0$ . Let  $X'$  be a *maximal* independent set in the graph.  $X'$  is the final set of facilities opened by the algorithm. Clients connect to the closest facility in  $X'$ .

To argue about the connection cost we consider the following possible sub-optimal assignment. For each facility  $i \in X$ , let  $\Gamma(i) := \{j : (i, j) \text{ is tight}\}$ . For any set  $T \subseteq X$ ,  $\Gamma(T) = \bigcup_{i \in T} \Gamma(i)$ . If  $j \in \Gamma(X')$ , then let  $\sigma(j) = i$  such that  $(i, j)$  is tight and  $\beta_{ij} > 0$ . If  $\beta_{ij} = 0$  for all  $i \in X'$ , then assign any one arbitrarily. For clients  $j \notin \Gamma(X')$ , let  $i \in X \setminus X'$  be a facility with  $(i, j)$  tight. Since  $i$  is not in  $X'$ , there must exist facility  $i' \in X'$  and  $j' \in \Gamma(i')$  such that  $\beta_{ij'} > 0$  and  $\beta_{i'j'} > 0$ . Assign client  $j$  to  $i'$ .

**Theorem 2.**  $\sum_{j \in C} c(j, \sigma(j)) + 3 \sum_{i \in X'} f_i \leq 3 \sum_{j \in C} \alpha_j$

*Proof.* Since we open facilities when  $\beta$ 's pay for the  $f_i$ 's we get

$$\sum_{i \in X'} f_i = \sum_{i \in X'} \sum_{j \in \Gamma(i)} \beta_{ij},$$

and since  $X'$  is an independent set, we get that for all  $i \in X'$ ,  $\beta_{ij} > 0$  for at most one client  $j$ . Furthermore,  $i = \sigma(j)$  for such a client. So,

$$\sum_{i \in X'} f_i = \sum_{j \in \Gamma(X')} \beta_{\sigma(j), j} = \sum_{j \in \Gamma(X')} (\alpha_j - c(j, \sigma(j))) \quad (11)$$

Let's now consider a client  $j$  who was assigned a facility  $i$  in  $X \setminus X'$ . It is now assigned client  $i' \in X'$ , and there is a client  $j'$  as in the description of the algorithm. By triangle inequality,  $c(j, i') \leq c(j, i) + c(i, j') + c(j', i')$ . Since  $\beta_{i'j'} > 0$  and  $\beta_{ij'} > 0$ , we know that  $c(i, j') < \alpha_{j'}$  and  $c(i', j') < \alpha_{j'}$ .  $c(i, j) \leq \alpha_j$  because  $j$  was assigned  $i$ . So,

$$c(i', j) \leq \alpha_j + 2\alpha_{j'}$$

What do we know about  $\alpha_j$  and  $\alpha_{j'}$ ?  $\alpha_j$  stopped growing when  $j$  was assigned  $i$ . This happened *after*  $i$  was declared tentatively open. Since  $\beta_{ij'} > 0$ ,  $(i, j')$  was tight when  $i$  was declared open; thus,  $\alpha_{j'}$  stopped growing *before* or *at the time*  $i$  was opened. In any case,  $\alpha_{j'}$  couldn't have grown any longer than  $\alpha_j$  implying  $\alpha_{j'} \leq \alpha_j$ . Putting this in the above inequality and summing over all  $j$  not in  $\Gamma(X')$ , we get

$$\sum_{j \in C \setminus \Gamma(X')} c(j, \sigma(j)) \leq 3 \sum_{j \in C \setminus \Gamma(X')} \alpha_j \quad (12)$$

Adding (11) (after multiplying by 3 on both sides) and (12) gives us the theorem.  $\square$

Since  $\alpha, \beta$  was a feasible dual, we get a 3 approximation. In fact, we get something stronger than a 3 approximation as the facilities are paid in one-is-to-one ratio. This fact will be extremely essential in the following procedure to obtain an algorithm for the  $k$ -median problem via the Lagrangean relaxation technique.

## Lagrangean Relaxations

The following is an LP for the  $k$ -median problem.

$$LP := \quad \min \quad \sum_{i \in F, j \in C} c(i, j) x_{ij} \quad (x_{ij}, y_i \geq 0) \quad (13)$$

$$\text{subject to} \quad \sum_{i \in F} x_{ij} \geq 1 \quad \forall j \in C \quad (14)$$

$$y_i \geq x_{ij} \quad \forall i \in F, \forall j \in C \quad (15)$$

$$\sum_{i \in F} y_i \leq k \quad (16)$$

The Lagrangean relaxation to the above LP is the following function

$$\mathcal{L}(\lambda) := \min\left\{ \sum_{i \in F, j \in C} c(i, j)x_{ij} + \lambda\left(\sum_{i \in F} y_i - k\right) : (14), (15) \right\}$$

For any  $\lambda \geq 0$ , note that  $\mathcal{L}(\lambda) \leq LP$ . This is because for any feasible solution  $(x, y)$  to (13) leads to a solution of  $\mathcal{L}(\lambda)$  of smaller value. In other words,

$$\max_{\lambda} \mathcal{L}(\lambda) \leq LP$$

Let's fix  $\lambda$  and consider  $\mathcal{L}(\lambda)$ . Observe this is a facility location LP (with a  $k\lambda$  subtracted from the objective) now with each facility having cost  $\lambda$ . Using the approximation algorithm developed in the previous section, we can obtain a solution  $\mathbf{alg}(\lambda)$  which opens facilities  $k_{\lambda}$  and obtains a facility opening cost of  $\lambda \cdot k_{\lambda}$  and connection cost of  $C_{\mathbf{alg}}(\lambda)$ . We have

$$C_{\mathbf{alg}}(\lambda) + 3\lambda \cdot k_{\lambda} \leq 3(\mathcal{L}(\lambda) + k\lambda)$$

So, the connection cost of this solution is

$$C_{\mathbf{alg}}(\lambda) \leq 3(\mathcal{L}(\lambda) + \lambda \cdot (k - k_{\lambda})) \leq 3LP + 3\lambda \cdot (k - k_{\lambda})$$

Suppose we could find a  $\lambda$  such that  $k_{\lambda} = k$ . Then the above algorithm run for that  $\lambda$  implies a 3 approximation. If  $\lambda = 0$ , then obviously  $k_{\lambda} = |F|$ . If  $\lambda$  is very high (larger than  $nc_{max}$  where  $n$  is the number of clients and  $c_{max}$  the largest connection cost), then  $k_{\lambda}$  is 1. So, by doing a binary search, we can reach in  $O(\log(nc_{max}/\varepsilon))$  steps, a  $\lambda$  and  $\varepsilon > 0$  such that  $k_1 := k_{\lambda - \varepsilon} > k$  and  $k_2 := k_{\lambda + \varepsilon} < k$ . Let  $C_1$  and  $C_2$  denote  $C_{\mathbf{alg}}(\lambda - \varepsilon)$  and  $C_{\mathbf{alg}}(\lambda + \varepsilon)$  respectively. Then we get

$$C_1 \leq 3LP + 3(\lambda - \varepsilon)(k - k_1); \quad C_2 \leq 3LP + 3(\lambda + \varepsilon)(k - k_2)$$

Let  $\rho := \frac{k - k_2}{k_1 - k_2} < 1$ . Using the fact that  $k \leq n$ , we get the following.

**Observation 1.**  $\rho \cdot C_1 + (1 - \rho)C_2 \leq 3(1 + 2n^2\varepsilon)LP$ .

Let  $\varepsilon$  be small enough that we can ignore  $2n^2\varepsilon$  and for the rest of the discussion we ignore this error term. Let  $X_1$  and  $X_2$  be the set of facilities opened by the algorithm when run with  $\lambda - \varepsilon$  and  $\lambda + \varepsilon$ . We now describe how to combine these two solutions to obtain a 6-approximation for the  $k$ -median problem.

Firstly note that the solution which opens  $k_2 < k$  facilities is a feasible solution. So, if  $\rho \leq 1/2$ , then this solution itself is a 6-approximation. So we assume  $\rho > 1/2$ . We now choose  $k$  facilities from  $X_1$  and argue this is a 6-approximation. For a facility  $i \in X_2$ , let  $\mathbf{near}(i)$  be the facility in  $X_1$  closest to it. Open all the facilities  $\{\mathbf{near}(i) : i \in X_2\}$ . This opens at most  $r \leq k_2$  facilities from  $X_1$ . Choose a *random* subset of  $(k - r)$  facilities among the remaining  $(k_1 - r)$  facilities and opens them. This is the set of  $k$  facilities opened. Every client connects to the nearest open facility. Let  $C_{\mathbf{alg}}$  be the connection cost

**Lemma 1.**  $E[C_{\mathbf{alg}}] \leq 6LP$ .

*Proof.* We construct a solution whose expected cost is at most  $6LP$ . Consider a client  $j$  and let  $\sigma_1(j) \in X_1$  and  $\sigma_2(j) \in X_2$  be the facilities it were connected to in the two solutions. Let  $c_1(j)$  be  $c(j, \sigma_1(j))$  and  $c_2(j)$  be similarly defined. In the new solution, we connect  $j$  to  $\sigma_1(j)$  if it is open, or to  $\mathbf{near}(\sigma_2(j))$  if not. The probability that  $\sigma_1(j)$  is opened is at least

$$p := \frac{k-r}{k_1-r} \geq \frac{k-k_2}{k_1-k_2} = \rho$$

The cost of connecting  $j$  to  $\mathbf{near}(\sigma_2(j))$  is at most

$$c(j, \sigma_2(j)) + c(\sigma_2(j), \mathbf{near}(\sigma_2(j))) \leq c_2(j) + c(\sigma_2(j), \sigma_1(j)) \leq 2c_2(j) + c_1(j)$$

Thus, the expected cost of connecting  $j$  is

$$pc_1(j) + (1-p)(2c_2(j) + c_1(j)) = c_1(j) + 2(1-p)c_2(j) < 2[pc_1(j) + (1-p)c_2(j)]$$

where we use  $p \geq \rho > 1/2$ . So, the expected connection cost of our solution is at most  $2(pC_1 + (1-p)C_2) \leq 2(\rho C_1 + (1-\rho)C_2)$  since  $p \geq \rho$  and  $C_1 \leq C_2$ .  $\square$