601.435/635 Approximation Algorithms **Lecturer:** Michael Dinitz
Topic: LP Duality **Date:** 4/1/21
Scribe: Michael Dinitz

19.1 Introduction

Today we're going to talk about something which might be review for many of you, but since it might be new for some people, and since we're going to use it extensively for the next week, it's worth spending some time on: LP duality. We'll be spending the next few lectures on using techniques from duality to design and analyze approximation algorithms, focusing on two related but different techniques: primal-dual algorithms, and dual fitting. But first, we need to understand the underlying mathematics about linear programming which gives duality.

19.2 LP Duality: Motivation and Definition

An example LP in canonical form (called the primal):

$$\begin{aligned} \min \quad & 3x_1 + x_2 + 4x_3 \\ (I1) \text{ s.t.} \quad & x_1 + 2x_2 \geq 3 \\ (I2) \quad & x_1 + 2x_3 \geq 2 \\ (I3) \quad & 2x_1 + 3x_2 + x_3 \geq 4 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

A possible solution to this LP is $x_1 = 0, x_2 = 3/2, x_3 = 1$, which gives a cost of $3(0)+(3/2)+4(1) = 11/2$. However, how do we know that $11/2$ is the best answer? Our strategy is to combine constraints to get a lower bound on the objective. For instance, since x_1, x_2, x_3 are all nonnegative and any solution must satisfy constraints $(I1), (I2), (I3)$, we know that any solution must satisfy any nonnegative combination of constraints $(I2)$ and $(I3)$. For example, taking $1/2$ of $(I2)$ and $1/3$ of $(I3)$ implies that any solution must satisfy

$$\begin{aligned} & 1/2(I2) + 1/3(I3) \geq 1/2(2) + (1/3)4 \\ \Leftrightarrow \quad & 1/2(x_1 + 2x_3) + 1/3(2x_1 + 3x_2 + x_3) \geq 7/3 \\ \Leftrightarrow \quad & (1/2 + 2/3)x_1 + x_2 + (1 + 1/3)x_3 \geq 7/3 \\ \Leftrightarrow \quad & 7/6x_1 + x_2 + 4/3x_3 \geq 7/3 \end{aligned}$$

Note that all of the coefficients in this expression are less than the coefficients in the objective function. Together with nonnegativity, this implies that in any solution, the objective function is at least $3x_1 + x_2 + 4x_3 \geq 7/6x_1 + x_2 + 4/3x_3 \geq 7/3$.

We can generalize this to try to find the best such lower bound on the LP. Our goal is to find multipliers y_1, y_2, y_3 for $y_1(I1)+y_2(I2)+y_3(I3)$ that give the best way of combining the inequalities

in the primal. In order to get a valid lower bound, the coefficients of x_1, x_2, x_3 in the combined bound have to all be smaller than their coefficients in the true objective function. So in this case, we need that $y_1 + y_2 + 2y_3 \leq 3$, $2y_1 + 3y_3 \leq 1$, and $2y_2 + y_3 \leq 4$. Subject to these constraints, we want to maximize the lower bound on the primal, i.e. we want to maximize $3y_1 + 2y_2 + 4y_3$. This gives the following dual LP:

$$\begin{aligned} \max\ & 3y_1 + 2y_2 + 4y_3 \\ \text{s.t. } & y_1 + y_2 + 2y_3 \leq 3 \\ & 2y_1 + 3y_3 \leq 1 \\ & 2y_2 + y_3 \leq 4 \\ & y_1, y_2, y_3 \geq 0 \end{aligned}$$

The dual is also an LP. We can solve this LP and find that the value is optimized by $y_1 = 1/2, y_2 = 2, y_3 = 0$, which gives an objective value of $3(1/2) + 2(2) + 4(0) = 11/2$. By construction, this dual objective value is a lower bound on the primal objective value.

19.2.1 Matrix notation

More generally, the primal and dual formulation can be rewritten in matrix notation, for $c, x \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in R^{m \times n}$. The following sides are equivalent and give the primal. The general process is to start with the primal, then make new variables, which becomes the dual.

$$\begin{aligned} \min\ & c^T x \\ \text{s.t. } & Ax \geq b \\ & x \geq 0 \end{aligned} \qquad \begin{aligned} \min\ & \sum_{j=1}^{n} c_j x_j \\ \text{s.t. } & \sum_{j=1}^{n} a_{ij} x_j \geq b_i \quad \forall i \in [m] \\ & x_j \geq 0 \quad \forall j \in [n] \end{aligned}$$

The dual is then formulated as follows, with variables $y_1, \ldots, y_m \geq 0$.

$$\begin{aligned} \max\ & \sum_{j=1}^{m} b_i y_i \\ \text{s.t. } & \sum_{i=1}^{m} a_{ij} y_i \leq c_j \quad \forall j \in [n] \\ & y_i \geq 0 \quad \forall i \in [m] \end{aligned}$$

Or equivalently

$$\begin{aligned} &\max\ b^T y \\ &\text{s.t. } A^T y \le c \\ &\quad y \ge 0 \end{aligned} \qquad\qquad \begin{aligned} &\min\ (-b)^T y \\ &\text{s.t. } (-A)^T y \ge -c \\ &\quad y \ge 0 \end{aligned}$$

Theorem 19.2.1 *If LP D is dual of P, then P is dual of D.*

19.2.2 Weak Duality

Weak duality says that the value of the dual is a lower bound on the value of the primal LP.

Theorem 19.2.2 *For any feasible primal dual solution pair (x, y), $b^T y \le c^T x$ (i.e., OPT(dual)$\le$ OPT(primal)).*

Proof:

$\sum_{j=1}^n c_j x_j \ge \sum_{j=1}^n \big(\sum_{i=1}^m a_{ij} y_i\big) x_j = \sum_{i=1}^m \big(\sum_{j=1}^n a_{ij} x_j\big) y_i \ge \sum_{i=1}^m b_i y_i.$ ■

19.2.3 Strong Duality

Strong duality says that if both primal and dual LPs are feasible, then they have the same optimal value. We're not going to actually prove this today, and often weak duality will be enough for us, but it's a very useful fact to keep in mind.

Theorem 19.2.3 *If x^* optimal for primal, y^* optimal for dual, then $b^T y = c^T x$ (i.e., y^* gives the best possible lower bound).*

Note: If the primal is infeasible, the dual is unbounded.

Theorem 19.2.4 *Let (x, y) be feasible (primal, dual) solution pair, then x, y are both optimal iff*

- *For all $j \in [n]$: either $\sum_{i=1}^m a_{ij} y_i = c_j$ or $x_j = 0$ (or both)*
- *For all $i \in [m]$: either $\sum_{j=1}^n a_{ij} x_j = b_i$ or $y_i = 0$ (or both)*

These are called the *complementary slackness* conditions.

Proof of Theorem 19.2.4: Assume the complementary slackness conditions hold. Then the inequalities in the proof of Theorem 19.2.2 are equalities, which implies that $\sum_{j=1}^n c_j x_j = \sum_{i=1}^m b_i y_i$. Since weak duality says that $\sum_{j=1}^n c_j x'_j \ge \sum_{i=1}^m b_i y'_i$ for any feasible x' and y', (x, y) must be optimal.

Assume (x, y) are optimal solutions. Then strong duality implies that $b^T y = c^T x$. This implies that the inequalities in the proof of Theorem 19.2.2 hold with equality, which in turn implies that the complementary slackness conditions hold. ■

19.3 Example algorithms with LP duality

Let's interpret some of the things that we already know through the lens of duality (next lecture we'll actually see some new things)

19.3.1 Max-flow, min-cut

Let $P_{s,t}$ be all $s-t$ paths. Let x_P represent the flow along path $P \in P_{s,t}$.

The LP for max-flow is

$$\begin{aligned} \max \quad & \sum_{P \in P_{s,t}} x_P \\ \text{s.t.} \quad & \sum_{P \in P_{s,t}: e \in P} x_P \leq c(e) \quad \forall e \in E \\ & x_P \geq 0 \quad \forall P \in P_{s,t} \end{aligned}$$

The dual of this LP is

$$\begin{aligned} \min \quad & \sum_{e \in E} c(e) y_e \\ \text{s.t.} \quad & \sum_{e \in P} y_e \geq 1 \quad \forall P \in \mathcal{P}_{s,t} \\ & y_e \geq 0 \quad \forall e \in E \end{aligned}$$

This is clearly the min $s-t$ cut LP! As we saw in a previous class, this LP can be rounded to an integral solution without any loss in the objective function. Hence strong duality, combined with this rounding, gives a proof that the maximum $s-t$ flow equals the min $s-t$ cut.

19.3.2 Max-multicommodity flow, Multicut

Input: $G = (V, E)$, capacities $c : e \to \mathbb{R}$, k source-sink pairs $(s_1, t_1), \ldots, (s_k, t_k)$.

Feasible: Some assignment of flow from each s_i to the corresponding t_i such that the total flow on edge e is less than equal to $c(e)$ for all $e \in E$.

Objective: maximize the total flow

An LP formulation of the max-multicommodity flow problem is the following. Think of x_P^i as the amount of flow along path P from s_i to t_i.

$$\begin{aligned} \max \quad & \sum_{i=1}^{k} \sum_{P \in P_{s_i,t_i}} x_P \\ \text{s.t.} \quad & \sum_{i=1}^{k} \sum_{P \in P_{s_i,t_i}: e \in P} x_P \leq c(e) \quad \forall e \in E \\ & x_P \geq 0 \quad \forall i \in [k], \forall P \in P_{s_i,t_i} \end{aligned}$$

Dual:

$$\begin{aligned} \min \ & \sum_{e \in E} c(e) y_e \\ \text{s.t.} \ & \sum_{e \in P} y_e \geq 1 \quad \forall i \in [k], \forall P \in P_{s_i,t_i} \\ & y_e \geq 0 \quad \forall e \in E \end{aligned}$$

Note that the dual is exactly the multicut LP! Let F^* equal the max-multicommodity flow, and let $\hat{C}$ equal the actual min-multicut and C^* be the min fractional multicut.

Theorem 19.3.1 $F^* \leq \hat{C} \leq 4(\ln(k+1))F^*$.

Proof: Let (x^*, y^*) be the optimal primal dual pair and $\hat{y}$ be the optimal multicut. We want to show that $\hat{C}$ is bounded on either side in terms of the optimal primal value F^*.

From weak duality we get that $F^* \leq C^*$. Because C^* is a relaxation of $\hat{C}$, we get that $C^* \leq \hat{C}$. Using the previously established rounding method from class for min-multicut, we know that $\hat{C} \leq 4(\ln(k+1))C^*$. Strong duality then gives us that $4(\ln(k+1))C^* = 4(\ln(k+1))F^*$.

Therefore the optimal min-multicut solution to the dual is within $4(\ln(k+1))$ of the optimal solution of max-multicommodity flow. ■

This type of result is known as a *flow-cut gap*. At a high level, the dual of any flow problem is a (fractional) cut problem. But unlike $s-t$ min cut, that cut problem cannot always be rounded without loss. So in general there is a gap between the value of the flow and the value of the cut. We've now seen this in two settings: $s-t$ flow/cut, and multicommodity flow / multicut. There are many others, though, most notably maximum concurrent multicommodity flow and a problem known as Sparsest Cut. We may see these later in the semester, but you can see the following section if you're interested.

19.3.3 Max Concurrent Multicommodty Flow, Sparsest Cut

This one is a bit less trivial, but still not too bad. Recall that in the max concurrent flow problem, the setup is the same as max multicommodity flow except that for every demand i we are also given demand $d : [k] \to \mathbb{R}^+$, and the goal is to find the maximum λ so that we can send $\lambda d(i)$ flow from s_i to t_i for every i simultaneously. This can be phrased as the following LP:

$$\begin{aligned} \max \ & \lambda \\ \text{s.t.} \ & \sum_{i=1}^{k} \sum_{p \in P_{s_i,t_i} : e \in p} x_p \leq c(e) && \forall e \in E \\ & \lambda d(i) - \sum_{p \in P_{s_i,t_i}} x_p \leq 0 && \forall i \in [k] \\ & x_p \geq 0 && \forall i \in [k], \forall p \in P_{s_i,t_i} \end{aligned}$$

When we take the dual of this, we get:

$$\begin{aligned}
\min & \sum_{e \in E} c(e) y_e \\
\text{s.t.} & \sum_{i=1}^{k} d(i) z_i \geq 1 \\
& \sum_{e \in p} y_e - z_i \geq 0 && \forall i \in [k], p \in P_{s_t, t_i} \\
& y_e \geq 0 && \forall e \in E \\
& z_i \geq 0 && \forall i \in [k]
\end{aligned}$$

This seems like a weird-looking cut problem, but let's think about it for a minute. First, consider some solution and let $L = \sum_{i=1}^{k} d(i) z_i$ (so $L \geq 1$). If $L > 1$, then we can rescale everything by setting $y'_e = y_e / L$ and $z'_i = z_i / L$ and we still get a feasible solution, and one which has smaller cost. Thus $L = 1$ in any optimal solution. This means that we can actually rescale to get the following, which doesn't look like an LP since it has quotients of variables, but because of the above argument is equivalent to our LP:

$$\begin{aligned}
\min & \frac{\sum_{e \in E} c(e) y_e}{\sum_{i=1}^{k} d(i) z_i} \\
\text{s.t.} & \sum_{e \in p} y_e - z_i \geq 0 && \forall i \in [k], p \in P_{s_t, t_i} \\
& y_e \geq 0 && \forall e \in E \\
& z_i \geq 0 && \forall i \in [k]
\end{aligned}$$

This is a fractional cut problem, since we have a constraint that the length of every $s_i - t_i$ path is at least something (z_i in this case, rather than 1 as usual). If we think about this *integrally* what do we get? Then $z_i = 1$ would say that we have to cut commodity i, while if $z_i = 0$ then we don't have to cut the commodity. And the y_e's (as usual) say whether we include an edge in the cut. So this becomes the problem of finding a cut $A \subseteq E$ such which minimizes not the cost of A, but the quantity

$$\frac{c(A)}{\sum_{i \in [k] : s_i, t_i \text{ not connected in } G \setminus A} d(i)}$$

This is known as the *sparsest cut* problem, or sometimes the *nonuniform sparsest cut* problem. We didn't talk about this problem, but it's quite important. There's a known algorithm to round this LP to achieve an $O(\log k)$-approximation, which proves that the flow-cut gap is at most $O(\log k)$. However, if we don't care about the flow-cut gap but rather care just about sparsest cut, there's a $O(\sqrt{\log n})$-approximation which does not go through the LP.