CS787: Advanced Algorithms

Scribe: Amanda Burton, Leah Kluegel **Lecturer:** Shuchi Chawla
Topic: Primal-Dual Algorithms **Date:** 10-17-07

14.1 Last Time

We finished our discussion of randomized rounding and began talking about LP Duality.

14.2 Constructing a Dual

Suppose we have the following primal LP.

min $\sum_i c_i x_i$ *s.t.*

$\sum_i A_{ij} x_i \geq b_j \quad \forall j = 1, ..., m$

$x_i \geq 0$

In this LP we are trying to minimize the cost function subject to some constraints. In considering this canonical LP for a minimization problem, lets look at the following related LP.

max $\sum_j b_j y_j$ *s.t.*

$\sum_j A_{ij} y_j \leq c_i \quad \forall i = 1, ..., n$

$y_j \geq 0$

Here we have a variable y_j for every constraint in the primal LP. The objective function is a linear combination of the b_j multiplied by the y_j. To get the constraints of the new LP, if we multiply each of the constraints of the primal LP by the multiplier y_j, then the coefficients of every x_i must sum up to no more than c_i. In this way we can construct a dual LP from a primal LP.

14.3 LP Duality Theorems

Duality gives us two important theorems to use in solving LPs.

Theorem 14.3.1 ***(Weak LP Duality Theorem)*** *If x is any feasible solution to the primal and y is any feasible solution to the dual, then $Val_P(x) \geq Val_D(y)$.*

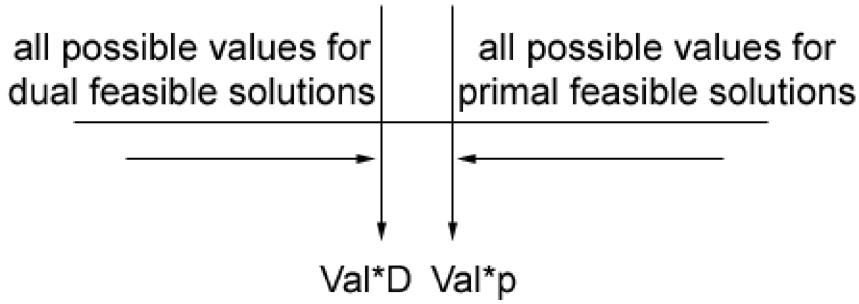

Figure 14.3.1: Primal vs. Dual in the Weak Duality Theorem.

On the number line we see that all possible values for dual feasible solutions lie to the left of all possible values for primal feasible solutions.

Last time we saw an example in which the optimal value for the dual was exactly equal to some value for the primal. This introduces the question: was it a coincidence that this was the case? The following theorem claims that no, it was not a coincidence. In fact, this is always the case.

Theorem 14.3.2 ***(Strong LP Duality Theorem)*** *When P and D have non-empty feasible regions, then their optimal values are equal, i.e.* $Val_P^* = Val_D^*$.

On the number line we see that the maximum value for dual feasible solutions is equivalent to the minimum value for primal feasible solutions.

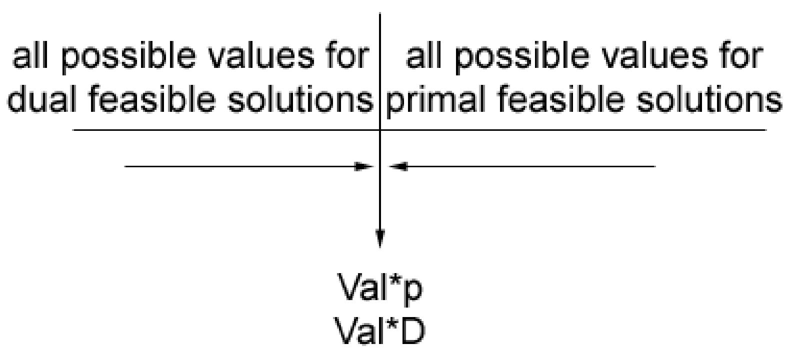

Figure 14.3.2: Primal vs. Dual in the Strong Duality Theorem.

It could happen that there is some LP with no solution that satisfies all of the constraints. In this case the feasible region would be empty. In this case the LP's dual will be unbounded in the sense that you can achieve any possible value in the dual.

The Strong LP Duality Theorem has a fairly simple geometric proof that will not be shown here due to time constraints. Recall from last time the proof of the Weak LP Duality Theorem.

Proof: (Theorem 14.3.1)

Start with some feasible solution to the dual LP, say y. Let y's objective function value be $Val_D(y)$. Let x be a feasible solution to the primal LP with objective function value $Val_P(x)$. Since y is a feasible solution for the dual and x is a feasible solution to the primal we have

$$\begin{aligned}
Val_D(y) &= \textstyle\sum_j b_j y_j \\
&\le \textstyle\sum_j (\sum_i A_{ij} x_i) y_j \\
&= \textstyle\sum_i \sum_j A_{ij} x_i y_j \\
&= \textstyle\sum_i (\sum_j A_{ij} y_j) x_i \\
&\le \textstyle\sum_i c_i x_i \\
&= Val_P(x)
\end{aligned}$$

■

So just rewriting the equations shows us that the value of the dual is no more than the value of the primal. What happens if these values are exactly equal to one another? This happens when x and y are optimal solutions for the primal and the dual respectively. What can we deduce from this?

First, both of the inequalities in the proof of 14.3.1 must both be equalities. If these are equal we have

$$\sum_j b_j y_j = \sum_j (\sum_i A_{ij} x_i) y_j \tag{14.3.1}$$

and

$$\sum_i (\sum_j A_{ij} y_j) x_i = \sum_i c_i x_i \tag{14.3.2}$$

Then each term on the left hand side of Equation 14.3.1 must equal the corresponding term on the right hand side of Equation 14.3.1 and each term on the left hand side of Equation 14.3.2 must equal the corresponding term on the right hand side of Equation 14.3.2. This is the case if, in Equation 14.3.1, $b_j = \sum_i A_{ij} x_i$ and, in Equation 14.3.2, $c_i = \sum_j A_{ij} y_j$. What if $b_j \neq \sum_i A_{ij} x_i$ or $c_i \neq \sum_j A_{ij} y_j$, can we still have the equalities in Equation 14.3.1 and Equation 14.3.2? Equation 14.3.1 can if $y_j = 0$ and Equation 14.3.2 can if $x_i = 0$.

If x and y are primal feasible and dual feasible, respectively, such that $Val_D(y) = Val_P(x)$, then they must satisfy the following:

1. $\forall j$: either $y_j = 0$ or $b_j = \sum_i A_{ij} x_i$.
2. $\forall i$: either $x_i = 0$ or $c_i = \sum_j A_{ij} y_j$.

If $b_j = \sum_i A_{ij} x_i$, we say that the j^{th} constraint in the primal is tight. Condition 1 is called dual complementary slackness (DCS). Similarly, if $c_i = \sum_j A_{ij} y_j$, we say the the i^{th} constraint in the dual is tight. Condition 2 is called primal complementary slackness (PCS).

For a nicer view of the correspondence between the primal and the dual consider the following way of viewing this property.

Primal LP	**Dual LP**
min $\sum_i c_i x_i$ *s.t.*	max $\sum_j b_j y_j$ *s.t.*
$\sum_i A_{i1} x_i \geq b_1$	$y_1 \geq 0$
$\sum_i A_{i2} x_i \geq b_2$	$y_2 \geq 0$
$\vdots$	$\vdots$
$\sum_i A_{im} x_i \geq b_m$	$y_m \geq 0$
$x_1 \geq 0$	$\sum_j A_{1j} y_j \geq c_1$
$x_2 \geq 0$	$\sum_j A_{2j} y_j \geq c_2$
$\vdots$	$\vdots$
$x_n \geq 0$	$\sum_j A_{nj} y_j \geq c_n$

Here we have associated regular constraints in the primal to non-negativity constraints in the dual and non-negativity constraints in the primal to regular constraints in the dual along the rows of this table. Notice that the primal has m regular constraints and n non-negativity constraints on it and the dual has a variable for each of the m regular constraints and a regular constraint for each of the n non-negativity constraints. Therefore, there is a one-to-one correspondence between constraints in the primal and constraints in the dual. If x and y are optimal solutions for the primal and the dual, then for each row in this table, either the left side or the right side of the table must be tight, i.e. an equality rather than an inequality.

From the Strong LP Duality Theorem we have

Corollary 14.3.3 (x^*, y^*) *are primal and dual optimal solutions respectively if and only if they satisfy DCS and PCS.*

14.4 Simple Example

Consider the following minimization LP.

$$
\begin{aligned}
\text{min} \quad & x_1 + 3x_2 + 4x_3 + x_5 \quad s.t. \\
& 5x_1 + 2x_4 \geq 1 \\
& 4x_2 + 3x_3 + x_4 + x_5 \geq 2 \\
& x_1 + x_3 + x_5 \geq 7 \\
& x_1 \geq 0 \\
& \vdots \\
& x_5 \geq 0
\end{aligned}
$$

We want to compute the dual of this LP. We want to introduce a variable for each regular constraint in the primal, so we will have variables y_1, y_2, and y_3 in the dual. We want to introduce a constraint for each variable in the primal, so we will have 5 constraints in the dual. Each constraint in the dual is written as a function of the variables y_1, y_2, and y_3 in such a way that the coefficients of x_i sum up to no more than the coefficient of x_i in the objective function of the primal. Finally, we determine the objective function of the dual by letting the right hand sides of the constraints in the primal be the coefficients of the y_j. This gives us the following dual

$$\begin{aligned}
\max \ & y_1 + 2y_2 + 7y_3 \ \ s.t. \\
& 5y_1 + y_3 \leq 1 \\
& 4y_2 \leq 3 \\
& 3y_2 + y_3 \leq 4 \\
& 2y_1 + y_2 \leq 0 \\
& y_2 + y_3 \leq 1 \\
& y_1 \geq 0 \\
& y_2 \geq 0 \\
& y_3 \geq 0
\end{aligned}$$

In matrix form this transposes the coefficient matrix

x_1	x_2	x_3	x_4	x_5
5	0	0	2	0
0	4	3	1	1
1	0	1	0	1

into a coefficient matrix by transposing the first column of coefficients of the primal into the first row of coefficients of the dual, the second column into the second row, and so on.

y_1	y_2	y_3
5	0	1
0	4	0
0	3	1
2	1	0
0	1	1

14.5 The Power of LP-Duality

We would now like to use LP-Duality for designing algorithms. The basic idea behind why we might want to use LP-Duality to design algorithms is to solve a problem more quickly than using Linear Programming on its own. When we need to solve an algorithm with Linear Programming, we relax some Integer Program so that the corresponding LP gives us a good lower bound, solve the LP for some optimal linear solution, and round it off. It is possible to solve Linear Programs in polynomial time for its optimal solution, but the known algorithms still run fairly slowly in practice.

Also, we sometimes want to reduce some very large Integer Program and relax it to some exponential sized Linear Program with a small integrality gap, so that you get a good approximation. In which case, you cannot solve the linear program to optimality in polynomial time. LP-duality lets us take these programs, and instead of finding the optimal point of the LP exactly, we use some feasible solution to the Dual LP to get some lower bound on the optimal solution and often times that suffices. In fact, sometimes this process gives us a way to design a purely combinatorial for approximating a problem.

Instead of writing down the LP and solving the LP, we simultaneously construct feasible solutions to both the Primal and Dual LPs in such a way that these are within a small factor of each other, and then construct an integer solution using those. This is much faster and much simpler than solving the LP to optimality. Sometimes, this can even be used even for LPs of exponential size.

Another nice thing about LP-duality, is that it is such a non-trivial relationship between two seemingly different LPs, that it often exposes very beautiful min-max kinds of theorems about combinatorial objects. One such theorem is the Max-flow Min-cut Theorem, and we will shortly prove that using LP-Duality.

14.6 Max-flow Min-cut

Max-flow Min-cut is one of the most widely stated examples of LP-duality, when in fact there are many other examples of the theorems of that kind that arise from this particular relationship. For this example, though, we will talk about Max-flow Min-cut.

To explore this relationship, we would first like to create some LP that solves a Max-flow problem, find its Dual, and then see how the Dual relates to the Min-cut problem.

For the Max-flow problem, are given a graph $G = (V, E)$ with source s, sink t, and some capacities on edges c_e.

14.6.1 Max-flow LP

First we define the variables of our Max-flow LP:

x_e : amount of flow on edge e

Next we define the constraints on the variables:

$x_e \leq c_e \;\; \forall e$ - the flow on x_e does not exceed the capacity on e

$x_e \geq 0 \;\; \forall e$ - the flow on e is non-negative

$\sum_{e \in \delta^+(v)} x_e = \sum_{e \in \delta^-(v)} x_e \;\; \forall v \neq s, t$ - the flow entering v equals the flow leaving v

Subject to these constraints, our objective function maximizes the amount of flow from s to t, by summing over all the flow entering t or all the flow leaving s:

$\max \; \sum_{e \in \delta^-(s)} x_e$

14.6.2 Alternate Max-flow LP

There exists another equivalent way of writing the Max-flow LP in terms of the flow on paths:

x_p : amount of flow on an s-t path p

$\mathcal{P}$: set if all paths from s-t

This LP has the constraints on the variables:

$x_p \geq 0 \;\; \forall p \in \mathcal{P}$ - the flow on p is non-negative

$\sum_{p \ni e} x_p \leq c_e \;\; \forall e$ - the flow on e is no larger than the edge's capacity

Subject to these constraints, our objective function maximizes the amount of flow on the x_p paths:

$\max \; \sum_{p \in \mathcal{P}} x_p$

This is an equivalent LP, and the Primal we are going to talk about, because this one will be easier to work with when constructing the Dual.

14.6.3 The Dual to the Max-flow LP

First, we must define our variables for the Dual, where each variable in the Dual corresponds to a constraint in the Primal. Since the Primal contains one non-trivial constraint for every edge, the Dual must contain one variable for every edge in the Primal:

y_e : variable for edge $e \in E$

The Dual LP must have one constraint for every variable. Like we did in the Simple Example, for each variable in the Primal, we look at the coefficient that you get when you multiply the Primal constraints by the new variables, y_e, and sum over the number of constraints e. We want to find the coefficient of any fixed variable, and write the constraint coresponding to that variable. For any fixed path p, we analyze the coefficients using the constraints on x_p:

$\sum_e \sum_{p \ni e} x_p y_e \leq \sum_e c_e y_e$

$\sum_p x_p \sum_{e \in p} y_e \leq \sum_e c_e y_e$

From this analysis, we get the constraints on the variable y_e:

$y_e \geq 0 \;\; \forall e$

$\sum_{e \in p} y_e \leq 1 \;\; \forall p$

Subject to these constraints, our objective function is

$\min \; \sum_e y_e c_e$

So far, we have mechanically applied a procedure to a Primal LP and determined its Dual. When you start from a combinatorial problem and obtain its Dual, it is a good question to ask what these variables and constraints actually mean. Do they have a nice combinatorical meaning? Usually, if you start from some combinatorial optimization problem, then its Dual does turn out to have a nice combinatorial meaning.

So what is the meaning of this Dual? It is saying that it wants to assign some value to every edge, such that looking at any s-t path in the graph, the sum total of the values on the edges in that path should be at least one. In order to understand what kind of solution this LP is asking for, think of an integral solution to the same LP. Say that we require $y_e \in \{0,1\}$. What, then, is a integral feasible solution to this program? If $y_e \in \{0,1\} \;\; \forall e$, then we are picking some subset of the edges, specifically the subset of edges where $y_e = 1$. What properties should those edges satisfy? It should be that for all s-t paths in the graph, the collection of edges that we pick should have at least one edge from that path, and these edges form a cut in the graph G. If we were to minimize the solution over all the cuts in the graph, this would give us a Min-cut.

Fact 14.6.1 *all integral feasible solutions to the Dual form the set of all s-t cuts.*

Because the Dual is minimizing over some values, any feasible solution to the Dual is going to be greater than or equal to any feasible solution to the Primal, by the Weak Duality Theorem.

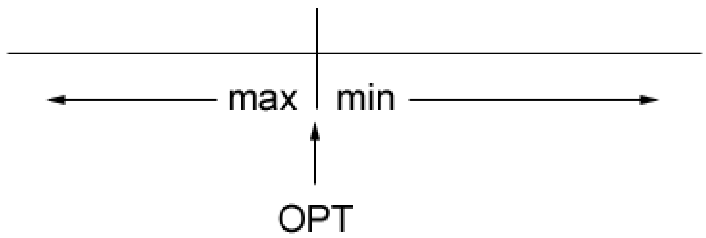

Figure 14.6.3: Ranges of minimization vs. maximazation feasible solutions.

From this we determine the following corollary:

Corollary 14.6.2 *Min-cut* $\geq$ *Max-flow*

This result is unsurprising, though determining that Min-cut = Max-flow is still non-trivial. Thankfully, the Strong Duality Theorem tells us:

Corollary 14.6.3 *Min-fractional-cut* $=$ *Max-flow*

The optimal value of the Dual will be exactly equal to the optimal value of the Primal. So far, we do not know if the optimal solution to the Dual is an integral solution or not. If the solution to the Dual is integral, than it will prove the Max-flow Min-cut Theorem, but if it turns out to be a fractional solution, than we cannot determine the best solution to the problem. The Strong Duality Theorem gives us this weaker version of Max-flow Min-cut, that any fractional Min-cut is equal to the Max-flow of the graph. A fractional Min-cut is definted by this LP, where we assign some fractional value to edges, so that the total value of a path is less than or equal to 1. In order to get the Max-flow Min-cut Theorem from this weaker version, we need to show that for every fractional solution there exists integral solution that is no less optimal. We will give a sketch of the proof of this idea.

Theorem 14.6.4 *There exists an integral optimal solution to the Min-cut LP.*

Proof: To prove this, we will start with any fractional feasible solution to this LP, and derive from it an integral solution that is no worse than the fractional solution. Assume some fractional solution to the LP, and think of the values on each edge as lengths. Then the LP is assigning a length to every edge, with the property that any s-t path is going to have a length of at least one. This means that the distance from s to t is at least one.

We are going to lay out the graph in such a way that these y_e's exactly represent the lengths of the edges.

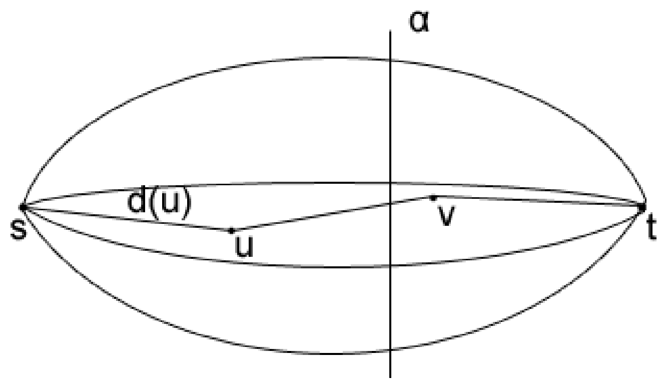

Figure 14.6.4: Distances y_e's from s to t.

Then, we will assign the distance to any point v from the source s as the minimum length of an s-v path. For any point v, we can look at the length of all paths from s to v, and the shortest such path gives me the distance from s to v, $d(v)$, with lengths according to the y_e's. From this we get the distances from s to all the points in the path, and we also know that $d(t) \geq 1$.

Then, we pick some value $\alpha \in [0, 1]$ uniformly at random, and look at all of the vertices that are within distance α of s. This will create a cut, because there are some number of edges crossing over between the set of vertices within distance α those with distances greater than α.

Suppose that we pick α uniformly at random from this range, than if we look at any particular edge in the graph, say (u, v), than the probability that the edge (u, v) is cut is no larger than its length. Why is this? The probability that (u, v) was cut is the probability that $d(u) \leq \alpha$ and $d(v) > \alpha$ (assuming $d(u) \leq d(v)$).

As we draw concentric circles of the distance α from s, then Figure 14.6.5 shows that the difference in the radii of these circles is no more than the distance from u to v. The only way that the edge can be cut, is if we chose an α somewhere between these two concentric circles. So the probability that (u, v) is cut is the difference in the radii, which is no more than the length of (u, v):

$$Pr\left[(u, v)\ is\ cut\right] \leq y_{u,v}$$

Then, the expectation of the size of the cut will be:

$$E\left[size\ of\ cut\right] \leq \sum_{e \in E} c_e Pr\left[e\ is\ cut\right]$$

$$E\left[size\ of\ cut\right] \leq \sum_{e \in E} c_e y_e$$

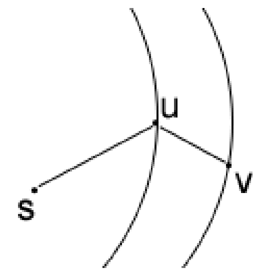

Figure 14.6.5: Distance from u to v vs. possible distances on α.

This expectation represents the size of the cut. If an edge e is in the cut, it contributes c_e to the size of the cut, and 0 otherwise. So we can assign indicator random variables to every edge that takes on a value 1 if it is in the cut, and 0 if it is not. So then the size of the cut is the weighted sum of the indicator variables. Then its expectation is the sum of its expected values of its variables, so it will turn out to be just the cost of each edge times the probability that the edge is in the cut, which is summed over all the edges.

As $\sum_{e \in E} c_e y_e$ is the value of the fractional solution, the expected value of the integer cut is no greater than that of the fractional solution. If we are picking a cut with some probability and the expected value of that cut is small, then there has to be at least one cut in the graph which has small value. So, there exists one integral cut with value at most the value of y.

■

14.7 Next Time

We will see more applications of LP-duality next time, as well as an algorithm based on LP-duality.