1 Duality for network flow

In lecture, we talked about applying LP duality to network flow, and seeing how we get the maxflow-mincut theorem as a special case of the strong duality theorem for LPs.

First, let's set up the primal. As usual, we'll have a variable f_{uv} for each edge (u, v) indicating the amount of material flowing on that edge, with constraints $0 \leq f_{uv} \leq c_{uv}$ where c_{uv} is the capacity of that edge. Now, to simplify our lives, let's add in an edge (t, s) of very large capacity M (greater than the sum of capacities of all other edges) and ask to maximize the flow on *that* edge f_{ts} subject to having flow-in equal to flow-out *everywhere*, including at s and t. Convince yourself that this is equivalent to the original max-flow problem. As mentioned in class, we can think of this like a fountain, where the (t, s) edge is the hidden pipe bringing water from the bottom back up to the top.

The next trick to simplify our lives is that instead of requiring flow-in = flow-out everywhere, we can require flow-in $\leq$ flow out everywhere. I.e.,

$$\text{for all } v, \sum_u f_{uv} \leq \sum_w f_{vw}.$$

This sounds dangerous at first, but notice that this can only be satisfied if indeed we have flow-in = flow-out everywhere. That's because if we sum up the LHS for all v we get the same thing as summing up the RHS for all v; i.e., $\sum_v \sum_u f_{uv} = \sum_v \sum_w f_{vw}$. So, the only way that all the left-hand-sides can be $\leq$ their corresponding right-hand-side is if indeed we have equality.

So, now we can write this as maximizing $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq 0$, where:

- $\mathbf{x}$ is the vector of f_{uv} variables, with, say, f_{ts} as the first variable for convenience.
- $\mathbf{c}^T$ is the vector $(100\ldots0)$.
- A has $|E| + |V|$ rows and $|E|$ columns, and $\mathbf{b}$ has $|E| + |V|$ entries.

  We can put the contraints in whatever order we like, but let's have the first $|E|$ constraints be the $f_{uv} \leq c_{uv}$ constraints, so the first $|E|$ rows of A look like the identity matrix and the first $|E|$ entries in $\mathbf{b}$ are the c_{uv}'s.

  Then the next $|V|$ rows of A correspond to the constraints "for all v, $\sum_u f_{uv} - \sum_w f_{vw} \leq 0$". So the vth row has 1's for all the edges entering into v and -1's for all the edges exiting from v. The corresponding entry in $\mathbf{b}$ is 0.

  Let's use A_{ve} to index the vth row and eth column of this bottom portion of A. So, $A_{ve} = 1$ if v is the head of edge e (e is pointing into v) and $A_{ve} = -1$ if v is the tail of e.

1.1 The dual

For the dual, we have one variable per row of A. Let's call the first $|E|$ variables z_{uv} (we have one per edge) and the next $|V|$ variables y_v (we have one per vertex). We want to minimize $\mathbf{y}^T\mathbf{b}$

subject to $\mathbf{y}^T A \geq \mathbf{c}^T$ and $\mathbf{y} \geq 0$. So, this means we want to:

- Minimize $\sum_{uv} c_{uv} z_{uv}$ subject to:
- For the first column of A we get the constraint $z_{ts} + y_s - y_t \geq 1$.
- For the rest of the columns of A we get $z_{uv} + y_v - y_u \geq 0$, i.e., $z_{uv} \geq y_u - y_v$.
- And we need all variables to be ≥ 0.

Now, we can simplify this a bit and interpret it. First of all, additive offsets to all the y_v's don't change anything, so we can set $y_t = 0$. Secondly, since c_{ts} is very large, the optimal solution will set $z_{ts} = 0$. So this means that $y_s \geq 1$, and in fact it's not hard to see that the optimal solution will set $y_s = 1$. Finally, if we look at the constraint $z_{uv} \geq y_u - y_v$, we can notice that if $y_u > y_v$ then this will be satisfied at equality ($z_{uv} = y_u - y_v$) in the optimal solution, since we are trying to minimize.

So, this means we can think of the dual as follows. Think of y_v as the "height" of v. We have the sink t at height 0, the source s at height 1, and we are trying to solve for heights y_v of all the other nodes to minimize the total sum, over all edges (u, v) that go downhill, of $c_{uv}(y_u - y_v)$.

Notice that one solution is to take some subset of the nodes S and put them at height 1, and put the remaining nodes $V - S$ at height 0, where $s \in S$ and $t \notin S$. In this case, the value of the objective will be exactly the capacity of the cut $(S, V - S)$. So, the minimum *integral* solution is the minimum cut in the graph.

Interestingly, the minimum interal solution is also the minimum fractional solution, i.e., you can't do better with fractions. We will prove this in a minute, but first notice that this then gives us the maxflow-mincut theorem from strong duality, since the optimum value of the primal LP is the maximum flow, and we are saying that the optimum value of the dual LP is the minimum cut.

OK, so now, why can't you do better with fractions? Let's prove it by contradiction. Suppose you can do better with fractions, and consider some optimal (fractional) solution, and out of all possible optimal fractional solutions (if there is more than one) pick the one that uses the fewest different heights. Let $h \in (0, 1)$ be some nonintegral height used in this solution, and let S_h be the set of all nodes of height h. Let C_1 be the sum of capacities of all edges coming into S_h from nodes of higher height. Let C_0 be the sum of capacities of all edges leaving S_h going to nodes of lower height. If $C_1 \geq C_0$, then we can get an equally good solution (or better if this inequality is strict) by raising the height of all nodes in S_h to the next higher height used in the solution. If $C_1 < C_0$ then we can get a better solution by lowering the height of all nodes in S_h to the next lower height used in the solution. (These statements follow from the same reasoning used to show that the optimal coffee shops in the coffee-shop problem will be located at people's houses.) Either way, this contradicts the assumption that this was the optimal solution that uses the *fewest* different heights.