15-451/651 Algorithms, Fall 2013

Recitation notes **October 23, 2013**

- Coding up shortest paths as an LP
- Another example of solving a matrix game

Coding up shortest paths as an LP: Let's see if we can code up the shortest-path problem as an LP. So our input is a directed graph G with weights $w(e)$ on all the edges, a start node s, and a destination t. We want to find a path from s to t of least weight.

There are actually 2 natural ways of doing this.

1. Let's have a variable x_e for each edge e, with $0 \leq x_e \leq 1$. (Think of $x_e = 1$ meaning we use that edge, and $x_e = 0$ meaning we don't, but of course the LP might assign fractional values.)

   Our goal is to minimize $\sum_e w(e)x_e$, subject to:

   - One unit of "flow" leaves s: $\sum_{e=(s,v)} x_{sv} = 1$
   - One unit of "flow" enters t: $\sum_{e=(v,t)} x_{vt} = 1$.
   - For all $v \notin \{s, t\}$, we have flow-in = flow-out: $\sum_{e=(u,v)} x_{uv} = \sum_{e=(v,u)} x_{vu}$.

   OK, so we really coded this problem up as a min-cost flow (send 1 unit of s-t-flow with the least cost. You can imagine that edges have capacity 1, but since we're sending at most 1 unit of flow, it does not make sense to write down the capacity constraints). But it is interesting to think of it as an LP.

   Now, what if the LP solver returns fractional values? The claim is that in that case, all paths from s to t that you get by following non-zero x_e's are shortest paths (otherwise you could get a better LP solution by rerouting that flow on the shortest path). We got lucky here that the fractional values didn't hurt us. For other problems (e.g., like the ones we will see in the next lectures like 3SAT and vertex-cover) you can't necessarily convert a fractional solution into an integer one.

2. Here is another approach. Let's solve for *distances* on all nodes v, representing the distance from s to v along the shortest path. If we can solve for these, we can then recover the path just like we did with Bellman-Ford.

   Variables: we will have a variable d_v for every vertex v, representing its distance from s.

   Constraints:

   - For any edge $e = (u, v)$ we put the constraint that $d_v \leq d_u + w(e)$. In other words, since one way to reach v is to reach u first and then go on edge e to v, the shortest-path distances must satisfy this property.
   - We also add the constraint $d_s = 0$.

Objective: now, we *maximize* $\sum_v d_v$ subject to those constraints.

This will never produce values d_v that are *larger* than the shortest path from s to v, since by induction (on nodes in order of their true distance from s) the node u that comes right before v in the true shortest path from s will never have too large a value, and we have constrained $d_v \leq d_u + w(u, v)$. It also will never produce values d_v that are *smaller* than the shortest path from s to v. We can prove this by contradiction (to make this proof a bit easier, assume all $w(e)$ are non-negative). Suppose for sake of contradiction there are some vertices whose distance values are too low, and let v be the vertex with the smallest incorrect value d_v. It cannot be the case that any of the constraints $d_v \leq d_u + w(u, v)$ are "tight", meaning that they are equalities, since otherwise u would be a vertex with a smaller incorrect value (can you see why d_u must be too low if d_v is too low and $d_v = d_u + w(u, v)$?). This means that we haven't correctly maximized our objective since this value d_v can be raised without breaking any of the constraints.

A matrix game: In the game of evens-odds, two players E (Eve) and O (Odelia) simultaneously show one or two fingers. Eve wins if the sum is even and Odelia wins if the sum is odd. For example, they might do this to decide who has to do some chore. Let's view winning as getting a score of $+1$ and losing as getting a score of -1.

Recall that a *strategy* is a deterministic or randomized method for picking what to play. The *value* of a strategy is the score you get (or the expected score if the strategy is randomized) against an opponent who plays optimally knowing your strategy (she has spies). For instance, considering the case of Odelia, the deterministic strategy "play one" has value -1 since Eve knowing this is Odelia's strategy would just play one. Similarly, the deterministic strategy "play two" has value -1 since Eve would just play two. The strategy "flip a coin and with probability 1/2 play one and with probability 1/2 play two" has value 0 since whatever Eve chooses, the expected score will be 0.

Here is a variation on evens-odds: suppose that we change the game so that both players playing two is a draw (both get a score of 0). In other words, we can summarize the game from Odelia's point of view by the following payoff matrix:

		Eve plays	
		one	two
Odelia plays	one	-1	1
	two	1	0

1. What is the value to Odelia of the strategy "with probability 1/2 play one and with probability 1/2 play two"? Remember, to solve this, figure out what Eve would do knowing that this is her strategy.

2. What is the value to Odelia of the strategy "always play two"?

3. What is the strategy for Odelia that has the highest value, and what is its value?

4. What strategy for Eve has the highest value to Eve, and what is its value? Remember, we are assuming now that *Odelia* has spies and will play *her* best response to this strategy. Also, a win for Odelia is a loss for Eve and vice-versa.