ORIE 6300 Mathematical Programming I	November 5, 2008

Recitation 9

Lecturer: Maurice Cheung — *Topic: Minimum-Cost Spanning Trees*

1 Minimum-Cost Spanning Trees

Given a connected graph $G = (V, E)$ with non-negative edge costs c_e, we want to find a minimum-cost subgraph so that every pair of nodes are connected.

We observe that there must exist an optimal solution where the subgraph does not contain any cycle. If the subgraph is connected and a cycle exists, one can delete any edge from the cycle and the resulting graph is still connected with cost no greater than before. Hence we can solve this problem by finding the **minimum-cost spanning tree** (MST). There are several well known algorithms for finding an MST. Here we describe the Kruskal's Algorithm. Kruskal's Algorithm is an example of greedy algorithm: it makes the cheapest choice at each step:

1. Sort the edges E in non-decreasing order, i.e. $c_{e_1} \leq c_{e_2} \leq ... \leq c_{e_m}$.
2. Let $H = (V, F)$ be a spanning forest of G. Initially $F = \emptyset$.
3. For i= 1 to m,
   If $F \cup e_i$ does not contain a cycle, add e_i to F

The straightforward approach for proving correctness of Kruskal's algorithm relies on the fact that if an edge e is the minimum-cost edge in some cut S, then every minimum spanning tree contains the edge e. Instead of using that proof, we will now discuss the connection between MST and linear programming. In particular, we will show that there is a linear programming relaxation for MST where the spanning tree returned by Kruskal's algorithm is an optimal solution. As a consequence, this gives an LP-based proof of the correctness of Kruskal's algorithm.

2 An LP Relaxation for Minimum-Cost Spanning Tree

We have a variable x_e for each edge. We will introduce some notation for writing the LP compactly. For any subset of edges B, let $x(B) := \sum_{e \in B} x_e$. For any subset of vertices S, we let $E(S) :=$ set of edges with both endpoints in S . One possible formulation for an LP relaxation of MST is as follows:

$$
\begin{aligned}
(P): \quad \min \quad & c^T x \\
\text{s.t.} \quad & x(E(S)) \leq |S| - 1, \forall S \subset V \\
& x(E) = |V| - 1 \\
& x_e \geq 0, \forall e \in E
\end{aligned}
$$

From the first set of constraints, taking S to be the two endpoints of any edges ensures that $x_e \leq 1$. The constraints of this LP captures the properties of a spanning tree, namely that it contain no cycle and has $n - 1$ edges ($|V| = n$).

Given a spanning tree T, let x be its characteristic vector (i.e. $x_e = 1$ if $e \in T$ and 0 otherwise). Then x is a feasible solution to the LP above, hence (P) a valid LP-relaxation, and the optimal value of (P) gives an lower bound of the cost of an MST.

For a subset of edges A, we let $\kappa(A)$ be the number of connected components of the subgraph (V, A) of G. Using this notation, we can write another LP-relaxation for MST:

$$\begin{aligned}(P'): \quad \min \quad & c^T x \\ \text{s.t.} \quad & x(A) \leq |V| - \kappa(A), \forall A \subset E \\ & x(E) = |V| - 1 \\ & x_e \geq 0, \forall e \in E\end{aligned}$$

Claim 1 *(P) and (P') are equivalent.*

Proof: First, suppose we have a feasible solution x to (P'). We wish to show it's feasible for (P). Take $A = E(S)$, then we have $x(E(S)) \leq |V| - \kappa(E(S))$. Observe that for the subgraph $(V, E(S))$, the number of connected component is at least $|S| + 1$: one for each vertex not in S, and at least one connected component for S. Hence we have $\kappa(E(S)) \geq |V \setminus S| + 1$. Substituting this into the constraint above gives $x(E(S)) \leq |S| - 1$. The other constraints are common to both LPs; hence x is feasible for (P).

Conversely, suppose x is feasible for (P). For any $A \subseteq E$, let $\kappa(A) = k$ and $S_1, ..., S_k$ be the vertex-set for the components for the subgraph (V, A). Then from the constraint in (P), we have $x(E(S_i)) \leq |S_i| - 1$, but:

$$x(A) \leq \sum_{i=1}^{k} x(E(S)) \leq \sum_{i=1}^{k} |S_i| - 1 = |V| - k$$

Hence x is feasible for (P'), and the two LPs are equivalent. $\square$

3 Analysis of Kruskal's Algorithm

To prove the correctness of Kruskal's algorithm, it suffices to show the characteristic vector of the spanning tree produced by Kruskal's algorithm, x^0, gives an optimal solution to (P') or equivalently (P). This combined with the fact that the optimal value for (P) is a lower bound for the cost of any spanning tree will show x^0 is a minimum-cost spanning tree.

The primal objective can be written as max $-c^T x$. Associating a dual variable y_A for each constraint in (P') and taking the dual gives:

$$\begin{aligned}(D'): \quad \min \quad & \textstyle\sum_{A \subseteq E} (|V| - \kappa(A)) y_A \\ \text{s.t.} \quad & \textstyle\sum (y_A : e \in A) \geq -c_e, \forall e \in E \\ & y_A \geq 0, \forall A \subset E \\ & y_E \text{ free}\end{aligned}$$

To prove x^0 is optimal, we will construct a dual feasible y^0 that satisfies the complementary slackness condition with x^0. Let $e_1, ..., e_m$ be the order in which Kruskal's algorithm considers the edges. Let R_i denote $e_1, ..., e_i$. We define a dual solution y^0 as follows: set $y^0_{R_i} = c_{e_{i+1}} - c_{e_i}$ for $1 \leq i \leq m - 1$ and $y^0_{R_m} = -c_{e_m}$. For all other A, set $y^0_A = 0$.

To check that y^0 is feasible for (D'), notice that $y_A \geq 0, \forall A \subset E$, by construction. Next, since every edge $e = e_i$ for some i:

$$\sum (y^0_A : e \in A) = \sum_{j=i}^{m} y^0_{R_j} = \sum_{j=i}^{m-1} (c_{e_{j+1}} - c_{e_j}) - c_{e_m} = -c_{e_i} = -c_e.$$

Hence all the dual constraints hold at equality. So we know y^0 is feasible. This also gives the complementary slackness condition

$$x_e^0 > 0 \Rightarrow \sum (y_A^0 : e \in A) = -c_e.$$

Finally, if $y_A^0 > 0$, then $A = R_i$ for some $1 \le i \le m-1$. But each stage of Kruskal's Algorithm maintains a spanning forest F, so from properties of spanning forest $|F| = |V| - \kappa(F)$. Now consider the spanning forest when Kruskal's Algorithm scanned $\{e_1, ..., e_i\}$, since the algorithm adds edges to decrease the number of components whenever possible, then we must have

$$x^0(R_i) = |V| - \kappa(F) = |V| - \kappa(R_i)$$

giving us the other complementary slackness condition:

$$y_A^0 > 0 \Rightarrow x_A^0 = |V| - \kappa(A)$$

It follows that x^0 is an optimal solution to (P'), and hence also optimal for (P).