CS 684: Approximation and Network Algorithms
10/12/01
Primal Dual Algorithms for Arborescences

scribe: Anirban Dasgupta

1 Arborescences

In this lecture we look at the problems of finding minimum cost directed trees in a directed graph. Throughout, we'll assume that $G(V, E)$ is our given directed graph. G has a special node, the **root node** r, and a nonnegative cost c_e assigned to each of its edges $e \in E$. We wish to find the minimum cost directed spanning tree (i.e. an *arborescence*) of G, such that, the tree is rooted at the root node r and is directed away from it.

In 681 we solved this problem using a greedy approach. What we show in this lecture is that, the greedy approach was in essence, a primal dual method of solving the problem. In addition, we will also bring in an economic interpretation of the algorithm in terms of "prices paid" in "buying edges" to form the arborescence. the following claim is useful in developing our algorithm.

Claim 1.1 *A* **mincost arborescence** *is a minimum cost set of edges, say F, such that for all subsets S, $S \subseteq V$ satisfying $r \notin S$ and $S \neq \emptyset$, there is at least one edge from F entering S.*

The claim is easy to justify once we note that F is a spanning subgraph and is the minimum cost one of all such set of edges.

2 Intuitive Idea

We will work on an economic interpretation which is approximately as follows. Nodes $v \in V$ will set prices y_v to buy edges in F. True to the economic interpretation, every node would set its price equal to the minimum cost of any edge entering it. Set price of v as $y_v = \min\{c_e \mid e \text{ enters } v\}$. But this may not give us a directed spanning tree yet. So, following our primal-dual scheme, we define a reduced cost for the edges as $c_e^Y = c_e - y_v$ if edge $e = (w, v)$ i.e. edge e enters node v. The idea is that, this is the least amount that the nodes can hope to pay and be connected to any arborescence. Later though, we will see that this definition of reduced cost would have to be modified slightly.

In the reduced cost graph we then consider the **zero-cost** subgraph and see if we can find out a arborescence from there. If we can, we're done. Else we would have to increment the costs and look at the graph again. The following figure gives us a typical example of the sort of zero-cost structure we might end up with once each node pays for the cheapest edge entering it.

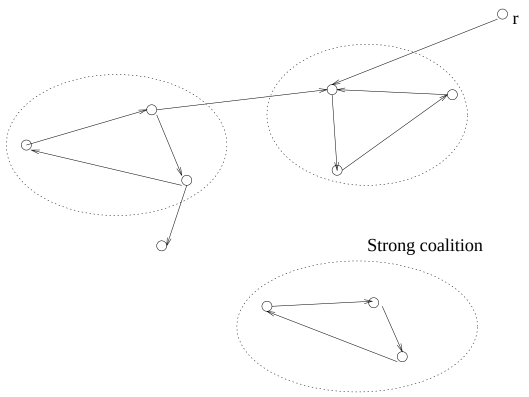

Figure 1: After choosing the zero cost edges

We might end up with few isolated directed circles and possibly circles hanging off a directed tree rooted at r. For each of the "groups"(strongly connected components) that has formed, we know that we can find out a arborescence for it. In view of this, we now slightly change our idea of pricing. We now assign prices to subsets of nodes. For each set of nodes, $S \subseteq V$, $S \neq \emptyset$ and $r \notin S$, assign a price y_S. If an edge e enters the set S, then S pays for e. Conversely, edge e is paid for by all the sets S such that e enters S. The prices then satisfy the following conditions:

- The prices are nonnegative. $\forall S \neq \emptyset,\ r \notin S\ y_S \geq 0$
- We do not overpay for any edge. $c_e \geq \sum_{e \text{ enters } S} y_S$.

In keeping with our primal-dual scheme, we bring in the concept of equilibrium in the prices and the edge costs.

Definition 2.1 *We say that the arborescence edges and the prices are in equilibrium if the following conditions hold :*

1. *The prices and edge costs satisfy the above requirements.*
2. *The arborescence uses only those edges for which the following equality holds.* $c_e = \sum_{e \text{ enters } S} y_S$.
3. *The sets with* $y_S > 0$ *should have exactly one edge entering from arborescence.*

As an aside, when we approach the problem from this pricing viewpoint, we should make sure that we do not have too many sets with $y_S > 0$, as we would definitely want a polytime algorithm.

We have the following theorem justifying our idea of equilibrium.

Theorem 2.2 *If the arborescence* T *and the prices* y_S *are in equilibrium, then* T *is the mincost arborescence.*

Proof. Let $P(e) = \{S \subseteq V \mid S \neq \emptyset,\ r \notin S, e \text{ enters } S\}$. Now, $\sum_{e \in T} c_e = \sum_{e \in T} \sum_{S \in P(e)} y_S = \sum_S y_S(\#T \text{ edges entering } S) = \sum_S y_S$, as according to equilibrium, there is an unique edge entering each set S for which $y_S > 0$. Let T' be another arborescence. Then, $\sum_{e \in T'} c_e \geq \sum_{e \in T'} \sum_{S \in P(e)} y_S = \sum_S y_S(\#T' \text{ edges entering } S) \geq \sum_S y_S$. Hence, T gives us the minimum cost arborescence.

3 Primal Dual Algorithm

The algorithm for the mincost arborescence can be formalized as follows.

1. At each step, maintain the costs y and the set of edges T, which is not an arborescence yet. The invariant throughout the steps is that y and T are in equilibrium.

2. Start with $y_S = 0$ for all S. And $T = \emptyset$. The algorithm works greedily on the dual variables, i.e. the prices y_S.

3. Consider all the edges that can be in the tree, i.e. all edges for which $c_e = \sum_{e \text{ enters } S} y_S$. If there is a directed tree using these edges, we're done.

4. If not, consider the strongly connected components formed by these edges. . View these strongly connected components as strong coalitions. If all these components S are such that no edge enters S, we need to change the prices.

5. Set the reduced cost for each edge e as $c_e^Y = c_e - \sum_{S \in P(e)} y_S$. Also set $y_S = min\{c_e^Y \mid S \in P(e)\}$.

6. Contract each such strong coalition S into a simple node.

7. Solve the contracted problem recursively with reduced cost.

The correctness of the contraction step of the algorithm is based on the following claim.

Claim 3.1 *A spanning tree of* 0*-reduced cost edges in contracted graph extends to a mincost arborescence in the original graph.*

An important point to take home from this algorithm is that contraction does not really play a role in the price setting. We use contraction as we want to ensure the property of exactly one edge entering a set S when $y_S > 0$. Contraction of strong coalitions into simple nodes provides us with a clean way of maintaining that.

4 The LP Interpretation

Finally, we take a look at the linear program corresponding to the problem and justify our choice of the sets as the entities paying the prices. If we use the initial **Claim** 1.1 about the arborescence, we can write down the linear program for the arborescence as follows. Remember $P(e) = \{S \subseteq V \mid S \neq \emptyset,\ r \notin S, e \text{ enters } S\}$.

$X_e \in \{0,1\} \forall e \in E$.
X_e =1 if $e \in T$
    0 else.
For all $S \subseteq V$, such that $S \neq \emptyset$, $r \notin S$,
$\sum_{S \in P(e)} X_e \geq 1$.
And finally $min \sum_{e \in E} c_e X_e$.

If we look at the dual linear program of this one, we'll get a dual variable for each of the inequalities. Since the inequalities are in terms of sets, the dual variables y are defined as the prices paid by sets.