COMPSCI 530: Design and Analysis of Algorithms	11/14/2013
Lecture 22	
Lecturer: Debmalya Panigrahi	*Scribe: Abhinandan Nath*

1 Overview

In the last class, the primal-dual method was introduced through the metric facility location problem. This lecture discusses two more problems where the primal-dual schema is used to give exact and approximate algorithms respectively for the shortest path and the steiner forest problem.

2 Shortest $s-t$ path

Given an undirected graph with non-negative costs associated with the edges, we want to find a shortest path between two specified vertices s and t. In this section, we will develop a primal-dual algorithm for this problem, which will turn out to be Dijkstra's algorithm in a different guise.

2.1 The LP

The LP for the problem and its dual are given below

$$\begin{aligned} &\min \sum_{e\in E} c_e x_e \\ &\sum_{e\in\delta(S)} x_e \geq 1 \quad \forall S : s\in S, t\notin S \\ &x_e \geq 0 \quad \forall e\in E \end{aligned}$$

$$\begin{aligned} &\max \sum_{S} y_S \\ &\sum_{S:e\in\delta(S)} y_S \leq c_e \quad \forall e\in E \\ &y_S \geq 0 \quad \forall S : s\in S, t\notin S \end{aligned}$$

2.2 The Algorithm

The primal-dual method increases the dual variables gradually until some dual constraint becomes tight. Then, the primal variable corresponding to the tight dual constraint is 'bought' (or selected), and the process continues till we get a feasible primal solution. Next, we compare the value of the primal solution to the value of the dual solution to get an appropriate approximation factor (or an exact algorithm). By weak duality, the ratio between the primal and the dual value is an upperbound on the approximation factor.

In the current problem, at each step we have to choose what dual variables we increase. We increase the dual variable corresponding to the connected component containing s. Initially, it is the set $\{s\}$. Once a constraint becomes tight, the corresponding edge is selected and the component grows. This continues till the component contains t. However we might have included too many edges. So we do a clean up step at the end to retain only those edges necessary to ensure an $s-t$ path. The algorithm is stated below

Algorithm 1 Primal-Dual s-t path

$y \leftarrow 0$
$A \leftarrow \phi$
$i \leftarrow 0$
while A is not feasible **do**
  $i \leftarrow i+1$
  Let S be the connected component containing s
  Increase y_S until $\exists e_i \in \delta(S) : \sum_{S:e_i\in\delta(S)} y_s = c_{e_i}$
  $A \leftarrow A \cup \{e_i\}$
end while
for $j \leftarrow i$ down to 1 **do**
  if $A - \{e_j\}$ is still feasible **then**
    $A \leftarrow A - \{e_j\}$
  end if
end for
return $A' \leftarrow A$

2.3 Analysis

We will argue that the cost of the edges included in A' is exactly equal to the dual value, thereby giving an exact algorithm. We have

$$\sum_{e\in A'} c_e = \sum_{e\in A'} \sum_{S:e\in\delta(S)} y_S = \sum_S y_S |A' \cap \delta(S)|$$

The following lemma will then imply that the value of the solution is exactly equal to the dual value.

Lemma 1. *For all S such that $y_S > 0$, $|A' \cap \delta(S)| = 1$.*

Proof. Consider the stage when we increase a particular y_S. Let A_S be the edges in S at this stage. Letting $B = A' - A_S$, we see that in the clean up phase, all the edges in B were considered for deletion before all the edges in A_S and were retained because deleting them would make the solution infeasible. Now, $A' \cap \delta(S)$ will contain edges only from B. Consider an $s-t$ path P in $A_S \cup B$. Suppose it crosses the cut S multiple times. Consider the last such edge (u,v). Since u is in the same connected component as s, there exists a path entirely in A_S from s to u, and this path along with the remaining part of P from v to t gives an $s-t$ path. Thus, all the remaining edges of P crossing S are unnecessary, and would have been deleted in the clean up phase. Thus, all $s-t$ paths in A_S cross S only once. Also, all these paths converge and cross S at only one edge, because if we have multiple crossing edges, all but one would have been deleted in the clean up phase because deleting them still retains an $s-t$ path. Any other crossing edge not part of an $s-t$ path will also be deleted in the clean-up phase, thus implying that $|A' \cap \delta(S)| = 1$. $\square$

3 Steiner forest

In the Steiner forest problem, we are given an undirected graph $G = (V,E)$ with costs $c_e \geq 0$ for each edge e. We are also given l pairs of vertices $\{(s_i,t_i) : i = 1,\dots,l)\}$. Our goal is to choose a minimum cost subset of edges so that the for all i, s_i and t_i are are connected.

3.1 The LP

Let $S_i = \{S \subseteq V : |S \cap \{s_i, t_i\}| = 1\}$. A linear program and its dual for the Steiner forest problem are given below

$$\min \sum_{e \in E} c_e x_e$$
$$\sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S : S \in S_i \text{ for some } i$$
$$x_e \geq 0 \quad \forall e \in E$$

$$\max \sum_{S : S \in S_i \text{ for some } i} y_S$$
$$\sum_{S : e \in \delta(S)} y_S \leq c_e \quad \forall e \in E$$
$$y_S \geq 0 \quad \forall S : S \in S_i \text{ for some } i$$

3.2 The Algorithm

We use an approach similar to the previous problem, but now we increase the values simultaneously and unifromly for all the connected components not having both s_j and t_j for any j. The algorithm is stated below

Algorithm 2 Primal-Dual Steiner forest

$y \leftarrow 0$
$A \leftarrow \phi$
$i \leftarrow 0$
while A is not feasible **do**
  $i \leftarrow i + 1$
  $C_i \leftarrow \{S : S \text{ is a connected component of } (V, A) : |S \cap \{s_j, t_j\} = 1 \text{ for some } j\}$
  Increase y_S for all $S \in C_i$ until $\exists e_i \notin A : \sum_{S : e_i \in \delta(S)} y_s = c_{e_i}$
  $A \leftarrow A \cup \{e_i\}$
end while
for $j \leftarrow i$ down to 1 **do**
  if $A - \{e_j\}$ is still feasible **then**
    $A \leftarrow A - \{e_j\}$
  end if
end for
return $A' \leftarrow A$

4 Analysis

Lemma 2. *For any i,*

$$\sum_{C \in C_i} |A' \cap \delta(C)| \leq 2|C_i|$$

Proof. Let A_i denote the set of edges chosen till iteration i. Let $B = A' - A_i$. Then $A_i \cup B$ is a feasible solution, but removing any edges from it renders the set infeasible.

Consider the graph (V', B) obtained by contracting the connected components induced by A_i. We call the vertices corresponding to components in C_i *red* vertices, and the remaining are called *blue* vertices. The edges in B form a forest, because if not then removing an edge of a cycle in B still maintains feasibility, cotradicting the definition of B. Then, $|A' \cap \delta(C)|$ is same as the degree of the vertex corresponding to $C \in C_i$ in the graph (V', B). Thus we need to show that

$$\sum_{v \in Red} deg(v) \leq 2|Red|$$

For this, we need the following claim

Claim 3. *If $v \in Blue$ then $deg(v) \neq 1$*

Proof. Suppose there exists a blue vertex v such that $deg(v) = 1$. Suppose we delete the edge e incident to v. Since v is a blue vertex, there does not exist any (s_i, t_i) such that $|C \cap \{s_i, t_i\}| = 1$, where C is the connected component corresponding to v. Thus, deleting e still keeps the set of edges feasible, contradicting the definition of B. $\square$

Thus we have,

$$\begin{aligned}\sum_{v \in Red} deg(v) &= \sum_{v \in Red \cup Blue} deg(v) - \sum_{v \in Blue} deg(v) \\ &\leq 2(|Red| + |Blue|) - 2|Blue| \\ &= 2|Red|\end{aligned}$$

The second step uses the fact that the sum of the degrees in a forest is at most twice the number of vertices. $\square$

Using lemma 2 we prove the following, thereby showing a 2-approximation factor.

Theorem 4.

$$\sum_{e \in A'} c_e \leq 2 \sum_S y_S$$

Proof. We have,

$$\sum_{e \in A'} c_e = \sum_{e \in A'} \sum_{S : e \in \delta(S)} y_S = \sum_S y_S |A' \cap \delta(S)|$$

Thus we have to show

$$\sum_S y_S |A' \cap \delta(S)| \leq 2 \sum_S y_S$$

We use induction on the number of iterations of the algorithm to prove the claim. Initially, $A = \phi$ and $y = 0$ and the inequality holds trivially.

Suppose the claim holds for the ith iteration. In the $(i+1)$th iteration, suppose we increase the dual variables unifromly by ε. Then the LHS of the inequality increases by $\varepsilon \sum_{S \in C_{i+1}} |A' \cap \delta(S)|$. The RHS increases by $2\varepsilon |C_{i+1}|$, which is at least the increase in the LHS by lemma 2. Thus the inequality holds after the $(i+1)$th iteration too. $\square$

5 Summary

In this lecture, we have illustrated the primal-dual method by giving an exact algorithm for the single pair shortest path problem and a 2-approximation algorithm for the Steiner forest problem.

References

[1] *http://courses.csail.mit.edu/6.891-s00/lecture8.ps*