COMPSCI 638: Graph Algorithms	October 11, 2019
Lecture 14	
Lecturer: Debmalya Panigrahi	*Scribe: Kevin Sun*

1 Overview

In this lecture, we continue studying the Generalized Steiner Forest problem. Last lecture, we saw a primal-dual algorithm, and now, we will use a new technique known as iterative rounding.

2 Iterative Rounding for Generalized Steiner Forest

Recall that in the Generalized Steiner Forest (GSF) problem, we are given an undirected graph $G = (V, E)$ where each edge e has weight $c(e) \geq 0$, and a set of k terminals pairs (s_i, t_i) for $i = 1, \ldots, k$. Each (s_i, t_i) pair has a demand value r_i. We seek a minimum-cost subset of edges F such that for every (s_i, t_i) pair, there are at least r_i edge-disjoint paths in (V, F).

Recall from Lecture 13 that the LP relaxation of the GSF problem is the following:

$$\text{(P): } \min \sum_{e \in E} c(e) x_e$$
$$\sum_{e \in \delta(S)} x_e \geq f(S) \quad \forall S \subseteq V$$
$$1 \geq x_e \geq 0 \quad \forall e \in E,$$

where $f(S) = \max_{i: S \in \mathcal{S}_i} r_i$, $\mathcal{S}_i$ denotes the subsets of V that separate s_i and t_i, and $\delta(S)$ denotes the subset of edges with exactly one endpoint in S. Also recall that this demand function f is *weakly supermodular*, that is, $f(\emptyset) = f(V) = 0$ and for every $A, B \subseteq V$,

$$f(A) + f(B) \leq \max\{f(A \cup B) + f(A \cap B), f(A \setminus B) + f(B \setminus A)\}$$

Our algorithm today, due to Jain [Jai01], relies on a remarkable property about solutions to (P) whenever the demand function f is weakly supermodular.

Basic feasible solutions: In any linear program, a *basic feasible solution* (BFS) corresponds to an extreme point of the polytope defined by the constraints. That is, a BFS cannot be written as the convex combination of two other feasible solutions. Algebraically, if the LP has n variables, then a BFS satisfies n linearly independent constraints with equality.

Many algorithms, including the one we present today, use the fact that if an LP has an optimal solution, then there exists a BFS that is also optimal, also known as a *basic optimal solution*. For our purposes, we assume that a basic optimal solution of (P) can quickly be obtained.

We now state the crucial property that gives rise to Jain's algorithm. We defer the proof of this lemma until the next lecture.

Lemma 1. *Let $f : 2^V \to \mathbb{Z}$ be a weakly supermodular function and let x be any basic feasible solution to (P). If $x_e \in (0, 1)$ for all $e \in E$, then there exists some e such that $x_e \geq 1/2$.*

The intuition behind the algorithm is similar to the algorithms for the GSF problem presented in Lecture 13. We begin with a weakly supermodular demand function f, and in each iteration, we solve (P) to obtain a BFS x that (fractionally) satisfies the demand. By Lemma 1, there exists some $x_e \geq 1/2$, so we add at least this edge to our overall solution. This iterative rounding procedure is repeated until all demands are satisfied, we'll show that the approximation ratio is at most two.

Algorithm 1 Iterative Rounding for GSF

Initialize: $F = \emptyset, f_{res} = f$.
while F is not a feasible solution **do**
    Let x be a BFS to (P) with demand function f_{res} and edge set $E \setminus F$.
    if there exists e such that $x_e = 0$ **then**
        Remove e from E and continue (to the next iteration).
    Add any (single) edge e satisfying $x_e \geq 1/2$ to F.
    For each $S \subseteq V$, set $f_{res}(S) = \max\{0, f(S) - |\delta(S) \cap F|\}$.
return F

Theorem 2. *Algorithm 1 is a 2-approximation for the GSF problem.*

Proof. By Lemma 1, in every iteration, we either remove some edge from E or add some edge to F. Edges that we remove can be ignored, and whenever we add an edge to F, $f_{res}(S)$ decreases by at least one for some S. This implies the algorithm terminates and returns a feasible solution.

Now we induct on the number of iterations k to bound the approximation ratio. Note that we are only concerned with iterations that add an edge to F. For any such iteration i, let $x^{(i)}$ denote the BFS obtained in that iteration, and let e_i denote the edge added to F. Furthermore, for any $E' \subseteq E$, let $c(E') = \sum_{e \in E'} c(e)$. Finally, let OPT denote the value of an optimal solution.

If $k = 1$, then $F = \{e_1\}$ is our solution and $c(e_1) \leq 2c(e_1)x_{e_1}^{(1)} \leq 2 \cdot OPT$. Now suppose Algorithm 1 runs for $k+1$ iterations. After removing e_1 from E and adding it to F, the algorithm then runs for k iterations and adds all the remaining edges of F. By the induction hypothesis, the cost of these edges is at most twice the cost of (P) with solution $x^{(2)}$. In other words, we have

$$c(F) = c(e_1) + c(F \setminus \{e_1\}) \leq 2\left(c(e_1)x_{e_1}^{(1)} + \sum_{e \in E \setminus \{e_1\}} c(e)x_e^{(2)}\right). \tag{1}$$

Now we claim that $x^{(1)}$ is a feasible solution to the second LP (which has edge set $E \setminus \{e_1\}$) by considering the difference between the two linear programs. Constraints that do not involve e_1 remain unchanged, so they are still satisfied by $x^{(1)}$. Any constraint that involves e_1 has its demand reduced by one, so removing $x_{e_1}^{(1)} \leq 1$ from the sum does not affect feasibility either. Thus, we have

$$\sum_{e \in E \setminus \{e_1\}} c(e)x_e^{(2)} \leq \sum_{e \in E \setminus \{e_1\}} c(e)x_e^{(1)}$$

Combining this with (1), we can conclude

$$c(F) \leq 2\sum_{e \in E} c(e)x_e^{(1)} \leq 2 \cdot OPT. \qquad \square$$

2.1 Proof of Lemma 1

Let x be a BFS, and for contradiction, assume $x_e \in (0, 1/2)$ for every $e \in E$. At a high level, our proof has two steps: we will first find a laminar family $\mathcal{L}$ on V containing $|E|$ sets that satisfy some key properties, and then we will apply a token-counting argument on $\mathcal{L}$ to obtain a contradiction. (Recall that a family of sets is *laminar* if, for any two non-disjoint sets, one is a subset of the other.)

First, we establish some terminology and notation. Two subsets *cross* if their intersection is non-empty and neither is a subset of the other. For any $E' \subseteq E$, we let $x(E') = \sum_{e \in E'} x_e$. We say a set $S \subseteq V$ is *tight* if its corresponding constraint in (P) is tight, that is, $x(\delta(S)) = f(S)$. Assuming a fixed order of E, for any $E' \subseteq E$, we let $\chi(E')$ denote the *incidence vector* of E': the e-th coordinate of $\chi(E')$ is 1 if $e \in E'$ and 0 otherwise. Finally, we say $E_1, E_2 \subseteq E$ are *linearly independent* if $\chi(E_1), \chi(E_2)$ are linearly independent.

Lemma 3. *There exists a laminar family $\mathcal{L}$ containing $|E|$ tight subsets of V. Furthermore, for any $A \neq B \in \mathcal{L}$, the incidence vectors of $\delta(A)$ and $\delta(B)$ are linearly independent.*

Proof. Since x is a BFS, there exist $|E|$ tight subsets of V, but they do not necessarily form a laminar family. To rectify this, we apply a process known as *uncrossing*. In particular, suppose A and B are tight crossing subsets of V, and f (being weakly supermodular) satisfies

$$f(A) + f(B) \leq f(A \cup B) + f(A \cap B). \tag{2}$$

(The proof is essentially identical if f satisfies the other inequality for weak supermodularity.) Recall from Lecture 13 that x, being a cut function, satisfies

$$x(\delta(A \cup B)) + x(\delta(A \cap B)) \leq x(\delta(A)) + x(\delta(B)) = f(A) + f(B), \tag{3}$$

where the equality holds because A and B are tight. Finally, since x is feasible to (P), it satisfies the primal inequalities corresponding to the sets $A \cup B$ and $A \cap B$, which means

$$f(A \cup B) + f(A \cap B) \leq x(\delta(A \cup B)) + x(\delta(A \cap B)). \tag{4}$$

By combining (2), (3), and (4), we see that they all hold with equality. In particular, equality in (4) and feasibility of x imply that $A \cup B$ and $A \cap B$ are tight. Thus, we can repeatedly replace any two sets A and B with $A \cup B$ and $A \cap B$, and the result is a laminar family $\mathcal{L}$ containing $|E|$ tight subsets.

We now show that the subsets in $\mathcal{L}$ are linearly independent by showing that $\mathcal{L}$ spans the vector space spanned by the tight subsets. For contradiction, let A be a tight subset such that $\chi(\delta(A))$ is outside the span of $\mathcal{L}$ and the number of sets in $\mathcal{L}$ that cross A is minimized. Let B be any subset in $\mathcal{L}$ that intersects A. (If no such subset existed, then A would be a member of $\mathcal{L}$.)

From the discussion above, we know $A \cup B$ and $A \cap B$ are both tight. Furthermore, we have

$$\chi(\delta(A)) + \chi(\delta(B)) = \chi(\delta(A \cup B)) + \chi(\delta(A \cap B)).$$

Since $\chi(\delta(A))$ is not in $\text{span}(\mathcal{L})$ while B is in $\mathcal{L}$, it must be the case that at least one of the vectors $\chi(\delta(A \cup B)), \chi(\delta(A \cap B))$ lies outside $\text{span}(\mathcal{L})$. But both $A \cup B$ and $A \cap B$ cross fewer subsets of $\mathcal{L}$ than A, so whichever is not in $\text{span}(\mathcal{L})$ violates our choice of A. □

Let $\mathcal{L}$ be the laminar family containing $|E|$ tight linearly independent subsets of V given by Lemma 3. For any $x, y \in V$, let S_x denote the smallest set in $\mathcal{L}$ that contains x (if such a set exists) and let $S_{x,y}$ denote the smallest set in $\mathcal{L}$ that contains both x and y (if such a set exists).

Token counting: Recall that $x_e \in (0, 1/2)$ for every $e \in E$. Now consider the following procedure: each edge $e = \{u, v\}$ is initially given one token. The edge then distributes x_e tokens to S_u, x_e tokens to S_v, and the remaining $1 - 2x_e$ tokens to $S_{u,v}$. If S_u, S_v, or $S_{u,v}$ is not defined, then e retains the corresponding tokens. We will show the following facts about this process:

Fact 4. *At least one edge retains some positive amount of tokens.*

Fact 5. *Every set in $\mathcal{L}$ receives at least 1 token.*

Observe that these two facts prove Lemma 1: there are $|E|$ edges and each gets one token, so Fact 4 implies that fewer than $|E|$ tokens are distributed to the sets. However, $\mathcal{L}$ has $|E|$ sets, so Fact 5 implies that at least $|E|$ tokens are received by the sets, contradicting the previous statement.

Proof of Fact 4. Let S be a maximal set in $\mathcal{L}$. Since S corresponds to a tight constraint of (P), $\delta(S)$ must contain some edge $e = \{u, v\}$. Since S is maximal and $\mathcal{L}$ is laminar, there is no set that contains both u and v, so $S_{u,v}$ is not defined. This means e retains at least $1 - 2x_e$ tokens. □

Proof of Fact 5. Let S be a set in $\mathcal{L}$ and let $t(S)$ denote the amount of tokens received by S; we want to show $t(S) \geq 1$. To do this, we will show $t(S) > 0$ and $t(S)$ is integer. (Note that this is quite a counterintuitive approach to proving such an inequality.)

Let $R_1, R_2, \ldots$ denote the maximal sets of $\mathcal{L}$ contained in S, $R = \cup_i R_i$, and $S' = S \setminus \cup_i R_i$. Then any edge $e = \{u, v\}$ that contributes a fractional amount to $t(S)$ is one of the following four types:

(a) $u \in S', v \notin S$: Edge e contributes x_e tokens to S.

(b) $u \in R_i, v \in R_j$ where $i \neq j$: Edge e contributes $1 - 2x_e$ tokens to S.

(c) $u \in S', v \in R$: Edge e contributes $x_e + 1 - 2x_e = 1 - x_e$ tokens to S.

(d) $u \in R, v \notin S$: These edges contribute 0 tokens to S, but they are in $\delta(S)$.

Note that it is not possible for all edges in $\delta(S)$ to be type (d), because that would imply $\chi(S) = \sum_i \chi(R_i)$, contradicting the fact that the sets in $\mathcal{L}$ are linearly independent. Therefore, an edge of the form (a), (b), or (c) must exist, so $t(S) > 0$.

More specifically, let a, b, c denote the number of edges of type (a), (b), (c), respectively, and x_a, x_b, x_c denote the total amount of tokens distributed to S by edge of type (a), (b), (c) respectively. Then adding the three kinds of contributions, we see that

$$t(S) = x_a + b - 2x_b + c - x_c.$$

We want to show $t(S)$ is an integer, so it suffices to show $x_a - 2x_b - x_c$ is an integer. Since S is tight, $x_a + x_d = f(S)$ is an integer. Furthermore, since every R_i is tight, $2x_b + x_c + x_d = \sum_i f(R_i)$ is also an integer. Subtracting these two quantities, we see

$$x_a - 2x_b - x_c = x_a + x_d - (2x_b + x_c + x_d) = f(S) - \sum_i f(R_i)$$

is also an integer, as desired. □

3 Summary

In this lecture, we gave an iterative rounding algorithm for the Steiner forest problem due to Jain [Jai01]. The algorithm relies on a key property of basic feasible solutions, and in proving it, we saw techniques involving laminar families, uncrossing, and linear independence.

References

[Jai01] Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.