CMPUT 675: Approximation Algorithms **Fall 2011**

Lecture 10,11 (Oct 11 and 13, 2011): Survivable Network Design

Lecturer: Mohammad R. Salavatipour *Scribe: Seyed Sina Khankhajeh*

10.1 Survivable Network Design

Steiner Network or Survivable Network Design problem is a generalization of the Steiner forest problem and is the most general form of connectivity requirement network design problem. The problem definition is as follows:

Survivable Network Design: Given an undirected graph $G = (V, E)$ and cost function on edges $c : E \to \mathbb{R}^+$ and a connectivity requirement $r(u, v)$ for any pair $u, v \in V$.
Goal: find a minimum cost subgraph H which has at least $r(u, v)$ edge-disjoint paths between u, v for every pair u, v.

Examples: If $r(u, v) = 1$ for all pairs then our problem is the Minimum Spanning Tree problem.
If $r(u, v) = 1$ for all $u, v \in T$ for a set $T \subseteq V$ then our problem is actually Steiner Tree.
If $r(u, v) \in \{0, 1\}$ for all pairs then our problem is actually Steiner Forest(Generalized Steiner Tree).
If $r(u, v) = k$ for some given k, for all pairs then our problem is actually Minimum Cost k-edge-connected subgraph problem.
If $r(u, v) = 2$ for all pairs then our problem is actually Minimum Cost 2-edge-connected subgraph problem.

If $k = \max r(u, v)$, a $2k$-approximation is given by [1]. Later, a $2H_k$-approximation came out by Goemans et al. [2] where $H_k = \sum_{i=1}^{k} \frac{1}{i}$. After that, Jain gave a 2-approximation algorithm [3] in which he used iterative rounding that is now known as Jain's rounding. It's an inefficient algorithms but that is the only 2-approximation algorithm that we know so far.

10.1.1 Jain's rounding

Before we express an LP-relaxation of the problem we need a few definitions.

Definition 1 *The cut requirement function $f : 2^V \to \mathbb{Z}^+, \forall S \subseteq V$ is the largest requirement across the cut $S, \bar{S}$. In other words, $f(s) = \max_{u,v} r(u, v)$ s.t. $u \in S, v \notin S$.*

Definition 2 *We define U_e to be maximum number of copies of edge e that you can pick. In a simple graph it would be 1. But in this problem, it is possible to have a graph with multiple edges.*

The following is a natural LP relaxation of the problem:

x_e is a variable that shows if we should pick edge e or not. We also define $\delta_x(S) = \sum_{e\in\delta(S)} x_e$.

Definition 3 *Suppose that we have picked a set of edges so far, say $H \subseteq E$. For each set $S \subseteq V$ we define the residual cut requirement as $f'(S) = f(S) - |\delta_H(S)|$.*

$$\begin{array}{lllll} \min & \sum_{e\in E} c_e \cdot x_e & & & \\ \text{s.t.} & \sum_{e\in\delta(S)} x_e \geq & f(S) & \forall S \in V & (1) \\ & 0 \leq x_e \leq U_e & \forall e \in E & & (2) \end{array}$$

Note that f' may not correspond to any connectivity requirement function $r(u, v)$. We now take a look at Jain's iterative rounding algorithm:

Jain's rounding

```
1. $H \leftarrow \emptyset$
2. $f' \leftarrow f$
3. **while** $H$ is not feasible **do**
4.   find an optimum extreme point solution (basic feasible solution) $x$ for graph $G = (V, E - H)$ using $f'$
5.   delete edges with $x_e = 0$
6.   for each edge $e$ with $x_e \geq \frac{1}{2}$ do
7.       add $\lceil x_e \rceil$ copies of $e$ to $H$
8.   for every $S \subseteq V : f'(S) \leftarrow f(S) - |\delta_H(S)|$
9. **return** $H$
```

Figure 10.1: Jain's iterative rounding

What is interesting about this method is that, not only we can use it to derive some approximation algorithms, but also you can prove some of the classical results in optimization theory as we'll see in the next lecture.

Note that although the LP has exponentially many constraints we can solve this using Ellipsoid algorithm if we have a separation oracle for it. For every pair of vertices $u, v \in V(G)$ we can run a max-flow-min-cut from u to v with these x values as flows to find a min-cut between u, v. This min-cut gives you a cut S and then you need to have at least $r(u, v)$ in this cut and the capacity of the cut with these values has to be at least $r(u, v)$. If a min-cut with less capacity than $r(u, v)$ is found, then this cut represents a violated constraint.

For updating f', imagine an arbitrary iteration. Given x' we want to check feasibility using ellipsoid algorithm. We define $x_e = x'_e + e_H$, where e_H is the number of copies of e added to H so far.

Lemma 1 *Cut $(S, \bar{S})$ is violated by x' under f', if and only if it is violated by x under f.*

Proof. We know that:

$$\delta_x(S) = \delta'_x(S) + |\delta_H(S)|$$

$$f(S) = f'(S) + |\delta_H(S)|$$

So we can easily conclude that:

$$\delta_x(S) \geq f(S) \Longleftrightarrow \delta'_x(S) \geq f'(S)$$

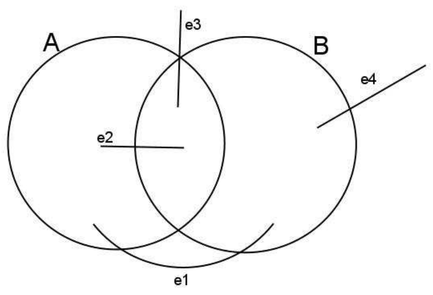

Figure 10.2: Types of edges for two crossing sets

■

So, you don't need to update f' in each iteration, instead you can work with function f. The update statement in Jain's rounding is for easier description of algorithm.

Until now we have proved that Jain's algorithm can be implemented to run in polynomial time. Now we should prove that algorithm gives us a 2-approximation. But first we need to define some tools to help us in our proof.

Definition 4 *A function $g : 2^V \to \mathbb{Z}^+$ is (strongly) sub-modular if $g(V) = 0$ and $\forall A, B \subseteq V$ the following two hold:*

1. $g(A) + g(B) \geq g(A \cup B) + g(A \cap B)$
2. $g(A) + g(B) \geq g(A - B) + g(B - A)$

Definition 5 *Two sets A, B cross each other if $A - B \neq \emptyset$, $B - A \neq \emptyset$ and $A \cap B \neq \emptyset$.*

Note that containment is not considered crossing.

Lemma 2 *The function $|\delta(S)|$ is sub-modular.*

Proof. Given $A, B \subseteq V$, we prove the first condition holds. The proof of second condition is very similar to the first one. If $A \cap B = \emptyset$ or $A \subseteq B$ or $B \subseteq A$ then the statement is easy to check. Otherwise, there are four type of edges we need to consider as shown in Figure 10.2 .

our statement is: $|\delta(A)| + |\delta(B)| \geq |\delta(A \cup B)| + |\delta(A \cap B)|$. Edges of type e_1 only contribute to the left hand side and edges of type e_2, e_3, e_4 contribute equally to both sides. The reason is, edges of type e_1 are in cuts of A and B but not in cuts of $A \cup B$ or $A \cap B$, and edges of type e_2, e_3, e_4 appear the same number of times in cuts of A and B as in cuts of $A \cup B$ and $A \cap B$. So the statement must be true. ■

Definition 6 *A function $g : 2^V \to \mathbb{Z}^+$ is weakly super-modular if $g(V) = 0$ and $\forall A, B \subseteq V$ at least one of the following two holds:*

1. $g(A) + g(B) \leq g(A \cup B) + g(A \cap B)$
2. $g(A) + g(B) \leq g(A - B) + g(B - A)$

Lemma 3 *f, f' are weakly super-modular.*

Proof. Since f is coming from $r(u, v)$, it is easy to prove that f is weakly super-modular. We show that f' is weakly super-modular. Suppose that for a given pair $A, B \subseteq V$, first condition holds for f (the proof when second condition is true is the same is the first). So we know that:

$$f(A) + f(B) \leq f(A \cup B) + f(A \cap B)$$

By Lemma 2, we know that $|\delta(S)|$ is sub-modular (for any graph H):

$$|\delta_H(A)| + |\delta_H(B)| \geq |\delta_H(A \cup B)| + |\delta_H(A \cap B)|$$

Since the in the first statement left hand side is less than or equal to right hand side and for second statement it is inverse, so we can subtract second statement from first statement and we will have:

$$(f(A) - |\delta_H(A)|) + (f(B) - |\delta_H(B)|) \leq (f(A \cup B) - |\delta_H(A \cup B)|) + (f(A \cap B) - |\delta_H(A \cap B)|)$$

And we know that as we defined before, $f(S) - |\delta_H(S)| = f'(S)$. Therefore:

$$f'(A) + f'(B) \leq f'(A \cup B) + f'(A \cap B)$$

■

We will show the following:

Theorem 1 *For any weakly super-modular function f' and a best feasible solution x to the LP with f', there is some edge e with $x_e \geq \frac{1}{2}$.*

Assuming this theorem we complete the remaining part of our proof. Note that this theorem shows that Jain's algorithm has an approximation ratio of 2 for any weakly super-modular function f', not just one that comes from a cut requirement function f.

Theorem 2 *Jain's algorithm is a 2-approximation.*

Proof. We prove this theorem by induction on the number of iterations. Let Z_{LP} be the optimum value of the original LP. If there is only one iteration then clearly our solution costs at most $2Z_{LP}$. For the induction step suppose there is an edge e with $x_e \geq \frac{1}{2}$ that is added to H. By induction hypothesis the algorithm finds a solution of cost at most $2Z'_{LP}$ where Z'_{LP} is the optimum value of the new residual LP (which has one less iterations than the original LP). We claim that $Z'_{LP} \leq Z_{LP} - c_e x_e$. This is because the restriction of x to the edges in $E - e$ is feasible for the residual LP. In other words:

$$\forall S \subseteq V,\ \delta_x(S) \geq f(S) \Longrightarrow \delta_x(S) - x_e \geq f'(S)$$

Therefore,

$$Z'_{LP} \leq Z_{LP} - c_e x_e \leq Z_{LP} - \frac{1}{2}c_e$$

So, if we show the whole cost of Graph H with $c(H)$, considering our induction hypothesis, we will have:

$$c(H) \leq c_e + 2Z'_{LP} \leq 2Z_{LP}$$

■

10.1.2 Characterization of basic feasible solutions

From now on, we prove Theorem 1.

Definition 7 *A collection $\mathscr{L}$ of sets is a Laminar family if no two of them cross.*

You can look at a Laminar family as a forest: there is a node corresponding to each set in $\mathscr{L}$ and a node u is the parent of a node v is the set corresponding to u is the minimal set that contains the set corresponding to v. Assume that $0 < x_e < 1$ for all edges because we can delete edges with $x_e = 0$ and add those with $x_e = 1$ to the solution without any cost increase w.r.t optimum. Let $m = |E|$ be the number of edges after deleting 0 and 1 values, i.e. the size of the totally fractional solution. For every set $F \subseteq E$ we use $x(F)$ to denote the sum of the x_e values for $e \in F$.

Definition 8 *A set S is called tight if $x(\delta(S)) = f(S)$.*

Definition 9 *For each set $F \subseteq E$ the characteristic vector is defined as $\chi_F \in \{0,1\}^{|E|=m}$, such that $\chi_F(e) = \begin{cases} 1 & if\ e \in F \\ 0 & otherwise \end{cases}$*

Lemma 4 *For any basic feasible solution x that is totally fractional (i.e. $0 < x < 1$) there is a collection of $m = |E|$ tight sets $\mathscr{L}$ s.t.*

1. *$\mathscr{L}$ is Laminar.*
2. *characteristic vectors of $\delta(S)$ ($S \in \mathscr{L}$) are linearly independent.*

To prove this lemma we need to prove some other statements first. Let x be a basic feasible solution and τ be a collection of m tight constraints corresponding to this solution. (note that they are all linearly independent)

Definition 10 *Span of τ is the vector space generated by these m linearly independent vectors, i.e. $\{\chi_{\delta(S)} : S \in \tau\}$. So, we will have $span(\tau) = \mathbb{R}^m$*

Let $\mathscr{L}$ be a maximal laminar family that is a subset of τ. If $|\mathscr{L}| = m$ then we have what we need in Lemma 4. Otherwise, we show there is a tight set S which can be added to $\mathscr{L}$ while increasing the span of $\mathscr{L}$ and keeping it Laminar.

Lemma 5 *If $A, B \in \tau$ and are crossing (tight and linearly independent), then one of following two holds:*

1. *$A \cup B$ and $A \cap B$ are tight and $\chi_{\delta(A)} + \chi_{\delta(B)} = \chi_{\delta(A\cup B)} + \chi_{\delta(A\cap B)}$*
2. *$A - B$ and $B - A$ are tight and $\chi_{\delta(A)} + \chi_{\delta(B)} = \chi_{\delta(A-B)} + \chi_{\delta(B-A)}$*

Remark: This lemma can be used to uncross two sets A and B that cross by replacing them with $A \cup B$ and $A \cap B$ or $A - B$ and $B - A$.

Proof. We know that f is weakly super-modular. Suppose that the second condition holds: $f(A) + f(B) \leq f(A-B) + f(B-A)$ (the proof when the first condition holds is similar). Because $A, B \in \mathcal{T}$ both are tight we must have $x(\delta(A)) = f(A)$ and $x(\delta(B)) = f(B)$.

So $x(\delta(A)) + x(\delta(B)) = f(A) + f(B) \leq f(A-B) + f(B-A)$. Because x is feasible $x(\delta(A-B)) \geq f(A-B)$ and $x(\delta(B-A)) \geq f(B-A)$; thus $x(\delta(A-B)) + x(\delta(B-A)) \geq f(A-B) + f(B-A)$. Combining these two yields $x(\delta(A)) + x(\delta(B)) \leq x(\delta(A-B)) + x(\delta(B-A))$.

We show that this holds with equality by the same technique as Lemma 2. Consider the four edge types in the figure. Edge e_3 contributes twice to the left hand side but edges e_1, e_2 and e_4 contribute equally to both sides. Therefore the inequality holds in other direction too; so we must have equality.

Because this holds with equality and we also have $x(\delta(A-B)) \geq f(A-B)$ and $x(\delta(B-A)) \geq f(B-A)$ (because x is a feasible solution) we must have $x(\delta(A-B)) = f(A-B)$ and $x(\delta(B-A)) = f(B-A)$. Also, there is only one type of edge, namely e_3 in the figure, that does not contribute to both sides equally; since we have $x(\delta(A)) + x(\delta(B)) = x(\delta(A-B)) + x(\delta(B-A))$, these edges must have value 0, i.e. don't exist. Thus $\chi_{\delta(A)} + \chi_{\delta(B)} = \chi_{\delta(A-B)} + \chi_{\delta(B-A)}$, as wanted. ■

We now return to prove lemma 4.

Proof. for Lemma 4

Assume $|\mathscr{L}| < m$. There must be a tight set S s.t. $\chi_{\delta(S)} \in span(\tau)$ but $\chi_{\delta(S)} \notin span(\mathscr{L})$. Pick such S that crosses the fewest number of sets in $\mathscr{L}$. Suppose that S crosses $S' \in \mathscr{L}$. Apply Lemma 5 to S, S'. Either $S \cap S'$ and $S \cup S'$ or $S - S'$ and $S' - S$ are tight and at least one of the following holds:

1. $\chi_{\delta(S)} + \chi_{\delta(S')} = \chi_{\delta(S\cup S')} + \chi_{\delta(S\cap S')}$
2. $\chi_{\delta(S)} + \chi_{\delta(S')} = \chi_{\delta(S-S')} + \chi_{\delta(S'-S)}$

We know that $S \notin \mathscr{L}$ and $S' \in \mathscr{L}$. If from the two situation mentioned above the first one is true, then at least one of $S \cup S'$ and $S \cup S'$ are not in $\mathscr{L}$ and if the second statement is true, then at least one of $S - S'$ and $S' - S$ are not in $\mathscr{L}$. In either case it is easy to show that for example $S \cup S' \notin \mathscr{L}$ crosses fewer sets in $\mathscr{L}$ than S'.

■

We have proved all of our lemmas and theorems except Theorem 1. We will prove this theorem with a counting argument.

Proof. For Theorem 1 Assume by contradiction that all $x_e < \frac{1}{2}$. Note that $|\mathscr{L}| = |E| = m$.
We assign one token to each edge e. The total number of tokens will be m. Then, we redistribute the token

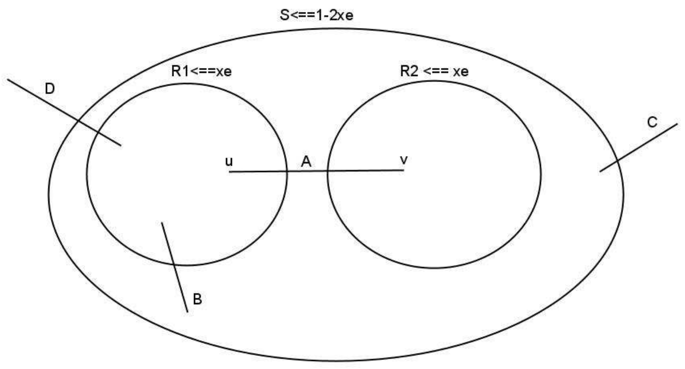

Figure 10.3: Redistributing tokens among sets

such that each $S \in \mathscr{L}$ gets one token and we will show that there will be some leftover tokens, which is clearly a contradiction. Each edge $e = (u, v)$ gives x_e tokens to each set which is the first (smallest) set containing an end-point of it, and also gives $1 - 2x_e$ tokens to the smallest set containing both end-points (as shown in Figure 10.3).

It is possible for an edge not to give $1 - 2x_e$ to the outer set because sometimes there would be no outer set to give the token to. So the total token that an edge e gives is at most $2 \times x_e + (1 - 2x_x) \leq 1$. For the root sets (i.e. sets that don't belong to any other set), the tokens of edges coming out are not totally used. So the total tokens that a set S gets is (considering Figure 10.3) is:

$$x(C) + x(B) + \sum_{e \in B}(1 - 2x_e) + \sum_{e \in A}(1 - 2x_e)$$

And because $\sum_{e \in B}(1 - 2x_e) = |B| - 2x(B)$ the total tokens will be:

$$x(c) - x(B) - 2x(A) + |B| + |A|.$$

We claim that this is a positive integer. We have assumed that set S has only two children. It may have more but the calculation doesn't change and same reasoning will work for more than two children. We know that $|A| + |B|$ is integer. Since our sets are all tight, so:

$$f(S) = x(\delta(S)) \qquad (10.1)$$
$$f(R_1) = x(\delta(R_1)) \qquad (10.2)$$
$$f(R_2) = x(\delta(R_2)) \qquad (10.3)$$

If we subtract the last two statements from the first one:

$$f(S) - f(R_1) - f(R_2) = x(\delta(S)) - x(\delta(R_1)) - x(\delta(R_2)) = x(C) - x(B) - 2x(A).$$

But $f(S) - f(R_1) - f(R_2)$ is an integer. So $x(C) - x(B) - 2x(A)$ should be an integer as well. Also, because S, R_1, R_2 are linearly independent, this value cannot be zero. So $x(C) - x(B) - 2x(A) > 0$, so it must at least 1. Thus, each set S gets at least 1 token. If S is a maximal set in the laminar family, clearly there are some edges coming out of S (those that are in $\delta(S)$ but in none of its children). There is at least $1 - 2x_e$ token of those edges left-over that are not accounted for. So the number of tokens would be larger than m, which is a contradiction. ■

10.2 Minimum Spanning Tree polytope

We know that the minimum spanning tree problem is solvable in polynomial time. In this section we prove that the spanning tree polytope is integral using an iterative argument. Let's start with the following LP relaxation of the problem.

$$\begin{array}{llcll} \min & \sum_{e\in E} c_e \cdot x_e & & & \\ \text{s.t.} & \delta_x(S) & \geq & 1 \quad \forall S \subset V & (1) \\ & x_e & \geq & 0 \quad \forall e \in E & (2) \end{array}$$

It is easy to see that all the constraints of this LP are satisfied by any spanning tree. However this LP has integral gap.

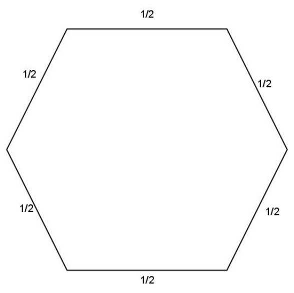

Figure 10.4: A cycle and a fractional spanning tree

Figure 10.4 shows an example with integrality gap. For a cycle of size n, the fractional solution will be $\frac{n}{2}$ but the integral solution will require $n - 1$ edges. So the size of integrality gap is almost 2. We can model our problem in a way that we wouldn't encounter to this problem. As an observation, it is obvious that in a graph G with a spanning tree T:

$$\forall S \subseteq V : E(T) \cap E(S) \leq |S| - 1$$

where $E(T)$ is the edges of tree and $E(S)$ is the edges both end-points in S. As a matter of fact this condition is also sufficient for spanning trees. So we have:

$$\begin{array}{ll} \min & \sum w_e x_e \\ \text{s.t.} & x(E(S)) \leq |S| - 1, \forall S \subset V \\ & x(E(V)) = |V| - 1 \\ & x_e \geq 0. \end{array}$$

We call this LP_{MST}. Our goal is to prove that:

Theorem 3 *LP_{MST} is integral, i.e. every bfs of this LP is integral.*

References

[1] Williamson, D.P. and Goemans, M.X. and Mihail, M. and Vazirani, V.V., *A primal-dual approximation algorithm for generalized Steiner network problems.* Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, ACM, 708–717, 1993.

[2] Goemans, M.X. and Goldberg, A.V. and Plotkin, S. and Shmoys, D.B. and Tardos, E. and Williamson, D.P., *Improved approximation algorithms for network design problems.* Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 223–232, 1994.

[3] Jain, K., *A factor 2 approximation algorithm for the generalized Steiner network problem.* Combinatorica, Springer, 39–60, 2001.