6.891 Approximation Algorithms	April 28, 2000
Lecture 12	
Lecturer: David P. Williamson	*Scribe: Ozlem Ergun*

1 Jain's Technique

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.
– T. S. Eliot, "Four Quartets"

For our last full lecture, we in essence return to a technique that we saw in our first full lecture; namely, approximation algorithms via the deterministic rounding of linear programming relaxations. We saw with the algorithm **Round** for the Set Cover problem that we could set up a linear programming relaxation for the problem, and round up every variable greater than a certain value $1/f$ to obtain an f-approximation algorithm for the problem. Today we return to a much more sophisticated use of the same technique, due to Jain.

1.1 The Survivable Network Design Problem

To illustrate this technique, we will consider the following problem, which is a generalization of the generalized Steiner tree problem we saw earlier.

Survivable Network Design Problem (SNDP)

- **Input:**
  - Undirected graph $G = (V, E)$.
  - Edge costs $c_e \geq 0$, for each $e \in E$.
  - Request counts $r(u, v) \in \mathbb{N}$, for each $u, v \in V, u \neq v$.
- **Goal:** Find $F \subseteq E$ of minimum cost, such that $G' = (V, F)$ contains at least $r(u, v)$ edge-disjoint paths between u and v, for each $u, v \in V, u \neq v$.

There had been substantial work on this problem prior to Jain's result. Define $k = \max_{u,v} r(u, v)$. A $2k$-approximation algorithm due to W, Goemans, Mihail and

Vazirani appeared in 1993 (conference version) and 1995 (journal version). This was improved to a $2H_k$- approximation algorithm due to Goemans, Goldberg, Plotkin, Shmoys, Tardos and W in 1994. We will see today a 2-approximation algorithm due to Jain, a graduate student at Georgia Tech, that appeared in January 1998.

The first two results use the primal-dual method in stages, taking care of one of the edge-disjoint paths required between each u and v per stage. Jain's result uses an entirely different proof strategy.

1.2 The Model

All three results use a relaxation of a certain integer programming formulation of the problem. First, we define $f(S) = \max_{u \in S, v \notin S} r(u,v)$ and $\delta(S)$ to be the edges across the cut S. Then we can formulate the survivable network design problem as the following integer program:

$$\begin{aligned} \text{Min} \quad & \sum_{e \in E} c_e x_e \\ \text{subject to:} \quad & \\ & \sum_{e \in \delta(S)} x_e \geq f(S) && \text{for each } S \subseteq V, \\ & x_e \in \{0,1\} && \text{for each } e \in E. \end{aligned}$$

A maximum flow-minimum cut argument shows that this models the surviviable network design problem.

As usual, we will relax this integer program to a linear program, replacing $x_e \in \{0,1\}$ with $0 \leq x_e \leq 1$. The resulting linear program has exponentially many constraints, but we can construct a polynomial time separation oracle for it. The oracle computes the maximum flow between each u, v pair in turn using the current solution, $x_e, e \in E$, as arc capacities. If the maximum flow between some pair u, v is less than $r(u,v)$, then the corresponding minimum cut S defines a violated constraint in our linear program.

We'll use the shorthand notation $x(F) = \sum_{e \in F} x_e$ throughout, in particular, notice that the constraints above become $x(\delta(S)) \geq f(S)$.

Our goal is to show that the linear programming relaxation is within a factor 2 of the integer program.

1.3 Weak Supermodularity

The function f we defined has some particularly useful properties.

Definition 1 *A function $g : 2^V \longrightarrow \mathbb{N}$ is weakly supermodular if $g(V) = 0$ and for each $A, B \subseteq V$, one of the following holds:*

(i) $g(A) + g(B) \le g(A \cup B) + g(A \cap B)$,

(ii) $g(A) + g(B) \le g(A - B) + g(B - A)$.

Claim 1 *f is weakly supermodular.*

The following result is the cornerstone Jain's 2-approximation algorithm.

Theorem 2 (Jain 1998) *For any weakly supermodular f and any extreme point x of our linear program, there exists $e \in E$ such that $x_e \ge 1/2$.*

We first show how this theorem implies a 2-approximation algorithm for the survivable network design problem, and then turn to the proof of the theorem. In fact, we will end up showing a weaker result: that there exists some edge e such that $x_e \ge 1/3$, which will imply a 3-approximation algorithm.

1.4 Jain's Algorithm

Given this theorem, we can obtain a 2-approximation algorithm for our problem in only a slightly more complicated fashion than in the **Round** algorithm for set cover. The basic idea is that we can solve the LP, obtain an extreme point, find some edge of value at least 1/2, round it up to 1 (thereby losing a factor of 2 for that edge), then recurse on a subproblem with a modified function f and the edge removed from the graph. We formalize this idea in the following algorithm.

Jain

> $f_0(S) \leftarrow f(S)$ for each $S \subseteq V$
> $i \leftarrow 0$
> **while** $\cup_{j=1}^{i} F_j$ is not a feasible solution
>   Solve the linear program on the graph $(V, E - \bigcup_{j=1}^{i} F_j)$ using the
>   function f_i to obtain an extreme point solution x^*
> $F_{i+1} \leftarrow \{e \in E - \bigcup_{j=1}^{i} F_j : x_e \ge 1/2\}$
> $f_{i+1}(S) \leftarrow \max(0, f(S) - |\delta(S) \cap \bigcup_{j=1}^{i+1} F_j|)$ for each $S \subseteq V$
> $i \leftarrow i + 1$
> **return** $\cup_{j=1}^{i} F_j$

To show that the algorithm works and that we can apply Jain's theorem, we need to show two things: first, that we can solve the LP in every iteration, and secondly, that the function f_i is in fact weakly supermodular every iteration. For the survivable network design problem the first is easy, via the following separation oracle: for every edge in $\bigcup_{j=1}^{i} F_j$ we install an edge of capacity 1, and again do a maxflow computation between every u, v pair in the graph G. We now show the second.

Lemma 3 *For any $F \subseteq E$ and $f(S)$ weakly supermodular, $f_i(S) = f(S) - |\delta(S) \cap F|$ for each $S \subseteq V$, is weakly supermodular.*

Proof: Define $z_e = \begin{cases} 1 & \text{if } e \in F, \\ 0 & \text{otherwise;} \end{cases}$ and $z(F') = \sum_{e \in F'} z_e$ for $F' \subseteq E$. We will show for general z that z is submodular, that is,

$$z(\delta(A)) + z(\delta(B)) \geq z(\delta(A \cup B)) + z(\delta(A \cap B))$$

and

$$z(\delta(A)) + z(\delta(B)) \geq z(\delta(A - B)) + z(\delta(B - A)),$$

for all $A, B \subseteq V$.

We claim that this implies the lemma. Notice that $f_i(S) = f(S) - z(\delta(S))$ for each $S \subseteq V$. Then suppose that for the pair $A, B \subseteq V$, condition (i) of weak supermodularity holds for f. Then

$$\begin{aligned} f_i(A) + f_i(B) &= f(A) - z(\delta(A)) + f(B) - z(\delta(B)) \\ &\leq f(A \cup B) - z(\delta(A \cup B)) + f(A \cap B) - z(\delta(A \cap B)) \\ &= f_i(A \cup B) + f_i(A \cap B). \end{aligned}$$

which shows that condition (i) holds for f_i. On the other hand, if condition (ii) of weak supermodularity holds for f and the pair A, B, a similar argument shows condition (ii) holds for f_i. Hence f_i is weakly supermodular.

To show these conditions on z we use a simple case analysis on the edges of F. It is possible to show that every edge that contributes z_e to the right-hand side of the inequalities will also contribute z_e to the left-hand side. For example, if edge $e \in F$ has one end point in $A - B$ and the other end point in $\overline{A \cup B}$, then e contributes z_e to $z(\delta(A))$ and $z(\delta(A \cup B))$; and to $z(\delta(A))$ and $z(\delta(A - B))$. Notice that only one type of edge contributes unequally to each side of these conditions on z: an edge e with one end point in $A - B$ and the other in $B - A$ contributes $2z_e$ to the left-hand side of first condition, but zero to the right-hand side. Similarly, an edge e from $A \cap B$ to $\overline{A \cup B}$ contributes $2z_e$ to the left-hand side of the second condition, but zero to the right-hand side. □

We can now show the following.

Theorem 4 *The algorithm* Jain *is a 2-approximation algorithm for any such integer program with weakly supermodular functions f_i such that we can solve the associated linear programs.*

Proof: The proof proceeds by induction on the number of iterations in the while loop of the algorithm Jain.

We want to show that the cost of the edges we choose in all iterations is no more than twice the linear program optimal value. That is:

$$\sum_{\substack{e\in F_i\\ i\geq 1}} c_e \leq 2\sum_{e\in E} c_e x_e^0$$

where x^0 is the solution of the linear program - the solution at the zeroth iteration. More generally, define x^i as the solution after the ith iteration.
Base case: Suppose that exactly one iteration of the while loop is necessary to achieve feasibility. Then

$$\sum_{e\in F_1} c_e \leq 2\sum_{e\in E} c_e x_e^0$$

since the solution F_1 involves rounding up exactly those $x_e^0 \geq 1/2$.
Inductive step: The inductive assumption is:

$$\sum_{\substack{e\in F_i\\ i\geq 2}} c_e \leq 2\sum_{e\in E-F_1} c_e x_e^1.$$

By similar reasoning as in the base case above, we know that

$$\sum_{e\in F_1} c_e \leq 2\sum_{e\in F_1} c_e x_e^0.$$

The crucial observation is that x^0 is a feasible solution to our linear program with weakly supermodular function f_1 on the graph $(V, E-F_1)$. Notice that

$$\sum_{e\in\delta(S)} x_e^0 \geq f(S).$$

Therefore

$$\sum_{e\in\delta(S)-F_1} x_e^0 \geq f(S) - |\delta(S)\cap F_1| = f_1(S).$$

Since x^0 satisfies the constraints for our LP with function f_1 on the graph $(V, E-F_1)$ and x^1 is the optimal solution to this LP, we have

$$\sum_{e\in E-F_1} c_e x_e^1 \leq \sum_{e\in E-F_1} c_e x_e^0.$$

Putting this all together, we have

$$\sum_{\substack{e\in F_i\\ i\geq 1}} c_e \leq 2\sum_{e\in F_1} c_e x_e^0 + 2\sum_{e\in E-F_1} c_e x_e^1 \leq 2\sum_{e\in F_1} c_e x_e^0 + 2\sum_{e\in E-F_1} c_e x_e^0 \leq 2\sum_{e\in E} c_e x_e^0.$$

□

1.5 Proof of Theorem 2

Throughout this section we'll assume that an extreme point solution x satisfies $0 < x_e < 1$ for each $e \in E$, since if $x_e = 1$ for some $e \in E$, then we are done, and if $x_e = 0$ we can remove e from E without loss of generality.

Define m to be the number of edges in the graph, or equivalently, the number of fractional variables in our linear programming extreme point solution. We need the following definitions for our proof.

Definition 2 *The sets $A, B \subseteq V$ are said to cross if $A - B$, $B - A$ and $A \cap B$ are non-empty.*

Definition 3 *A collection $\mathcal{L}$ of sets is laminar if no pair of sets in $\mathcal{L}$ cross.*

Definition 4 *A set $S \subseteq V$ is tight if $x(\delta(S)) = f(S)$.*

Definition 5 *The edge incidence vector $\chi_F \in \{0,1\}^m$ for $F \subseteq E$ is defined componentwise as* $\chi_F(e) = \begin{cases} 1 & \text{if } e \in F, \\ 0 & \text{otherwise.} \end{cases}$

Theorem 5 *For an extreme point solution x to our linear program, there exists a collection $\mathcal{L}$ of m sets such that*

1. *S is tight for each $S \in \mathcal{L}$.*

2. *The set of vectors $\{\chi_{\delta(S)} : S \in \mathcal{L}\}$ is linearly independent.*

3. *$\mathcal{L}$ is laminar.*

Note that our linear program has $\chi^T_{\delta(S)} x \geq f(S)$ as constraints, that is, $\chi_{\delta(S)}$ is a row of our constraint matrix for each $S \subseteq V$. Clearly, then, since x is an extreme point of our linear program, it follows that there exists a collection $\mathcal{L}$ of m sets such that Conditions 1 and 2 of Theorem 5 hold. To show that there exists $\mathcal{L}$ such that Condition 3 holds, we'll need the following lemma.

Lemma 6 *If A, B cross and are tight, then either*

(i) $A \cup B$ and $A \cap B$ are tight and $\chi_{\delta(A)} + \chi_{\delta(B)} = \chi_{\delta(A \cup B)} + \chi_{\delta(A \cap B)}$,

or (ii) $A - B$ and $B - A$ are tight and $\chi_{\delta(A)} + \chi_{\delta(B)} = \chi_{\delta(A - B)} + \chi_{\delta(B - A)}$.

Proof: f is weakly supermodular, and suppose that for A, B, we have that

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

However,

$$\begin{aligned}
f(A)+f(B) &= x(\delta(A))+x(\delta(B)) && \text{since } A \text{ and } B \text{ are tight,}\\
&\geq x(\delta(A\cup B))+x(\delta(A\cap B)) && \text{see the proof of Lemma 3,}\\
&\geq f(A\cup B)+f(A\cap B) && \text{since } x \text{ is feasible.}
\end{aligned}$$

Therefore, by the feasibility of x, $x(\delta(A\cup B)) = f(A\cup B)$ and $x(\delta(A\cap B)) = f(A\cap B)$.

To show the second part of condition (i), we recall the proof of Lemma 3. In that analysis, there was only one type of edge that didn't contribute to both sides of the inequality equally. However, we just showed that

$$x(\delta(A))+x(\delta(B)) = x(\delta(A\cup B))+x(\delta(A\cap B))$$

which indicates that any of these exceptional edges e must have $x_e = 0$. Since we discarded all edges with $x_e = 0$, we must have no exceptional edges in E. Therefore, the second part of condition (i) holds.

The proof is similar if the other condition of weak supermodularity holds for f and A, B. □

Proof of Theorem 5: By the properties of basic (feasible) solutions there exists a family $\mathcal{T}$ of m sets satisfying conditions 1 and 2 of Theorem 5. Let span($\mathcal{T}$) be the span of the vectors $\{\chi_{\delta(S)} : S \in \mathcal{T}\}$. Note that $\text{span}(\mathcal{T}) = \mathbb{R}^m$. Let $\mathcal{L}$ be a maximal collection of the sets in $\mathcal{T}$ that satisfy Conditions 1, 2 and 3 of Theorem 5.

If $|\mathcal{L}| = m$, we are done. Otherwise, we can choose a tight set S such that $\chi_{\delta(S)} \in \text{span}(\mathcal{T})$ but $\chi_{\delta(S)} \notin \text{span}(\mathcal{L})$ and such that no other such tight set crosses fewer members of $\mathcal{L}$. Note that S must cross at least one set in $\mathcal{L}$, or it could be added to $\mathcal{T}$ without violating Conditions 1, 2 and 3 of Theorem 5. We obtain a proof by contradiction by obtaining another set U such that $\chi_{\delta(U)} \in \text{span}(\mathcal{T})$, $\chi_{\delta(U)} \notin \text{span}(\mathcal{L})$, and such that U crosses fewer members of $\mathcal{L}$ than T.

Pick $T \in \mathcal{L}$ such that S and T cross, and apply Lemma 6. Suppose that for the pair S, T we have that $S-T$ and $T-S$ are tight and

$$\chi_{\delta(S)} + \chi_{\delta(T)} = \chi_{\delta(S-T)} + \chi_{\delta(T-S)}.$$

Then it can't be the case that both $\chi_{\delta(S-T)}$ and $\chi_{\delta(T-S)}$ are in span($\mathcal{L}$) since $\chi_{\delta(S)} \notin \text{span}(\mathcal{L})$. On the other hand, for the pair S, T, we may have the other alternative from Lemma 6, with $S\cup T$ and $S\cap T$ tight and

$$\chi_{\delta(S)} + \chi_{\delta(T)} = \chi_{\delta(S\cup T)} + \chi_{\delta(S\cap T)}.$$

In this case, at least one of $\chi_{\delta(S\cup T)}$ and $\chi_{\delta(S\cap T)}$ are not in span($\mathcal{L}$). Therefore, at least one of $S-T$, $T-S$, $S\cup T$ and $S\cap T$ is not in span($\mathcal{L}$).

We'll show that $S-T$, $T-S$, $S\cup T$ and $S\cap T$ all cross fewer sets in $\mathcal{L}$ than S, and hence we will have a set U such that $\chi_{\delta(U)} \in \text{span}(\mathcal{T})$, $\chi_{\delta(U)} \notin \text{span}(\mathcal{L})$, with U crossing fewer members of $\mathcal{L}$ than T.

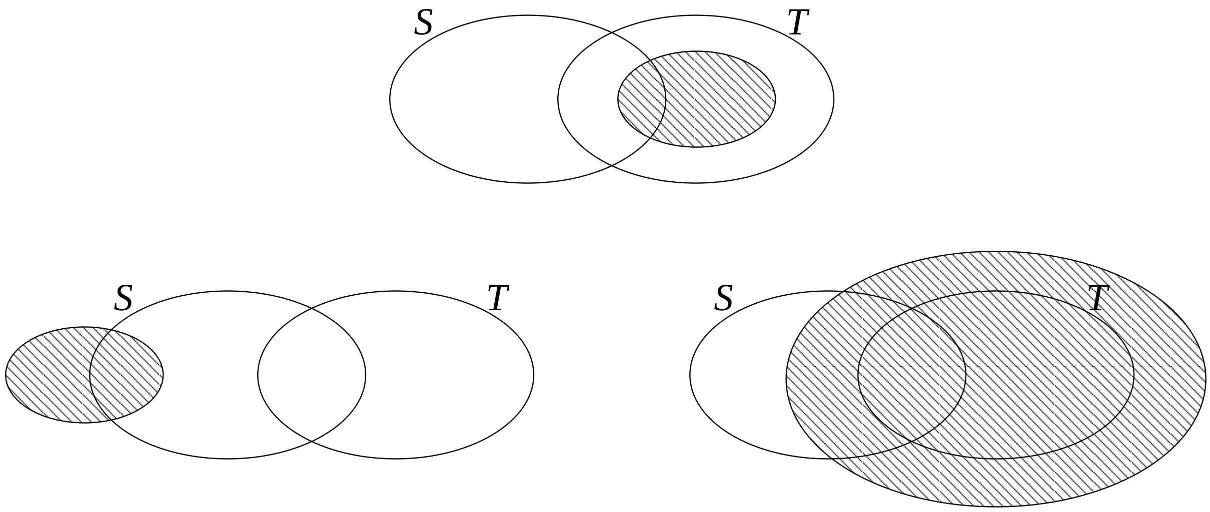

Figure 1: Sets of $\mathcal{L}$ that don't cross T but do cross one of $S-T$, $T-S$, $S \cup T$ or $S \cap T$.

Any set crossing $S-T$, $T-S$, $S \cup T$ or $S \cap T$ but not T must also cross S. See Figure 1 for a schematic of the possible sets in $\mathcal{L}$ that must be considered. Note too that S crosses T, but none of $S-T$, $T-S$, $S \cup T$ or $S \cap T$ do. Therefore, each of $S-T$, $T-S$, $S \cup T$ and $S \cap T$ crosses strictly fewer sets of $\mathcal{L}$ than S. $\square$

Observation: $f(S) \geq 1$ for each $S \in \mathcal{L}$ since S is tight, $\chi_{\delta(S)} \neq 0$, all edges such that $x_e = 0$ have been discarded, and $f(S)$ is an integer.

Finally, we can state the theorem that will complete our proof of Theorem 2. Note however, that instead of showing that there exists an edge e such that $x_e \geq 1/2$, we'll show there exists an edge e such that $x_e \geq 1/3$. This will make the proof of Theorem 7 simpler, but won't change anything else of substance.

Theorem 7 *There exists $S \in \mathcal{L}$ such that $|\delta(S)| \leq 3$.*

Note that, in combination with the observation above, this proves (our slightly weaker version of) Theorem 2. We will need the following definition regarding the edges and vertices of G.

Definition 6 *A socket is an edge-vertex pair (e, v) such that v is one of the two endpoints of e.*

Observe that since the graph has m edges, it has $2m$ distinct sockets.

We'll consider the tree defined on the elements on $\mathcal{L}$ by the parent relation: for each $S \in \mathcal{L}$, Parent(S) = the smallest set $T \in \mathcal{L}$ that contains S. See Figure 2 for an example.

Proof of Theorem 7: Suppose that the theorem is false. Then all $S \in \mathcal{L}$ has $|\delta(S)| \geq 4$. We'll use this to charge sockets uniquely to the elements of $\mathcal{L}$ so that each

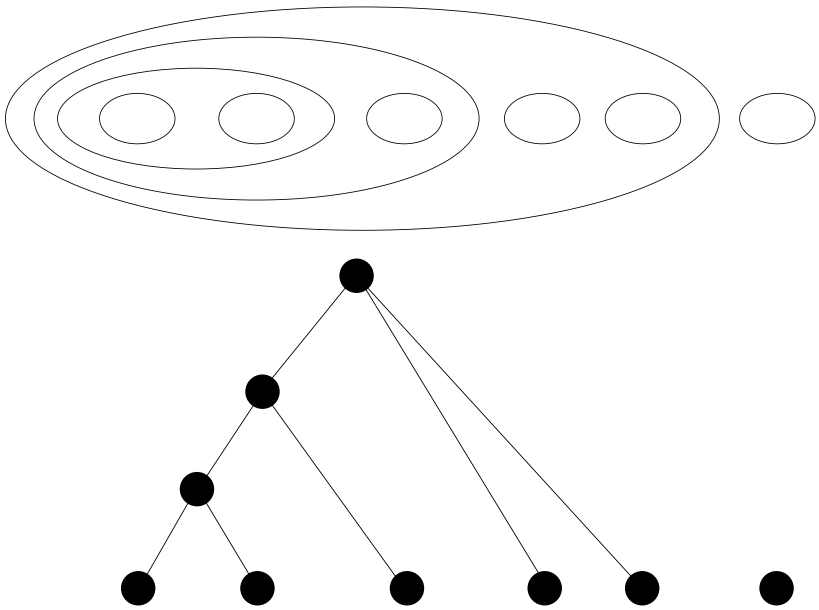

Figure 2: Example tree structure for a collection of laminar sets.

element of $\mathcal{L}$ is charged 2 sockets except for the root of the tree, which is charged 4 sockets. Since $\mathcal{L}$ has m elements, we have $2m + 2$ sockets, a contradiction, and the theorem is proved.

We'll show that we can charge sockets as specified above by induction on the tree from the leaves on up.

Base case: We charge 4 sockets to each of the leaves of the tree. We can do this because of our assumption that the theorem is false, and each $S \in \mathcal{L}$ has $|\delta(S)| \geq 4$. Thus each such S is associated with four sockets for the edges in $\delta(S)$ and the endpoint of each edge in $\delta(S)$ that S contains.

Inductive step: For any parent, assume inductively that in each child's subtree, 2 sockets are charged per non-root node and 4 sockets are charged to the root.

If there are 2 or more children, then each child can pass a charge of 2 to the parent.

If there is exactly one child, but the parent has at least two additional sockets then the child can pass a charge of 2 to the parent and the inductive assumption will be satisfied.

If the parent P has exactly one child C and no additional sockets, then $\delta(P) = \delta(C)$ and hence $\chi_{\delta(P)} = \chi_{\delta(C)}$ and the elements of $\mathcal{L}$ are no longer linearly independent. Therefore, this case cannot happen.

If the parent P has exactly one child C and exactly one additional socket representing edge e, then either $f(P) = f(C) - x_e$ or $f(P) = f(C) + x_e$. In either case, we have a contradiction since $0 < x_e < 1$ and f is integer-valued. Therefore, this case cannot happen.

Thus we can, by induction, charge the sockets to the elements of the tree to satisfy the condition that 2 sockets are charged to each non-root node and 4 sockets to the root.

Considering where the falsehood of the theorem was used in constructing the charging scheme, we notice that it was only used in charging 4 to each of the leaves of the tree. Therefore, one of the leaves S of the tree must be such that $|\delta(S)| \leq 3$. $\square$

It would be nice to have a simple, intuitive proof that there exists e such that $x_e \geq 1/2$. It would also be nice to have a combinatorial 2-approximation algorithm for the problem, rather than one that needs to solve a linear program.