# 23 Steiner Network

The following generalization of the Steiner forest problem to higher connectivity requirements has applications in network design and is also known as the *survivable network design problem.* In this chapter, we will give a factor 2 approximation algorithm for this problem by enhancing the LP-rounding technique to *iterated rounding.* A special case of this problem was considered in Exercise 22.10.

**Problem 23.1 (Steiner network)** We are given an undirected graph $G = (V, E)$, a cost function on edges $c : E \to \mathbf{Q}^+$ (not necessarily satisfying the triangle inequality), a connectivity requirement function $r$ mapping unordered pairs of vertices to $\mathbf{Z}^+$, and a function $u : E \to \mathbf{Z}^+ \cup \{\infty\}$ stating an upper bound on the number of copies of edge $e$ we are allowed to use; if $u_e = \infty$, there is no upper bound for edge $e$. The problem is to find a minimum cost multigraph on vertex set $V$ that has $r(u, v)$ edge disjoint paths for each pair of vertices $u, v \in V$. Each copy of edge $e$ used for constructing this graph will cost $c(e)$.

## 23.1 The LP-relaxation and half-integrality

In order to give an integer programming formulation for this problem, we will first define a *cut requirement function*, $f : 2^V \to \mathbf{Z}^+$, as we did for the metric Steiner forest problem. For every $S \subseteq V$, $f(S)$ is defined to be the largest connectivity requirement separated by the cut $(S, \overline{S})$, i.e., $f(S) = \max\{r(u, v) | u \in S \text{ and } v \in \overline{S}\}$.

$$\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} c_e x_e && && (23.1)\\
\text{subject to} \quad & \sum_{e:\ e \in \delta(S)} x_e \geq f(S), && S \subseteq V \\
& x_e \in \mathbf{Z}^+, && e \in E \text{ and } u_e = \infty \\
& x_e \in \{0, 1, \ldots, u_e\}, && e \in E \text{ and } u_e \neq \infty
\end{aligned}$$

The LP-relaxation is:

$$\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} c_e x_e && && (23.2)\\
\text{subject to} \quad & \sum_{e:\ e \in \delta(S)} x_e \geq f(S), && S \subseteq V \\
& x_e \geq 0, && e \in E \text{ and } u_e = \infty \\
& u_e \geq x_e \geq 0, && e \in E \text{ and } u_e \neq \infty
\end{aligned}$$

Since LP (23.2) has exponentially many constraints, we will need the ellipsoid algorithm for finding an optimal solution. Exercise 23.1 develops a polynomial-sized LP.

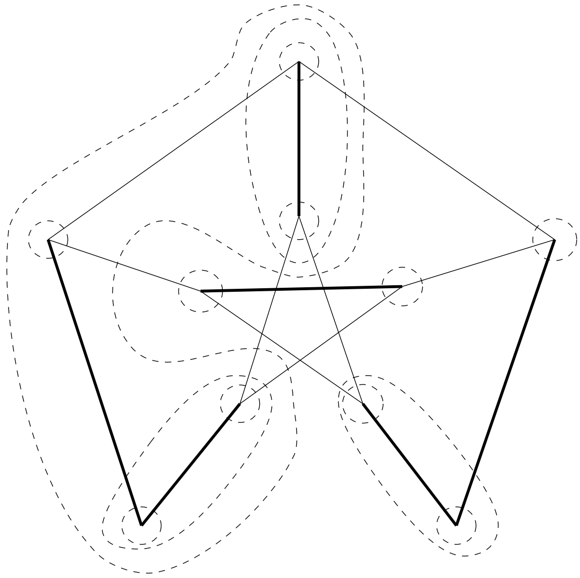
As shown in Chapters 14 and 19, certain **NP**-hard problems, such as vertex cover and node multiway cut, admit LP-relaxations having the remarkable property that they always have a half-integral optimal solution. Rounding up all halves to 1 in such a solution leads to a factor 2 approximation algorithm. Does relaxation (23.2) have this property? The following lemma shows that the answer is "no".

**Lemma 23.2** *Consider the Petersen graph (see Section 1.2) with a connectivity requirement of 1 between each pair of vertices and with each edge of unit cost. Relaxation (23.2) does not have a half-integral optimal solution for this instance.*

**Proof:** Consider the fractional solution $x_e = 1/3$ for each edge $e$. Since the Petersen graph is 3-edge connected (in fact, it is 3-vertex connected as well), this is a feasible solution. The cost of this solution is 5. In any feasible solution, the sum of edge variables incident at any vertex must be at least 1, to allow connectivity to other vertices. Therefore, any feasible solution must have cost at least 5 (since the Petersen graph has 10 vertices). Hence, the solution given above is in fact optimal.

Any solution with $x_e = 1$ for some edge $e$ must have cost exceeding 5, since additional edges are required to connect the endpoints of $e$ to the rest of the graph. Therefore, any half-integral solution of cost 5 would have to pick, to the extent of one half each, the edges of a Hamiltonian cycle. Since the Petersen graph has no Hamiltonian cycles, there is no half-integral optimal solution. □

Let us say that an *extreme point solution*, also called a vertex solution or a basic feasible solution, for an LP is a feasible solution that cannot be written as the convex combination of two feasible solutions. The solution $x_e = 1/3$, for each edge $e$, is not an extreme point solution. An extreme optimal solution is shown in the figure below; thick edges are picked to the extent of 1/2, thin edges to the extent of 1/4, and the missing edge is not picked.



The isomorphism group of the Petersen graph is edge-transitive, and there are 15 related extreme point solutions; the solution $x_e = 1/3$ for each edge $e$ is the average of these.

Notice that although the extreme point solution is not half-integral, it picks some edges to the extent of half. We will show below that in fact this is a property of any extreme point solution to LP (23.2). We will obtain a factor 2 algorithm by rounding up these edges and iterating. Let $H$ be the set of edges picked by the algorithm at some point. Then, the residual requirement of cut $(S, \overline{S})$ is $f'(S) = f(S) - |\delta_H(S)|$, where $\delta_H(S)$ represents the set of edges of $H$ crossing the cut $(S, \overline{S})$. In general, the *residual cut requirement function*, $f'$, may not correspond to the cut requirement function for any set of connectivity requirements. We will need the following definitions to characterize it:

Function $f : 2^V \rightarrow \mathbf{Z}^+$ is said to be *submodular* if $f(V) = 0$, and for every two sets $A, B \subseteq V$, the following two conditions hold:

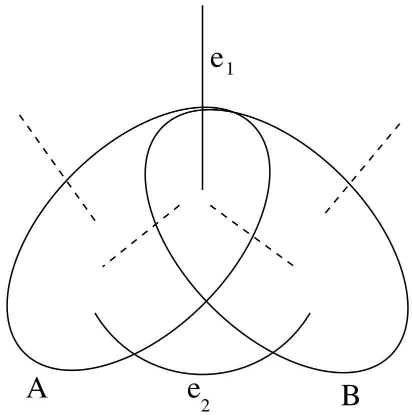
- $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$
- $f(A) + f(B) \geq f(A - B) + f(B - A)$.

**Remark 23.3** Sometimes submodularity is defined only with the first condition. We will need to work with the stronger definition given above.

Two subsets of $V$, $A$ and $B$, are said to *cross* if each of the sets, $A - B$, $B - A$, and $A \cap B$, is nonempty. If $A$ and $B$ don't cross then either they are disjoint or one of these sets is contained in the other.

**Lemma 23.4** *For any graph $G$ on vertex set $V$, the function $|\delta_G(.)|$ is submodular.*

**Proof:** If sets $A$ and $B$ do not cross, then the two conditions given in the definition of submodular functions hold trivially. Otherwise, edges having one endpoint in $A \cap B$ and the other in $\overline{A \cup B}$ (edge $e_1$ in the figure below) contribute to $\delta(A)$ and $\delta(B)$ but not to $\delta(A-B)$ or $\delta(B-A)$. Similarly, edge $e_2$ below does not contribute to $\delta(A \cap B)$ or to $\delta(A \cup B)$. The remaining edges contribute equally to both sides of both conditions. □




Function $f : 2^V \rightarrow \mathbf{Z}$ is said to be *weakly supermodular* if $f(V) = 0$, and for every two sets $A, B \subseteq V$, at least one of the following conditions holds:

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

It is easy to check that the original cut requirement function is weakly supermodular; by Lemma 23.5, so is the residual cut requirement function.

**Lemma 23.5** *Let $H$ be a subgraph of $G$. If $f : 2^{V(G)} \rightarrow \mathbf{Z}^+$ is a weakly supermodular function, then so is the residual cut requirement function $f'$.*

**Proof:** Suppose $f(A) + f(B) \leq f(A - B) + f(B - A)$; the proof of the other case is similar. By Lemma 23.4, $|\delta_H(A)| + |\delta_H(B)| \geq |\delta_H(A-B)| + |\delta_H(B-A)|$. Subtracting, we get $f'(A) + f'(B) \leq f'(A - B) + f'(B - A)$. □

We can now state the central polyhedral fact needed for the factor 2 algorithm in its full generality.

**Theorem 23.6** *For any weakly supermodular function $f$, any extreme point solution, $\boldsymbol{x}$, to LP (23.2) must pick some edge to the extent of at least a half, i.e., $x_e \geq 1/2$ for at least one edge $e$.*

## 23.2 The technique of iterated rounding

In this section, we will give an iterated rounding algorithm for the Steiner network problem, using Theorem 23.6.

**Algorithm 23.7 (Steiner network)**

1. **Initialization:** $H \leftarrow \emptyset$: $f' \leftarrow f$.
2. While $f' \not\equiv \mathbf{0}$, do:
   Find an extreme optimal solution,
   $\boldsymbol{x}$, to LP (23.2) with cut requirements given by $f'$.
   For each edge $e$ such that $x_e \geq 1/2$, include $\lceil x_e \rceil$ copies of $e$ in $H$,
   and decrement $u_e$ by this amount.
   Update $f'$: for $S \subseteq V$, $f'(S) \leftarrow f(S) - |\delta_H(S)|$.
3. Output $H$.

The algorithm presented above achieves an approximation guarantee of factor 2 for an arbitrary weakly supermodular function $f$. Establishing a polynomial running time involves showing that an extreme optimal solution to LP (23.2) can be found efficiently. We do not know how to do this for an arbitrary weakly supermodular function $f$. However, if $f$ is the original cut requirement function for some connectivity requirements, then a polynomial time implementation follows from the existence of a polynomial time separation oracle for each iteration.

For the first iteration, a separation oracle follows from a max-flow subroutine. Given a solution $\boldsymbol{x}$, construct a graph on vertex set $V$ with capacity $x_e$ for each edge $e$. Then, for each pair of vertices $u, v \in V$, check if this graph admits a flow of at least $r(u, v)$ from $u$ to $v$. If not, we will get a violated cut, i.e., a cut $(S, \overline{S})$ such that $\delta_{\boldsymbol{x}}(S) < f(S)$, where

$$\delta_{\boldsymbol{x}}(S) = \sum_{e:\ e \in \delta(S)} x_e.$$

Let $f'$ be the cut requirement function of a subsequent iteration. Given a solution to LP (23.2) for this function, say $\boldsymbol{x}'$, define $\boldsymbol{x}$ as follows: for each edge $e$, $x_e = x'_e + e_H$, where $e_H$ is the number of copies of edge $e$ in $H$. The following lemma shows that a separation oracle for the original function $f$ leads to a separation oracle for $f'$. Furthermore, this lemma also shows that there is no need to update $f'$ explicitly after each iteration.

**Lemma 23.8** *A cut $(S, \overline{S})$ is violated by solution $\boldsymbol{x}'$ under cut requirement function $f'$ iff it is violated by solution $\boldsymbol{x}$ under cut requirement function $f$.*

**Proof:** Notice that $\delta_{\boldsymbol{x}}(S) = \delta_{\boldsymbol{x}'}(S) + |\delta_H(S)|$. Since $f(S) = f'(S) + |\delta_H(S)|$, $\delta_{\boldsymbol{x}}(S) \geq f(S)$ iff $\delta_{\boldsymbol{x}'}(S) \geq f'(S)$. □

Lemma 23.8 implies that solution $\boldsymbol{x}'$ is feasible for the cut requirement function $f'$ iff solution $\boldsymbol{x}$ is feasible for $f$. Assuming Theorem 23.6, whose proof we will provide below, let us show that Algorithm 23.7 achieves an approximation guarantee of 2.

**Theorem 23.9** *Algorithm 23.7 achieves an approximation guarantee of 2 for the Steiner network problem.*

**Proof:** By induction on the number of iterations executed by the algorithm when run with a weakly supermodular cut requirement function $f$, we will prove that the cost of the integral solution obtained is within a factor of two of the cost of the optimal fractional solution. Since the latter is a lower bound on the cost of the optimal integral solution, the claim follows.

For the base case, if $f$ requires one iteration, the claim follows, since the algorithm rounds up only edges $e$ with $x_e \geq 1/2$.

For the induction step, assume that $\boldsymbol{x}$ is the extreme optimal solution obtained in the first iteration. Obtain $\hat{\boldsymbol{x}}$ from $\boldsymbol{x}$ by zeroing out components that are strictly smaller than $1/2$. By Theorem 23.6, $\hat{\boldsymbol{x}} \neq 0$. Let $H$ be the set of edges picked in the first iteration. Since $H$ is obtained by rounding up nonzero components of $\hat{\boldsymbol{x}}$ and each of these components is $\geq 1/2$, $\text{cost}(H) \leq 2{\cdot}\text{cost}(\hat{\boldsymbol{x}})$.

Let $f'$ be the residual requirement function after the first iteration and $H'$ be the set of edges picked in subsequent iterations for satisfying $f'$. The key observation is that $\boldsymbol{x} - \hat{\boldsymbol{x}}$ is a feasible solution for $f'$, and thus by the induction hypothesis, $\text{cost}(H') \leq 2 \cdot \text{cost}(\boldsymbol{x} - \hat{\boldsymbol{x}})$. Let us denote by $H + H'$ the edges of $H$ together with those of $H'$. Clearly, $H + H'$ satisfies $f$. Now,

$$\begin{aligned} \text{cost}(H + H') &\leq \text{cost}(H) + \text{cost}(H') \\ &\leq 2 \cdot \text{cost}(\hat{\boldsymbol{x}}) + 2 \cdot \text{cost}(\boldsymbol{x} - \hat{\boldsymbol{x}}) \leq 2 \cdot \text{cost}(\boldsymbol{x}). \end{aligned}$$ □

**Corollary 23.10** *The integrality gap of LP (23.2) is bounded by 2.*

Notice that previous algorithms obtained using LP-rounding solved the relaxation once and did the entire rounding based on this solution. These algorithms did not exploit the full power of rounding – after part of the solution is rounded, the remaining fractional solution may not be the best solution to continue the rounding process. It may be better to assume integral values for the rounded variables and recompute fractional values for the remaining variables, as is done above. We will call this technique *iterated rounding.*

**Example 23.11** The tight example given for the metric Steiner tree problem, Example 3.4, is also a tight example for this algorithm. Observe that after including a subset of edges of the cycle, an extreme optimal solution to the resulting problem picks the remaining edges of the cycle to the extent of one half each. The algorithm finds a solution of cost $(2 - \varepsilon)(n - 1)$, whereas the cost of the optimal solution is $n$. □

## 23.3 Characterizing extreme point solutions

From polyhedral combinatorics we know that a feasible solution for a set of linear inequalities in $\mathbf{R}^m$ is an extreme point solution iff it satisfies $m$ linearly independent inequalities with equality. Extreme solutions of LP (23.2) satisfy an additional property which leads to a proof of Theorem 23.6.

We will assume that the cut requirement function $f$ in LP (23.2) is an arbitrary weakly supermodular function. Given a solution $\boldsymbol{x}$ to this LP, we will say that an inequality is *tight* if it holds with equality. If this inequality corresponds to the cut requirement of a set $S$, then we will say that *set $S$ is tight*. Let us make some simplifying assumptions. If $x_e = 0$ for some edge $e$, this edge can be removed from the graph, and if $x_e \geq 1$, $\lfloor x_e \rfloor$ copies of edge $e$ can be picked and the cut requirement function be updated accordingly. We may assume without loss of generality that an extreme point solution $\boldsymbol{x}$ satisfies $0 < x_e < 1$, for each edge $e$ in graph $G$. Therefore, each tight inequality corresponds to a tight set. Let the number of edges in $G$ be $m$.

We will say that a collection, $\mathcal{L}$, of subsets of $V$ forms a *laminar family* if no two sets in this collection cross. The inequality corresponding to a set $S$ defines a vector in $\mathbf{R}^m$: the vector has a 1 corresponding to each edge $e \in \delta_G(S)$, and 0 otherwise. We will call this the *incidence vector* of set $S$, and will denote it by $\mathcal{A}_S$.

**Theorem 23.12** *Corresponding to any extreme point solution to LP (23.2) there is a collection of $m$ tight sets such that*

- *their incidence vectors are linearly independent, and*
- *collection of sets forms a laminar family.*

**Example 23.13** The extreme point solution for the Peterson graph assigns nonzero values to 14 of the 15 edges. By Theorem 23.12, there should be 14 tight sets whose incidence vectors are linearly independent. These are marked in figure. □

Fix an extreme point solution, $\boldsymbol{x}$, to LP (23.2). Let $\mathcal{L}$ be a laminar family of tight sets whose incidence vectors are linearly independent. Denote by span($\mathcal{L}$) the vector space generated by the set of vectors $\{\mathcal{A}_S | S \in \mathcal{L}\}$. Since $\boldsymbol{x}$ is an extreme point solution, the span of the collection of all tight sets is $m$. We will show that if span($\mathcal{L}$) $< m$, then there is a tight set $S$ whose addition to $\mathcal{L}$ does not violate laminarity and also increases the span. Continuing in this manner, we will obtain $m$ tight sets as required in Theorem 23.12.

We begin by studying properties of crossing tight sets.

**Lemma 23.14** *Let $A$ and $B$ be two crossing tight sets. Then, one of the following must hold:*

- *$A - B$ and $B - A$ are both tight and $\mathcal{A}_A + \mathcal{A}_B = \mathcal{A}_{A-B} + \mathcal{A}_{B-A}$*
- *$A \cup B$ and $A \cap B$ are both tight and $\mathcal{A}_A + \mathcal{A}_B = \mathcal{A}_{A\cup B} + \mathcal{A}_{A\cap B}$.*

**Proof:** Since $f$ is weakly supermodular, either $f(A)+f(B) \leq f(A-B)+f(B-A)$ or $f(A)+f(B) \leq f(A\cup B)+f(A\cap B)$. Let us assume the former holds; the proof for the latter is similar. Since $A$ and $B$ are tight, we have

$$\delta_{\boldsymbol{x}}(A)+\delta_{\boldsymbol{x}}(B)=f(A)+f(B).$$

Since $A-B$ and $B-A$ are not violated,

$$\delta_{\boldsymbol{x}}(A-B)+\delta_{\boldsymbol{x}}(B-A)\geq f(A-B)+f(B-A).$$

Therefore,

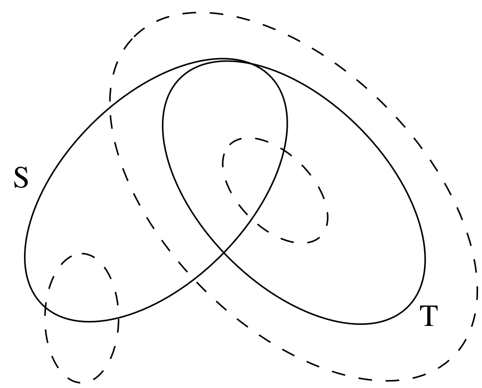
$$\delta_{\boldsymbol{x}}(A)+\delta_{\boldsymbol{x}}(B)\leq \delta_{\boldsymbol{x}}(A-B)+\delta_{\boldsymbol{x}}(B-A).$$

As argued in Lemma 23.4 (which established the submodularity of function $|\delta_G(.)|$), edges having one endpoint in $A\cup B$ and the other in $A\cap B$ can contribute only to the left-hand side of this inequality. The rest of the edges must contribute equally to both sides. So, this inequality must be satisfied with equality. Furthermore, since $x_e > 0$ for each edge $e$, $G$ cannot have any edge having one endpoint in $A\cup B$ and the other in $A\cap B$. Therefore, $\mathcal{A}_A+\mathcal{A}_B=\mathcal{A}_{A-B}+\mathcal{A}_{B-A}$. □

For any set $S\subseteq V$, define its *crossing number* to be the number of sets of $\mathcal{L}$ that $S$ crosses.

**Lemma 23.15** *Let $S$ be a set that crosses set $T\in\mathcal{L}$. Then, each of the sets $S-T, T-S, S\cup T$ and $S\cap T$ has a smaller crossing number than $S$.*

**Proof:** The figure below illustrates the three ways in which a set $T'\in\mathcal{L}$ can cross one of these four sets without crossing $T$ itself ($T'$ is shown dotted). In all cases, $T'$ crosses $S$ as well. In addition, $T$ crosses $S$ but not any of the four sets. □



**Lemma 23.16** *Let $S$ be a tight set such that $\mathcal{A}_S \notin \text{span}(\mathcal{L})$ and $S$ crosses some set in $\mathcal{L}$. Then, there is a tight set $S'$ having a smaller crossing number than $S$ and such that $\mathcal{A}_{S'} \notin \text{span}(\mathcal{L})$.*

**Proof:** Let $S$ cross $T \in \mathcal{L}$. Suppose the first possibility established in Lemma 23.14 holds; the proof of the second possibility is similar. Then, $S-T$ and $T-S$ are both tight sets and $\mathcal{A}_S + \mathcal{A}_T = \mathcal{A}_{S-T} + \mathcal{A}_{T-S}$. This linear dependence implies that $\mathcal{A}_{S-T}$ and $\mathcal{A}_{T-S}$ cannot both be in span($\mathcal{L}$), since otherwise $\mathcal{A}_S \in \text{span}(\mathcal{L})$. By Lemma 23.15, $S-T$ and $T-S$ both have a smaller crossing number than $S$. The lemma follows. □

**Corollary 23.17** *If $\text{span}(\mathcal{L}) \neq \mathbf{R}^m$, then there is a tight set $S$ such that $\mathcal{A}_S \notin \text{span}(\mathcal{L})$ and $\mathcal{L} \cup \{S\}$ is a laminar family.*

By Corollary 23.17, if $\mathcal{L}$ is a maximal laminar family of tight sets with linearly independent incidence vectors, then $|\mathcal{L}| = m$. This establishes Theorem 23.12.

## 23.4 A counting argument

The characterization of extreme point solutions given in Theorem 23.12 will yield Theorem 23.6 via a counting argument. Let $\boldsymbol{x}$ be an extreme point solution and $\mathcal{L}$ be the collection of tight sets established in Theorem 23.12. The number of sets in $\mathcal{L}$ equals the number of edges in $G$, i.e., $m$. The proof is by contradiction. Suppose that for each edge $e, x_e < 1/2$. Then, we will show that $G$ has more than $m$ edges.

Since $\mathcal{L}$ is a laminar family, it can be viewed as a forest of trees if its elements are ordered by inclusion. Let us make this precise. For $S \in \mathcal{L}$, if $S$ is not contained in any other set of $\mathcal{L}$, then we will say that $S$ is a *root set*. If $S$ is not a root set, we will say that $T$ is the *parent* of $S$ if $T$ is a minimal set in $\mathcal{L}$ containing $S$; by laminarity of $\mathcal{L}$, $T$ is unique. Further, $S$ will be called a *child* of $T$. Let the relation *descendent* be the reflexive transitive closure of the relation "child". Sets that have no children will be called *leaves*. In this manner, $\mathcal{L}$ can be partitioned into a forest of trees, each rooted at a root set. For any set $S$, by the *subtree rooted at $S$* we mean the set of all descendents of $S$.

Edge $e$ is *incident* at set $S$ if $e \in \delta_G(S)$. The *degree* of $S$ is defined to be $|\delta_G(S)|$. Set $S$ *owns* endpoint $v$ of edge $e = (u, v)$ if $S$ is the smallest set of $\mathcal{L}$ containing $v$. The subtree rooted at set $S$ *owns* endpoint $v$ of edge $e = (u, v)$ if some descendent of $S$ owns $v$.

Since $G$ has $m$ edges, it has $2m$ endpoints. Under the assumption that $\forall e, x_e < 1/2$, we will prove that for any set $S$, the endpoints owned by the subtree rooted at $S$ can be redistributed in such a way that $S$ gets at least 3 endpoints, and each of its proper descendents gets 2 endpoints. Carrying out this procedure for each of the root sets of the forest, the total number of endpoints in the graph must exceed $2m$, leading to a contradiction.

We have assumed that $\forall e : 0 < x_e < 1/2$. For edge $e$, define $y_e = 1/2 - x_e$, the *halves complement* of $e$. Clearly, $0 < y_e < 1/2$. For $S \in \mathcal{L}$ define its *corequirement* to be

$$\text{coreq}(S) = \sum_{e \in \delta(S)} y_e = \frac{1}{2}|\delta_G(S)| - f(S).$$

Clearly, $0 < \text{coreq}(S) < |\delta_G(S)|/2$. Furthermore, since $|\delta_G(S)|$ and $f(S)$ are both integral, $\text{coreq}(S)$ is half-integral. Let us say that *coreq*$(S)$ is *semi-integral* if it is not integral, i.e., if $\text{coreq}(S) \in \{1/2, 3/2, 5/2, \ldots\}$. Since $f(S)$ is integral, *coreq*$(S)$ is semi-integral iff $|\delta_G(S)|$ is odd.

Sets having a corequirement of 1/2 play a special role in this argument. The following lemma will be useful in establishing that certain sets have this corequirement.

**Lemma 23.18** *Suppose $S$ has $\alpha$ children and owns $\beta$ endpoints, where $\alpha + \beta = 3$. Furthermore, each child of $S$, if any, has a corequirement of 1/2. Then,* $\text{coreq}(S) = 1/2$.

**Proof:** Since each child of $S$ has corequirement of 1/2, it has odd degree. Using this and the fact that $\alpha + \beta = 3$, one can show that $S$ must have odd degree (see Exercise 23.3). Therefore the corequirement of $S$ is semi-integral. Next, we show that $\text{coreq}(S)$ is strictly smaller than 3/2, thereby proving the lemma. Clearly,

$$\text{coreq}(S) = \sum_{e \in \delta(S)} y_e \leq \sum_{S'} \text{coreq}(S') + \sum_{e} y_e,$$

where the first sum is over all children $S'$ of $S$, and the second sum is over all edges $e$ having an endpoint in $S$. Since $y_e$ is strictly smaller than 1/2, if $\beta > 0$, then $\text{coreq}(S) < 3/2$. If $\beta = 0$, all edges incident at the children of $S$ cannot also be incident at $S$, since otherwise the incidence vectors of these four sets will be linearly dependent. Therefore,

$$\text{coreq}(S) < \sum_{S'} \text{coreq}(S') = 3/2.$$

□

The next two lemmas place lower bounds on the number of endpoints owned by certain sets.

**Lemma 23.19** *If set $S$ has only one child, then it must own at least two endpoints.*

**Proof:** Let $S'$ be the child of $S$. If $S$ has no endpoint incident at it, the set of edges incident at $S$ and $S'$ must be the same. But then $\mathcal{A}_S = \mathcal{A}_{S'}$, leading to a contradiction. $S$ cannot own exactly one endpoint, because then $\delta_{\boldsymbol{x}}(S)$ and $\delta_{\boldsymbol{x}}(S')$ will differ by a fraction, contradicting the fact that both these sets are tight and have integral requirements. The lemma follows. □

**Lemma 23.20** *If set $S$ has two children, one of which has a corequirement of $1/2$, then it must own at least one endpoint.*

**Proof:** Let $S'$ and $S''$ be the two children of $S$, with $\text{coreq}(S') = 1/2$. Suppose $S$ does not own any endpoints. Since the three vectors $\mathcal{A}_S, \mathcal{A}_{S'}$, and $\mathcal{A}_{S''}$ are linearly independent, the set of edges incident at $S'$ cannot all be incident at $S$ or all be incident at $S''$. Let $a$ denote the sum of $y_e$'s of all edges incident at $S'$ and $S$, and let $b$ denote the sum of $y_e$'s of all edges incident at $S'$ and $S''$. Thus, $a > 0$, $b > 0$, and $a + b = \text{coreq}(S) = 1/2$.

Since $S'$ has a semi-integral corequirement, it must have odd degree. Therefore, the degrees of $S$ and $S''$ have different parities, and these two sets have different corequirements. Furthermore, $\text{coreq}(S) = \text{coreq}(S'') + a - b$. Therefore, $\text{coreq}(S) - \text{coreq}(S'') = a - b$. But $-1/2 < a - b < 1/2$. Therefore, $S$ and $S''$ must have the same corequirement, leading to a contradiction. □

**Lemma 23.21** *Consider a tree $T$ rooted at set $S$. Under the assumption that $\forall e, x_e < 1/2$, the endpoints owned by $T$ can be redistributed in such a way that $S$ gets at least 3 endpoints, and each of its proper descendents gets 2 endpoints. Furthermore, if $\text{coreq}(S) \neq 1/2$, then $S$ must get at least 4 endpoints.*

**Proof:** The proof is by induction on the height of tree $T$. For the base case, consider a leaf set $S$. $S$ must have degree at least 3, because otherwise an edge $e$ incident at it will have $x_e \geq 1/2$. If it has degree exactly 3, $\text{coreq}(S)$ is semi-integral. Further, since $\text{coreq}(S) < |\delta_G(S)|/2 = 3/2$, the corequirement of $S$ is $1/2$. Since $S$ is a leaf, it owns an endpoint of each edge incident at it. Therefore, $S$ has the required number of endpoints.

Let us say that a set has a *surplus* of 1 if 3 endpoints have been assigned to it and a surplus of 2 if 4 endpoints have been assigned to it. For the induction step, consider a nonleaf set $S$. We will prove that by moving the surplus of the children of $S$ and considering the endpoints owned by $S$ itself, we can assign the required number of endpoints to $S$. There are four cases:

1. If $S$ has 4 or more children, we can assign the surplus of each child to $S$, thus assigning at least 4 endpoints to $S$.
2. Suppose $S$ has 3 children. If at least one of them has a surplus of 2, or if $S$ owns an endpoint, we can assign 4 endpoints to $S$. Otherwise, each child must have a corequirement of half, and by Lemma 23.18, $\text{coreq}(S) = 1/2$ as well. Thus, assigning $S$ the surplus of its children suffices.

3. Suppose $S$ has two children. If each has a surplus of 2, we can assign 4 endpoints to $S$. If one of them has surplus 1, then by Lemma 23.20, $S$ must own at least one endpoint. If each child has a surplus of 1 and $S$ owns exactly one endpoint, then we can assign 3 endpoints to $S$, and this suffices by Lemma 23.18. Otherwise, we can assign 4 endpoints to $S$.
4. If $S$ has one child, say $S'$, then by Lemma 23.19, $S$ owns at least 2 endpoints. If $S$ owns exactly 2 endpoints and $S'$ has surplus of exactly 1, then we can assign 3 endpoints to $S$; by Lemma 23.18, $\text{coreq}(S) = 1/2$, so this suffices. In all other cases, we can assign 4 endpoints to $S$.

□