1 Problem Statement

Let $G = (E, V)$ be a graph and $c : E \to \mathbb{R}^+$ be a cost function on edges. Let $\{(s_i, t_i, r_i), 1 \leq i \leq k\}$ such that $s_i, t_i \in V$ and $r_i \in \mathbb{Z}^+$ be a set of requests. Our goal is to find a subgraph $H$ of $G$ of minimum cost such that for all $(s_i, t_i)$ pairs, there are at least $r_i$ edge disjoint paths between $s_i$ and $t_i$ in $H$.

2 Solving Generalized Steiner Forest with Repeated Edges

As a warm up, we start by allowing the algorithm to use multiple copies of the same edge. We will solve this problem by repeatedly solving Steiner forest problems. That is, in round $k$, create a Steiner forest instance with request pairs that have a requirement of at least $k$ and are not already $k$-connected.

Claim 1. The approximation factor of this algorithm is $2 \cdot r_{\max}$, where $r_{\max} = \max_i r_i$.

Proof. We are running $r_{\max}$ Steiner forest instances, each with an approximation factor of 2. Letting $OPT_k$ be the optimal solution to the $k$th Steiner forest instance, and $OPT$ be the optimal solution to the whole problem, it is enough to show that $c(OPT_k) \leq c(OPT)$. For this, note that $OPT \setminus OPT_{k-1} \setminus \cdots \setminus OPT_1$ must be feasible for $k$th Steiner forest instance, since OPT must meet all terminals’ requests. \hfill \square

The following LP encodes this problem.

$$\min \sum_{e \in E} c_e x_e$$
\[\text{subject to } \sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \subset V \text{ s.t. } \exists s_i, t_i \text{ with } s_i \in S, t_i \notin S, \text{ and } r_i \geq k \] (1)
\[x_e \geq 0 \quad \forall e \in E\]

Here, $\delta(S)$ is the cut defined by $S$.

3 Generalized Steiner Forest [2]

Now, we attempt to solve generalized Steiner forest without repeated edges. What do we need to change to our approach in Section 2? The solution to the LP given in [1] may buy edges that it already bought in a previous round. We could attempt to fix this problem by removing those edges, but then we cannot guarantee that our LP is still feasible. In fact, there are very simple examples where this happens. In the following instance, suppose there is a demand of 2 between $s_3$ and $t_3$ and a demand of 1 between $s_1, t_1$ and $s_2, t_2$. After making all pairs 1-connected and deleting the edges used, there is no way to connect $s_3$ to $t_3$ with the remaining edges!
However, note that buying the last remaining edge is all that is required to provide 2-connectivity to $s_3, t_3$. Therefore, we should instead define requirements on cuts. Let

$$r(S) = \max \{ r_i : s_i \in S, t_i \notin S \},$$  \hspace{1cm} (2)

$$r_k(S) = \begin{cases} 1 & \text{if } r(S) \geq k \text{ and } |\delta_{F_{k-1}}(S)| = k - 1, \\ 0 & \text{otherwise.} \end{cases}$$  \hspace{1cm} (3)

Here, $F_{k-1}$ is the set of edges bought in the first $k - 1$ rounds. $\delta_{F_{k-1}}(S)$ is the set of edges in $F_{k-1}$ crossing the cut defined by $S$.

We can change the LP from 1 to reflect this new requirement:

$$\min \sum_{e \in E \setminus F_{k-1}} c_e x_e$$

subject to

$$\sum_{e \in \delta(S)} x_e \geq r_k(S) \hspace{1cm} \forall S \subset V$$

$$x_e \geq 0 \hspace{1cm} \forall e \in E$$  \hspace{1cm} (4)

Note that this is still linear: for LP$_k$, $r_k(S)$ is a constant.

**Claim 2.** $c(\text{LP}_k) \leq c(\text{OPT}).$

**Proof.** Because OPT must satisfy all requirements, it must satisfy all requirements that have not been satisfied by $F_{k-1}$. Therefore, OPT $\setminus F_{k-1}$ is a feasible solution to LP$_k$, and therefore OPT is a feasible solution as well. \hfill \square

We will next show some properties of $r_k(S)$. From there, showing that the primal dual method for solving the Steiner forest LP can be extended to the LP given by 4 is straightforward.

**Definition 1.** A symmetric function $f$ is proper if

1. $f(V) = 0$, and
2. $f(A \cup B) \leq \max \{ f(A), f(B) \}$ for disjoint $A, B$.

**Claim 3.** $r(S)$ is proper.

**Proof.** (1) is trivial. For (2), $r(A \cup B) = k$ if and only if $\exists s_i \in A \cup B, t_i \notin A \cup B$ such that $r_i \geq k$. $s_i$ must be in either $A$ or $B$, and therefore either $A$ or $B$ must have requirement at least $k$. \hfill \square

**Definition 2.** A symmetric function $f : 2^V \to \{0, 1\}$ is uncrossable if
(1) \( f(V) = 0 \)

(2) \( f(A) = f(B) = 1 \) implies either \( f(A \cup B) = f(A \cap B) = 1 \) or \( f(A \setminus B) = f(B \setminus A) = 1 \).

**Claim 4.** Proper functions with value either 0 or 1 are necessarily uncrossable.

**Proof.** Note that \((A \cap B), (A \setminus B), (B \setminus A), (A \cup B)\) are disjoint, and \(f(S) = f(S)\) since \(f\) is symmetric. Therefore, we have the following:

\[
\begin{align*}
f(A) &= f((A \cap B) \cup (A \setminus B)) \\
&= f((B \setminus A) \cup (A \cup B)) \\
f(B) &= f((A \cap B) \cup (B \setminus A)) \\
&= f((A \setminus B) \cup (A \cup B))
\end{align*}
\]

From the definition of proper functions, if \( f(A) = 1 \) and \( f(B) = 1 \), we have

\[
\begin{align*}
\max \{ (A \cap B), (A \setminus B) \} &= 1, \\
\max \{ (B \setminus A), (A \cup B) \} &= 1, \\
\max \{ (A \cap B), (B \setminus A) \} &= 1, \text{ and} \\
\max \{ (A \setminus B), (A \cup B) \} &= 1.
\end{align*}
\]

This implies that \( f \) is uncrossable. \( \square \)

**Theorem 5.** There is an algorithm for network design on uncrossable demand functions (the LP given by \( \mathcal{J} \) where the requirement function is uncrossable) with an approximation factor of 2.

**Proof.** It is easy to extend a primal-dual algorithm to solve the problem. This is left for the homework. \( \square \)

All that is left to show is that \( r_k \) is, in fact, uncrossable.

**Claim 6.** \( r_k \) is uncrossable.

**Proof.** The proof is by case analysis. We show one case here. Suppose \( r_k(A) = r_k(B) = 1 \), and \( r_k(A \setminus B) = 0 \). Then, \( \exists s_l \in A \cup B, t_l \notin A \cup B \), such that \( r_l \geq k \). Suppose \( s_l \in A \cap B \) (the other cases are similar). From a simple counting argument, we have the following general property of graphs:

\[
|\delta_{F_{k-1}}(A)| + |\delta_{F_{k-1}}(B)| \geq |\delta_{F_{k-1}}(A \cap B)| + |\delta_{F_{k-1}}(A \cup B)|.
\]

Now, since we know that \( |\delta_{F_{k-1}}(A)| = |\delta_{F_{k-1}}(B)| = k - 1 \), we have

\[
|\delta_{F_{k-1}}(A)| = |\delta_{F_{k-1}}(B)| = |\delta_{F_{k-1}}(A \cap B)| = |\delta_{F_{k-1}}(A \cup B)| = k - 1,
\]

and therefore \( r_k(A \cap B) = 1 \). Showing \( r_k(A \cup B) = 1 \) is similar. \( \square \)

This immediately implies a \( 2 \cdot r_{\max} \) approximation for generalized Steiner forest.
4 Removing Linear Dependence on $r_{\text{max}}$ [1]

In this section, we outline how the algorithm given in the previous section can be improved from $2 \cdot r_{\text{max}}$ to $2 \cdot H(r_{\text{max}})$ where $H(n)$ is the harmonic function, $1 + 1/2 + 1/3 + \cdots + 1/n = \Theta(\log n)$. This change basically amounts to running the sequence of LPs in the opposite direction – first starting with the terminals that have maximum requirement, and proceeding to terminals with lower requirement. In the original algorithm after $k$ stages we required $|\delta F_k(S) \geq \min \{r(S), k\}$. Now, we will run in the reverse direction – after $k$ stages we will require $|\delta F_k(S) \geq r(S) - r_{\text{max}} + k$. For LP$_k$, our new requirement function is therefore

$$r_k(S) = \max \{r(S) - r_{\text{max}} + k - |\delta F_{k-1}(S)|, 0\}. \quad (13)$$

It is not hard to show that this function is also uncrossable. To get the desired approximation factor, our goal is to show that

$$c(\text{LP}_k) \leq \frac{c(\text{OPT})}{r_{\text{max}} - k + 1}. \quad (14)$$

Consider the first round, with $k = 1$. We require

$$\sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S : r(S) = r_{\text{max}}. \quad (15)$$

However, OPT is a setting $x_e = x_e^* \forall e$, which satisfies

$$\sum_{e \in \delta(S)} x_e^* \geq r_{\text{max}} \quad \forall S : r(S) = r_{\text{max}}. \quad (16)$$

Therefore, $x_e = x_e^*/r_{\text{max}} \forall e$ satisfies (15) and has cost $c(\text{OPT})/r_{\text{max}}$, which gives $c(\text{LP}_1) \leq c(\text{OPT})/r_{\text{max}}$. The same argument shows that Equation (14) holds for all $k$. Finally, summing Equation (14) over all $k$ gives an approximation factor of $2 \cdot H(r_{\text{max}})$ as desired.

References
