Steiner Forest (Generalized Steiner Tree)

Input: - \( G = (V, E) \)
- \( c : E \rightarrow \mathbb{R}^+ \)
- \( K \) pairs of nodes \( (s_1, t_1), (s_2, t_2), \ldots, (s_K, t_K) \)

Feasible solution: \( F \subseteq E \) s.t. \( \exists s_i, t_i \) path in \( (V, F) \) for all \( i \in \{1, \ldots, K\} \)

Objective: \( \min c(F) = \sum_{e \in F} c(e) \)

Def: \( S_i = \{ s \in V : |s \cap \{s_i, t_i\}| = 1 \} \)
\( S = \bigcup_{i=1}^{K} S_i \)

LP Relaxation:
\[
\begin{align*}
\min & \quad \sum_{e \in E} c(e) x_e \\
\text{s.t.} & \quad \sum_{e \in S(s)} x_e \geq 1 \quad \forall s \in S \\
& \quad x_e \geq 0 \quad \forall e \in E
\end{align*}
\]

Dual:
\[
\begin{align*}
\max & \quad \sum_{s \in S} y_s \\
\text{s.t.} & \quad \sum_{s \in S(s)} y_s \leq c(e) \quad \forall e \in E \\
& \quad y_s \geq 0 \quad \forall s \in S
\end{align*}
\]
Algorithm: Primal-Dual, with interesting features:
- Raise multiple dual variables simultaneously
- "Reverse Clean-Up" step

Init: $F_1 = \emptyset$, $y = \theta$, $j = 1$

while $F_j$ not feasible {
  - Let $E_j = \{ S \in \mathcal{A} : S$ a connected component of $(U, F_j) \}$
    "active components"
  - Increase all $y_S : S \in E_j$ uniformly until $f$ some $e \in S(s)$, $S \in E_j$; where constraint for $e$ becomes tight:
    \[
    \sum_{S \in \mathcal{A} : e \in S(s)} y_S = c(e_j)
    \]
  - Let $D_j$ be amount raised each $y_S$
  - $F_{j+1} = F_j \cup \{ e_j \}$
  - $j = j + 1$
}

$F = F_j$
for ($k = j - 1$ down to $1$)
  - if $F \setminus \{ e_k \}$ feasible
    Remove $e_k$ from $F$
return $F$
Lemma: Alg is polynomial

\[ pf: \exists 1 \leq \ell \leq \rho \text{ iterations} \exists n \text{ active components each iteration} \]

Once constraint tight for \( \ell \) added to \( B \)

\[ pf: \exists \text{ feasible solution for } \mathcal{B} \]

Initially \( Y \) is always dual feasible

Easy Observations:

\[ Y_0 = (1, 0, 0) \]

\[ Y_{n+1} = (1, 0, 0) \]

\[ Y_0 = (1, 0, 0) \]
\[ \sum \leq 1 \text{Eln nonzero dual vars total} \]
\[ \sum \text{each iteration takes polytime} \]

**Main Thm:** Alg is a 2-approximation

**Lemma:** For all iterations \( i \), \( \sum \lfloor \Delta \rfloor \leq 21 \text{E}_i \)

Assume lemma for now. Start trying to prove thm

**Claim:** \( \sum \lceil \Delta \rfloor \leq 2 \sum \text{se} \)

**Proof:** Induction on iterations of alg (alg invariant)

**Init:** LHS = RHS = 0

**In some iteration \( i \):**

LHS increases by

\[ \sum \lfloor \Delta \rfloor \leq 21 \text{E}_i \]  
(by lemma)

RHS increases by

\[ 2 \sum \Delta_i = 21 \text{E}_i \]
\[(C_F) = \sum_{e \in F} c(e)\]
\[= \sum_{e \in F} \sum_{S \in \mathcal{G}: e \in e(S)} y_S\]
\[= \sum_{S \in \mathcal{G}} \sum_{e \in e(S)} y_S\]
\[= \sum \left\{ \bigcap_{S \in \mathcal{G}} \bigvee_l y_S \right\}\]
\[\leq 2 \sum_{S \in \mathcal{G}} y_S\]
\[\leq 2 \cdot \text{OPT}\]

(Dual constraint tight \(\forall e \in F\))

(switch order of summation)

(claim)

(weak duality)

So just need to prove lemma:

Lemma: For all iterations \(i\), \(\sum_{e \in e(S)} y_S \leq 2 \cdot \text{OPT}\)

Claim: \(F_i\) a forest \(A_j\)

Proof: Induction. True initially, each iteration add 1 edge between components.

Fix some \(i\). Define new graph \(G_i = (V_j, E_j)\):
- \(V_j\): vertex for each connected component of \((V, F_i)\)
- \(E_j = \{e \in T \cap \{v, u\} \in E\} \text{ with } v \in S, u \in T\)
Notes:
- Every edge of $E_j$ corresponds to exactly one edge in $F$ (or else cycle)
- $G_j$ is a forest
- $E_j \subseteq V_j$ (some components are active)

So $|F \cap S(S)| = \text{degree of } S \text{ in } G_j \quad (S \text{ component of } (U, F_j))$

$\exists \sum |F \cap S(S)| = \sum \deg_{G_j}(S) \leq 2|E_j|$

In other words: wT5 average degree in $G_j$ of components in $E_j$ is $\leq 2$

Claim: Let $S \in V_j$ have degree 1 in $G_j$. Then $S \in E_j$
PE: $S \subseteq S \subseteq E \implies S \not\subseteq \emptyset \implies S$ does not separate any $S$-tie pair.

Since $e$ only edge in $F$ leaving $S$, no $S$-tie both outside $S$ connected through $S$.

$\implies$ Final reverse cleanup would have removed $e$.

$\implies \therefore S \subseteq E$.

Claim: Let $T$ be a tree. If $S \subseteq V(T)$ contains all leaves of $T$, then $\sum_{v \in S} \deg(v) \leq 2|S|$

Proof:

$\sum_{v \in S} \deg(v) = \sum_{v \in V(T)} \deg(v) - \sum_{v \in S} \deg(v)$

$= 2(|V(T)| - 1) - \sum_{v \in S} \deg(v)$ (edges in $T$)

$\leq 2(|V(T)| - 1) - 2(1) (v \notin S$ has $\deg \geq 2)$

$= 2|S| - 2 \leq 2|S|$

Done!
Extensions / Thoughts:

- Open Question: is it possible to do better than 2?
  - Is $SF$ as easy as $ST$?

- Steiner $k$-Forest: Given $k < \# \text{demands}$, connect
  $k$ of them
  - Much harder! Best approx $O(n^{\sqrt{k}})$ [Ahn, Hajiaghayi, Nagarajan, Ravi '10]
  - If $c(e) = 1$: $\forall e: O(n^{0.44722})$ [CD, Kortsarz, Nutov '14]

- Survivable Network Design: connectivities $\geq 1$
  - 2-approx [Jain '01]

- $k$-Edge Fault Tolerant Subgraph: build a subgraph $H$ where
  connected components of $H\nF$ are same as $G\nF \cup F\nE$, $F\nE$
  - 2-approx [CD, Kortsarz, Kortsarz '22]