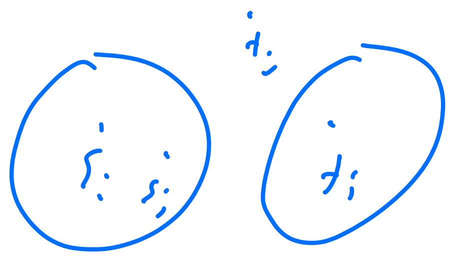
Steiner Forest (Generalized Steiner Tree)

Input: - $G = (V, E)$
- $c : E \to \mathbb{R}^+$
- k pairs of nodes $(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$

Feasible Solution: $F \subseteq E$ s.t. $\exists\, s_i - t_i$ path in (V, F) for all $i \in [k]$

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



Def: $\mathcal{S}_i = \{ S \subseteq V : |S \cap \{s_i, t_i\}| = 1 \}$

$\mathcal{S} = \bigcup_{i=1}^{k} \mathcal{S}_i$

LP Relaxation:

$$\min \sum_{e \in E} c(e)\, x_e$$

$$\text{s.t.} \quad \sum_{e \in \delta(S)} x_e \geq 1 \qquad \forall S \in \mathcal{S}$$

$$x_e \geq 0 \qquad \forall e \in E$$

Dual:

$$\max \sum_{S \in \mathcal{S}} y_S$$

$$\text{s.t.} \quad \sum_{S \in \mathcal{S} : e \in \delta(S)} y_S \leq c(e) \qquad \forall e \in E$$

$$y_S \geq 0 \qquad \forall S \in \mathcal{S}$$

Algorithm: Primal-Dual, with interesting features:

- Raise multiple dual variables simultaneously
- "Reverse Cleanup" step

Init: $F_1 = \emptyset$, $y = \vec{0}$, $j = 1$

while F_j not feasible {

- Let $\mathcal{C}_j = \{S \in \mathcal{S} : S$ a connected component of $(V, F_j)\}$

"active components"

- Increase all $y_S : S \in \mathcal{C}_j$ uniformly until $\exists$ some $e_j \in \delta(S)$, $S \in \mathcal{C}_j$ where constraint for e_j becomes tight:

$$\sum_{S \in \mathcal{S} : e_j \in \delta(S)} y_S = c(e_j)$$

- Let Δ_j be amount raised each y_S

- $F_{j+1} = F_j \cup \{e_j\}$
- $j = j + 1$

}

$F = F_j$

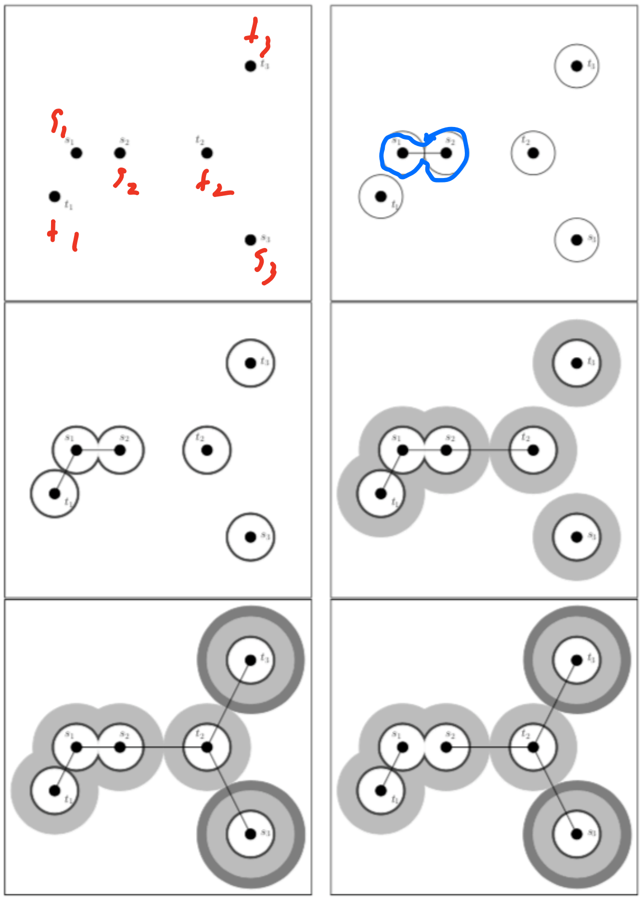
for ($k = j-1$ down to 1)

  if ($F \setminus \{e_k\}$ feasible)

    Remove e_k from F

return F

$y_{\{s_1\}} + y_{\{s_2\}} + y_{\{s_1,s_2\}} + y_{\{t_1\}} = c(\{s_1, t_1\})$



$y_{\{s_1\}} + y_{\{s_2\}} = c(\{s_1, s_2\})$

Easy Observations:

Lemma: y is always dual feasible

Pf: Consider some e. Initially $\sum_{S \subseteq V: e \in \delta(S)} y_S = 0 \leq c(e)$

Once constraint tight for e, added to F

$\Rightarrow$ inside a connected component, no S s.t. $e \in \delta(S)$ ever increased again

Lemma: Alg is polytime

Pf:

$\leq |E|$ iterations, $\leq n$ active components each iteration

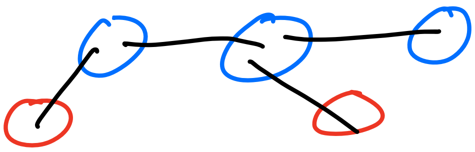
$\Rightarrow \leq |E|n$ nonzero dual vars total

$\Rightarrow$ each iteration takes poly time

<u>Main Thm</u>: Alg is a 2-approximation

<u>Lemma</u>: For all iterations j, $\sum_{S \in \mathcal{C}_j} |F \cap \delta(S)| \leq 2|\mathcal{C}_j|$

final F



Assume lemma for now. Start trying to prove thm

<u>Claim</u>: $\sum_{S \in \mathcal{S}} |\delta(S) \cap F|\, y_S \leq 2 \sum_{S \in \mathcal{S}} y_S$

<u>Pf</u>: Induction on iterations of alg (alg invariant)

Init: LHS = RHS = 0

In some iteration j:

LHS increases by

$$\sum_{S \in \mathcal{C}_j} |\delta(S) \cap F|\, \Delta_j = \Delta_j \sum_{S \in \mathcal{C}_j} |\delta(S) \cap F|$$

$$\leq 2|\mathcal{C}_j|\, \Delta_j \qquad \text{(by lemma)}$$

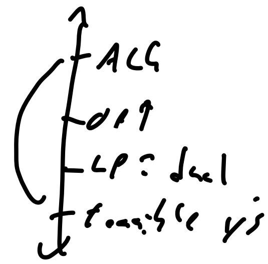
RHS increases by $2 \sum_{S \in \mathcal{C}_j} \Delta_j = 2|\mathcal{C}_j|\, \Delta_j$

$$c(F) = \sum_{e\in F} c(e)$$

$$= \sum_{e\in F}\ \sum_{S\in\mathcal{S}:\, e\in\delta(S)} y_S$$ (Dual constraint tight $\forall e\in F$)

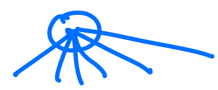
$$= \sum_{S\in\mathcal{S}}\ \sum_{e\in F:\, e\in\delta(S)} y_S$$ (switch order of summations

$$= \sum_{S\in\mathcal{S}} |\delta(S)\cap F|\, y_S$$



$$\le 2\sum_{S\in\mathcal{S}} y_S$$ (claim)

$$\le 2\cdot OPT$$ (weak duality)



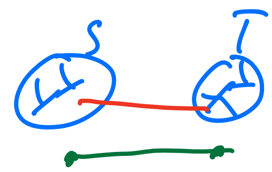
So just need to prove lemma:

<u>Lemma</u>: For all iterations j, $\sum_{S\in\mathcal{C}_j} |F\cap\delta(S)| \le 2|\mathcal{C}_j|$

↑ final F

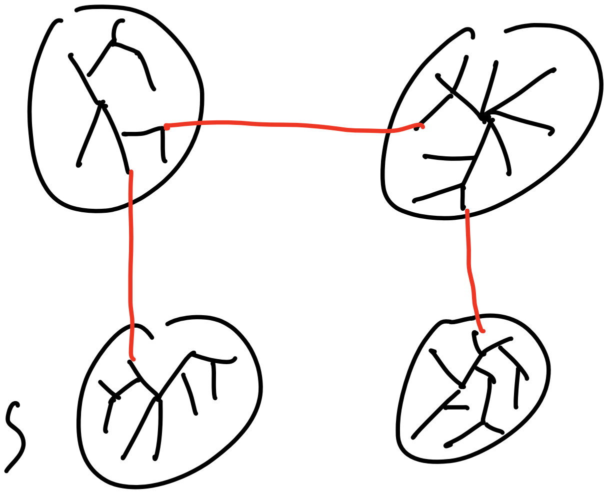
<u>Claim</u>: F_j a forest $\forall j$

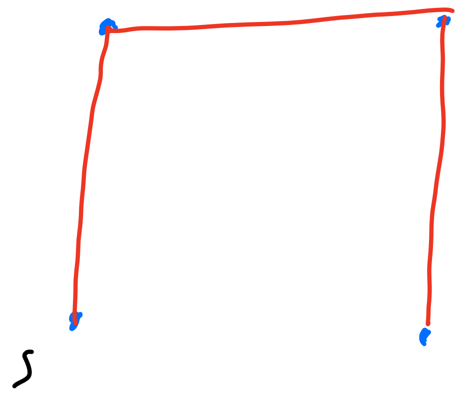
<u>Pf</u>: Induction. True initially, each iteration add 1 edge between components.



Fix some j. Define new graph $G_j = (V_j, E_j)$:

- V_j: vertex for each connected component of (V, F_j)
- $E_j = \{\{S,T\} : \exists \{u,v\}\in F \text{ with } u\in S,\ v\in T\}$





Notes:

- Every edge of E_j corresponds to exactly one edge in F (or else cycle)
- G_j a forest
- $\mathcal{C}_j \subseteq V_j$ (some components are active)

So $|F \cap \delta(S)| =$ degree of S in G_j (S component of (V, F_j))

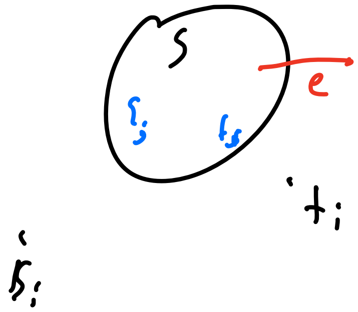
$$\Rightarrow \sum_{S \in \mathcal{C}_j} |F \cap \delta(S)| = \sum_{S \in \mathcal{C}_j} \deg_{G_j}(S) \leq 2|\mathcal{C}_j|$$

WTS

In other words: WTS average degree in G_j of components in $\mathcal{C}_j$ is ≤ 2

Claim: Let $S \in V_j$ have degree 1 in G_j. Then $S \in \mathcal{C}_j$

Pf: Sps $S \notin \mathcal{C}_i \Rightarrow S \notin \mathcal{S} \Rightarrow S$ does not separate any $s_i - t_i$ pair



Since e only edge in F leaving S, an $s_i - t_i$ both outside S connected through S

$\Rightarrow$ final reverse cleanup would have removed e

$\Rightarrow\Leftarrow$, so $S \in \mathcal{C}_i$

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

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

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

$$\le 2(|V(T)| - 1) - 2(|V(T)| - |S|) \qquad (v \notin S \text{ has deg.} \ge 2)$$

$$= 2|S| - 2 \le 2|S|$$

Done!

Extensions / Thoughts:

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

- Steiner k-Forest: Given $k <$ # demands, connect k of them
  - Much harder! Best approx $O(\sqrt{n})$ [Gupta, Hajiaghayi, Nagarajan, Ravi '10]
  - If $c(e) = 1 \ \forall e$: $O(n^{0.44772})$ [D, Kortsarz, Nutov '14]

- Survivable Network Design: connectivities ≥ 1
  - 2-approx [Jain '01]

- f-Edge Fault Tolerant Subgraph: build a subgraph H where connected components of $H \setminus F$ are same as $G \setminus F$ $\forall F \subseteq E, |F| \leq f$
  - 2-approx [D, Koranteng, Kortsarz '22]