Steiner Network

Teodora Dobos

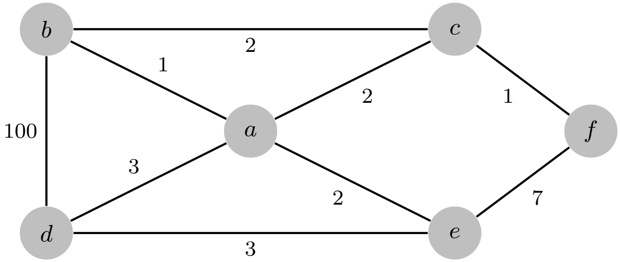
Technical University of Munich
Seminar Approximation Algorithms

July 2, 2021

What is given

- $G = (V, E)$ undirected
- cost function $c : E \to \mathbb{Q}^+$
- connectivity requirement function $r : V \times V \to \mathbb{Z}^+$
- upper bound function $u : E \to \mathbb{Z}^+ \cup \{\infty\}$

Example: $r(a, b) = 2, r(a, d) = 2,$
$u(\{a, b\}) = 2$

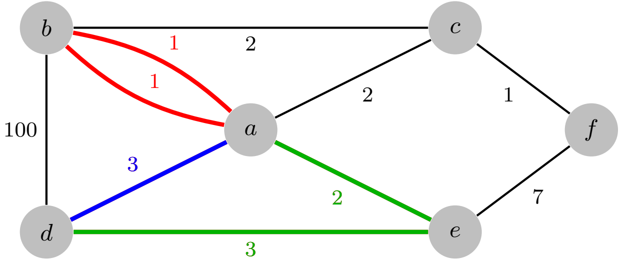


What we want

Minimal cost multigraph (V, H) such that:

- $\forall\ u, v \in V : \exists r(u, v)$ **edge disjoint paths** between u, v
- edge e is used at most $u(e)$ times

Example: $r(a, b) = 2, r(a, d) = 2,$
$u(\{a, b\}) = 2$



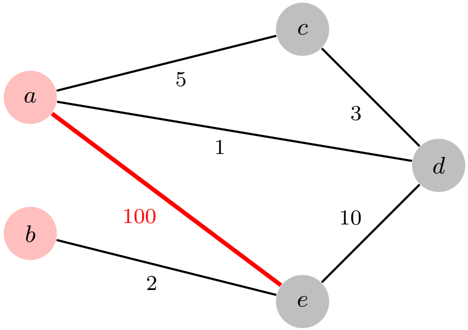
Cost: $2 \cdot 1 + 3 + (2 + 3) = 10.$

Motivation

- telecommunication industry
- low-cost networks that can survive failures of the edges
- $r(u,v)-1$ edge failures $\implies$ u and v are still connected
- wish to have certain pairs of vertices highly connected

- $f : 2^V \to \mathbb{Z}^+$
- $f(S) = \max\{r(u,v) | u \in S, v \in \bar{S}\}$

Example: $S = \{a, b\}, \bar{S} = \{c, d, e\},\ f(S) = 100$



The weight of an edge $e = \{u, v\}$ denotes connectivity requirement $r(u, v)$.

IP

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

$\rightarrow$ solution: m-dimensional vector x, where $m = |E|$.

LP relaxation

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

$\rightarrow$ LP relaxation solution $\widehat{=}$ **lower bound** for the initial min. problem

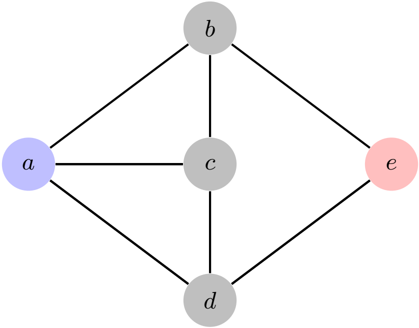
Why is the LP correct?

By the MaxFlow/MinCut theorem:

$\exists r(u,v)$ edge-disjoint paths $u-v$ in $(V,H) \iff \forall$ cut $(S,\bar{S})$ with $u \in S$, $v \in \bar{S}$ contains at least $r(u,v)$ edges of H.

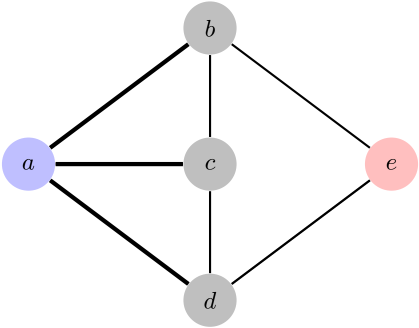
Are there 2 edge-disjoint paths between a and e? $\rightarrow$ analyse **every** cut $(S, \bar{S})$ with $a \in S$ and $e \in \bar{S}$!

$\rightarrow$ 'solution' graph with $x_e = 1 \quad \forall e \in H$



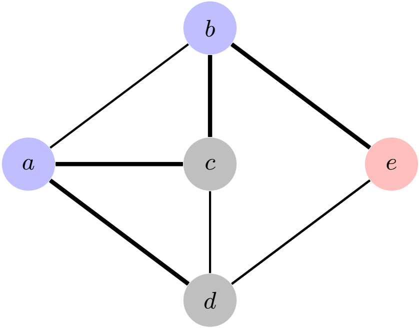
Are there 2 edge-disjoint paths between a and e in this graph? $\rightarrow$ analyse every cut $(S, \bar{S})$ with $a \in S$ and $e \in \bar{S}$!

$S = \{a\}$



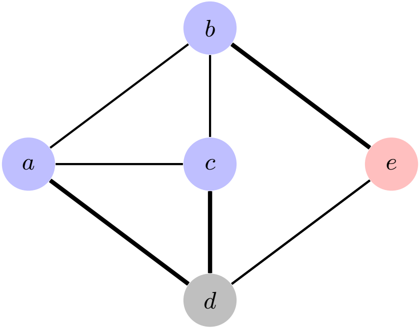
Are there 2 edge-disjoint paths between a and e in this graph? $\rightarrow$ analyse every cut $(S, \bar{S})$ with $a \in S$ and $e \in \bar{S}$!

$S = \{a, b\}$



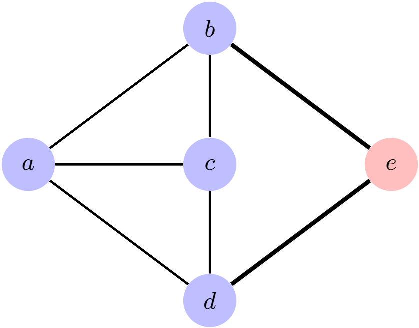
Are there 2 edge-disjoint paths between a and e in this graph? $\rightarrow$ analyse every cut $(S, \bar{S})$ with $a \in S$ and $e \in \bar{S}$!

$S = \{a, b, c\}$



Are there 2 edge-disjoint paths between a and e in this graph? $\rightarrow$ analyse every cut $(S, \bar{S})$ with $a \in S$ and $e \in \bar{S}$!

$S = \{a, b, c, d\}$



Why is the LP correct?

$\rightarrow \delta_H(S) :=$ set of edges in H crossing the cut $(S, \bar{S})$.

Set H of edges is feasible iff

$$|\delta_H(S)| \geq \max\{r(u,v) | u \in S, v \in \bar{S}\} = f(S)$$

for all $S \subseteq V$.

LP analysis

- exponentially many possible cuts $(2^{n-1} - 1)$
- polytope P has unmanageable many extreme points
- using Simplex is not feasible
- design approximation algorithm that runs in polynomial time

  $\rightarrow$ use **ellipsoid method** with a **polynomial time separation oracle**

Algorithm 1: Steiner network

1. $H \leftarrow \emptyset, f' \ \leftarrow f$
2. While H is not a feasible solution do:
   2.1 Solve LP on edge set $E - H$ with cut requirements f' to obtain BFS x.
   2.2 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.
   2.3 Update f': for $S \subseteq V, f'(S) \leftarrow f(S) - |\delta_H(S)|$.
3. Return H.

Iterated rounding technique

We iteratively round up the LP solution to create the final feasible solution.

Each iteration:

- a set of variables x_e are made integral $\Rightarrow$ *residual problem*
- solve the LP relaxation of the *residual problem*
- always round variables up by at most a factor of 2

  $\rightarrow$ **2-approximation algorithm**

Goal: Prove that the algorithm gives a **2-approximation** for the SNP in **polynomial time**.

Weakly supermodular functions

$f : 2^V \to \mathbb{Z}^+$ is *weakly supermodular* if $f(\emptyset) = f(V) = 0$, $\forall A, B \subseteq V$ **at least** one of the following conditions holds:

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

Lemma
The original cut requirement function f is weakly supermodular.

Why does this help?

Theorem

For any BFS x to our LP such that f is a weakly supermodular function, there exists some edge $e \in E$ such that $x_e \geq 1/2$.

We will prove this later...

$\rightarrow$ at most $|E|$ iterations

Theorem

For any BFS x to our LP such that f is a weakly supermodular function, there exists some edge $e \in E$ such that $x_e \geq 1/2$.

We will prove this later...

$\rightarrow$ at most $|E|$ iterations

We know that the *original* cut requirement function f is weakly supermodular, but are all $f' = f(S) - |\delta_H(S)|$ weakly supermodular?

Yes!

Lemma

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

Original and residual cut requirement functions

Lemma

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

$\rightarrow$ we need to characterize $|\delta_H(.)|$ in order to be able to prove this lemma

$f : 2^V \to \mathbb{Z}^+$ is *submodular* if $f(V) = 0$, $\forall A, B \subseteq V$:

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

Two sets of V, A and B, are said to *cross* if the sets $A - B$, $B - A$, and $A \cup B$ are not empty.

Lemma

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

Proof

Case 1: A and B do not cross.

Both conditions hold trivially.

Case 2: A and B cross.

Case 2: A and B cross.

Three possible types of edges:

-
-
-

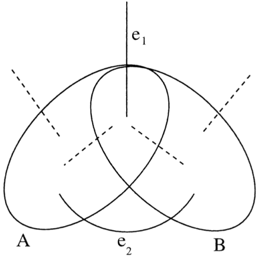


Figure: Sets A and B (Source: [1]).

- counting argument
- $|\delta(A)| + |\delta(B)| \geq |\delta(A \cap B)| + |\delta(A \cup B)|$
- $|\delta(A)| + |\delta(B)| \geq |\delta(A - B)| + |\delta(B - A)|$

Case 2: A and B cross.

Three possible types of edges:

- $e_1 = \{x, y\}, x \in A \cap B, y \in \overline{A \cup B}$
-
-

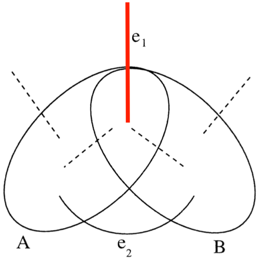


Figure: Sets A and B (Source: [1]).

- counting argument
- $|\delta(A)| + |\delta(B)| \geq |\delta(A \cap B)| + |\delta(A \cup B)|$
- $|\delta(A)| + |\delta(B)| \geq |\delta(A - B)| + |\delta(B - A)|$

Case 2: A and B cross.

Three possible types of edges:

- $e_1 = \{x, y\}, x \in A \cap B, y \in \overline{A \cup B}$
- $e_2 = \{x, y\}, x \in A - B, y \in B - A$
-

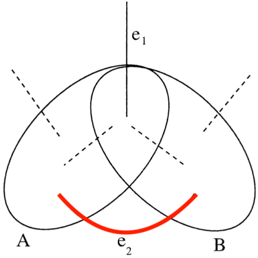


Figure: Sets A and B (Source: [1]).

- counting argument
- $|\delta(A)| + |\delta(B)| \geq |\delta(A \cap B)| + |\delta(A \cup B)|$
- $|\delta(A)| + |\delta(B)| \geq |\delta(A - B)| + |\delta(B - A)|$

Case 2: A and B cross.

Three possible types of edges:

- $e_1 = \{x, y\}, x \in A \cap B, y \in \overline{A \cup B}$
- $e_2 = \{x, y\}, x \in A - B, y \in B - A$
- neither e_1, nor e_2

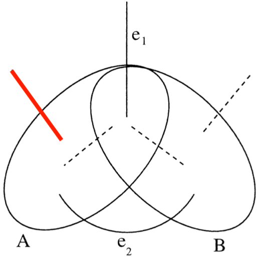


Figure: Sets A and B (Source: [1]).

- counting argument
- $|\delta(A)| + |\delta(B)| \geq |\delta(A \cap B)| + |\delta(A \cup B)|$
- $|\delta(A)| + |\delta(B)| \geq |\delta(A - B)| + |\delta(B - A)|$

Lemma
If $f : 2^{V(G)} \to \mathbb{Z}^+$ is weakly supermodular, then so is the residual cut requirement function f'.

Proof
Suppose $f(A) + f(B) \leq f(A - B) + f(B - A)$.

Back to previous lemma

Lemma

If $f : 2^{V(G)} \to \mathbb{Z}^+$ *is weakly supermodular, then so is the residual cut requirement function* f'*.*

Proof

Suppose $f(A) + f(B) \leq f(A - B) + f(B - A)$.

Since $|\delta(.)|$ is submodular: $|\delta(A)| + |\delta(B)| \geq |\delta(A - B)| + |\delta(B - A)|$.

Lemma

If $f : 2^{V(G)} \to \mathbb{Z}^+$ is weakly supermodular, then so is the residual cut requirement function f'.

Proof

Suppose $f(A) + f(B) \leq f(A - B) + f(B - A)$.

Since $|\delta(.)|$ is submodular: $|\delta(A)| + |\delta(B)| \geq |\delta(A - B)| + |\delta(B - A)|$.

By definition, $f'(A) = f(A) - |\delta(A)|$ and $f'(B) = f(B) - |\delta(B)|$.

Lemma

If $f : 2^{V(G)} \to \mathbb{Z}^+$ is weakly supermodular, then so is the residual cut requirement function f'.

Proof

Suppose $f(A) + f(B) \leq f(A - B) + f(B - A)$.

Since $|\delta(.)|$ is submodular: $|\delta(A)| + |\delta(B)| \geq |\delta(A - B)| + |\delta(B - A)|$.

By definition, $f'(A) = f(A) - |\delta(A)|$ and $f'(B) = f(B) - |\delta(B)|$.

Then:
$f'(A) + f'(B) = (f(A) - |\delta(A)|) + (f(B) - |\delta(B)|) \leq$
$(f(A - B) - |\delta(A - B)|) + (f(B - A) - |\delta(B - A)|) = f'(A - B) + f'(B - A)$.

Lemma

If $f : 2^{V(G)} \to \mathbb{Z}^+$ is weakly supermodular, then so is the residual cut requirement function f'.

Proof

Suppose $f(A) + f(B) \leq f(A - B) + f(B - A)$.

Since $|\delta(.)|$ is submodular: $|\delta(A)| + |\delta(B)| \geq |\delta(A - B)| + |\delta(B - A)|$.

By definition, $f'(A) = f(A) - |\delta(A)|$ and $f'(B) = f(B) - |\delta(B)|$.

Then:
$f'(A) + f'(B) = (f(A) - |\delta(A)|) + (f(B) - |\delta(B)|) \leq$
$(f(A - B) - |\delta(A - B)|) + (f(B - A) - |\delta(B - A)|) = f'(A - B) + f'(B - A)$.

Inequality $f(A) + f(B) \leq f(A \cap B) + f(A \cup B)$ follows in a similar manner.

Central theorem

Theorem

For any BFS x to our LP such that f is a weakly supermodular function, there exists some edge $e \in E$ such that $x_e \geq 1/2$.

Central theorem

Theorem

For any BFS x to our LP such that f is a weakly supermodular function, there exists some edge $e \in E$ such that $x_e \geq 1/2$.

$\rightarrow$ in order to prove this we need another theorem...

Tight set of vertices

Given a solution x to the LP.

Set $S \subseteq V$ is *tight* if:

$$\delta_x(S) = \sum_{e : e \in \delta(S)} x_e = f(S)$$

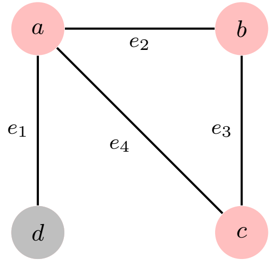
.

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

$$\chi_{\delta(S)_i} = \begin{cases} 1 & \text{if } e_i \in \delta(S), \\ 0 & \text{otherwise} \end{cases}$$

Example:

$S = \{a, b, c\}$, $\chi_{\delta(S)} = (1, 0, 0, 0)$



Laminar collection

A collection $\{S_1, S_2, ...\}$ is called *laminar* if for every i, j the intersection of the sets S_i and S_j is either empty, or equals S_i, or equals S_j.

Theorem

For any BFS x to our LP with f a weakly supermodular function, there exists a collection $\mathcal{L}$ of subsets of vertices with:

1. *For all $S \in \mathcal{L}$, S is tight.*
2. *The vectors $\chi_{\delta(S)}$ for $S \in \mathcal{L}$ are linearly independent.*
3. $|\mathcal{L}| = |E|$.
4. *The collection $\mathcal{L}$ is laminar.*

Theorem

For any BFS x to our LP with f a weakly supermodular function, there exists a collection $\mathcal{L}$ of subsets of vertices with:

1. *For all $S \in \mathcal{L}$, S is tight.*
2. *The vectors $\chi_{\delta(S)}$ for $S \in \mathcal{L}$ are linearly independent.*
3. $|\mathcal{L}| = |E|$.
4. *The collection $\mathcal{L}$ is laminar.*

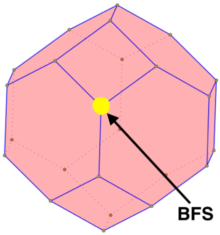


Figure: Polytope

Suppose that for all $e \in E$, $0 < x_e < 1/2$.

Proof idea:

Suppose that for all $e \in E$, $0 < x_e < 1/2$.

Proof idea:

Part 1:

- formulate a charging scheme for the sets $S \in \mathcal{L}$
- total charge $< |E|$

Suppose that for all $e \in E$, $0 < x_e < 1/2$.

Proof idea:

Part 1:

- formulate a charging scheme for the sets $S \in \mathcal{L}$
- total charge $< |E|$

Part 2:

- show that each $S \in \mathcal{L}$ receives a charge of at least one
- total charge $\geq |E|$

Proof of Central theorem - Part 1

For each $e = \{u, v\} \in E$ distribute charge :

- $1 - 2x_e > 0$ to the smallest set $S \in \mathcal{L}$ such that $u, v \in S$
- $x_e > 0$ to the smallest set $S \in \mathcal{L}$ such that $u \in S$
- $x_e > 0$ to the smallest set $S \in \mathcal{L}$ such that $v \in S$

For each $e = \{u, v\} \in E$ distribute charge :

- $1 - 2x_e > 0$ to the smallest set $S \in \mathcal{L}$ such that $u, v \in S$
- $x_e > 0$ to the smallest set $S \in \mathcal{L}$ such that $u \in S$
- $x_e > 0$ to the smallest set $S \in \mathcal{L}$ such that $v \in S$

$\implies$ distributed charge per edge ≤ 1

Proof of Central theorem - Part 1

Consider $S \in \mathcal{L}$ such that S is not contained in any other set.

- $\exists e \in \delta(S)$

Consider $S \in \mathcal{L}$ such that S is not contained in any other set.

- $\exists e \in \delta(S)$
- charge $1 - 2x_e$ is not distributed $\implies$ distributed charge for e is < 1

Proof of Central theorem - Part 1

Consider $S \in \mathcal{L}$ such that S is not contained in any other set.

- $\exists e \in \delta(S)$
- charge $1 - 2x_e$ is not distributed $\implies$ distributed charge for e is < 1

$\implies$ total distributed charge $< |E|$

Proof of Central theorem - Part 2

- $C \in \mathcal{L}$ is a *child* of S if $C \in S$ and $\nexists C' \in \mathcal{L}$ such that $C \in C'$ and $C' \in S$
- divide the edges related to S and its children into four sets

What is the charge contribution to S for each edge in the four sets?

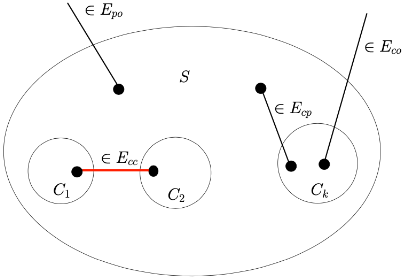


Figure: Illustration of the edge sets (Source: [2])

- E_{cc}: $1 - 2x_e$
- E_{cp}: $1 - x_e$
- E_{po}: x_e
- E_{co}: 0

What is the charge contribution to S for each edge in the four sets?

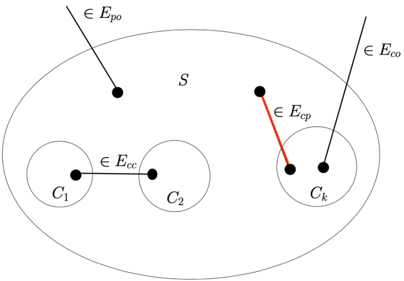


Figure: Illustration of the edge sets (Source: [2])

- E_{cc}: $1 - 2x_e$
- E_{cp}: $1 - x_e$
- E_{po}: x_e
- E_{co}: 0

What is the charge contribution to S for each edge in the four sets?

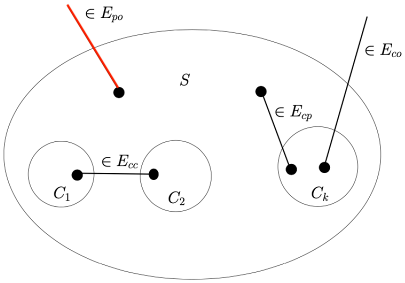


Figure: Illustration of the edge sets (Source: [2])

- E_{cc}: $1 - 2x_e$
- E_{cp}: $1 - x_e$
- E_{po}: x_e
- E_{co}: 0

What is the charge contribution to S for each edge in the four sets?

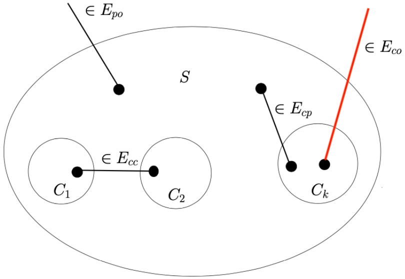


Figure: Illustration of the edge sets (Source: [2])

- E_{cc}: $1 - 2x_e$
- E_{cp}: $1 - x_e$
- E_{po}: x_e
- E_{co}: 0

Proof of Central theorem - Part 2

- impossible that all edges of E_S are in E_{co}
- otherwise: $e \in \delta(S)$ iff $e \in \delta(C_i)$ for some $i \implies \chi_{\delta(S)} = \sum_{i=1}^{k} \chi_{\delta(C_i)} \lightning$
- at least one edge of E_S must be in E_{cc}, E_{cp} or E_{po}

Total charge received by S:

$$|E_{cc}| - 2x(E_{cc}) + |E_{cp}| - x(E_{cp}) + x(E_{po}) > 0$$

By definition:

$$x(\delta(S)) - \sum_{i=1}^{k} x(\delta(C_i)) = x(E_{po}) - x(E_{cp}) - 2x(E_{cc}).$$

Total charge received by S:

$$|E_{cc}| + |E_{cp}| + \left(x(\delta(S)) - \sum_{i=1}^{k} x(\delta(C_i))\right) =$$

Total charge received by S:

$$|E_{cc}| - 2x(E_{cc}) + |E_{cp}| - x(E_{cp}) + x(E_{po}) > 0$$

By definition:

$$x(\delta(S)) - \sum_{i=1}^{k} x(\delta(C_i)) = x(E_{po}) - x(E_{cp}) - 2x(E_{cc}).$$

Total charge received by S:

$$|E_{cc}| + |E_{cp}| + \left(x(\delta(S)) - \sum_{i=1}^{k} x(\delta(C_i))\right) =$$

$$|E_{cc}| + |E_{cp}| + \left(f(S) - \sum_{i=1}^{k} f(C_i)\right)$$

$\implies$ total charge is at least 1

Proof of Central theorem - Part 2

- each $S \in \mathcal{L}$ receives a charge of at least 1
- $|\mathcal{L}| = |E|$

$\implies$ total charge $\geq |E|$

But in Part 1 we showed that the distributed charge $< |E|$ ↯

LP solvable in polynomial time

Lemma

For any $H \subseteq E$ we can solve the LP in polynomial time with edge set $E - H$ and function $g(S) = f(S) - |\delta(S) \cap H|$ when $f(S) = max\{r(u,v)|u \in S, v \in \bar{S}\}$.

$\rightarrow$ **polynomial-time separation oracle** based on maxFlow applied in the ellipsoid method

Theorem

The algorithm achieves an approximation guarantee of 2 for the Steiner network problem.

Proof sketch

Notation:

- x_i^* := BFS computed in the i-th iteration
- H_i := set of edges chosen by the algorithm in the i-th iteration
- cost$(H_i) = \sum_{e \in H_i} c_e x_e$, where $x_e = \lceil {x_i}_e^* \rceil$ if ${x_i}_e^* \geq 1/2$

Approximation guarantee of 2

Assume that the algorithm has k iterations.

$\text{cost}(x_1^*) \leq 2OPT$

Assume that the algorithm has k iterations.

$\text{cost}(x_1^*) \leq 2OPT$

$\text{cost}(x_2^*) \leq \text{cost}(x_1^*) - \text{cost}(H_1) \implies \text{cost}(H_1) \leq \text{cost}(x_1^*) - \text{cost}(x_2^*)$

Assume that the algorithm has k iterations.

$$\text{cost}(x_1^*) \leq 2OPT$$

$$\text{cost}(x_2^*) \leq \text{cost}(x_1^*) - \text{cost}(H_1) \implies \text{cost}(H_1) \leq \text{cost}(x_1^*) - \text{cost}(x_2^*)$$

$$\text{cost}(x_3^*) \leq \text{cost}(x_2^*) - \text{cost}(H_2) \implies \text{cost}(H_2) \leq \text{cost}(x_2^*) - \text{cost}(x_3^*)$$

$$\text{cost}(x_4^*) \leq \text{cost}(x_3^*) - \text{cost}(H_3) \implies \text{cost}(H_3) \leq \text{cost}(x_3^*) - \text{cost}(x_4^*)$$

$$\text{cost}(x_5^*) \leq \text{cost}(x_4^*) - \text{cost}(H_4) \implies \text{cost}(H_4) \leq \text{cost}(x_4^*) - \text{cost}(x_5^*)$$

$$\vdots$$

$$\text{cost}(x_{k+1}^*) \leq \text{cost}(x_k^*) - \text{cost}(H_k) \implies \text{cost}(H_k) \leq \text{cost}(x_k^*) - \text{cost}(x_{k+1}^*)$$

Approximation guarantee of 2

Assume that the algorithm has k iterations.

$\text{cost}(x_1^*) \leq 2OPT$

$\text{cost}(x_2^*) \leq \text{cost}(x_1^*) - \text{cost}(H_1) \implies \text{cost}(H_1) \leq \text{cost}(x_1^*) - \text{cost}(x_2^*)$

$\text{cost}(x_3^*) \leq \text{cost}(x_2^*) - \text{cost}(H_2) \implies \text{cost}(H_2) \leq \text{cost}(x_2^*) - \text{cost}(x_3^*)$

$\text{cost}(x_4^*) \leq \text{cost}(x_3^*) - \text{cost}(H_3) \implies \text{cost}(H_3) \leq \text{cost}(x_3^*) - \text{cost}(x_4^*)$

$\text{cost}(x_5^*) \leq \text{cost}(x_4^*) - \text{cost}(H_4) \implies \text{cost}(H_4) \leq \text{cost}(x_4^*) - \text{cost}(x_5^*)$

$$\vdots$$

$\text{cost}(x_{k+1}^*) \leq \text{cost}(x_k^*) - \text{cost}(H_k) \implies \text{cost}(H_k) \leq \text{cost}(x_k^*) - \text{cost}(x_{k+1}^*)$

Total cost:

$\text{cost}(H_1) + \text{cost}(H_2) + \ldots + \text{cost}(H_k) \leq$

Approximation guarantee of 2

Assume the algorithms needs k iterations.

$$\text{cost}(x_1^*) \leq 2OPT$$

$$\text{cost}(x_2^*) \leq \text{cost}(x_1^*) - \text{cost}(H_1) \implies \text{cost}(H_1) \leq \text{cost}(x_1^*) - \text{cost}(x_2^*)$$

$$\text{cost}(x_3^*) \leq \text{cost}(x_2^*) - \text{cost}(H_2) \implies \text{cost}(H_2) \leq \text{cost}(x_2^*) - \text{cost}(x_3^*)$$

$$\text{cost}(x_4^*) \leq \text{cost}(x_3^*) - \text{cost}(H_3) \implies \text{cost}(H_3) \leq \text{cost}(x_3^*) - \text{cost}(x_4^*)$$

$$\text{cost}(x_5^*) \leq \text{cost}(x_4^*) - \text{cost}(H_4) \implies \text{cost}(H_4) \leq \text{cost}(x_4^*) - \text{cost}(x_5^*)$$

$$\vdots$$

$$\text{cost}(x_{k+1}^*) \leq \text{cost}(x_k^*) - \text{cost}(H_k) \implies \text{cost}(H_k) \leq \text{cost}(x_k^*) - \text{cost}(x_{k+1}^*)$$

Total cost:

$$\text{cost}(H_1) + \text{cost}(H_2) + \ldots + \text{cost}(H_k) \leq \text{cost}(x_1^*) \leq 2OPT. \ \square$$

Vijay V. Vazirani.
Approximation Algorithms.
Springer Publishing Company, Incorporated, 2010.

David P. Williamson and David B. Shmoys.
The Design of Approximation Algorithms.
Cambridge University Press, 2011.