Matroids and Integrality Gaps for Hypergraphic Steiner Tree Relaxations

Rico Zenklusen

Johns Hopkins University

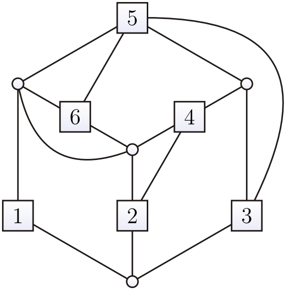
Joint work with:

Michel Goemans, Neil Olver, Thomas Rothvoß

The Steiner Tree problem

Given:
- Graph $G = (V, E)$,
- terminals $R \subseteq V$,
- edge costs $c : E \to \mathbb{R}_+$.

Goal: Find $T \subseteq E$ connecting R & minimizing $c(T) := \sum_{e \in T} c(e)$.



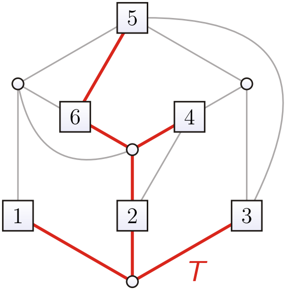
□ : R (terminals)

○ : $V \setminus R$ (Steiner nodes)

The Steiner Tree problem

Given:
- Graph $G = (V, E)$,
- terminals $R \subseteq V$,
- edge costs $c : E \to \mathbb{R}_+$.

Goal: Find $T \subseteq E$ connecting R & minimizing $c(T) := \sum_{e \in T} c(e)$.



□ : R (terminals)

∘ : $V \setminus R$ (Steiner nodes)

Approximation results

	Approximation guarantee
Folklore	2
Zelikovsky [1991]	$11/6 \leq 1.83$
Berman & Ramaiyer [1991]	$16/9 \leq 1.78$
Zelikovsky [1993]	$1 + \ln(2) + \epsilon \leq 1.70$
Karpinski & Zelikovsky [1997]	≤ 1.65
Prömel & Steger [2000]	$5/3 \leq 1.67$
Hougardy & Prömel [1999]	≤ 1.59
Robins & Zelikovsky [2005]	$1 + \ln(3)/2 \leq 1.55$
Byrka, Grandoni, Rothvoß, Sanità [2010]	$\ln(4) + \epsilon \leq 1.39$

Approximation results

	Approximation guarantee
Folklore	2
Zelikovsky [1991]	$11/6 \leq 1.83$
Berman & Ramaiyer [1991]	$16/9 \leq 1.78$
Zelikovsky [1993]	$1 + \ln(2) + \epsilon \leq 1.70$
Karpinski & Zelikovsky [1997]	≤ 1.65
Prömel & Steger [2000]	$5/3 \leq 1.67$
Hougardy & Prömel [1999]	≤ 1.59
Robins & Zelikovsky [2005]	$1 + \ln(3)/2 \leq 1.55$
Byrka, Grandoni, Rothvoß, Sanità [2010]	$\ln(4) + \epsilon \leq 1.39$

Major challenge in design of strong approximation algorithms:

➙ Very bad understanding of LP relaxations.

Towards LP-based approaches

Until $(\ln(4) + \epsilon) \approx 1.39$-approx of [Byrka et al., 2010], best Steiner Tree approximations did not use LP relaxations.

Towards LP-based approaches

Until $(\ln(4) + \epsilon) \approx 1.39$-approx of [Byrka et al., 2010], best Steiner Tree approximations did not use LP relaxations.

Algorithm of Byrka, Grandoni, Rothvoß & Sanità

- Relies on a hypergraphic LP relaxation.
- However, only shown that returned solution has cost $\leq 1.39 OPT$, and not $\leq 1.39 LP$. In particular, no implication on integrality gap of LP!
- Algo can be adapted to use $O(\log(n))$ random bits $\Rightarrow$ derandomization.

Towards LP-based approaches

Until $(\ln(4) + \epsilon) \approx 1.39$-approx of [Byrka et al., 2010], best Steiner Tree approximations did not use LP relaxations.

Algorithm of Byrka, Grandoni, Rothvoß & Sanità

- Relies on a hypergraphic LP relaxation.
- However, only shown that returned solution has cost $\leq 1.39OPT$, and not $\leq 1.39LP$. In particular, no implication on integrality gap of LP!
- Algo can be adapted to use $O(\log(n))$ random bits $\Rightarrow$ derandomization.

- Furthermore they show: solution of 1.55-approx of Robins & Zelikovsky has cost $\leq 1.55LP$.

Towards LP-based approaches

Until $(\ln(4) + \epsilon) \approx 1.39$-approx of [Byrka et al., 2010], best Steiner Tree approximations did not use LP relaxations.

Algorithm of Byrka, Grandoni, Rothvoß & Sanità

- Relies on a hypergraphic LP relaxation.
- However, only shown that returned solution has cost $\leq 1.39 OPT$, and not $\leq 1.39 LP$. In particular, no implication on integrality gap of LP!
- Algo can be adapted to use $O(\log(n))$ random bits $\Rightarrow$ derandomization.

- Furthermore they show: solution of 1.55-approx of Robins & Zelikovsky has cost $\leq 1.55 LP$.

First result showing an integrality gap < 2
for an LP relaxation for Steiner Tree.

Towards LP-based approaches

Until $(\ln(4) + \epsilon) \approx 1.39$-approx of [Byrka et al., 2010], best Steiner Tree approximations did not use LP relaxations.

Algorithm of Byrka, Grandoni, Rothvoß & Sanità

- Relies on a hypergraphic LP relaxation.
- However, only shown that returned solution has cost $\leq 1.39OPT$, and not $\leq 1.39LP$. In particular, no implication on integrality gap of LP!
- Algo can be adapted to use $O(\log(n))$ random bits $\Rightarrow$ derandomization.

- Furthermore they show: solution of 1.55-approx of Robins & Zelikovsky has cost $\leq 1.55LP$.

First result showing an integrality gap < 2
for an LP relaxation for Steiner Tree.

➙ We want to get better understanding of hypergraphic relaxation.

Our results

- New algorithm inspired by Byrka et al.:
  - Returns solution that compares against LP.
    $\Rightarrow$ ln(4) bound on integrality gap of LP.
  - Deterministic.
  - Faster: don't need to repeatedly solve LP (as Byrka et al.).

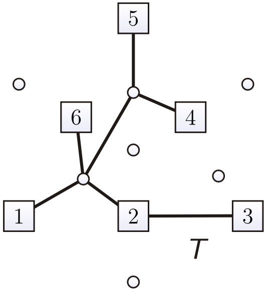
- Deeper understanding of hypergraphic LP.

- Further results for quasi-bipartite graphs.

Hypergraphic LP relaxation

Component relaxation

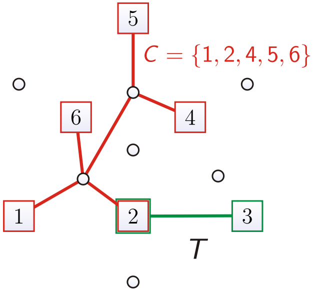
- Any Steiner Tree can be decomposed into components.
  - A component C is a tree in G whose leaves are terminals and non-leaves are Steiner nodes.
  - For simplicity: denote by C the terminals spanned by component.



Hypergraphic LP relaxation

Component relaxation

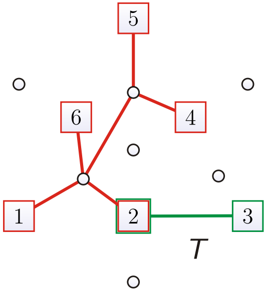
- Any Steiner Tree can be decomposed into components.
  - A component C is a tree in G whose leaves are terminals and non-leaves are Steiner nodes.
  - For simplicity: denote by C the terminals spanned by component.



Hypergraphic LP relaxation

Component relaxation

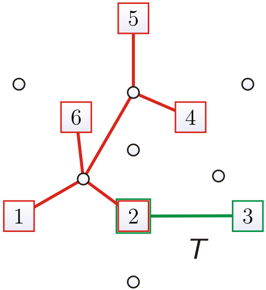
- Any Steiner Tree can be decomposed into components.



Hypergraphic LP relaxation

Component relaxation

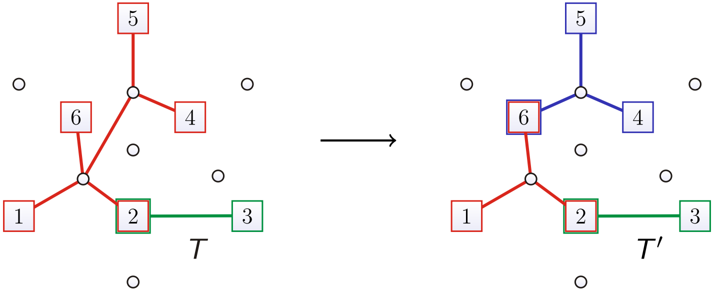
- Any Steiner Tree can be decomposed into components.
- Goal of hypergraphic LP: determine which components are used.



Hypergraphic LP relaxation

Component relaxation

- Any Steiner Tree can be decomposed into components.
- Goal of hypergraphic LP: determine which components are used.
- Borchers and Du [1997]: sufficient to consider "small" components C: if $|C| \leq 2^r$ is imposed $\Rightarrow$ best such Steiner Trees has cost $\leq 1 + \frac{1}{r}$.



Hypergraphic LP relaxation (component relaxation)

Several equivalent variants of hypergraphic relaxations.
(Warme [1998], Polzin & Vahdati Daneshmand [2003], Könemann et al [2009])

$$
\begin{aligned}
\min \quad & \sum_{C \in \mathcal{K}} x_C \operatorname{cost}(C) \\
& \sum_{C \in \mathcal{K}} x_C (|S \cap C| - 1)^+ \leq |S| - 1 \quad \forall S \subseteq R, S \neq \emptyset \\
& \sum_{C \in \mathcal{K}} x_C (|C| - 1) = |R| - 1 \\
& x_C \geq 0 \quad \forall C \in \mathcal{K}
\end{aligned}
$$

- $\mathcal{K}$: set of all components
  up to some size.
- x_C: variable for component $C \in \mathcal{K}$.
- $(\cdot)^+ := \max\{0, \cdot\}$

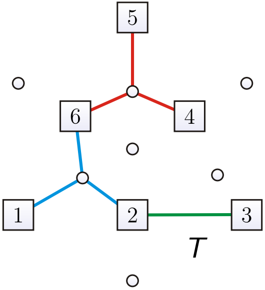
Hypergraphic LP relaxation (component relaxation)

Several equivalent variants of hypergraphic relaxations.

(Warme [1998], Polzin & Vahdati Daneshmand [2003], Könemann et al [2009])

$$
\begin{array}{lrcll}
\min & \sum_{C \in \mathcal{K}} x_C \operatorname{cost}(C) & & & \\
 & \sum_{C \in \mathcal{K}} x_C (|S \cap C| - 1)^+ & \leq & |S| - 1 & \forall S \subseteq R, S \neq \emptyset \\
 & \sum_{C \in \mathcal{K}} x_C (|C| - 1) & = & |R| - 1 & \\
 & x_C & \geq & 0 & \forall C \in \mathcal{K}
\end{array}
$$

- $\mathcal{K}$: set of all components up to some size.
- x_C: variable for component $C \in \mathcal{K}$.
- $(\cdot)^+ := \max\{0, \cdot\}$



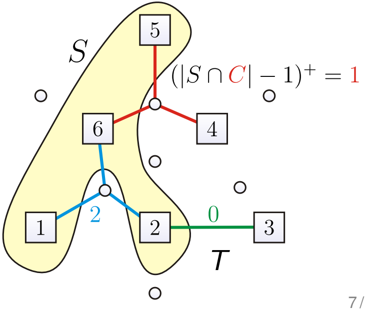
Hypergraphic LP relaxation (component relaxation)

Several equivalent variants of hypergraphic relaxations.

(Warme [1998], Polzin & Vahdati Daneshmand [2003], Könemann et al [2009])

$$
\begin{aligned}
\min \quad & \sum_{C\in\mathcal{K}} x_C \,\mathrm{cost}(C) & & \\
& \sum_{C\in\mathcal{K}} x_C(|S\cap C|-1)^+ \leq |S|-1 & & \forall S\subseteq R, S\neq\emptyset \\
& \sum_{C\in\mathcal{K}} x_C(|C|-1) = |R|-1 & & \\
& x_C \geq 0 & & \forall C\in\mathcal{K}
\end{aligned}
$$

- $\mathcal{K}$: set of all components up to some size.
- x_C: variable for component $C \in \mathcal{K}$.
- $(\cdot)^+ := \max\{0, \cdot\}$

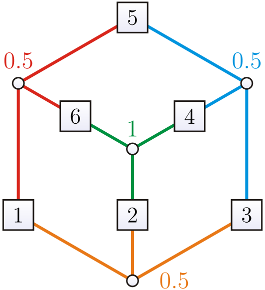


Hypergraphic LP relaxation (component relaxation)

Several equivalent variants of hypergraphic relaxations.
(Warme [1998], Polzin & Vahdati Daneshmand [2003], Könemann et al [2009])

$$\begin{array}{rrcll} \min & \sum_{C \in \mathcal{K}} x_C \operatorname{cost}(C) & & & \\ & \sum_{C \in \mathcal{K}} x_C(|S \cap C| - 1)^+ & \leq & |S| - 1 & \forall S \subseteq R, S \neq \emptyset \\ & \sum_{C \in \mathcal{K}} x_C(|C| - 1) & = & |R| - 1 & \\ & x_C & \geq & 0 & \forall C \in \mathcal{K} \end{array}$$

- $\mathcal{K}$: set of all components up to some size.
- x_C: variable for component $C \in \mathcal{K}$.
- $(\cdot)^+ := \max\{0, \cdot\}$



ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\rightarrow (x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

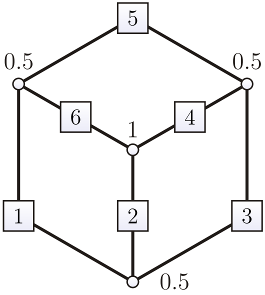
(iii) • If only one terminal left $\rightarrow$ return contracted components.
• Otherwise $\rightarrow$ back to **(i)**.

ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\to (x_C)_{C\in\mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C/(\sum_{C\in\mathcal{K}} x_C)$.

(iii)
- If only one terminal left $\to$ return contracted components.
- Otherwise $\to$ back to **(i)**.

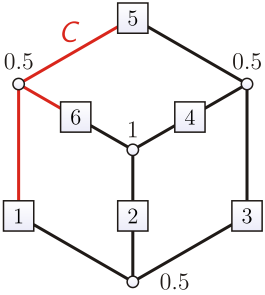


ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP → $(x_C)_{C\in\mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C/(\sum_{C\in\mathcal{K}} x_C)$.

(iii)
- If only one terminal left → return contracted components.
- Otherwise → back to **(i)**.

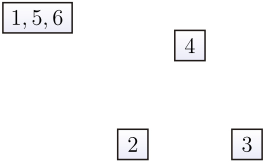


ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\rightarrow (x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii)
- If only one terminal left $\rightarrow$ return contracted components.
- Otherwise $\rightarrow$ back to **(i)**.

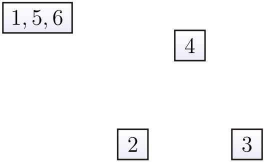


ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\rightarrow (x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii)
- If only one terminal left $\rightarrow$ return contracted components.
- Otherwise $\rightarrow$ back to **(i)**.

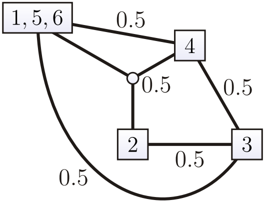


ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP → $(x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii) • If only one terminal left → return contracted components.
• Otherwise → back to **(i)**.

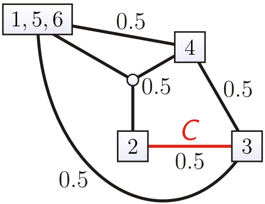


$\ln(4)$-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\rightarrow (x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii) • If only one terminal left $\rightarrow$ return contracted components.
• Otherwise $\rightarrow$ back to **(i)**.

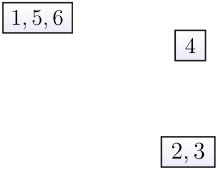


ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP → $(x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii)
- If only one terminal left → return contracted components.
- Otherwise → back to **(i)**.

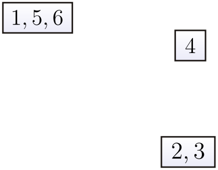


ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\rightarrow (x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii)
- If only one terminal left $\rightarrow$ return contracted components.
- Otherwise $\rightarrow$ back to **(i)**.

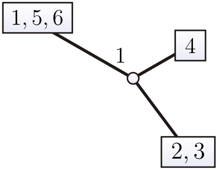


ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\rightarrow (x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii) • If only one terminal left $\rightarrow$ return contracted components.
• Otherwise $\rightarrow$ back to **(i)**.

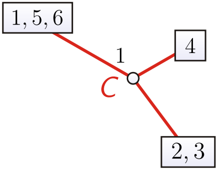


ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\rightarrow (x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii)
- If only one terminal left $\rightarrow$ return contracted components.
- Otherwise $\rightarrow$ back to **(i)**.



$\ln(4)$-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\to (x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii)
- If only one terminal left $\to$ return contracted components.
- Otherwise $\to$ back to **(i)**.

$1, 2, 3, 4, 5, 6$

$\ln(4)$-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\rightarrow (x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii)
- If only one terminal left $\rightarrow$ return contracted components.
- Otherwise $\rightarrow$ back to **(i)**.

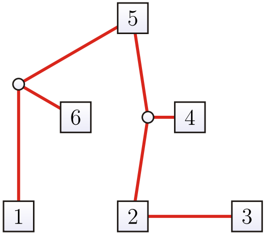
$$\boxed{1, 2, 3, 4, 5, 6}$$

ln(4)-approx of Byrka, Grandoni, Rothvoß & Sanità

(i) Solve hypergraphic LP $\rightarrow (x_C)_{C \in \mathcal{K}}$.

(ii) Contract randomly one component $C \in \mathcal{K}$ with prob $= x_C / (\sum_{C \in \mathcal{K}} x_C)$.

(iii)
- If only one terminal left $\rightarrow$ return contracted components.
- Otherwise $\rightarrow$ back to **(i)**.



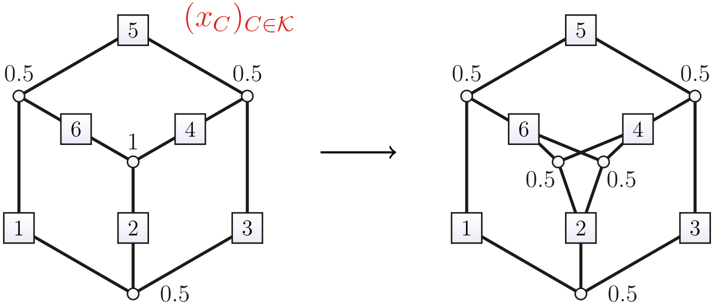
Towards a $ln(4)$ integrality gap

- We want to measure progress in terms of initial LP solution.
- After contraction, we modify current LP solution to obtain feasibility.
  - Modifications: we split up components into smaller ones.

Need better understanding of how components can be split after contraction to obtain feasibility.

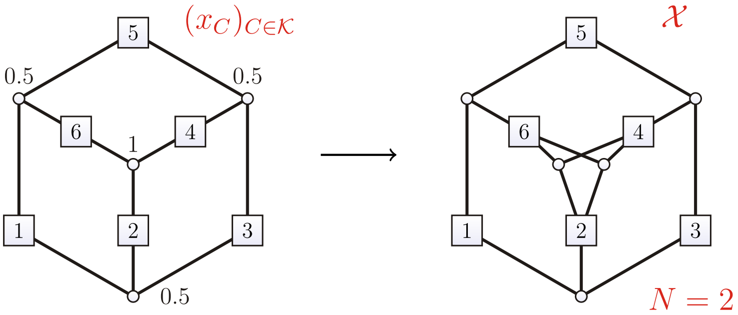
Blowup graph

Idea to simplify exposition: assume all components have same LP-value.



Blowup graph

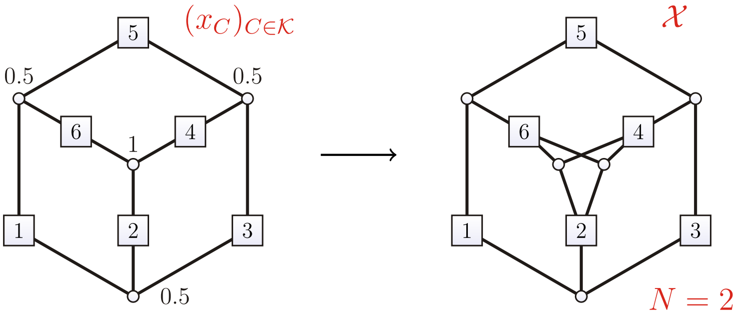
Idea to simplify exposition: assume all components have same LP-value.



- Choose N s.t. $N \cdot x_C \in \mathbb{N}$ for $C \in \mathcal{K}$.
- Make $N \cdot x_C$ copies of C.

Blowup graph

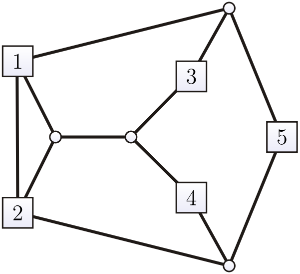
Idea to simplify exposition: assume all components have same LP-value.



- Choose N s.t. $N \cdot x_C \in \mathbb{N}$ for $C \in \mathcal{K}$.
- Make $N \cdot x_C$ copies of C.

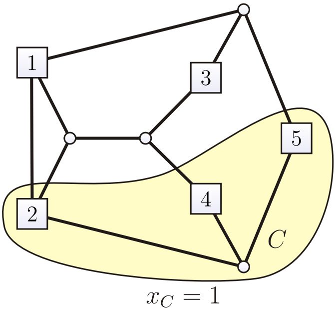
> - $\mathcal{X}$: blowup graph.
> - $\Gamma(\mathcal{X})$: components of blowup graph.
> - $E(\mathcal{X})$: edges of blowup graph.

Edge removals to obtain feasibility

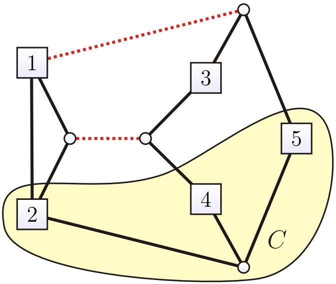


$x_C = 0.5 \; \forall C \in \Gamma(\mathcal{X})$

Edge removals to obtain feasibility

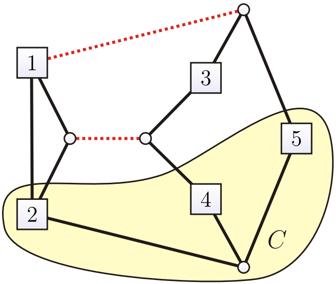


Edge removals to obtain feasibility

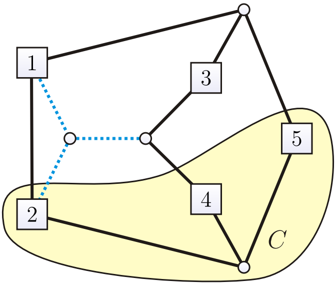


$x_C = 1$

Edge removals to obtain feasibility

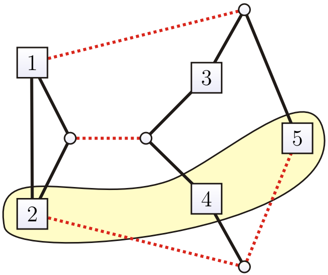


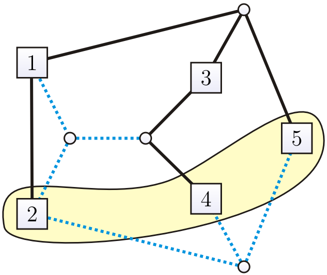
$x_C = 1$



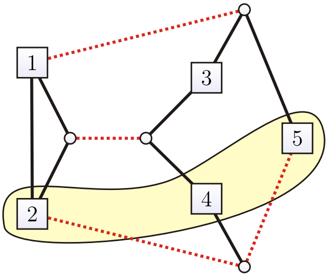
$x_C = 1$

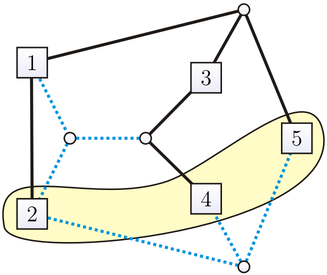
Edge removals to obtain feasibility





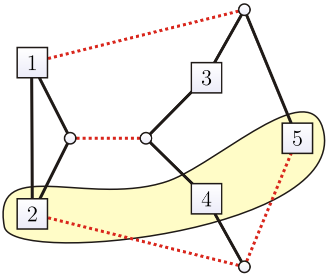
Edge removals to obtain feasibility





- Feasible sets to remove seem hard to describe directly.

Edge removals to obtain feasibility



- Feasible sets to remove seem hard to describe directly.
- It turns out that minimal removal sets are well structures:

Theorem

Minimal edge removals achieving feasibility form bases $\mathcal{B}_C$ of a matroid M_C.

Splitting sets and cleanup edges

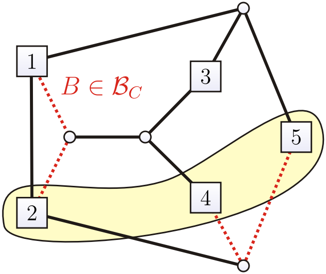
Possible removals can be described by following two-step procedure:

(i) Remove a basis B of M_C.
(ii) Clean up pendant edges.

Splitting sets and cleanup edges

Possible removals can be described by following two-step procedure:

(i) Remove a basis B of M_C.
(ii) Clean up pendant edges.

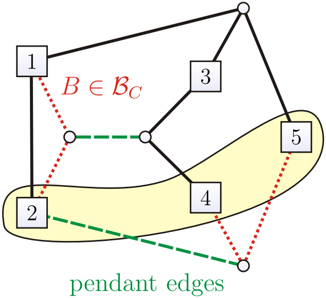


Splitting sets and cleanup edges

Possible removals can be described by following two-step procedure:

(i) Remove a basis B of M_C.

(ii) Clean up pendant edges.

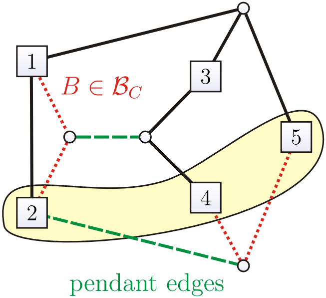


Splitting sets and cleanup edges

Possible removals can be described by following two-step procedure:

(i) Remove a basis B of M_C.

(ii) Clean up pendant edges.

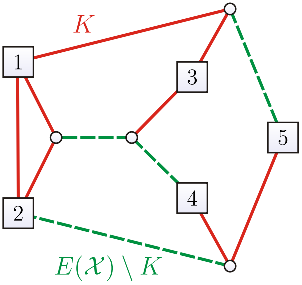


- During algo, each edge will be removed or cleaned up at some point.

Splitting sets and cleanup edges

Possible removals can be described by following two-step procedure:

(i) Remove a basis B of M_C.
(ii) Clean up pendant edges.

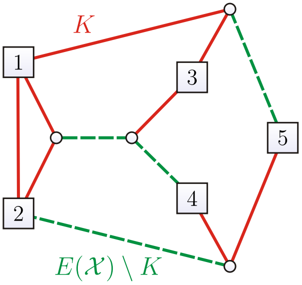


- During algo, each edge will be removed or cleaned up at some point.
- Analysis simplifies by partitioning edges at beginning into:
  - Edges to be removed eventually: $\rightarrow$ splitting set: $K \subseteq E(\mathcal{X})$.
  - Edges to be cleaned up eventually: $\rightarrow$ cleanup edges: $E(\mathcal{X}) \setminus K$.

Splitting sets and cleanup edges

Possible removals can be described by following two-step procedure:

(i) Remove a basis B of M_C.
(ii) Clean up pendant edges.



- During algo, each edge will be removed or cleaned up at some point.
- Analysis simplifies by partitioning edges at beginning into:
  - Edges to be removed eventually:
    → splitting set: $K \subseteq E(\mathcal{X})$.
  - Edges to be cleaned up eventually:
    → cleanup edges: $E(\mathcal{X}) \setminus K$.

Splitting set:	Any minimal edge-set whose removal leads to single-terminal connected components.

Witness sets

- When can an edge $f \in E(\mathcal{X}) \setminus K$ be cleaned up?

Witness sets

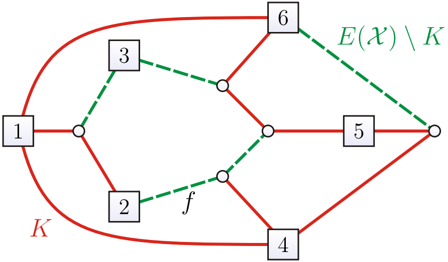
- When can an edge $f \in E(\mathcal{X}) \setminus K$ be cleaned up?
  - ➙ As soon as its witness set $W(f) \subseteq K$ is removed:

$W(f)$:	Minimal set of splitting edges in same component as f whose removal makes f pendant.

Witness sets

- When can an edge $f \in E(\mathcal{X}) \setminus K$ be cleaned up?
  - ➙ As soon as its witness set $W(f) \subseteq K$ is removed:

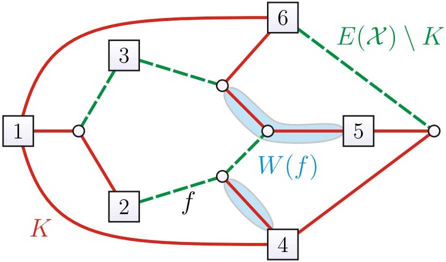
$W(f)$:	Minimal set of splitting edges in same component as f whose removal makes f pendant.



Witness sets

- When can an edge $f \in E(\mathcal{X}) \setminus K$ be cleaned up?
  - ➙ As soon as its witness set $W(f) \subseteq K$ is removed:

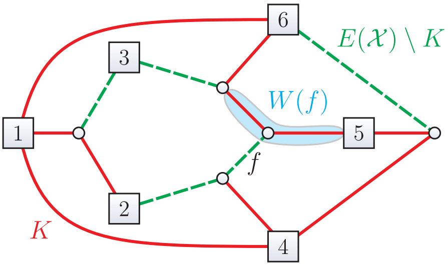
$W(f)$:	Minimal set of splitting edges in same component as f whose removal makes f pendant.



Witness sets

- When can an edge $f \in E(\mathcal{X}) \setminus K$ be cleaned up?
  - ➙ As soon as its witness set $W(f) \subseteq K$ is removed:

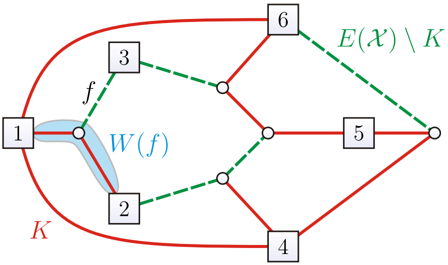
$W(f)$:	Minimal set of splitting edges in same component as f whose removal makes f pendant.



Witness sets

- When can an edge $f \in E(\mathcal{X}) \setminus K$ be cleaned up?
  - ➙ As soon as its witness set $W(f) \subseteq K$ is removed:

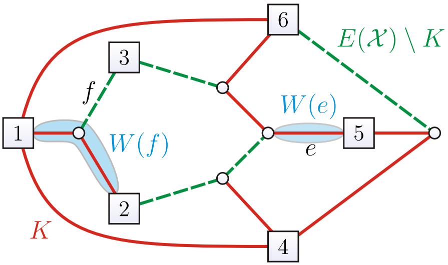
$W(f)$:	Minimal set of splitting edges in same component as f whose removal makes f pendant.



Witness sets

- When can an edge $f \in E(\mathcal{X}) \setminus K$ be cleaned up?
  - ➙ As soon as its witness set $W(f) \subseteq K$ is removed:

$W(f)$:	Minimal set of splitting edges in same component as f whose removal makes f pendant.

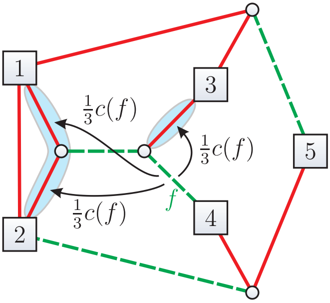


- We set $W(e) = \{e\}$ for $e \in K$.

Measuring progress via a potential function

- We understand quite well how to remove edges from splitting set K.
  - ➙ Measure progress in terms of removed splitting edges.
  - ➙ Plan: charge costs of cleanup edges to splitting edges:

$$w(e) := \frac{1}{N}\left(c(e) + \sum_{\substack{f \in E(\mathcal{X} \setminus K), \\ e \in W(f))}} \frac{c(f)}{|W(f)|} \right).$$



Measuring progress via a potential function

- We understand quite well how to remove edges from splitting set K.
  - → Measure progress in terms of removed splitting edges.
  - → Plan: charge costs of cleanup edges to splitting edges:

$$w(e) := \frac{1}{N}\left(c(e) + \sum_{\substack{f \in E(\mathcal{X} \setminus K),\\ e \in W(f))}} \frac{c(f)}{|W(f)|} \right).$$

Measuring progress via a potential function

- We understand quite well how to remove edges from splitting set K.
  - ➙ Measure progress in terms of removed splitting edges.
  - ➙ Plan: charge costs of cleanup edges to splitting edges:

$$w(e) := \frac{1}{N}\left(c(e) + \sum_{\substack{f \in E(\mathcal{X} \setminus K), \\ e \in W(f))}} \frac{c(f)}{|W(f)|} \right).$$

Potential for blowup graph $\mathcal{X}$ with splitting set K:

$$\Phi_K(\mathcal{X}) := \frac{1}{N} \sum_{e \in E(\mathcal{X})} c(e) H(|W(e)|),$$

- where $H(k) := \sum_{j=1}^{k} \frac{1}{j}$ (harmonic function).

Measuring progress via a potential function

Theorem

(a) When contracting any $C \in \mathcal{K}$, removing $B \subseteq \mathcal{B}_C, B \subseteq K$ and cleaning up: $|\Delta\Phi_K(\mathcal{X})| \geq w(B)$.

(b) $\exists\ C \in \mathcal{K}$ and basis B of M_C, $B \subseteq K$ s.t. $\text{cost}(C) \leq w(B)$.

Measuring progress via a potential function

Theorem

(a) When contracting any $C \in \mathcal{K}$, removing $B \subseteq \mathcal{B}_C$, $B \subseteq K$ and cleaning up: $|\Delta\Phi_K(\mathcal{X})| \geq w(B)$.

(b) $\exists\ C \in \mathcal{K}$ and basis B of M_C, $B \subseteq K$ s.t. $\text{cost}(C) \leq w(B)$.

$\Rightarrow$ Following algo returns Steiner tree of cost $\leq \Phi_K(\mathcal{X})$:

(i) Solve hypergraphic LP.

(ii) Find $C \in \mathcal{K}$, & basis B of M_C, $B \subseteq K$ s.t. $\text{cost}(C) \leq w(B)$.

(iii) Contract C, remove B & cleanup.

(iv)
- If only one terminal left $\rightarrow$ return contracted components.
- Otherwise $\rightarrow$ back to **(ii)**.

Measuring progress via a potential function

Theorem

(a) When contracting any $C \in \mathcal{K}$, removing $B \subseteq \mathcal{B}_C, B \subseteq K$ and cleaning up: $|\Delta\Phi_K(\mathcal{X})| \geq w(B)$.

(b) $\exists\ C \in \mathcal{K}$ and basis B of M_C, $B \subseteq K$ s.t. $\text{cost}(C) \leq w(B)$.

$\Rightarrow$ Algo returns Steiner Tree of cost $\leq \Phi_K(\mathcal{X})$.

Measuring progress via a potential function

Theorem

(a) When contracting any $C \in \mathcal{K}$, removing $B \subseteq \mathcal{B}_C, B \subseteq K$ and cleaning up: $|\Delta\Phi_K(\mathcal{X})| \geq w(B)$.

(b) $\exists$ $C \in \mathcal{K}$ and basis B of M_C, $B \subseteq K$ s.t. $\text{cost}(C) \leq w(B)$.

$\Rightarrow$ Algo returns Steiner Tree of cost $\leq \Phi_K(\mathcal{X})$.

(c) $\exists$ splitting set K s.t. $\Phi_K(\mathcal{X}) \leq \ln(4) cost(LP)$.

(d) Splitting set K minimizing $\Phi_K(\mathcal{X})$ can be found efficiently.

Measuring progress via a potential function

Theorem

(a) When contracting any $C \in \mathcal{K}$, removing $B \subseteq \mathcal{B}_C, B \subseteq K$ and cleaning up: $|\Delta\Phi_K(\mathcal{X})| \geq w(B)$.

(b) $\exists$ $C \in \mathcal{K}$ and basis B of M_C, $B \subseteq K$ s.t. cost$(C) \leq w(B)$.

$\Rightarrow$ Algo returns Steiner Tree of cost $\leq \Phi_K(\mathcal{X})$.

(c) $\exists$ splitting set K s.t. $\Phi_K(\mathcal{X}) \leq \ln(4) cost(LP)$.

(d) Splitting set K minimizing $\Phi_K(\mathcal{X})$ can be found efficiently.

$\Rightarrow$ Suggested algo leads to cost $\leq \ln(4)\,\text{cost}(LP)$.

Measuring progress via a potential function

Theorem

(a) When contracting any $C \in \mathcal{K}$, removing $B \subseteq \mathcal{B}_C$, $B \subseteq K$ and cleaning up: $|\Delta\Phi_K(\mathcal{X})| \geq w(B)$.

(b) $\exists\ C \in \mathcal{K}$ and basis B of M_C, $B \subseteq K$ s.t. $\text{cost}(C) \leq w(B)$.

$\Rightarrow$ Algo returns Steiner Tree of cost $\leq \Phi_K(\mathcal{X})$.

(c) $\exists$ splitting set K s.t. $\Phi_K(\mathcal{X}) \leq \ln(4) cost(LP)$.

(d) Splitting set K minimizing $\Phi_K(\mathcal{X})$ can be found efficiently.

$\Rightarrow$ Suggested algo leads to cost $\leq \ln(4)\,\text{cost}(LP)$.

Some insights how to exploit matroid structure

(b) $\exists\, C \in \mathcal{K}$ and basis B of M_C, $B \subseteq K$ s.t. $\text{cost}(C) \leq w(B)$.

We prove **(b)** via averaging argument by showing:

Lemma

$\exists\, B_C \subseteq K$ basis of M_C $\forall C \in \Gamma(\mathcal{X})$ s.t.:

$$N \cdot w(K) = \sum_{C \in \Gamma(\mathcal{X})} \text{cost}(C) \leq \sum_{C \in \Gamma(\mathcal{X})} w(B_c)$$

High-level proof outline.

(i) We prove $N \cdot \mathbf{1} \in \sum_{C \in \Gamma(\mathcal{X})} P_{M_C}$,

- $\mathbf{1}$ is all-one vector in $\{0,1\}^K$,
- P_{M_C} is matroid polytope of M_C.

(ii) Key insight: $\sum_{C \in \Gamma(\mathcal{X})} P_{M_C}$ is polymatroid with rank funct. $\sum_{C \in \Gamma(\mathcal{X})} r_{M_C}$.

(iii) We prove **(i)** by studying matroids M_C to obtain properties on r_{M_C}.

Conclusions

- Hypergraphic LP has rich combinatorial structures linked to matroids and submodular functions.
- By leveraging them many interesting results can be obtained.

Conclusions

- Hypergraphic LP has rich combinatorial structures linked to matroids and submodular functions.
- By leveraging them many interesting results can be obtained.

Open problems

- Faster methods to solve hypergraphic LP?
- Better understanding of simpler, bidirected cut relaxation?
- Use new insights to design stronger approximation algorithms?

Conclusions

- Hypergraphic LP has rich combinatorial structures linked to matroids and submodular functions.
- By leveraging them many interesting results can be obtained.

Open problems

- Faster methods to solve hypergraphic LP?
- Better understanding of simpler, bidirected cut relaxation?
- Use new insights to design stronger approximation algorithms?

Thank you!

References I

Borchers, A. and Du, D.-Z. (1997).
The k-Steiner ratio in graphs.
SIAM Journal on Computing, 26(3):857–869.

Byrka, J., Grandoni, F., Rothvoß, T., and Sanità, L. (2010).
In improved LP-based approximation for Steiner Tree.
In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 583–592.