Matroidal Degree-Bounded Minimum Spanning Trees

Rico Zenklusen

MIT

Degree-constrained MST problems

$\min\{c(T) \mid T \text{ spanning tree in } G = (V, E), \text{ satisfying degree constraints}\}$

Degree-constrained MST problems

$$\min\{c(T) \mid T \text{ spanning tree in } G = (V, E), \text{ satisfying degree constraints}\}$$

Types of degree constraints

- Degree-bounded MST (the classic):

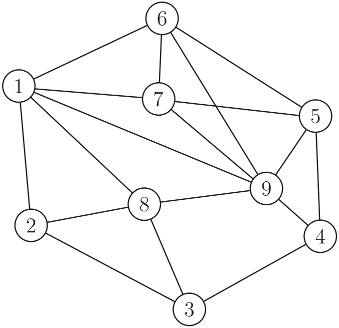
  $$|T \cap \delta(v)| \leq B_v \ \forall v \in V.$$

- Cut-constraints:

  $$|T \cap \delta(S)| \leq B_S \text{ for } S \in \mathcal{S} \subseteq 2^V.$$

- 0/1-packing constraints:

  $$|T \cap U| \leq B_U \text{ for } U \in \mathcal{U} \subseteq 2^E.$$



Degree-constrained MST problems

$\min\{c(T) \mid T \text{ spanning tree in } G = (V, E), \text{ satisfying degree constraints}\}$

Types of degree constraints

- Degree-bounded MST (the classic):

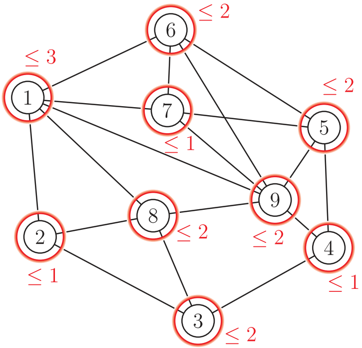
  $$|T \cap \delta(v)| \leq B_v \ \forall v \in V.$$

- Cut-constraints:

  $$|T \cap \delta(S)| \leq B_S \text{ for } S \in \mathcal{S} \subseteq 2^V.$$

- 0/1-packing constraints:

  $$|T \cap U| \leq B_U \text{ for } U \in \mathcal{U} \subseteq 2^E.$$



Degree-constrained MST problems

$\min\{c(T) \mid T$ spanning tree in $G = (V, E)$, satisfying degree constraints$\}$

Types of degree constraints

- Degree-bounded MST (the classic):

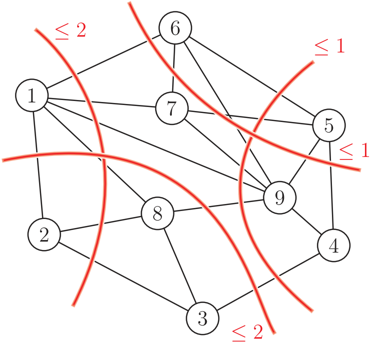
  $$|T \cap \delta(v)| \leq B_v \ \forall v \in V.$$

- Cut-constraints:

  $$|T \cap \delta(S)| \leq B_S \text{ for } S \in \mathcal{S} \subseteq 2^V.$$

- 0/1-packing constraints:

  $$|T \cap U| \leq B_U \text{ for } U \in \mathcal{U} \subseteq 2^E.$$



Degree-constrained MST problems

$\min\{c(T) \mid T$ spanning tree in $G = (V, E)$, satisfying degree constraints$\}$

Types of degree constraints

- Degree-bounded MST (the classic):

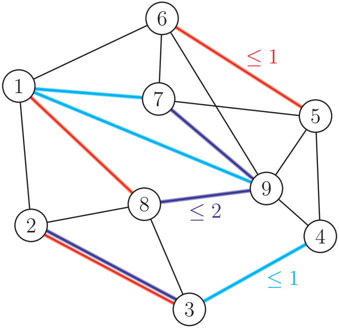
  $$|T \cap \delta(v)| \leq B_v \ \forall v \in V.$$

- Cut-constraints:

  $$|T \cap \delta(S)| \leq B_S \text{ for } S \in \mathcal{S} \subseteq 2^V.$$

- 0/1-packing constraints:

  $$|T \cap U| \leq B_U \text{ for } U \in \mathcal{U} \subseteq 2^E.$$



Degree-constrained MST problems

$\min\{c(T) \mid T \text{ spanning tree in } G = (V, E), \text{ satisfying degree constraints}\}$

Types of degree constraints

- Degree-bounded MST (the classic):

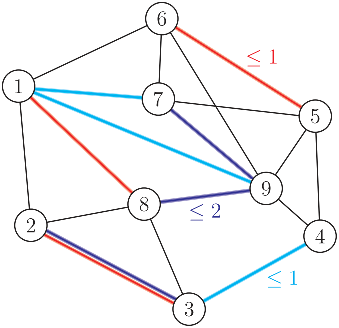
$$|T \cap \delta(v)| \leq B_v \ \forall v \in V.$$

- Cut-constraints:

$$|T \cap \delta(S)| \leq B_S \text{ for } S \in \mathcal{S} \subseteq 2^V.$$

- 0/1-packing constraints:

$$|T \cap U| \leq B_U \text{ for } U \in \mathcal{U} \subseteq 2^E.$$



Typical motivations for degree constraints

- Technical restrictions (VLSI design, telecommunication networks).
- Increase reliability by avoiding overloaded vertices.
- Reduce vulnerability against malicious attacks.

Trading slight infeasibility for "optimal" cost

At the example of degree-bounded MST

$$OPT = \min\{c(T) \mid T \in \underbrace{\mathcal{T}}_{\boxed{\text{all spanning trees } (\subseteq 2^E)}}, |T \cap \delta(v)| \le B_v \ \forall v \in V\}$$

- Even checking feasibility is NP hard
  (e.g. if $B_v = 2 \ \forall v \in V \rightarrow$ Hamiltonian path problem.)

Trading slight infeasibility for "optimal" cost

At the example of degree-bounded MST

$$OPT = \min\{c(T) \mid T \in \underbrace{\mathcal{T}}_{\boxed{\text{all spanning trees } (\subseteq 2^E)}}, |T \cap \delta(v)| \leq B_v \ \forall v \in V\}$$

- Even checking feasibility is NP hard
  (e.g. if $B_v = 2 \ \forall v \in V \to$ Hamiltonian path problem.)

Goal

Find tree T of cost $\leq$ LP relaxation ($c(T) \leq OPT_f$), minimizing deg violation.

Trading slight infeasibility for "optimal" cost

At the example of degree-bounded MST

$$OPT = \min\{c(T) \mid T \in \underbrace{\mathcal{T}}_{\text{all spanning trees } (\subseteq 2^E)}, |T \cap \delta(v)| \leq B_v \ \forall v \in V\}$$

- Even checking feasibility is NP hard
  (e.g. if $B_v = 2 \ \forall v \in V \rightarrow$ Hamiltonian path problem.)

Goal

Find tree T of cost $\leq$ LP relaxation ($c(T) \leq OPT_f$), minimizing deg violation.

- $OPT_f = \min c^T x$
  $$x \in P_{ST} = \text{conv}(\{\mathbf{1}_T \mid T \in \mathcal{T}\})$$
  $$x(\delta(v)) \leq B_v \quad \forall v \in V$$
  $$\cancel{x \in \{0,1\}^E}$$
- $\min\{c(T) \mid T \in \mathcal{T}, |T \cap \delta(v)| \leq B_v + k \ \forall v \in V\}$

Previous results

Degree-bounded MST

• $+1$ violation (no costs)	Fürer and Raghavachari (1994)
• various super-constant violations with cost $\leq OPT_f$	Könemann and Ravi (2002, 2003), Chaudhuri et al. (2005), ...
• $+2$	Goemans (2006)
• $+1$	Singh and Lau (2007)

Generalized bounds

Laminar cut bounds:

- $+O(\log |V|)$ — Bansal et al. (2010)

Bounds on arbitrary edge sets $x(U) \leq B_U \; \forall U \in \mathcal{U} \subseteq 2^E$:

- $+\max_{e \in E} |\{U \in \mathcal{U} \mid e \in U\}|$ — Bansal et al. (2009)

Thinness bounds wrt $y \in P_{ST}$: $x(\delta(S)) \leq y(\delta(S)) \;\; \forall \emptyset \neq S \subsetneq V$ (*thin trees*):

- $\times O\left(\frac{\log |V|}{\log \log |V|}\right)$ — Asadpour et al. (2010)

Previous results

Degree-bounded MST

- $+1$ violation (no costs)	Fürer and Raghavachari (1994)
- various super-constant violations with cost $\leq OPT_f$	Könemann and Ravi (2002, 2003), Chaudhuri et al. (2005), ...
- $+2$	Goemans (2006)
- $+1$	Singh and Lau (2007)

Generalized bounds

Laminar cut bounds:

- $+O(\log |V|)$ — Bansal et al. (2010)

Bounds on arbitrary edge sets $x(U) \leq B_U \; \forall U \in \mathcal{U} \subseteq 2^E$:

- $+\max_{e \in E} |\{U \in \mathcal{U} \mid e \in U\}|$ — Bansal et al. (2009)

Thinness bounds wrt $y \in P_{ST}$: $x(\delta(S)) \leq y(\delta(S)) \;\; \forall \emptyset \neq S \subsetneq V$ (*thin trees*):

- $\times O\left(\frac{\log |V|}{\log \log |V|}\right)$ — Asadpour et al. (2010)

Obtain constant violation for constraints beyond degree-bounded MST?

Our contributions

Main results

- Efficent algorithm with $+8$ guarantee for matroidal degree constraints.
- This is based on extensions to the iterative relaxation framework.

Our contributions

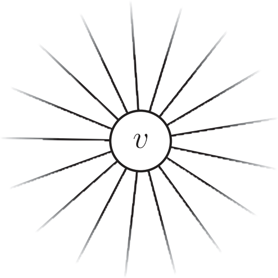
Main results

- Efficent algorithm with $+8$ guarantee for matroidal degree constraints.
- This is based on extensions to the iterative relaxation framework.

Definition (Matroidal degree constraints)

$$\min\{c(T) \mid T \in \mathcal{T}, T \cap \delta(v) \in \mathcal{I}_v \;\; \forall v \in V\},$$

where $\mathcal{M}_v = (\delta(v), \mathcal{I}_v)$ is a matroid $\forall v \in V$.



Our contributions

Main results

- Efficent algorithm with $+8$ guarantee for matroidal degree constraints.
- This is based on extensions to the iterative relaxation framework.

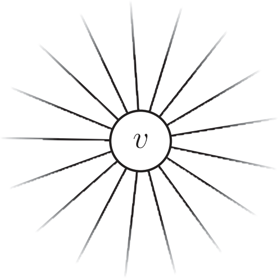
Definition (Matroidal degree constraints)

$$\min\{c(T) \mid T \in \mathcal{T}, T \cap \delta(v) \in \mathcal{I}_v \ \ \forall v \in V\},$$

where $\mathcal{M}_v = (\delta(v), \mathcal{I}_v)$ is a matroid $\forall v \in V$.

Examples

- deg-bounded MST ($\mathcal{M}_v$: uniform matroids)
- partition constraints (part. matroids)
- laminar constraints (lam. matroids)



Our contributions

Main results

- Efficent algorithm with $+8$ guarantee for matroidal degree constraints.
- This is based on extensions to the iterative relaxation framework.

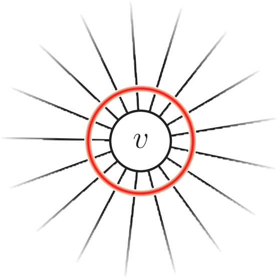
Definition (Matroidal degree constraints)

$$\min\{c(T) \mid T \in \mathcal{T}, T \cap \delta(v) \in \mathcal{I}_v \;\; \forall v \in V\},$$

where $\mathcal{M}_v = (\delta(v), \mathcal{I}_v)$ is a matroid $\forall v \in V$.

Examples

- deg-bounded MST ($\mathcal{M}_v$: uniform matroids)
- partition constraints (part. matroids)
- laminar constraints (lam. matroids)



Our contributions

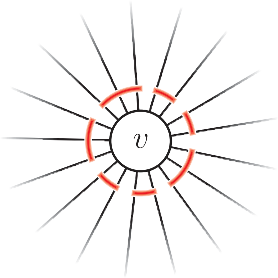
Main results

- Efficent algorithm with $+8$ guarantee for matroidal degree constraints.
- This is based on extensions to the iterative relaxation framework.

Definition (Matroidal degree constraints)

$$\min\{c(T) \mid T \in \mathcal{T}, T \cap \delta(v) \in \mathcal{I}_v \;\; \forall v \in V\},$$

where $\mathcal{M}_v = (\delta(v), \mathcal{I}_v)$ is a matroid $\forall v \in V$.



Examples

- deg-bounded MST ($\mathcal{M}_v$: uniform matroids)
- partition constraints (part. matroids)
- laminar constraints (lam. matroids)

Our contributions

Main results

- Efficent algorithm with $+8$ guarantee for matroidal degree constraints.
- This is based on extensions to the iterative relaxation framework.

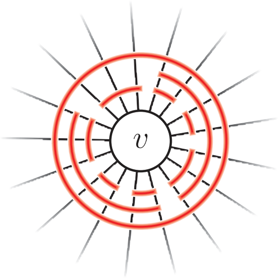
Definition (Matroidal degree constraints)

$$\min\{c(T) \mid T \in \mathcal{T}, T \cap \delta(v) \in \mathcal{I}_v \ \ \forall v \in V\},$$

where $\mathcal{M}_v = (\delta(v), \mathcal{I}_v)$ is a matroid $\forall v \in V$.

Examples

- deg-bounded MST ($\mathcal{M}_v$: uniform matroids)
- partition constraints (part. matroids)
- laminar constraints (lam. matroids)



Our contributions

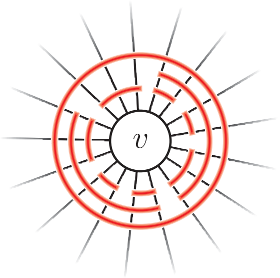
Main results

- Efficent algorithm with $+8$ guarantee for matroidal degree constraints.
- This is based on extensions to the iterative relaxation framework.

Definition (Matroidal degree constraints)

$$\min\{c(T) \mid T \in \mathcal{T}, T \cap \delta(v) \in \mathcal{I}_v \;\; \forall v \in V\},$$

where $\mathcal{M}_v = (\delta(v), \mathcal{I}_v)$ is a matroid $\forall v \in V$.



Examples

- deg-bounded MST ($\mathcal{M}_v$: uniform matroids)
- partition constraints (part. matroids)
- laminar constraints (lam. matroids)

Definition ($\leq k$ violation of degree constraints)

$T \in \mathcal{T}$ violates degree constraint at v by $\leq k$ units if:
feasibility can be obtained by ignoring the contribution of $\leq k$ edges.

Our contributions

Main results

- Efficent algorithm with $+8$ guarantee for matroidal degree constraints.
- This is based on extensions to the iterative relaxation framework.

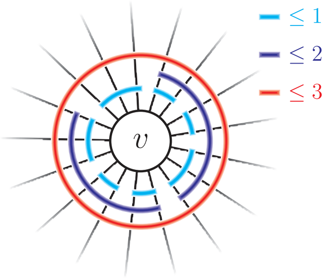
Definition (Matroidal degree constraints)

$$\min\{c(T) \mid T \in \mathcal{T}, T \cap \delta(v) \in \mathcal{I}_v \ \ \forall v \in V\},$$

where $\mathcal{M}_v = (\delta(v), \mathcal{I}_v)$ is a matroid $\forall v \in V$.

Examples

- deg-bounded MST ($\mathcal{M}_v$: uniform matroids)
- partition constraints (part. matroids)
- laminar constraints (lam. matroids)



Definition ($\leq k$ violation of degree constraints)

$T \in \mathcal{T}$ violates degree constraint at v by $\leq k$ units if:
feasibility can be obtained by ignoring the contribution of $\leq k$ edges.

Our contributions

Main results

- Efficent algorithm with $+8$ guarantee for matroidal degree constraints.
- This is based on extensions to the iterative relaxation framework.

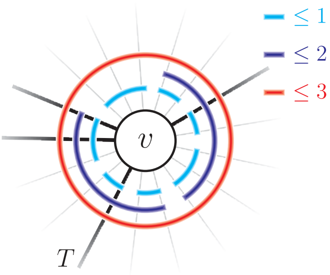
Definition (Matroidal degree constraints)

$$\min\{c(T) \mid T \in \mathcal{T}, T \cap \delta(v) \in \mathcal{I}_v \ \ \forall v \in V\},$$

where $\mathcal{M}_v = (\delta(v), \mathcal{I}_v)$ is a matroid $\forall v \in V$.

Examples

- deg-bounded MST ($\mathcal{M}_v$: uniform matroids)
- partition constraints (part. matroids)
- laminar constraints (lam. matroids)



Definition ($\leq k$ violation of degree constraints)

$T \in \mathcal{T}$ violates degree constraint at v by $\leq k$ units if:
feasibility can be obtained by ignoring the contribution of $\leq k$ edges.

Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

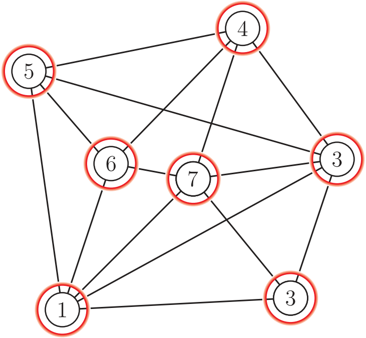
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

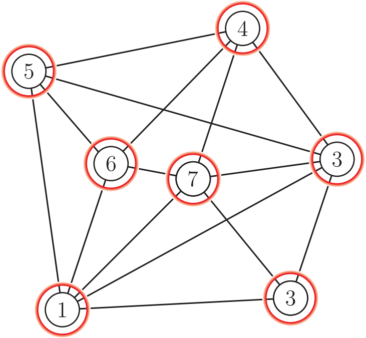
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

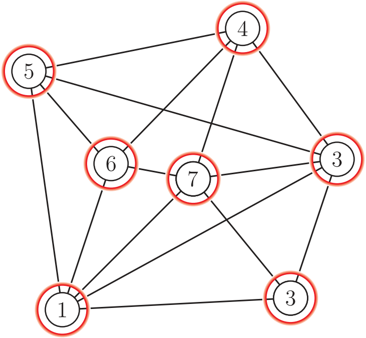
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

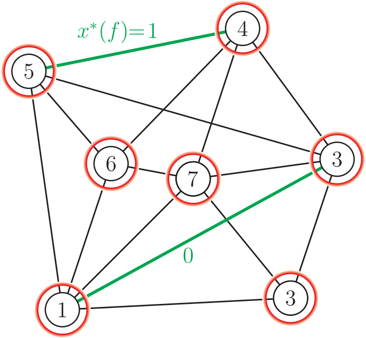
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.

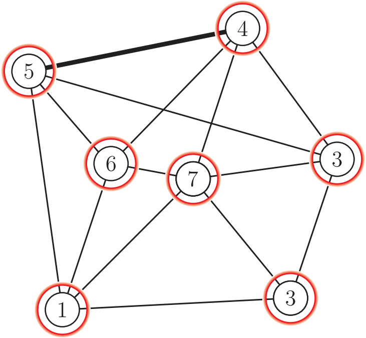


Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2
- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)
- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

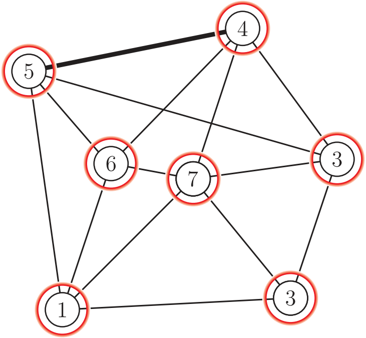
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

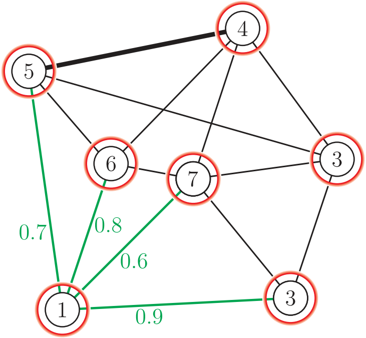
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

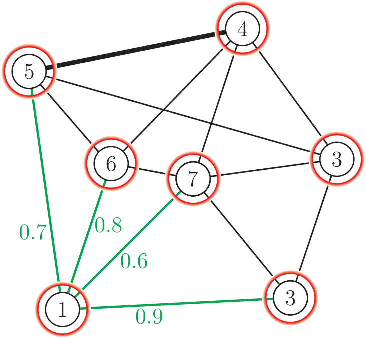
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



$z(\delta(1)) = 0.3 + 0.2 + 0.4 + 0.1 = 1 < 2$

Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

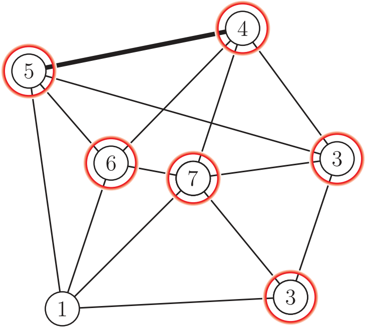
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

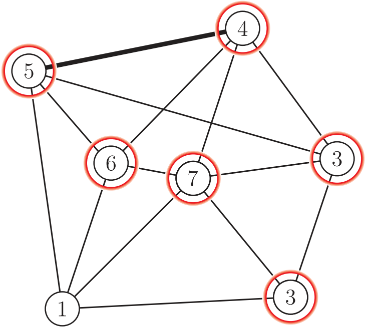
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

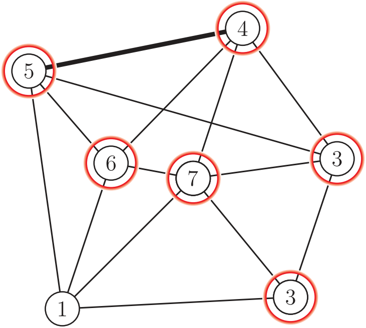
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

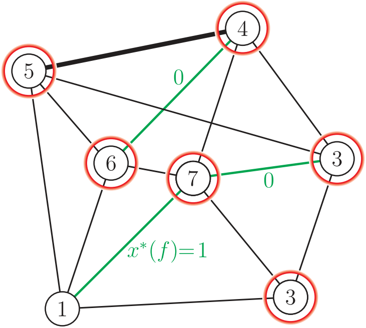
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

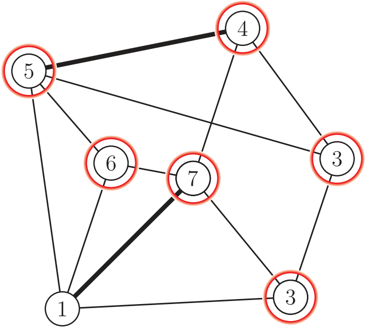
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

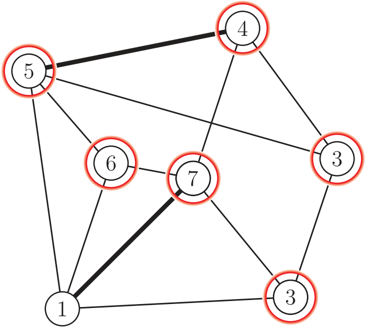
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

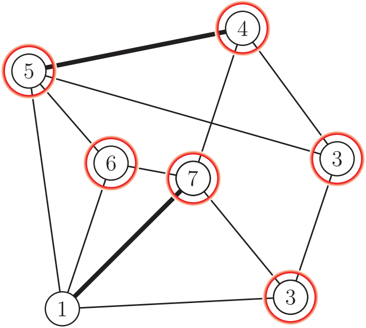
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Key property: x^* is sparse (in particular $|\text{supp}(x^*)| \leq 2|V| - 1$).

$$\begin{pmatrix} A_{ST} \\ \hdashline A_{deg} \end{pmatrix} x^* \leq \begin{pmatrix} b_{ST} \\ \hdashline b_{deg} \end{pmatrix}$$

$$x^* \geq 0$$

Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

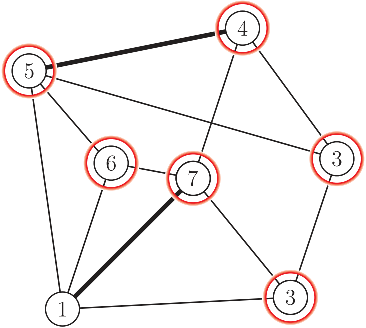
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Key property: x^* is sparse (in particular $|\mathrm{supp}(x^*)| \leq 2|V| - 1$).

$$\begin{pmatrix} A_{ST} \\ \hdashline A_{deg} \\ \hdashline -I \end{pmatrix} x^* \leq \begin{pmatrix} b_{ST} \\ \hdashline b_{deg} \\ \hdashline 0 \end{pmatrix}$$

Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

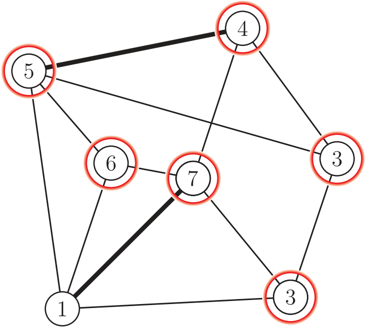
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Key property: x^* is sparse (in particular $|\mathrm{supp}(x^*)| \leq 2|V| - 1$).

$$\begin{pmatrix} A_{ST} \\ A_{deg} \\ -I \end{pmatrix} x^* \overset{=}{\leq} \begin{pmatrix} b_{ST} \\ b_{deg} \\ 0 \end{pmatrix}$$

Iterative relaxation by Singh and Lau (+1 guarantee)

Step 1

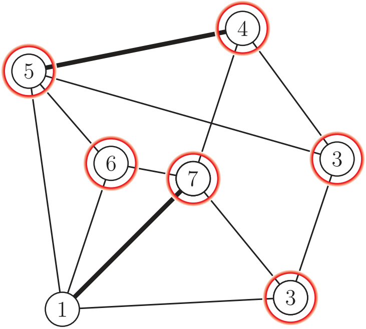
- Get basic LP solution x^*.
- If sol is integral: stop.

Step 2

- Delete 0-edges.
- Fix 1-edges.

Step 3 (let *spare*: $z = 1 - x^*$)

- Find $v \in V$ s.t. $z(\delta(v)) < 2$.
- Delete deg constraint at v.
- Back to Step 1.



Key property: x^* is sparse (in particular $|\text{supp}(x^*)| \leq 2|V| - 1$).

$$\begin{pmatrix} A_{ST} \\ \hline A_{deg} \\ \hline -I \end{pmatrix} x^* \overset{=}{\leq} \begin{pmatrix} b_{ST} \\ \hline b_{deg} \\ \hline 0 \end{pmatrix} \begin{matrix} \} \ \# \leq |V| - 1 \\ \} \ \# \leq |V| \\ \} \ \# \geq |E| - (2|V| - 1) \end{matrix}$$

Challenges with more general/matroidal constraints

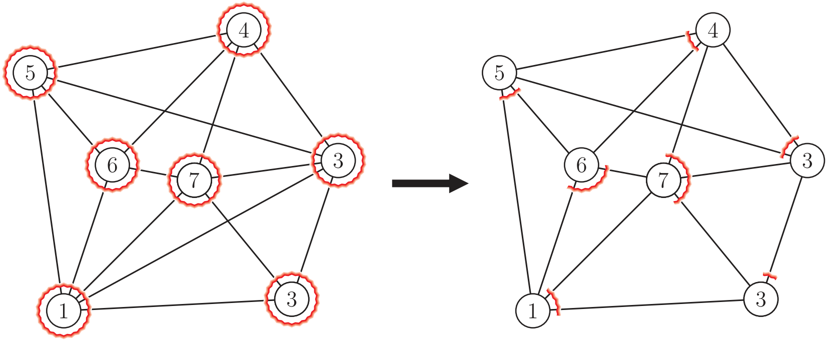
Previous iterative relaxation/rounding approaches are hard to generalize to matroidal deg constraints (or other generalized constraint).

Some issues with previous iterative relaxation approaches

- Not sufficient sparsity to drop full degree constraints at some vertex.
- Previous approaches relied on the fact that each edge is only in a constant number of linear constraints (belonging to degree constraints).

High-level goal of our algorithm

Iteratively change constraints to approach matroid intersection problem instead of targeting ST polytope (which is a matroid base polytope).
$\rightarrow$ Iteratively “remove” each edge $\{u, v\}$ either from deg constraint at u or v.
(this is similar in spirit to Goemans’ algorithm, but works iteratively.)



- If each edge belong to at most one degree constraints, all matroidal degree constraints together form one single matroid.
- Resulting optimization problem is matroid intersection and thus integral.

Summary of further technical contributions

Further contributions on algorithm design level

- When removing edges from constraints: old constraint gets replaced by a possibly more complicated matroidal constraint (s.t. violation is bounded by slack).
- We fix tight ST constraints $\rightarrow$ they help respect degree constraints.

New ideas for the analysis

- New argument to prove sparsity that exploits interplay of constraints.
- Exploit properties of low-dimensional faces of ST polytope (to deal with cases where many lin indep ST constraints are tight).

An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

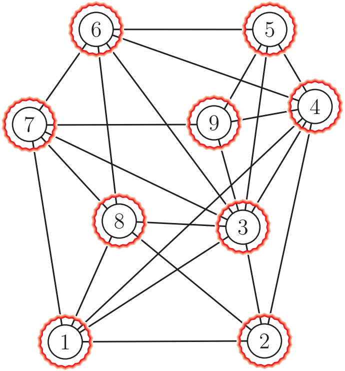
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

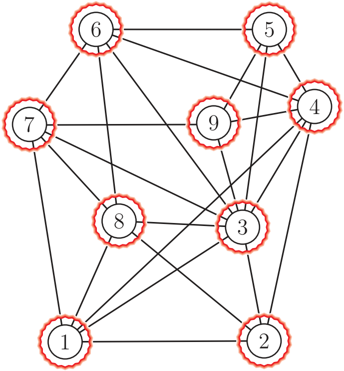
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

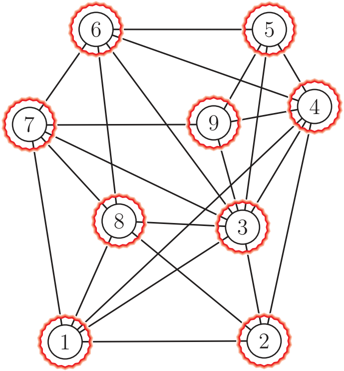
- Get basic solution x^* to LP: $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

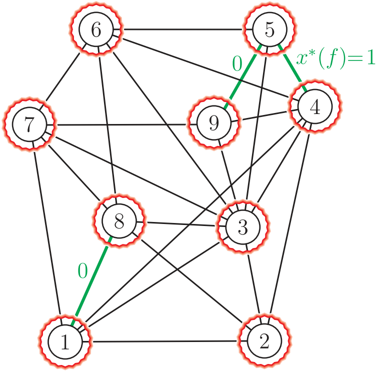
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

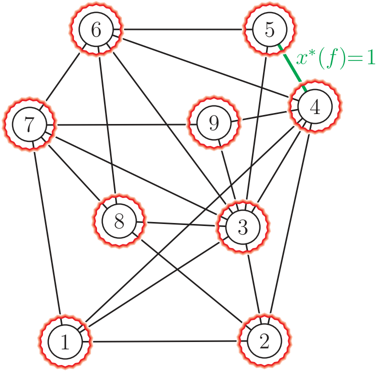
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

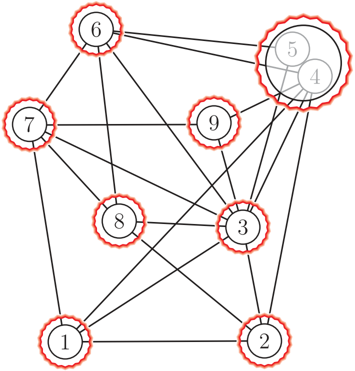
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

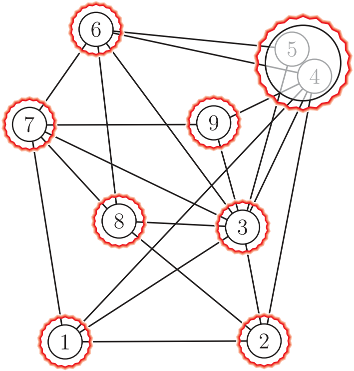
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

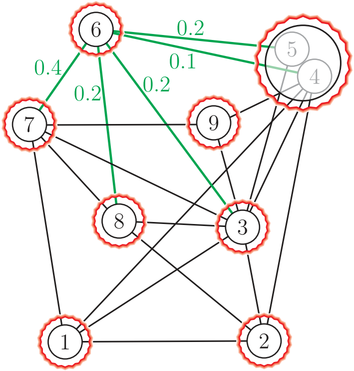
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

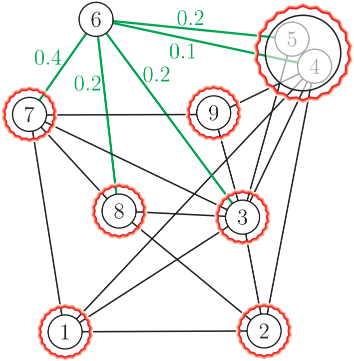
- Get basic solution x^* to LP: $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

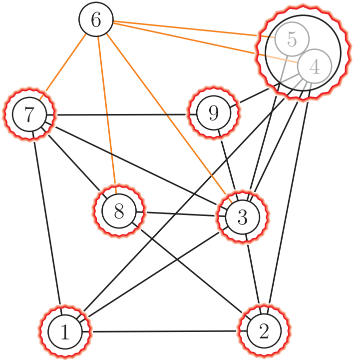
- Get basic solution x^* to LP: $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

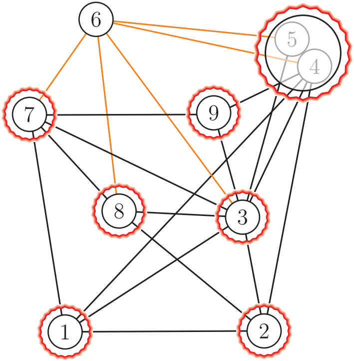
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

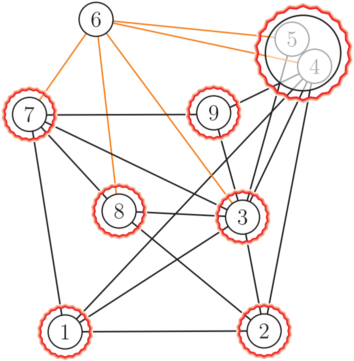
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

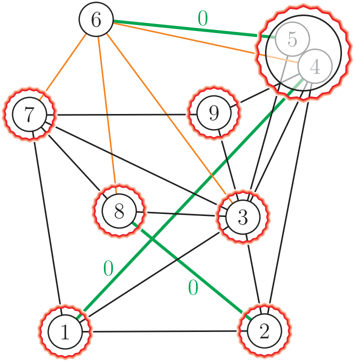
- Get basic solution x^* to LP: $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

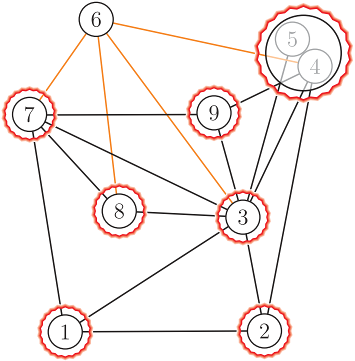
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

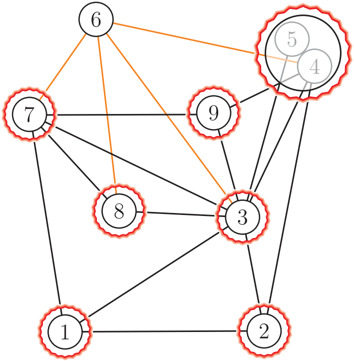
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

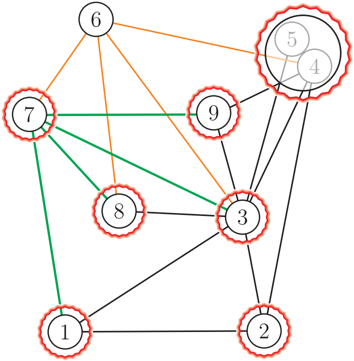
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

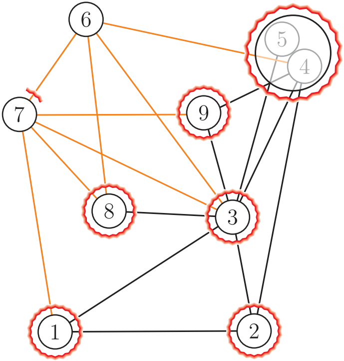
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

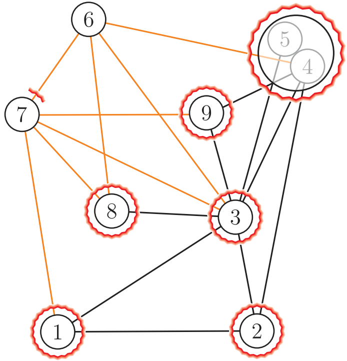
- Get basic solution x^* to LP: $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

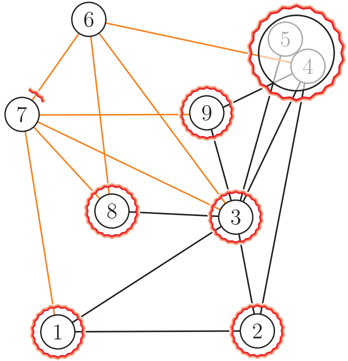
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

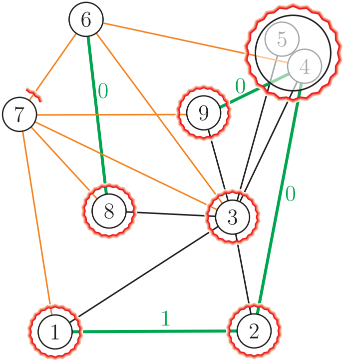
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

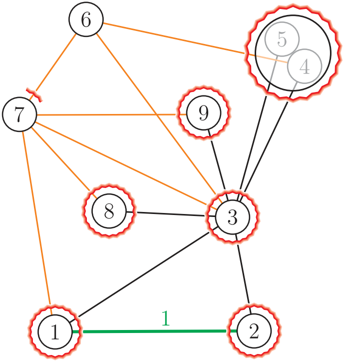
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

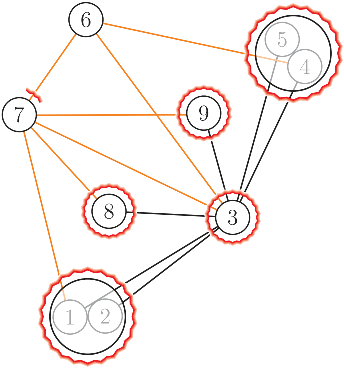
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

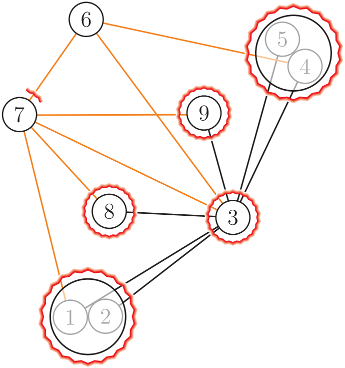
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

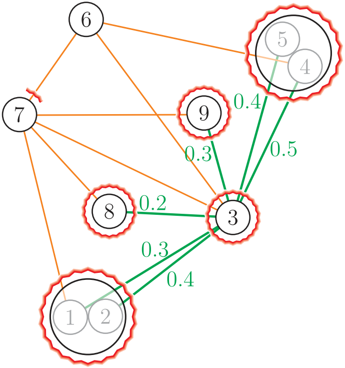
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

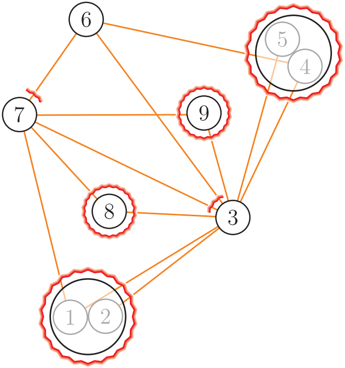
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

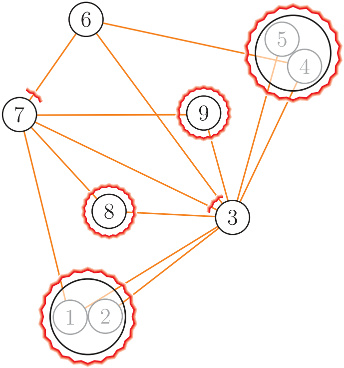
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

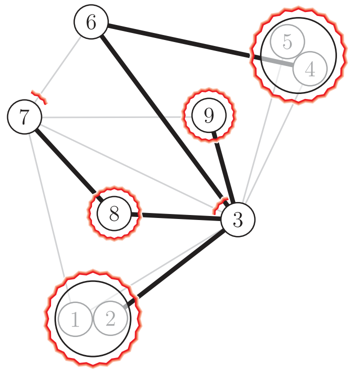
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

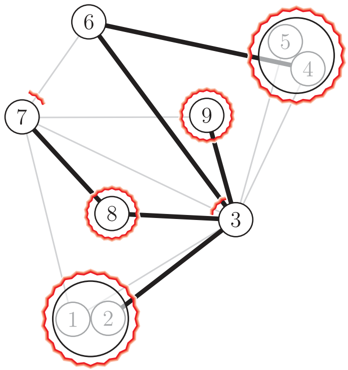
- Get basic solution x^* to LP:
  $\min\{c^T x \mid x \in P_{ST}, x\big|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



An oversimplified sketch of the algorithm

- $H = (W, F)$: current graph, $F_2 \subseteq F$: edges currently in both constraints.
- N_w: current matroidal deg constraints with corresp. matroid polytope P_{N_w}.

Step 1

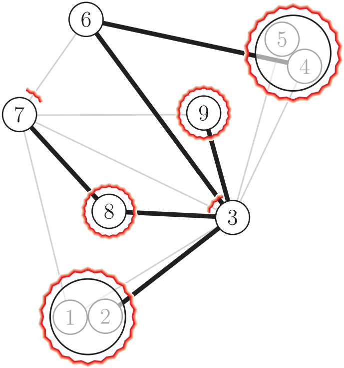
- Get basic solution x^* to LP: $\min\{c^T x \mid x \in P_{ST}, x|_{\delta(w)} \in P_{N_w} \forall w \in W\}$
- If x^* is integral: stop.

Step 2

- Delete 0-edges.
- Contract 1-edges.
- Fix tight spanning tree constraints.

Step 3: degree constraint adaptation

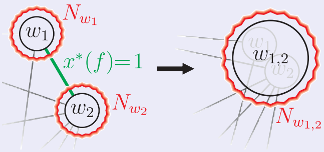
- Find $w \in W$ s.t. $z(\delta(w) \cap F_2) \leq 4$.
- "Remove" $\delta(w) \cap F_2$ from constraint at w.
- Go back to Step 1.



For $v \in V$, only ≤ 1 deg adaptations impacts $\delta(v) \Rightarrow$ violation ≤ 4.

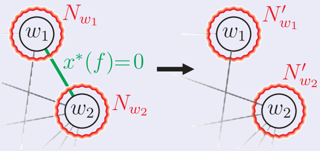
Updating degree constraints

Contraction of 1-edge



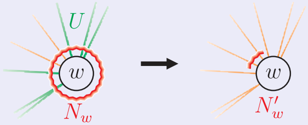
- Contract f in N_{w_1} and N_{w_2}.
- $N_{w_{1,2}}$ is disjoint union of N_{w_1} and N_{w_2}.

Deletion of 0-edge



- Delete f from N_{w_1} and N_{w_2}.

Removing edges from deg constr. (a bit more involved)



Update is done such that:

i) $T \in \mathcal{T}$ satisfies $N'_w \Rightarrow T$ violates N_w by $\leq \lceil z(U) \rceil$.

ii) Current LP sol remains feasible.

Proving sparsity to show $\exists$ adaptation step

- $\delta^D(w) \subseteq \delta(w)$: edges not yet removed from N_w.

Lemma

If k linearly indep degree constraints of P_{N_w} are tight wrt x^* $\Rightarrow$ $x^*(\delta^D(w)) \geq k$.

Proving sparsity to show $\exists$ adaptation step

- $\delta^D(w) \subseteq \delta(w)$: edges not yet removed from N_w.

Lemma

If k linearly indep degree constraints of P_{N_w} are tight wrt x^* $\Rightarrow x^*(\delta^D(w)) \geq k$.

- Total # of linearly indep tight deg constraints at start of algo.:
  $|\mathcal{D}| \leq \sum_{w \in W} x^*(\delta^D(w)) = \sum_{w \in W} x^*(\delta(w)) = 2x^*(F) \overset{x^* \in P_{ST}}{=} 2(|W| - 1)$.
- Total # of linearly indep & tight ST constraints: $|\mathcal{L}| \overset{\text{uncrossing}}{\leq} |W| - 1$.
- $|\operatorname{supp}(x^*)| = |F| \leq |\mathcal{L}| + |\mathcal{D}| \leq 3(|W| - 1)$.

Proving sparsity to show $\exists$ adaptation step

- $\delta^D(w) \subseteq \delta(w)$: edges not yet removed from N_w.

Lemma

If k linearly indep degree constraints of P_{N_w} are tight wrt x^* $\Rightarrow x^*(\delta^D(w)) \geq k$.

- Total # of linearly indep tight deg constraints at start of algo.: $|\mathcal{D}| \leq \sum_{w\in W} x^*(\delta^D(w)) = \sum_{w\in W} x^*(\delta(w)) = 2x^*(F) \overset{x^*\in P_{ST}}{=} 2(|W|-1)$.
- Total # of linearly indep & tight ST constraints: $|\mathcal{L}| \overset{\text{uncrossing}}{\leq} |W|-1$.
- $|\,\text{supp}(x^*)| = |F| \leq |\mathcal{L}| + |\mathcal{D}| \leq 3(|W|-1)$.

$\exists$ adaptation step in first iteration

$$\sum_{w\in W} z(\delta^D(w)) = \sum_{w\in W} z(\delta(w)) = \sum_{w\in W} (|\delta(w)| - x^*(\delta(w)))$$
$$\leq 2|F| - \sum_{w\in W} x^*(\delta(w)) = 2|F| - 2x^*(F) \leq 4(|W|-1).$$

Proving sparsity to show $\exists$ adaptation step

- $\delta^D(w) \subseteq \delta(w)$: edges not yet removed from N_w.

Lemma

If k linearly indep degree constraints of P_{N_w} are tight wrt x^* $\Rightarrow$ $x^*(\delta^D(w)) \geq k$.

- Total # of linearly indep tight deg constraints at start of algo.:
  $|\mathcal{D}| \leq \sum_{w \in W} x^*(\delta^D(w)) = \sum_{w \in W} x^*(\delta(w)) = 2x^*(F) \overset{x^* \in P_{ST}}{=} 2(|W|-1)$.
- Total # of linearly indep & tight ST constraints: $|\mathcal{L}| \overset{\text{uncrossing}}{\leq} |W|-1$.
- $|\operatorname{supp}(x^*)| = |F| \leq |\mathcal{L}| + |\mathcal{D}| \leq 3(|W|-1)$.

$\exists$ adaptation step in first iteration

$$\sum_{w \in W} z(\delta^D(w)) = \sum_{w \in W} z(\delta(w)) = \sum_{w \in W} (|\delta(w)| - x^*(\delta(w)))$$
$$\leq 2|F| - \sum_{w \in W} x^*(\delta(w)) = 2|F| - 2x^*(F) \leq 4(|W|-1).$$

In later iterations, this averaging argument does not work anymore

- For some nodes $w \in W$ we will have $\delta^D(w) = \emptyset$.
- We improve sparsity with 2nd type of deg adaptation ($\rightarrow$ another $+4$ in violation).

Conclusions

- Even for very general degree constraints (matroidal degree constraints), a tree of cost $\leq OPT$ can be obtained with a constant additive degree violation.
- Targeting matroid intersection instead of single matroid seems like an interesting plan in iterative relaxation framework.

- Extensions to other problems?
- Obtaining an additive violation < 8?
- Constant multiplicative errors for special families of cut constraints?
- Constant-thin spanning trees (implies constant factor approx for ATSP)?

Conclusions

- Even for very general degree constraints (matroidal degree constraints), a tree of cost $\leq OPT$ can be obtained with a constant additive degree violation.
- Targeting matroid intersection instead of single matroid seems like an interesting plan in iterative relaxation framework.

- Extensions to other problems?
- Obtaining an additive violation < 8?
- Constant multiplicative errors for special families of cut constraints?
- Constant-thin spanning trees (implies constant factor approx for ATSP)?

Thank you!

References I

Asadpour, A., Goemans, M. X., Madry, A., Oveis Gharan, S., and Saberi, A. (2010). An $O(\log n / \log \log n)$-approximation algrithm for the asymmetric traveling salesman problem. In *Proceedings of the 20th Annual ACM -SIAM Symposium on Discrete Algorithms (SODA)*.

Bansal, N., Khandekar, R., Könemann, J., Nagarajan, V., and Peis, B. (2010). On generalizations of network design problems with degree bounds. In *Proceedings of Integer Programming and Combinatorial Optimization (IPCO)*, pages 110–123.

Bansal, N., Khandekar, R., and Nagarajan, V. (2009). Additive guarantees for degree-bounded directed network design. *SIAM Journal on Computing*, 39(4):1413–1431.

Chaudhuri, K., Rao, S., Riesenfeld, S., and Talwar, K. (2005). What would edmonds do? augmenting paths and witnesses for degree-bounded MSTs. In Chekuri, C., Jansen, K., Rolim, J. D. P., and Trevisan, L., editors, *Approximation, Randomization and Combinatorial Optimization*, volume 3624 of *Lecture Notes in Computer Science*, pages 26–39. Springer Berlin / Heidelberg. 10.1007/11538462_3.

Fürer, M. and Raghavachari, B. (1994). Approximating the minimum-degree Steiner Tree to within one of optimal. *Journal of Algorithms*, 17(3):409–423.

References II

Goemans, M. X. (2006). Minimum bounded degree spanning trees. In *Proceedings of the 47th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 273–282.

Könemann, J. and Ravi, R. (2002). A matter of degree: Improved approximation algorithms for degree-bounded minimum spanning trees. *SIAM Journal on Computing*, 31:1783–1793.

Könemann, J. and Ravi, R. (2003). Primal-dual meets local search: approximating MST's with nonuniform degree bounds. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 389–395.

Singh, M. and Lau, L. C. (2007). Approximating minimum bounded degree spanning trees to within one of optimal. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 661–670.