Dependent Randomized Rounding on the Spanning Tree Polytope with Applications in Multi-Objective Optimization

Rico Zenklusen

Institute for Discrete Optimization, EPFL

Joint work with Chandra Chekuri and Jan Vondrák

Outline

1. **A short introduction to dependent rounding**
   - Motivation
   - Classical rounding approaches: a review by examples
   - Typical goals when designing rounding procedures

2. **Randomized swap rounding: rounding on matroid polytopes**
   - Motivating example
   - Randomized swap rounding on the spanning tree polytope
   - Example applications
   - Some short remarks about submodular functions

3. **Conclusion**

Outline

1. **A short introduction to dependent rounding**
   - Motivation
   - Classical rounding approaches: a review by examples
   - Typical goals when designing rounding procedures

2. **Randomized swap rounding: rounding on matroid polytopes**
   - Motivating example
   - Randomized swap rounding on the spanning tree polytope
   - Example applications
   - Some short remarks about submodular functions

3. **Conclusion**

Motivation

Many combinatorial problems become easy when relaxing integrality constraints.

How to profit from fractional solutions?

→ One promising approach: Rounding.

Useful problem properties for rounding procedures

- Small integrality gap
- "Weak constraints"
- Fractional solution has
  - few fractional values
  - fractional components generally have high values

→ Still for most problems it is not clear whether and how rounding procedures can be applied.

Motivation

Many combinatorial problems become easy when relaxing integrality constraints.

How to profit from fractional solutions?

→ One promising approach: Rounding.

Useful problem properties for rounding procedures

- Small integrality gap
- "Weak constraints"
- Fractional solution has
  - few fractional values
  - fractional components generally have high values

→ Still for most problems it is not clear whether and how rounding procedures can be applied.

Motivation

Many combinatorial problems become easy when relaxing integrality constraints.

How to profit from fractional solutions?

$\rightarrow$ One promising approach: Rounding.

Useful problem properties for rounding procedures

- Small integrality gap
- "Weak constraints"
- Fractional solution has
  - few fractional values
  - fractional components generally have high values

$\rightarrow$ Still for most problems it is not clear whether and how rounding procedures can be applied.

Motivation

Many combinatorial problems become easy when relaxing integrality constraints.

How to profit from fractional solutions?

$\rightarrow$ One promising approach: Rounding.

Useful problem properties for rounding procedures

- Small integrality gap
- "Weak constraints"
- Fractional solution has
  - few fractional values
  - fractional components generally have high values

$\rightarrow$ Still for most problems it is not clear whether and how rounding procedures can be applied.

An example for deterministic rounding

(Unweighted) Set Cover with bounded coverage

Given:
- A finite set U, collection of subsets $\mathcal{F} = \{S_1, \ldots, S_p\} \subseteq 2^U$.
- $|\{i \in [p] := \{1, \ldots, p\} \mid u \in S_i\}| \leq \ell$ for all $u \in U$.

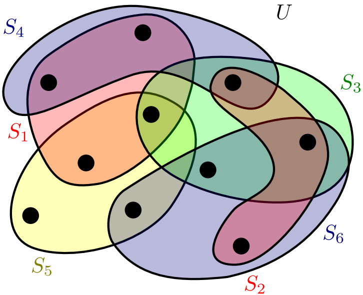
Task: Find $I \subseteq [p]$ with $|I|$ minimum such that $\cup_{i \in I} S_i = U$.

An example for deterministic rounding

(Unweighted) Set Cover with bounded coverage

Given:
- A finite set U, collection of subsets $\mathcal{F} = \{S_1, \ldots, S_p\} \subseteq 2^U$.
- $|\{i \in [p] := \{1, \ldots, p\} \mid u \in S_i\}| \leq \ell$ for all $u \in U$.

Task: Find $I \subseteq [p]$ with $|I|$ minimum such that $\cup_{i \in I} S_i = U$.



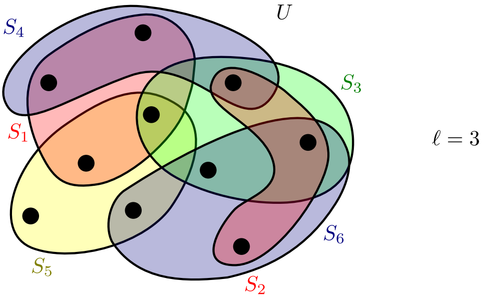
$\ell = 3$

An example for deterministic rounding

(Unweighted) Set Cover with bounded coverage

Given:
- A finite set U, collection of subsets $\mathcal{F} = \{S_1, \ldots, S_p\} \subseteq 2^U$.
- $|\{i \in [p] := \{1, \ldots, p\} \mid u \in S_i\}| \leq \ell$ for all $u \in U$.

Task: Find $I \subseteq [p]$ with $|I|$ minimum such that $\cup_{i \in I} S_i = U$.



A ℓ-approximation can be obtained by rounding an LP solution.

An example for deterministic rounding

(Unweighted) Set Cover with bounded coverage

Given:
- A finite set U, collection of subsets $\mathcal{F} = \{S_1, \ldots, S_p\} \subseteq 2^U$.
- $|\{i \in [p] := \{1, \ldots, p\} \mid u \in S_i\}| \leq \ell$ for all $u \in U$.

Task: Find $I \subseteq [p]$ with $|I|$ minimum such that $\cup_{i \in I} S_i = U$.

ILP
$$\begin{array}{llll} \min & \sum_{i \in [p]} x_i & & \\ & \sum_{i \in I : u \in S_i} x_i & \geq 1 & \forall u \in U \\ & x_i & \in \{0,1\} & \forall i \in [p] \end{array}$$

An example for deterministic rounding

(Unweighted) Set Cover with bounded coverage

Given:
- A finite set U, collection of subsets $\mathcal{F} = \{S_1, \ldots, S_p\} \subseteq 2^U$.
- $|\{i \in [p] := \{1, \ldots, p\} \mid u \in S_i\}| \leq \ell$ for all $u \in U$.

Task: Find $I \subseteq [p]$ with $|I|$ minimum such that $\cup_{i \in I} S_i = U$.

LP

$$\begin{array}{llll} \min & \sum_{i \in [p]} x_i & & \\ & \sum_{i \in I : u \in S_i} x_i & \geq 1 & \forall u \in U \\ & x_i & \geq 0 & \forall i \in [p] \end{array}$$

An example for deterministic rounding

(Unweighted) Set Cover with bounded coverage

Given:
- A finite set U, collection of subsets $\mathcal{F} = \{S_1, \dots, S_p\} \subseteq 2^U$.
- $|\{i \in [p] := \{1, \dots, p\} \mid u \in S_i\}| \leq \ell$ for all $u \in U$.

Task: Find $I \subseteq [p]$ with $|I|$ minimum such that $\cup_{i \in I} S_i = U$.

LP

$$
\begin{array}{llll}
\min & \sum_{i \in [p]} x_i & & \\
 & \sum_{i \in I : u \in S_i} x_i & \geq 1 & \forall u \in U \\
 & x_i & \geq 0 & \forall i \in [p]
\end{array}
$$

Rounding step

If $x_i^* \geq 1/\ell$, set it to 1, else to 0 $\quad \Rightarrow \quad I = \{i \in [p] \mid x_i^* \geq 1/\ell\}$.

An example for deterministic rounding

(Unweighted) Set Cover with bounded coverage

Given:
- A finite set U, collection of subsets $\mathcal{F} = \{S_1, \ldots, S_p\} \subseteq 2^U$.
- $|\{i \in [p] := \{1, \ldots, p\} \mid u \in S_i\}| \leq \ell$ for all $u \in U$.

Task: Find $I \subseteq [p]$ with $|I|$ minimum such that $\cup_{i \in I} S_i = U$.

LP

$$\begin{array}{lrll} \min & \sum_{i \in [p]} x_i & & \\ & \sum_{i \in I : u \in S_i} x_i & \geq 1 & \forall u \in U \\ & x_i & \geq 0 & \forall i \in [p] \end{array}$$

Rounding step

If $x_i^* \geq 1/\ell$, set it to 1, else to 0 $\quad \Rightarrow \quad I = \{i \in [p] \mid x_i^* \geq 1/\ell\}$.

Approximation quality and feasibility

i) Covering constraints are satisfied since for every inequality $\sum_{i \in I : u \in S_i} x_i^* \geq 1$ at least one x_i must be $\geq 1/\ell$.

ii) $|S| \leq \ell \cdot$ LP solution.

An example for randomized rounding (I)

Congestion minimization problem

Given: An undirected graph $G = (V, E)$, source destination pairs $(s_1, t_1), \dots, (s_p, t_p) \in V \times V$.

Task: For $i \in [p]$, find a s_i-t_i path $P^i \subseteq E$, such that the congestion $\max_{e \in E} |\{i \in [p] \mid e \in P^i\}|$ is minimized.

$$
\begin{aligned}
\min \quad & C & & \\
& f^i(\delta^+(v)) - f^i(\delta^-(v)) = \begin{cases} 1 & \text{if } v = s_i \\ -1 & \text{if } v = t_i \\ 0 & \text{otherwise} \end{cases} & & \forall i \in [p], v \in V \\
& \sum_{i=1}^{p} f^i(e) \leq C & & \forall e \in E \\
& f^i \in \{0,1\}^E & & \forall i \in [p]
\end{aligned}
$$

A $O(\log m / \log \log m)$-approximation, where $m := |E|$, can be obtained by randomly rounding an optimal solution to the above LP.

An example for randomized rounding (I)

Congestion minimization problem

Given: An undirected graph $G = (V, E)$, source destination pairs $(s_1, t_1), \ldots, (s_p, t_p) \in V \times V$.

Task: For $i \in [p]$, find a s_i-t_i path $P^i \subseteq E$, such that the congestion $\max_{e \in E} |\{i \in [p] \mid e \in P^i\}|$ is minimized.

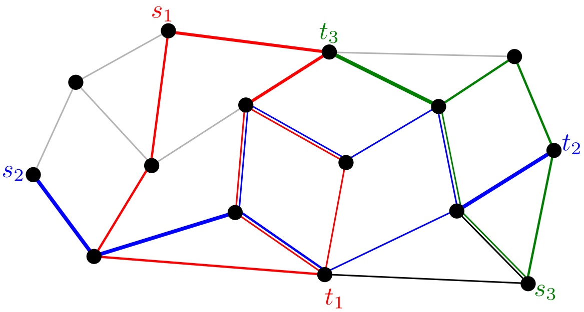
$$
\begin{aligned}
\min \quad & C & & \\
& f^i(\delta^+(v)) - f^i(\delta^-(v)) & = \begin{cases} 1 & \text{if } v = s_i \\ -1 & \text{if } v = t_i \\ 0 & \text{otherwise} \end{cases} & \quad \forall i \in [p], v \in V \\
& \sum_{i=1}^{p} f^i(e) & \leq C & \quad \forall e \in E \\
& f^i & \in [0,1]^E & \quad \forall i \in [p]
\end{aligned}
$$

A $O(\log m / \log\log m)$-approximation, where $m := |E|$, can be obtained by randomly rounding an optimal solution to the above LP.

An example for randomized rounding (II)

A $O(\log m / \log\log m)$-approximation

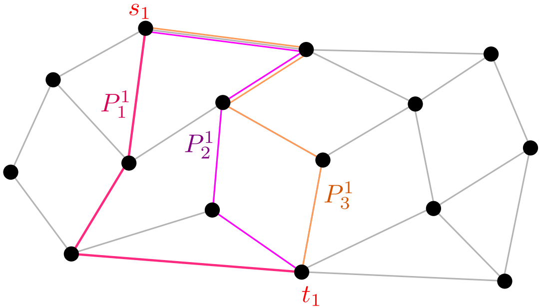
1. Determine an optimal solution $f^i \in [0,1]^E, i \in [p]$ to the LP.
2. Decompose each f^i into at most m paths $P^i_1, \dots P^i_{n_i}$ from s_i to t_i, i.e., $f^i = \sum_{k=1}^{n_i} \alpha^i_k \mathbf{1}_{P^i_k}$.
3. For each $i \in [p]$, choose randomly one path P^i out of $P^i_1, \dots, P^i_k$, where P^i_k is chosen with probability α^i_k.
4. Return $P^1, \dots, P^p$.



An example for randomized rounding (II)

A $O(\log m / \log\log m)$-approximation

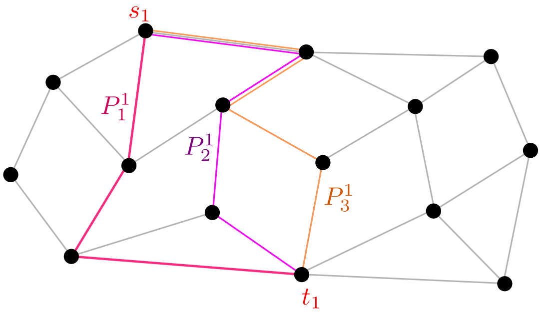
1. Determine an optimal solution $f^i \in [0,1]^E, i \in [p]$ to the LP.
2. Decompose each f^i into at most m paths $P^i_1, \dots P^i_{n_i}$ from s_i to t_i, i.e., $f^i = \sum_{k=1}^{n_i} \alpha^i_k \mathbf{1}_{P^i_k}$.
3. For each $i \in [p]$, choose randomly one path P^i out of $P^i_1, \dots, P^i_k$, where P^i_k is chosen with probability α^i_k.
4. Return $P^1, \dots, P^p$.



An example for randomized rounding (II)

A $O(\log m / \log\log m)$-approximation

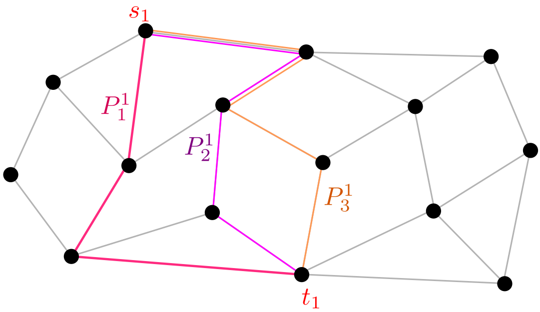
1. Determine an optimal solution $f^i \in [0,1]^E, i \in [p]$ to the LP.
2. Decompose each f^i into at most m paths $P^i_1, \dots P^i_{n_i}$ from s_i to t_i, i.e., $f^i = \sum_{k=1}^{n_i} \alpha^i_k \mathbf{1}_{P^i_k}$.
3. For each $i \in [p]$, choose randomly one path P^i out of $P^i_1, \dots, P^i_k$, where P^i_k is chosen with probability α^i_k.
4. Return $P^1, \dots, P^p$.



An example for randomized rounding (II)

A $O(\log m / \log\log m)$-approximation

1. Determine an optimal solution $f^i \in [0,1]^E, i \in [p]$ to the LP.
2. Decompose each f^i into at most m paths $P^i_1, \ldots P^i_{n_i}$ from s_i to t_i, i.e., $f^i = \sum_{k=1}^{n_i} \alpha^i_k \mathbf{1}_{P^i_k}$.
3. For each $i \in [p]$, choose randomly one path P^i out of $P^i_1, \ldots, P^i_k$, where P^i_k is chosen with probability α^i_k.
4. Return $P^1, \ldots, P^p$.



An example for randomized rounding (III)

Analysis of the algorithm

- Fix an edge e.
- $c^*(e) :=$ fractional congestion of e.
- $C(e) :=$ random congestion of e after rounding, i.e.,

$$C(e) = \sum_{i=1}^{p} \mathbf{1}_{e \in P^i}, \quad \text{where} \quad \mathbf{1}_{e \in P^i} = \begin{cases} 1 & \text{if } e \in P^i, \\ 0 & \text{otherwise.} \end{cases}$$

- The variables $\{\mathbf{1}_{e \in P^i} \mid i \in [p]\}$ are independent.
- $\mathbf{E}[C(e)] = c^*(e)$

Theorem (Chernoff bound)

Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0,1]$. For $\mu \geq \mathbf{E}[\sum_{i=1}^{n} X_i]$ and $\delta > 0$ we have:

$$\Pr[\sum_{i=1}^{n} X_i \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{\mu}.$$

An example for randomized rounding (III)

Analysis of the algorithm

- Fix an edge e.
- $c^*(e) :=$ fractional congestion of e.
- $C(e) :=$ random congestion of e after rounding, i.e.,

$$C(e) = \sum_{i=1}^{p} \mathbf{1}_{e \in P^i}, \quad \text{where} \quad \mathbf{1}_{e \in P^i} = \begin{cases} 1 & \text{if } e \in P^i, \\ 0 & \text{otherwise.} \end{cases}$$

- The variables $\{\mathbf{1}_{e \in P^i} \mid i \in [p]\}$ are independent.
- $\mathbf{E}[C(e)] = c^*(e)$

Theorem (Chernoff bound)

Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0,1]$. For $\mu \geq \mathbf{E}[\sum_{i=1}^{n} X_i]$ and $\delta > 0$ we have:

$$\Pr[\sum_{i=1}^{n} X_i \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{\mu}.$$

An example for randomized rounding (III)

Analysis of the algorithm

Theorem (Chernoff bound)

Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0,1]$. For $\mu \geq \mathbf{E}[\sum_{i=1}^{n} X_i]$ and $\delta > 0$ we have:

$$\Pr[\sum_{i=1}^{n} X_i \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}.$$

- $X_i := \mathbf{1}_{e \in P^i}$
- $\mu = \mu(e) = \max\{c^*(e), 1\} \rightarrow \mu$ is a lower bound of the optimal congestion.
- $1 + \delta := 4 \log m / \log \log m$.

$$\Pr[C(e) > (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \overset{\boxed{\mu \geq 1}}{\leq} \frac{e^{\delta}}{(1+\delta)^{1+\delta}} \leq \left(\frac{e}{1+\delta}\right)^{1+\delta}$$

$$= \left(\frac{e \log \log m}{4 \log m}\right)^{\frac{4 \log m}{\log \log m}} \leq \left(\frac{1}{\sqrt{\log m}}\right)^{\frac{4 \log m}{\log \log m}} = \frac{1}{m^2}.$$

- By union bound: $\Pr[\exists e \in E \text{ with } C(e) > (1+\delta)\mu(e)] \leq m \cdot \frac{1}{m^2} \leq 1/m$.
- With probability $\geq 1 - 1/m$ we have a $(4 \log m / \log \log m)$-approximation.

An example for randomized rounding (III)

Analysis of the algorithm

Theorem (Chernoff bound)

Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0,1]$. For $\mu \geq \mathbf{E}[\sum_{i=1}^{n} X_i]$ and $\delta > 0$ we have:

$$\Pr[\sum_{i=1}^{n} X_i \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{\mu}.$$

- $X_i := \mathbf{1}_{e \in P^i}$
- $\mu = \mu(e) = \max\{c^*(e), 1\} \to \mu$ is a lower bound of the optimal congestion.
- $1 + \delta := 4 \log m / \log\log m$.

$$\Pr[C(e) > (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{\mu} \overset{\boxed{\mu \geq 1}}{\leq} \frac{e^{\delta}}{(1+\delta)^{1+\delta}} \leq \left(\frac{e}{1+\delta} \right)^{1+\delta}$$

$$= \left(\frac{e \log\log m}{4 \log m} \right)^{\frac{4 \log m}{\log\log m}} \leq \left(\frac{1}{\sqrt{\log m}} \right)^{\frac{4 \log m}{\log\log m}} = \frac{1}{m^2}.$$

- By union bound: $\Pr[\exists e \in E \text{ with } C(e) > (1+\delta)\mu(e)] \leq m \cdot \frac{1}{m^2} \leq 1/m$.
- With probability $\geq 1 - 1/m$ we have a $(4 \log m / \log\log m)$-approximation.

An example for randomized rounding (III)

Analysis of the algorithm

Theorem (Chernoff bound)

Let $X_1, \dots, X_n$ be independent random variables taking values in $[0, 1]$. For $\mu \geq \mathbf{E}[\sum_{i=1}^{n} X_i]$ and $\delta > 0$ we have:

$$\Pr[\sum_{i=1}^{n} X_i \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}.$$

- $X_i := \mathbf{1}_{e \in P^i}$
- $\mu = \mu(e) = \max\{c^*(e), 1\} \rightarrow \mu$ is a lower bound of the optimal congestion.
- $1 + \delta := 4 \log m / \log \log m$.

$$\Pr[C(e) > (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \overset{\boxed{\mu \geq 1}}{\leq} \frac{e^{\delta}}{(1+\delta)^{1+\delta}} \leq \left(\frac{e}{1+\delta}\right)^{1+\delta}$$

$$= \left(\frac{e \log \log m}{4 \log m}\right)^{\frac{4 \log m}{\log \log m}} \leq \left(\frac{1}{\sqrt{\log m}}\right)^{\frac{4 \log m}{\log \log m}} = \frac{1}{m^2}.$$

- By union bound: $\Pr[\exists e \in E \text{ with } C(e) > (1+\delta)\mu(e)] \leq m \cdot \frac{1}{m^2} \leq 1/m$.
- With probability $\geq 1 - 1/m$ we have a $(4 \log m / \log \log m)$-approximation.

An example for randomized rounding (III)

Analysis of the algorithm

Theorem (Chernoff bound)

Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0, 1]$. For $\mu \geq \mathbf{E}[\sum_{i=1}^{n} X_i]$ and $\delta > 0$ we have:

$$\Pr[\sum_{i=1}^{n} X_i \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu.$$

- $X_i := \mathbf{1}_{e \in P^i}$
- $\mu = \mu(e) = \max\{c^*(e), 1\} \to \mu$ is a lower bound of the optimal congestion.
- $1 + \delta := 4 \log m / \log\log m$.

$$\Pr[C(e) > (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \overset{\boxed{\mu \geq 1}}{\leq} \frac{e^\delta}{(1+\delta)^{1+\delta}} \leq \left(\frac{e}{1+\delta}\right)^{1+\delta}$$

$$= \left(\frac{e \log\log m}{4 \log m}\right)^{\frac{4 \log m}{\log\log m}} \leq \left(\frac{1}{\sqrt{\log m}}\right)^{\frac{4 \log m}{\log\log m}} = \frac{1}{m^2}.$$

- By union bound: $\Pr[\exists e \in E \text{ with } C(e) > (1+\delta)\mu(e)] \leq m \cdot \frac{1}{m^2} \leq 1/m$.
- With probability $\geq 1 - 1/m$ we have a $(4 \log m / \log\log m)$-approximation.

An example for randomized rounding (III)

Analysis of the algorithm

Theorem (Chernoff bound)

Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0,1]$. For $\mu \geq \mathbf{E}[\sum_{i=1}^{n} X_i]$ and $\delta > 0$ we have:

$$\Pr[\sum_{i=1}^{n} X_i \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}.$$

- $X_i := \mathbf{1}_{e \in P^i}$
- $\mu = \mu(e) = \max\{c^*(e), 1\} \to \mu$ is a lower bound of the optimal congestion.
- $1 + \delta := 4 \log m / \log\log m$.

$$\Pr[C(e) > (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \overset{\boxed{\mu \geq 1}}{\leq} \frac{e^{\delta}}{(1+\delta)^{1+\delta}} \leq \left(\frac{e}{1+\delta}\right)^{1+\delta}$$

$$= \left(\frac{e \log\log m}{4 \log m}\right)^{\frac{4 \log m}{\log\log m}} \leq \left(\frac{1}{\sqrt{\log m}}\right)^{\frac{4 \log m}{\log\log m}} = \frac{1}{m^2}.$$

- By union bound: $\Pr[\exists e \in E \text{ with } C(e) > (1+\delta)\mu(e)] \leq m \cdot \frac{1}{m^2} \leq 1/m$.
- With probability $\geq 1 - 1/m$ we have a $(4 \log m / \log\log m)$-approximation.

Goals when designing rounding procedure?

- Maintain feasibility.
  → Can be obtained by careful dependent rounding.
- Rounded point should be "similar" to fractional one.
  → Typically obtained by independence of rounding and Chernoff bounds.

Main difficulty: these two goals are conflicting.

For example: how to round a point in

- the spanning tree polytope?
- a (poly-)matroid polytpe?
- the matching or b-matching polytope?
- the independent set polytope of the intersection of two (poly-)matroids?

Goals when designing rounding procedure?

- Maintain feasibility.
  $\rightarrow$ Can be obtained by careful dependent rounding.
- Rounded point should be "similar" to fractional one.
  $\rightarrow$ Typically obtained by independence of rounding and Chernoff bounds.

Main difficulty: these two goals are conflicting.

For example: how to round a point in

- the spanning tree polytope?
- a (poly-)matroid polytpe?
- the matching or b-matching polytope?
- the independent set polytope of the intersection of two (poly-)matroids?

Goals when designing rounding procedure?

- Maintain feasibility.
  → Can be obtained by careful dependent rounding.
- Rounded point should be "similar" to fractional one.
  → Typically obtained by independence of rounding and Chernoff bounds.

Main difficulty: these two goals are conflicting.

For example: how to round a point in

- the spanning tree polytope?
- a (poly-)matroid polytpe?
- the matching or *b*-matching polytope?
- the independent set polytope of the intersection of two (poly-)matroids?

Goals when designing rounding procedure?

- Maintain feasibility.
  → Can be obtained by careful dependent rounding.
- Rounded point should be "similar" to fractional one.
  → Typically obtained by independence of rounding and Chernoff bounds.

Main difficulty: these two goals are conflicting.

For example: how to round a point in

- the spanning tree polytope?
- a (poly-)matroid polytpe?
- the matching or *b*-matching polytope?
- the independent set polytope of the intersection of two (poly-)matroids?

Outline

Multi-criteria spanning tree (I)

Terminology

- $G = (V, E)$: undirected graph.
- $\mathcal{T} \subseteq 2^E$: set of all spanning trees.
- $\mathbf{1}_U \in \{0,1\}^E$: incidence vector of the set $U \subseteq E$.
- $P_{ST} = conv(\{\mathbf{1}_T \mid T \in \mathcal{T}\})$: spanning tree polytope.

Multi-budgeted spanning tree problem

Given: G, a weight function $w : E \to [0,1]$, and p length functions $\ell_i : E \to [0,1]$ for $i \in [p]$ with corresponding budget $B_i \in \mathbb{Q}_+$.

Task: Find spanning tree of minimum weight respecting all budgets.

$$\min_{T \in \mathcal{T}} \; w(T) \quad \ell_i(T) \leq B_i \quad \forall i \in [p] \qquad \Leftrightarrow \qquad \min_{x \text{ vertex of } P_{ST}} \; w^T x \quad \ell_i^T x \leq B_i \quad \forall i \in [p]$$

These constraints are often weak, i.e., it is ok to violate them slightly.

$\to$ We then talk of multi-criteria optimization.

Multi-criteria spanning tree (I)

Terminology

- $G = (V, E)$: undirected graph.
- $\mathcal{T} \subseteq 2^E$: set of all spanning trees.
- $\mathbf{1}_U \in \{0,1\}^E$: incidence vector of the set $U \subseteq E$.
- $P_{ST} = conv(\{\mathbf{1}_T \mid T \in \mathcal{T}\})$: spanning tree polytope.

Multi-budgeted spanning tree problem

Given: G, a weight function $w : E \to [0,1]$, and p length functions $\ell_i : E \to [0,1]$ for $i \in [p]$ with corresponding budget $B_i \in \mathbb{Q}_+$.

Task: Find spanning tree of minimum weight respecting all budgets.

$$\begin{array}{lll} \min\limits_{T \in \mathcal{T}} & w(T) & \\ & \ell_i(T) \le B_i & \forall i \in [p] \end{array} \quad \Leftrightarrow \quad \begin{array}{lll} \min\limits_{x \text{ vertex of } P_{ST}} & w^T x & \\ & \ell_i^T x \le B_i & \forall i \in [p] \end{array}$$

These constraints are often weak, i.e., it is ok to violate them slightly.

$\rightarrow$ We then talk of multi-criteria optimization.

Multi-criteria spanning tree (I)

Terminology

- $G = (V, E)$: undirected graph.
- $\mathcal{T} \subseteq 2^E$: set of all spanning trees.
- $\mathbf{1}_U \in \{0,1\}^E$: incidence vector of the set $U \subseteq E$.
- $P_{ST} = conv(\{\mathbf{1}_T \mid T \in \mathcal{T}\})$: spanning tree polytope.

Multi-budgeted spanning tree problem

Given: G, a weight function $w : E \to [0,1]$, and p length functions $\ell_i : E \to [0,1]$ for $i \in [p]$ with corresponding budget $B_i \in \mathbb{Q}_+$.

Task: Find spanning tree of minimum weight respecting all budgets.

$$\begin{array}{lll} \min\limits_{T\in\mathcal{T}} & w(T) & \\ & \ell_i(T) \leq B_i & \forall i \in [p] \end{array} \quad \Leftrightarrow \quad \begin{array}{lll} \min\limits_{x \text{ vertex of } P_{ST}} & w^T x & \\ & \ell_i^T x \preceq B_i & \forall i \in [p] \end{array}$$

These constraints are often weak, i.e., it is ok to violate them slightly.

$\rightarrow$ We then talk of multi-criteria optimization.

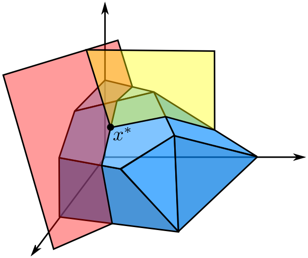
Multi-criteria spanning tree (II)

Multi-criteria problem:

$$\begin{array}{ll} \min\limits_{x \text{ vertex of } P_{ST}} & w^T x \\ & \ell_i^T x \preceq B_i \quad \forall i \in [p] \end{array}$$

$\xrightarrow{\text{LP relaxation}}$

$$\begin{array}{ll} \min\limits_{x \in P_{ST}} & w^T x \\ & \ell_i^T x \preceq B_i \quad \forall i \in [p] \end{array}$$



- The LP (with hard constraints) can be solved in polynomial time.
  $\rightarrow$ Let x^* be an optimal LP solution.
- We will show how x^* can be rounded to a good spanning tree.

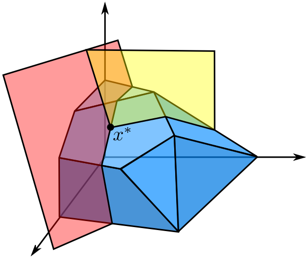
Multi-criteria spanning tree (II)

Multi-criteria problem:

$$\min_{x \text{ vertex of } P_{ST}} \quad w^T x$$
$$\ell_i^T x \preceq B_i \quad \forall i \in [p]$$

$\xrightarrow{\text{LP relaxation}}$

$$\min_{x \in P_{ST}} \quad w^T x$$
$$\ell_i^T x \preceq B_i \quad \forall i \in [p]$$



- The LP (with hard constraints) can be solved in polynomial time.
  $\rightarrow$ Let x^* be an optimal LP solution.
- We will show how x^* can be rounded to a good spanning tree.

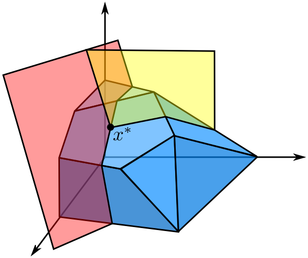
Multi-criteria spanning tree (II)

Multi-criteria problem:

$$\min_{x \text{ vertex of } P_{ST}} \quad w^T x \qquad \ell_i^T x \preceq B_i \quad \forall i \in [p]$$

$\xrightarrow{\text{LP relaxation}}$

$$\min_{x \in P_{ST}} \quad w^T x \qquad \ell_i^T x \preceq B_i \quad \forall i \in [p]$$



- The LP (with hard constraints) can be solved in polynomial time.
  $\rightarrow$ Let x^* be an optimal LP solution.
- We will show how x^* can be rounded to a good spanning tree.

Main framework

- Drop budget constraints and randomly round x^* to a vertex of P_{ST} close to x^*.

Problem: defining good rounding steps in P_{ST} is not easy.
Idea: profit from combinatorial knowledge about spanning trees.

- We work on a convex decomposition of the point $x^* \in P$ to round: $x^* = \sum_{i=1}^{m} \beta_i \mathbf{1}_{T_i}$, where $T_1, \ldots, T_m$ are spanning trees.
- We iteratively merge the spanning trees $T_1, \ldots, T_m$ to a single spanning tree.

Main framework

- Drop budget constraints and randomly round x^* to a vertex of P_{ST} close to x^*.

Problem: defining good rounding steps in P_{ST} is not easy.
Idea: profit from combinatorial knowledge about spanning trees.

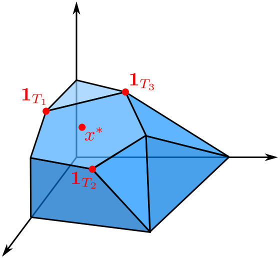
- We work on a convex decomposition of the point $x^* \in P$ to round: $x^* = \sum_{i=1}^{m} \beta_i \mathbf{1}_{T_i}$, where $T_1, \ldots, T_m$ are spanning trees.
- We iteratively merge the spanning trees $T_1, \ldots, T_m$ to a single spanning tree.

Main framework

- Drop budget constraints and randomly round x^* to a vertex of P_{ST} close to x^*.

Problem: defining good rounding steps in P_{ST} is not easy.
Idea: profit from combinatorial knowledge about spanning trees.

- We work on a convex decomposition of the point $x^* \in P$ to round: $x^* = \sum_{i=1}^{m} \beta_i \mathbf{1}_{T_i}$, where $T_1, \ldots, T_m$ are spanning trees.
- We iteratively merge the spanning trees $T_1, \ldots, T_m$ to a single spanning tree.

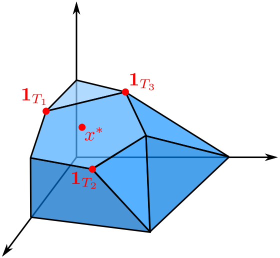


Main framework

- Drop budget constraints and randomly round x^* to a vertex of P_{ST} close to x^*.

Problem: defining good rounding steps in P_{ST} is not easy.
Idea: profit from combinatorial knowledge about spanning trees.

- We work on a convex decomposition of the point $x^* \in P$ to round: $x^* = \sum_{i=1}^{m} \beta_i \mathbf{1}_{T_i}$, where $T_1, \ldots, T_m$ are spanning trees.
- We iteratively merge the spanning trees $T_1, \ldots, T_m$ to a single spanning tree.



Rounding using the convex decomposition

- We start with $x_1 := x^*$ and iteratively reduce the number of terms in the convex representation using a Merge operation.

$$
\begin{aligned}
x_1 &= \underbrace{\beta_1 \mathbf{1}_{T_1} + \beta_2 \mathbf{1}_{T_2}} + \beta_3 \mathbf{1}_{T_3} + \dots \beta_m \mathbf{1}_{T_m} \\
x_2 &= \underbrace{(\beta_1 + \beta_2)\mathbf{1}_{T_{1:2}} + \beta_3 \mathbf{1}_{T_3}} + \dots \beta_m \mathbf{1}_{T_m} \quad & \mid T_{1:2} = Merge(\beta_1, T_1, \beta_2, T_2) \\
x_3 &= (\beta_1 + \beta_2 + \beta_3)\mathbf{1}_{T_{1:3}} + \dots \beta_m \mathbf{1}_{T_m} \quad & \mid T_{1:3} = Merge(\beta_1 + \beta_2, T_{1:2}, \beta_3, T_3) \\
&\vdots \\
x_k &= (\textstyle\sum_{i=1}^{k} \beta_i)\mathbf{1}_{T_{1:k}} + \textstyle\sum_{i=k+1}^{m} \beta_i \mathbf{1}_{T_i} \quad & \mid T_{1:k} = Merge(\textstyle\sum_{i=1}^{k-1} \beta_i, T_{1:k-1}, \beta_k, T_k) \\
&\vdots \\
x_m &= (\beta_1 + \dots + \beta_m)\mathbf{1}_{T_{1:m}} = \mathbf{1}_{T_{1:m}}
\end{aligned}
$$

Merging two spanning trees by swaps

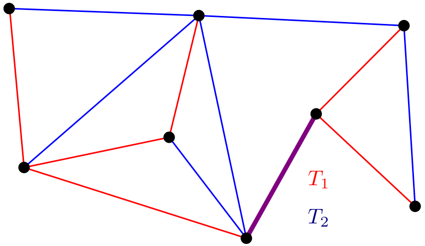
Algorithm Merge$(\beta_1, T_1, \beta_2, T_2)$

While $(T_1 \neq T_2)$ do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
    $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
    Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.

Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

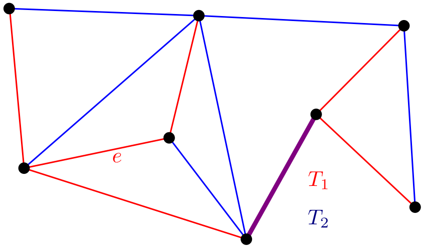
While ($T_1 \neq T_2$) do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
    $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
    Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

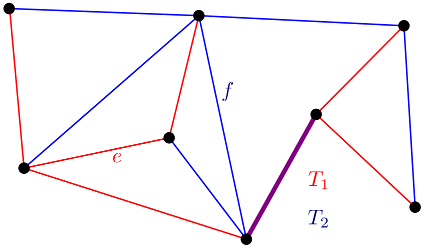
While ($T_1 \neq T_2$) do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
  $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
  Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge$(\beta_1, T_1, \beta_2, T_2)$

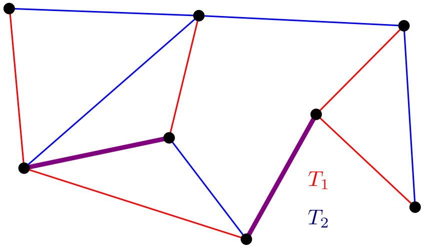
While $(T_1 \neq T_2)$ do
Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
$T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

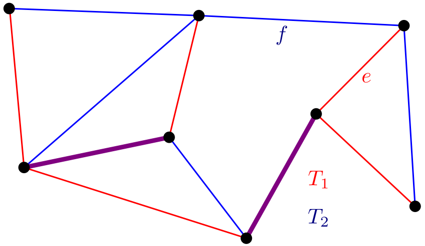
While ($T_1 \neq T_2$) do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
    $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
    Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

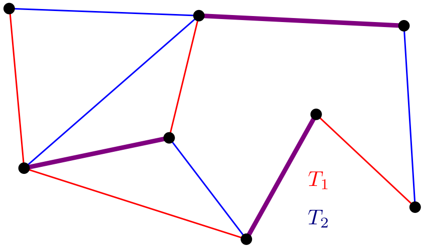
While ($T_1 \neq T_2$) do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
  $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
  Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

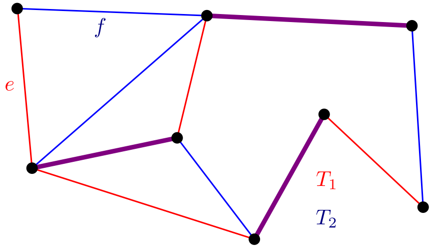
While ($T_1 \neq T_2$) do
Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
$T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

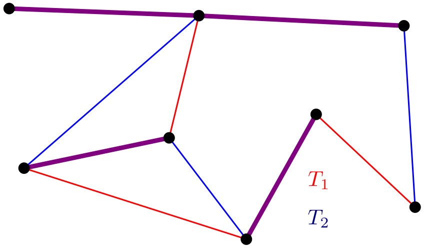
While ($T_1 \neq T_2$) do
Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
$T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge$(\beta_1, T_1, \beta_2, T_2)$

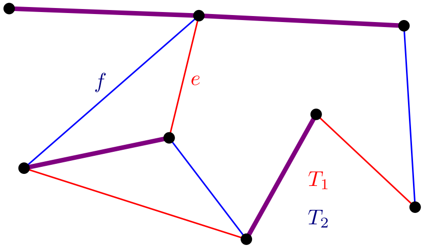
While $(T_1 \neq T_2)$ do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
    $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
    Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

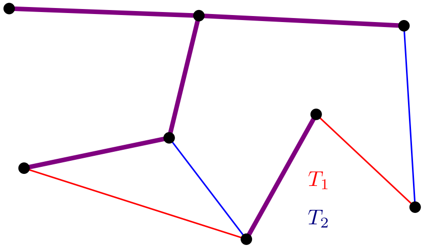
While ($T_1 \neq T_2$) do
Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
$T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

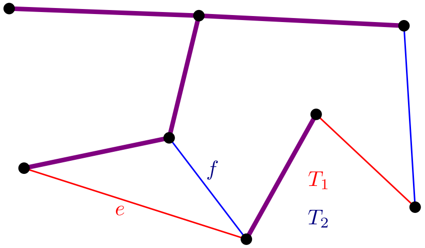
While ($T_1 \neq T_2$) do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
  $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
  Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

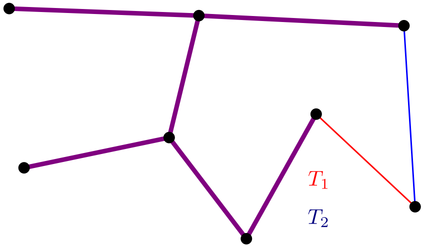
While ($T_1 \neq T_2$) do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
    $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
    Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

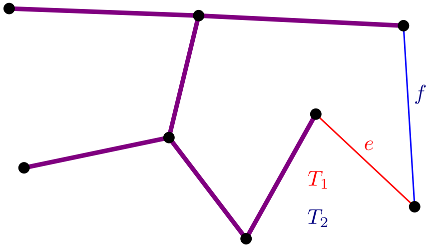
While ($T_1 \neq T_2$) do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
    $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
    Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge($\beta_1, T_1, \beta_2, T_2$)

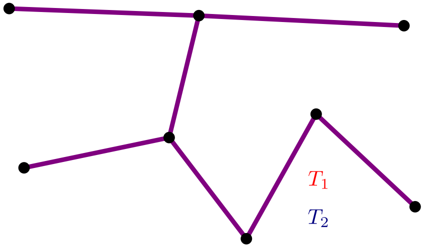
While ($T_1 \neq T_2$) do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
  $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
  Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Merging two spanning trees by swaps

Algorithm Merge$(\beta_1, T_1, \beta_2, T_2)$

While $(T_1 \neq T_2)$ do
  Pick $e \in T_1 \setminus T_2$ and find $f \in T_2 \setminus T_1$ such that
  $T_1 - e + f \in \mathcal{T}$ and $T_2 - f + e \in \mathcal{T}$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{T_2 \leftarrow T_2 - f + e\}$;
  Else $\{T_1 \leftarrow T_1 - e + f\}$;
EndWhile
Output T_1.



Concentration bounds (I)

- Assume for simplicity $x^* = \beta_1 \mathbf{1}_{T_1} + \beta_2 \mathbf{1}_{T_2}$.

Properties of the Merge procedure

Let y^1 be the point obtained after applying one merge operation to x^*.

i) $\mathbf{E}[y^1] = x^*$.

ii) Exactly two components change and their sum remains constant.

Theorem

From the above properties the following negative correlation property of a rounded tree T can be derived. For any $U \subseteq E$, we have

- $\Pr[U \subseteq T] \leq \prod_{e \in U} x^*(e)$,
- $\Pr[U \cap T = \emptyset] \leq \prod_{e \in U} (1 - x^*(e))$.

Such negative correlation is sufficient to get Chernoff bounds (Panconesi and Srinivasan [1997]).

Concentration bounds (I)

- Assume for simplicity $x^* = \beta_1 \mathbf{1}_{T_1} + \beta_2 \mathbf{1}_{T_2}$.

Properties of the Merge procedure

Let y^1 be the point obtained after applying one merge operation to x^*.

i) $\mathbf{E}[y^1] = x^*$.

ii) Exactly two components change and their sum remains constant.

Theorem

From the above properties the following negative correlation property of a rounded tree T can be derived. For any $U \subseteq E$, we have

- $\Pr[U \subseteq T] \leq \prod_{e \in U} x^*(e)$,
- $\Pr[U \cap T = \emptyset] \leq \prod_{e \in U} (1 - x^*(e))$.

Such negative correlation is sufficient to get Chernoff bounds (Panconesi and Srinivasan [1997]).

Concentration bounds (I)

- Assume for simplicity $x^* = \beta_1 \mathbf{1}_{T_1} + \beta_2 \mathbf{1}_{T_2}$.

Properties of the Merge procedure

Let y^1 be the point obtained after applying one merge operation to x^*.

i) $\mathbf{E}[y^1] = x^*$.

ii) Exactly two components change and their sum remains constant.

Theorem

From the above properties the following negative correlation property of a rounded tree T can be derived. For any $U \subseteq E$, we have

- $\Pr[U \subseteq T] \leq \prod_{e \in U} x^*(e)$,
- $\Pr[U \cap T = \emptyset] \leq \prod_{e \in U} (1 - x^*(e))$.

Such negative correlation is sufficient to get Chernoff bounds (Panconesi and Srinivasan [1997]).

Concentration bounds (I)

- Assume for simplicity $x^* = \beta_1 \mathbf{1}_{T_1} + \beta_2 \mathbf{1}_{T_2}$.

Properties of the Merge procedure

Let y^1 be the point obtained after applying one merge operation to x^*.

i) $\mathbf{E}[y^1] = x^*$.

ii) Exactly two components change and their sum remains constant.

Theorem

From the above properties the following negative correlation property of a rounded tree T can be derived. For any $U \subseteq E$, we have

- $\Pr[U \subseteq T] \leq \prod_{e \in U} x^*(e)$,
- $\Pr[U \cap T = \emptyset] \leq \prod_{e \in U} (1 - x^*(e))$.

Such negative correlation is sufficient to get Chernoff bounds (Panconesi and Srinivasan [1997]).

Concentration bounds (II)

Let $\ell : E \to [0,1]$ and let $\ell(x) := \sum_{e \in E} x_e \ell(e)$ for $x \in [0,1]^E$.

Theorem (Chernoff bounds for swap rounding)

Let T be a random tree obtained by randomized swap rounding. For $\mu \geq \ell(x^*)$ and $\delta > 0$ we have:

$$\Pr[\ell(T) \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu .$$

- Similar bounds can be shown for the lower tail (and even for submodular functions).

Linear functions do not change much through randomized swap rounding!

Concentration bounds (II)

Let $\ell : E \to [0,1]$ and let $\ell(x) := \sum_{e \in E} x_e \ell(e)$ for $x \in [0,1]^E$.

Theorem (Chernoff bounds for swap rounding)

Let T be a random tree obtained by randomized swap rounding. For $\mu \geq \ell(x^*)$ and $\delta > 0$ we have:

$$\Pr[\ell(T) \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu .$$

- Similar bounds can be shown for the lower tail (and even for submodular functions).

Linear functions do not change much through randomized swap rounding!

Concentration bounds (II)

Let $\ell : E \to [0,1]$ and let $\ell(x) := \sum_{e \in E} x_e \ell(e)$ for $x \in [0,1]^E$.

Theorem (Chernoff bounds for swap rounding)

Let T be a random tree obtained by randomized swap rounding. For $\mu \geq \ell(x^*)$ and $\delta > 0$ we have:

$$\Pr[\ell(T) \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu .$$

- Similar bounds can be shown for the lower tail (and even for submodular functions).

Linear functions do not change much through randomized swap rounding!

Application to multi-criteria spanning tree

$$\begin{aligned} \min_{x \in P_{ST}} \quad & w^T x \\ & \ell_i^T x \preceq B_i \quad \forall i \in [p] \end{aligned}$$

$\rightarrow$ Again, let x^* be an optimal solution to this LP.

Theorem

Let $\epsilon \in [0,1]$. It suffices to consider $O(1/\epsilon)$ independent outputs of the swap rounding algorithm, to obtain with high probability at least one spanning tree T that is a $(1+\epsilon, O(\log p / \log\log p))$-approximation for x^*, i.e., there is a constant $c(\epsilon)$ such that

$$\begin{aligned} w(T) &\leq (1+\epsilon) w(x^*) \\ \ell_i(T) &\leq \frac{c(\epsilon)\log p}{\log\log p} \ell_i(x^*) \quad \forall i \in [p] \end{aligned}$$

Remarks

- Since we have a $(1+\epsilon, O(\log p / \log\log p))$-approximation with respect to x^*, we have the same guarantee with respect to OPT.
- The above result holds even if p is not constant.
- If p is constant, a polynomial $(1+\epsilon, 1+\epsilon)$-approximation can be obtained by a preliminary guessing step.

Application to multi-criteria spanning tree

$$\begin{array}{lll} \min_{x \in P_{ST}} & w^T x & \\ & \ell_i^T x & \preceq B_i \quad \forall i \in [p] \end{array}$$

$\rightarrow$ Again, let x^* be an optimal solution to this LP.

Theorem

Let $\epsilon \in [0,1]$. It suffices to consider $O(1/\epsilon)$ independent outputs of the swap rounding algorithm, to obtain with high probability at least one spanning tree T that is a $(1+\epsilon, O(\log p / \log\log p))$-approximation for x^*, i.e., there is a constant $c(\epsilon)$ such that

$$\begin{aligned} w(T) &\leq (1+\epsilon) w(x^*) \\ \ell_i(T) &\leq \tfrac{c(\epsilon) \log p}{\log\log p} \ell_i(x^*) \quad \forall i \in [p] \end{aligned}$$

Remarks

- Since we have a $(1+\epsilon, O(\log p / \log\log p))$-approximation with respect to x^*, we have the same guarantee with respect to OPT.
- The above result holds even if p is not constant.
- If p is constant, a polynomial $(1+\epsilon, 1+\epsilon)$-approximation can be obtained by a preliminary guessing step.

Proof of $(1+\epsilon, 4\log p/\log\log p)$-approximation

- $T :=$ spanning tree obtained from x^* by randomized swap rounding.
- $1+\delta := 4\log p/\log\log p$ and we assume for simplicity $\epsilon \geq 4/p$.

Probability that budget i is violated:
Using Chernoff bound for swap rounding:

$$\Pr[\ell_i(T) > (1+\delta)B_i] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{B_i} \overset{\boxed{B_i \geq 1}}{\leq} \frac{e^\delta}{(1+\delta)^{1+\delta}} \leq \left(\frac{e}{1+\delta}\right)^{1+\delta}$$

$$= \left(\frac{e\log\log p}{4\log p}\right)^{\frac{4\log p}{\log\log p}} \leq \left(\frac{1}{\sqrt{\log p}}\right)^{\frac{4\log p}{\log\log p}} = \frac{1}{p^2}.$$

Probability that some budget is violated:
Using the union bound:

$$\Pr[\ell_i(T) > (1+\delta)\ell_i(x^*) \text{ for some } i] \leq p \cdot \frac{1}{p^2} = \frac{1}{p} \leq \frac{\epsilon}{4}.$$

Probability that objective function is not fine:
Using Markov's inequality:

$$\Pr[w(T) > (1+\epsilon)w(x^*)] \leq \frac{\mathbf{E}[w(T)]}{(1+\epsilon)w(x^*)} \overset{\boxed{\mathbf{E}[w(T)] = w(x^*)}}{=} \frac{1}{1+\epsilon} \leq 1 - \frac{\epsilon}{2}.$$

Proof of $(1+\epsilon, 4\log p/\log\log p)$-approximation

- $T :=$ spanning tree obtained from x^* by randomized swap rounding.
- $1+\delta := 4\log p/\log\log p$ and we assume for simplicity $\epsilon \geq 4/p$.

Probability that budget i is violated:
Using Chernoff bound for swap rounding:

$$\Pr[\ell_i(T) > (1+\delta)B_i] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{B_i} \overset{\boxed{B_i \geq 1}}{\leq} \frac{e^\delta}{(1+\delta)^{1+\delta}} \leq \left(\frac{e}{1+\delta}\right)^{1+\delta}$$

$$= \left(\frac{e\log\log p}{4\log p}\right)^{\frac{4\log p}{\log\log p}} \leq \left(\frac{1}{\sqrt{\log p}}\right)^{\frac{4\log p}{\log\log p}} = \frac{1}{p^2}.$$

Probability that some budget is violated:
Using the union bound:

$$\Pr[\ell_i(T) > (1+\delta)\ell_i(x^*) \text{ for some } i] \leq p \cdot \frac{1}{p^2} = \frac{1}{p} \leq \frac{\epsilon}{4}.$$

Probability that objective function is not fine:
Using Markov's inequality:

$$\Pr[w(T) > (1+\epsilon)w(x^*)] \leq \frac{\mathbf{E}[w(T)]}{(1+\epsilon)w(x^*)} \overset{\boxed{\mathbf{E}[w(T)] = w(x^*)}}{=} \frac{1}{1+\epsilon} \leq 1 - \frac{\epsilon}{2}.$$

Proof of $(1+\epsilon, 4\log p/\log\log p)$-approximation

- $T :=$ spanning tree obtained from x^* by randomized swap rounding.
- $1+\delta := 4\log p/\log\log p$ and we assume for simplicity $\epsilon \geq 4/p$.

Probability that budget i is violated:
Using Chernoff bound for swap rounding:

$$\Pr[\ell_i(T) > (1+\delta)B_i] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{B_i} \overset{\boxed{B_i \geq 1}}{\leq} \frac{e^\delta}{(1+\delta)^{1+\delta}} \leq \left(\frac{e}{1+\delta}\right)^{1+\delta}$$

$$= \left(\frac{e\log\log p}{4\log p}\right)^{\frac{4\log p}{\log\log p}} \leq \left(\frac{1}{\sqrt{\log p}}\right)^{\frac{4\log p}{\log\log p}} = \frac{1}{p^2}.$$

Probability that some budget is violated:
Using the union bound:

$$\Pr[\ell_i(T) > (1+\delta)\ell_i(x^*) \text{ for some } i] \leq p \cdot \frac{1}{p^2} = \frac{1}{p} \leq \frac{\epsilon}{4}.$$

Probability that objective function is not fine:
Using Markov's inequality:

$$\Pr[w(T) > (1+\epsilon)w(x^*)] \leq \frac{\mathbf{E}[w(T)]}{(1+\epsilon)w(x^*)} \overset{\boxed{\mathbf{E}[w(T)] = w(x^*)}}{=} \frac{1}{1+\epsilon} \leq 1 - \frac{\epsilon}{2}.$$

Proof of $(1+\epsilon, 4\log p/\log\log p)$-approximation

- $T :=$ spanning tree obtained from x^* by randomized swap rounding.
- $1+\delta := 4\log p/\log\log p$ and we assume for simplicity $\epsilon \geq 4/p$.

Probability that budget i is violated:

Using Chernoff bound for swap rounding:

$$\Pr[\ell_i(T) > (1+\delta)B_i] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{B_i} \overset{\boxed{B_i \geq 1}}{\leq} \frac{e^\delta}{(1+\delta)^{1+\delta}} \leq \left(\frac{e}{1+\delta}\right)^{1+\delta}$$

$$= \left(\frac{e\log\log p}{4\log p}\right)^{\frac{4\log p}{\log\log p}} \leq \left(\frac{1}{\sqrt{\log p}}\right)^{\frac{4\log p}{\log\log p}} = \frac{1}{p^2}.$$

Probability that some budget is violated:

Using the union bound:

$$\Pr[\ell_i(T) > (1+\delta)\ell_i(x^*) \text{ for some } i] \leq p \cdot \frac{1}{p^2} = \frac{1}{p} \leq \frac{\epsilon}{4}.$$

Probability that objective function is not fine:

Using Markov's inequality:

$$\Pr[w(T) > (1+\epsilon)w(x^*)] \leq \frac{\mathbf{E}[w(T)]}{(1+\epsilon)w(x^*)} \overset{\boxed{\mathbf{E}[w(T)] = w(x^*)}}{=} \frac{1}{1+\epsilon} \leq 1 - \frac{\epsilon}{2}.$$

Proof of $(1+\epsilon, 4\log p/\log\log p)$-approximation

- $T :=$ spanning tree obtained from x^* by randomized swap rounding.
- $1+\delta := 4\log p/\log\log p$ and we assume for simplicity $\epsilon \geq 4/p$.

Probability that some budget is violated:
Using the union bound:

$$\Pr[\ell_i(T) > (1+\delta)\ell_i(x^*) \text{ for some } i] \leq p \cdot \frac{1}{p^2} = \frac{1}{p} \leq \frac{\epsilon}{4}.$$

Probability that objective function is not fine:
Using Markov's inequality:

$$\Pr[w(T) > (1+\epsilon)w(x^*)] \leq \frac{\mathbf{E}[w(T)]}{(1+\epsilon)w(x^*)} \overset{\boxed{\mathbf{E}[w(T)] = w(x^*)}}{=} \frac{1}{1+\epsilon} \leq 1 - \frac{\epsilon}{2}.$$

Probability that objective function or some budget is not ok:
Using union bound on the two probabilities above:

$$\Pr[T \text{ not fine}] \leq 1 - \frac{\epsilon}{2} + \frac{\epsilon}{4} = 1 - \frac{\epsilon}{4}.$$

- With probability at least $\frac{\epsilon}{4}$, T is a $(1+\epsilon, 4\log p/\log\log p)$-approximation.
- In expectation, $O(1/\epsilon)$ iterations suffice to find a good spanning tree.

Proof of $(1+\epsilon, 4\log p/\log\log p)$-approximation

- $T :=$ spanning tree obtained from x^* by randomized swap rounding.
- $1+\delta := 4\log p/\log\log p$ and we assume for simplicity $\epsilon \geq 4/p$.

Probability that some budget is violated:
Using the union bound:

$$\Pr[\ell_i(T) > (1+\delta)\ell_i(x^*) \text{ for some } i] \leq p \cdot \frac{1}{p^2} = \frac{1}{p} \leq \frac{\epsilon}{4}.$$

Probability that objective function is not fine:
Using Markov's inequality:

$$\Pr[w(T) > (1+\epsilon)w(x^*)] \leq \frac{\mathbf{E}[w(T)]}{(1+\epsilon)w(x^*)} \overset{\boxed{\mathbf{E}[w(T)] = w(x^*)}}{=} \frac{1}{1+\epsilon} \leq 1 - \frac{\epsilon}{2}.$$

Probability that objective function or some budget is not ok:
Using union bound on the two probabilities above:

$$\Pr[T \text{ not fine}] \leq 1 - \frac{\epsilon}{2} + \frac{\epsilon}{4} = 1 - \frac{\epsilon}{4}.$$

- With probability at least $\frac{\epsilon}{4}$, T is a $(1+\epsilon, 4\log p/\log\log p)$-approximation.
- In expectation, $O(1/\epsilon)$ iterations suffice to find a good spanning tree.

□

Proof of $(1+\epsilon, 4\log p/\log\log p)$-approximation

- $T :=$ spanning tree obtained from x^* by randomized swap rounding.
- $1+\delta := 4\log p/\log\log p$ and we assume for simplicity $\epsilon \geq 4/p$.

Probability that some budget is violated:
Using the union bound:

$$\Pr[\ell_i(T) > (1+\delta)\ell_i(x^*) \text{ for some } i] \leq p \cdot \frac{1}{p^2} = \frac{1}{p} \leq \frac{\epsilon}{4}.$$

Probability that objective function is not fine:
Using Markov's inequality:

$$\Pr[w(T) > (1+\epsilon)w(x^*)] \leq \frac{\mathbf{E}[w(T)]}{(1+\epsilon)w(x^*)} \overset{\boxed{\mathbf{E}[w(T)] = w(x^*)}}{=} \frac{1}{1+\epsilon} \leq 1 - \frac{\epsilon}{2}.$$

Probability that objective function or some budget is not ok:
Using union bound on the two probabilities above:

$$\Pr[T \text{ not fine}] \leq 1 - \frac{\epsilon}{2} + \frac{\epsilon}{4} = 1 - \frac{\epsilon}{4}.$$

- With probability at least $\frac{\epsilon}{4}$, T is a $(1+\epsilon, 4\log p/\log\log p)$-approximation.
- In expectation, $O(1/\epsilon)$ iterations suffice to find a good spanning tree.

□

A short comment about submodular functions

Definition
Let S be a finite set. A function $f : 2^S \to \mathbb{R}$ is submodular if

$$f(A \cup \{s\}) - f(A) \geq f(B \cup \{s\}) - f(B) \quad \forall A \subseteq B \subseteq S, s \in S.$$

- Submodular functions are interesting candidates for utility functions since they can model diminishing returns.
- Since submodular functions are only defined on a discrete set, an extension is typically used in LP relaxations. A useful candidate is the multilinear extension F defined for $x \in [0,1]^E$ by

$$F(x) = \sum_{R \subseteq S} \left(\prod_{i \in R} x_i \right) \left(\prod_{i \notin R} (1 - x_i) \right).$$

- The upper tail concentration bounds of randomized swap rounding also hold for the above submodular extension.

A short comment about submodular functions

Definition
Let S be a finite set. A function $f : 2^S \to \mathbb{R}$ is submodular if

$$f(A \cup \{s\}) - f(A) \geq f(B \cup \{s\}) - f(B) \quad \forall A \subseteq B \subseteq S, s \in S.$$

- Submodular functions are interesting candidates for utility functions since they can model diminishing returns.
- Since submodular functions are only defined on a discrete set, an extension is typically used in LP relaxations. A useful candidate is the multilinear extension F defined for $x \in [0,1]^E$ by

$$F(x) = \sum_{R \subseteq S} \left(\prod_{i \in R} x_i \right) \left(\prod_{i \notin R} (1 - x_i) \right).$$

- The upper tail concentration bounds of randomized swap rounding also hold for the above submodular extension.

A short comment about submodular functions

Definition

Let S be a finite set. A function $f : 2^S \to \mathbb{R}$ is submodular if

$$f(A \cup \{s\}) - f(A) \geq f(B \cup \{s\}) - f(B) \quad \forall A \subseteq B \subseteq S, s \in S.$$

- Submodular functions are interesting candidates for utility functions since they can model diminishing returns.
- Since submodular functions are only defined on a discrete set, an extension is typically used in LP relaxations. A useful candidate is the multilinear extension F defined for $x \in [0,1]^E$ by

$$F(x) = \sum_{R \subseteq S} \left(\prod_{i \in R} x_i \right) \left(\prod_{i \notin R} (1 - x_i) \right).$$

- The upper tail concentration bounds of randomized swap rounding also hold for the above submodular extension.

A short comment about submodular functions

Definition

Let S be a finite set. A function $f : 2^S \to \mathbb{R}$ is submodular if

$$f(A \cup \{s\}) - f(A) \geq f(B \cup \{s\}) - f(B) \quad \forall A \subseteq B \subseteq S, s \in S.$$

- Submodular functions are interesting candidates for utility functions since they can model diminishing returns.
- Since submodular functions are only defined on a discrete set, an extension is typically used in LP relaxations. A useful candidate is the multilinear extension F defined for $x \in [0,1]^E$ by

$$F(x) = \sum_{R \subseteq S} \left(\prod_{i \in R} x_i \right) \left(\prod_{i \notin R} (1 - x_i) \right).$$

- The upper tail concentration bounds of randomized swap rounding also hold for the above submodular extension.

Outline

Conclusions

- Randomized rounding is a powerful tool in many settings.
- Randomized swap rounding allows to profit from combinatorial knowledge of the underlying problem by
  - **i)** Representing the point to round as a convex combination of vertices of the underlying polytope.
  - **ii)** Applying merging steps on the terms in the convex combination.
- The spanning tree polytope (or more generally matroid polytopes) have nice combinatorial properties for applying rounding procedures.

- In which other settings is the general swap rounding framework useful? (So far, we have some results on matroid intersection and b-matchings)
- Can the approach be derandomized in some non-trivial settings?

References

A. Panconesi and A. Srinivasan. Randomized distributed edge coloring via an extension of the chernoff–hoeffding bounds. *SIAM Journal on Computing*, 26(2):350–368, 1997. ISSN 0097-5397. doi: http://dx.doi.org/10.1137/S0097539793250767.