Randomized rounding in matroid polytopes

Chandra Chekuri[1] Jan Vondrák[2] Rico Zenklusen[3]

[1]U Illinois, Urbana-Champaign

[2]IBM Almaden Research Center

[3]ETH Zurich

April 13, 2010

6 problems

What do the following problems have in common?

- **Degree-bounded Spanning Tree**
- **Geometric Stabbing Spanning Tree**
- **Welfare Maximization in Combinatorial Auctions**
- **Max-Min Allocation Problem**
- **Generalized Assignment Problem**
- **Asymmetric Traveling Salesman Problem**

Matroids

Definition

A matroid on N is defined by a system of *bases* $\mathcal{B} \subset 2^N$, satisfying

1. $\forall B_1, B_2 \in \mathcal{B}; |B_1| = |B_2|$.
2. $\forall B_1, B_2 \in \mathcal{B}; \forall x \in B_1; \exists y \in B_2;$
$$B_1 - x + y \in \mathcal{B}, \quad B_2 - y + x \in \mathcal{B}.$$

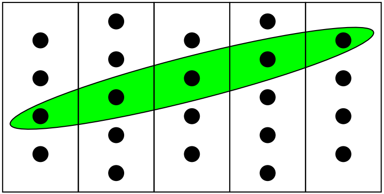
Matroids

Definition

A matroid on N is defined by a system of *bases* $\mathcal{B} \subset 2^N$, satisfying

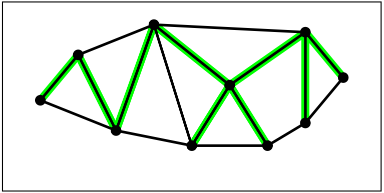
1. $\forall B_1, B_2 \in \mathcal{B}; |B_1| = |B_2|$.
2. $\forall B_1, B_2 \in \mathcal{B}; \forall x \in B_1; \exists y \in B_2;$
$$B_1 - x + y \in \mathcal{B}, \quad B_2 - y + x \in \mathcal{B}.$$

Examples:



partition matroid

(bases = transversals)



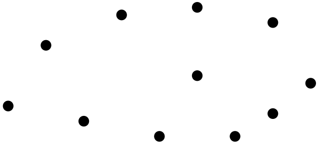
graphic matroid

(bases = spanning trees)

Minimum Stabbing Spanning Tree

Given n points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, find a spanning tree T such that no hyperplane cuts too many edges of T.

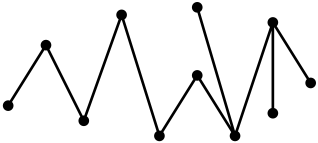
Def.: *stabbing number* $\sigma(T)$ = maximum number of edges of T intersected by any hyperplane.



Minimum Stabbing Spanning Tree

Given n points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, find a spanning tree T such that no hyperplane cuts too many edges of T.

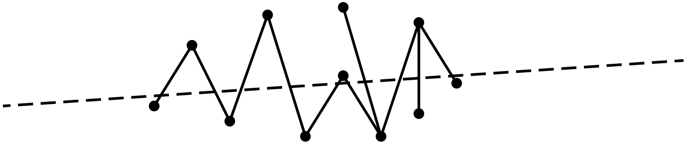
Def.: *stabbing number* $\sigma(T)$ = maximum number of edges of T intersected by any hyperplane.



Minimum Stabbing Spanning Tree

Given n points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, find a spanning tree T such that no hyperplane cuts too many edges of T.

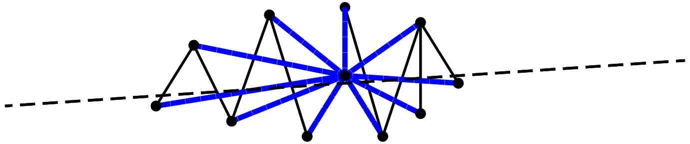
Def.: *stabbing number* $\sigma(T) =$ maximum number of edges of T intersected by any hyperplane.



Minimum Stabbing Spanning Tree

Given n points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, find a spanning tree T such that no hyperplane cuts too many edges of T.

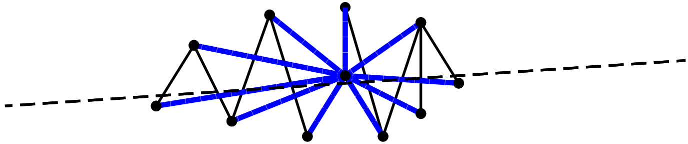
Def.: *stabbing number* $\sigma(T)$ = maximum number of edges of T intersected by any hyperplane.



Minimum Stabbing Spanning Tree

Given n points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, find a spanning tree T such that no hyperplane cuts too many edges of T.

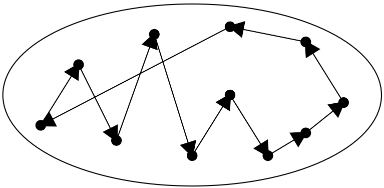
Def.: *stabbing number* $\sigma(T) =$ maximum number of edges of T intersected by any hyperplane.



Known: $O(\log n)$-approximation for finding a spanning tree minimizing the stabbing number $\sigma(T)$ [Har-Peled '08].

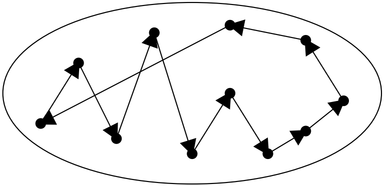
Asymmetric Traveling Salesman Problem

ATSP: Given a directed graph with edge costs c_e, find a directed circuit of minimum cost, visiting all vertices.



Asymmetric Traveling Salesman Problem

ATSP: Given a directed graph with edge costs c_e, find a directed circuit of minimum cost, visiting all vertices.



Asadpour, Goemans, Madry, Oveis-Gharan, Saberi '10:
$O(\log n / \log\log n)$-approximation, improving the previously long-standing $O(\log n)$-approximation.

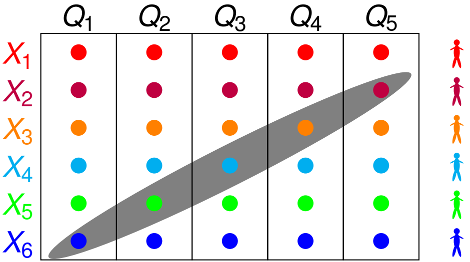
Key ingredient: Given a fractional spanning tree x, find a spanning tree T which does not contain too many edges across any cut $(S, \overline{S})$:

$$|T \cap e(S, \overline{S})| \leq \alpha x(S, \overline{S}).$$

Allocation problems

Social Welfare Maximization: *Given n agents with valuation functions $w_i : 2^X \to \mathbb{R}_+$. Allocate items to maximize $\sum_{i=1}^{n} w_i(S_i)$.*

Social Welfare Maximization: *Given n agents with valuation functions $w_i : 2^X \to \mathbb{R}_+$. Allocate items to maximize $\sum_{i=1}^n w_i(S_i)$.*



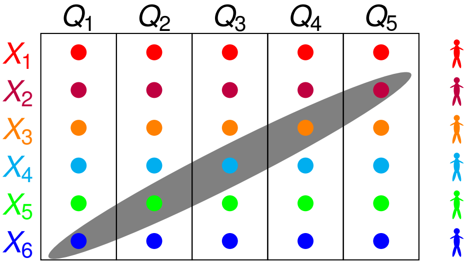
Reduction:

Create n clones of each item,
$f(S) = \sum w_i(S \cap X_i)$,
$\mathcal{B} = \{S : \forall i; |S \cap Q_i| = 1\}$
(a partition matroid).

Social Welfare Maximization is equivalent to $\max\{f(S) : S \in \mathcal{B}\}$.

Allocation problems

Social Welfare Maximization: *Given n agents with valuation functions $w_i : 2^X \to \mathbb{R}_+$. Allocate items to maximize $\sum_{i=1}^{n} w_i(S_i)$.*



Reduction:

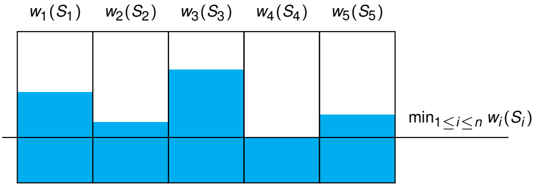
Create n clones of each item,
$f(S) = \sum w_i(S \cap X_i)$,
$\mathcal{B} = \{ S : \forall i; |S \cap Q_i| = 1 \}$
(a partition matroid).

Social Welfare Maximization is equivalent to $\max\{f(S) : S \in \mathcal{B}\}$.

Known: $(1 - 1/e)$-approximation for submodular functions [V. '08]

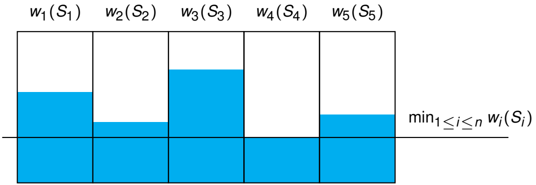
Max-min allocation

A different objective (a.k.a. Santa Claus problem):
maximize the least happy person, $\min_{1 \leq i \leq n} w_i(S_i)$.



Max-min allocation

A different objective (a.k.a. Santa Claus problem):
maximize the least happy person, $\min_{1 \le i \le n} w_i(S_i)$.



- much harder to approximate, even for linear valuation functions
- for constant n, there is an FPTAS [Woeginger 2000]
- for growing n, best known: n^ϵ-approximation in $O(n^{1/\epsilon})$ time [Chakrabarty, Chuzhoy, Khanna '09]
- for submodular valuations, $(2n-1)$-approximation [Khot, Ponnuswami '07]

Constrained matroid-base problems

General framework:
Find a matroid base $B \in \mathcal{B}$ under some additional constraints.

Constrained matroid-base problems

General framework:
Find a matroid base $B \in \mathcal{B}$ under some additional constraints.

Standard approach:

- Let $B(\mathcal{M}) = \text{conv}\{\mathbf{1}_B : B \in \mathcal{B}\}$ denote the *matroid base polytope*.

Constrained matroid-base problems

General framework:
Find a matroid base $B \in \mathcal{B}$ under some additional constraints.

Standard approach:

- Let $B(\mathcal{M}) = \text{conv}\{\mathbf{1}_B : B \in \mathcal{B}\}$ denote the *matroid base polytope*.
- Replace the problem by an LP relaxation:
$$\min\{c^T x : x \in B(\mathcal{M}), Ax \leq b\}.$$

Constrained matroid-base problems

General framework:
Find a matroid base $B \in \mathcal{B}$ under some additional constraints.

Standard approach:

- Let $B(\mathcal{M}) = \mathrm{conv}\{\mathbf{1}_B : B \in \mathcal{B}\}$ denote the *matroid base polytope*.
- Replace the problem by an LP relaxation:
$$\min\{c^T x : x \in B(\mathcal{M}), Ax \leq b\}.$$
- For a submodular objective, consider the *multilinear relaxation*:
$$\max\{F(x) : x \in B(\mathcal{M}), Ax \leq b\}.$$

Constrained matroid-base problems

General framework:
Find a matroid base $B \in \mathcal{B}$ under some additional constraints.

Standard approach:

- Let $B(\mathcal{M}) = \text{conv}\{\mathbf{1}_B : B \in \mathcal{B}\}$ denote the *matroid base polytope*.
- Replace the problem by an LP relaxation:
$$\min\{c^T x : x \in B(\mathcal{M}), Ax \leq b\}.$$
- For a submodular objective, consider the *multilinear relaxation*:
$$\max\{F(x) : x \in B(\mathcal{M}), Ax \leq b\}.$$
- Main issue in this talk: How to round a fractional solution?

Randomized rounding

Raghavan-Thompson '87: **randomized rounding.**

$$\max\{\sum_i c_i x_i : Ax \leq b; x \geq 0\}$$

- Take a fractional solution and round $x_i \rightarrow \{0, 1\}$ randomly, *without looking at the constraints*.

Randomized rounding

Raghavan-Thompson '87: **randomized rounding.**

$$\max\{\sum_i c_i x_i : Ax \leq b; x \geq 0\}$$

- Take a fractional solution and round $x_i \to \{0, 1\}$ randomly, *without looking at the constraints*.
- Maybe we violate some constraints, but not too much (due to Chernoff bounds).

Randomized rounding

Raghavan-Thompson '87: **randomized rounding.**

$$\max\{\sum_i c_i x_i : Ax \leq b; x \geq 0\}$$

- Take a fractional solution and round $x_i \rightarrow \{0, 1\}$ randomly, *without looking at the constraints*.
- Maybe we violate some constraints, but not too much (due to Chernoff bounds).

However, what if we want to satisfy some constraints *exactly*?

$$\max\{\sum_i c_i x_i : Ax \leq b; x \geq 0; \sum x_i = k\}$$

Randomized rounding in matroid polytopes

Goal: Given a fractional solution $x \in B(\mathcal{M})$,
produce a random base $R \in \mathcal{B}$ such that

1. $\forall i; \Pr[i \in R] = x_i$
2. Linear functions $a(R) = \sum_{i \in R} a_i$ are concentrated around $\sum a_i x_i$.

Randomized rounding in matroid polytopes

Goal: Given a fractional solution $x \in B(\mathcal{M})$, produce a random base $R \in \mathcal{B}$ such that

1. $\forall i; \Pr[i \in R] = x_i$
2. Linear functions $a(R) = \sum_{i \in R} a_i$ are concentrated around $\sum a_i x_i$.

Note: we cannot sample R independently, because we need to satisfy $R \in \mathcal{B}$.
However, it is known that Chernoff-type bounds hold under relaxed conditions, e.g. *negative correlation* [Panconesi, Srinivasan '97]

Randomized rounding in matroid polytopes

Goal: Given a fractional solution $x \in B(\mathcal{M})$, produce a random base $R \in \mathcal{B}$ such that

1. $\forall i; \Pr[i \in R] = x_i$
2. Linear functions $a(R) = \sum_{i \in R} a_i$ are concentrated around $\sum a_i x_i$.

Note: we cannot sample R independently, because we need to satisfy $R \in \mathcal{B}$. However, it is known that Chernoff-type bounds hold under relaxed conditions, e.g. *negative correlation* [Panconesi, Srinivasan '97]

Previously known techniques:

- *Dependent randomized rounding:* under the constraint $\sum_{i=1}^{n} x_i = k$, with negative correlation properties [Srinivasan '01]

Randomized rounding in matroid polytopes

Goal: Given a fractional solution $x \in B(\mathcal{M})$, produce a random base $R \in \mathcal{B}$ such that

1. $\forall i; \Pr[i \in R] = x_i$
2. Linear functions $a(R) = \sum_{i \in R} a_i$ are concentrated around $\sum a_i x_i$.

Note: we cannot sample R independently, because we need to satisfy $R \in \mathcal{B}$. However, it is known that Chernoff-type bounds hold under relaxed conditions, e.g. *negative correlation* [Panconesi, Srinivasan '97]

Previously known techniques:

- *Dependent randomized rounding:* under the constraint $\sum_{i=1}^{n} x_i = k$, with negative correlation properties [Srinivasan '01]
- *Randomized pipage rounding:* for x in the bipartite matching polytope, concentration for special subsets [Gandhi et al. '06]

Randomized rounding in matroid polytopes

Goal: Given a fractional solution $x \in B(\mathcal{M})$,
produce a random base $R \in \mathcal{B}$ such that

1. $\forall i; \Pr[i \in R] = x_i$
2. Linear functions $a(R) = \sum_{i \in R} a_i$ are concentrated around $\sum a_i x_i$.

Note: we cannot sample R independently, because we need to satisfy $R \in \mathcal{B}$.
However, it is known that Chernoff-type bounds hold under relaxed conditions, e.g. *negative correlation* [Panconesi, Srinivasan '97]

Previously known techniques:

- *Dependent randomized rounding:* under the constraint $\sum_{i=1}^{n} x_i = k$, with negative correlation properties [Srinivasan '01]
- *Randomized pipage rounding:* for x in the bipartite matching polytope, concentration for special subsets [Gandhi et al. '06]
- *Maximum entropy sampling:* for x in the spanning tree polytope, concentration for arbitrary subsets [Asadpour et al. '09]

Our work [Chekuri, V., Zenklusen '09]

We present a very simple rounding technique such that:

- Given $x \in B(\mathcal{M})$, it produces a random matroid base $R \in \mathcal{B}$.

Our work [Chekuri, V., Zenklusen '09]

We present a very simple rounding technique such that:

- Given $x \in B(\mathcal{M})$, it produces a random matroid base $R \in \mathcal{B}$.
- Marginals are preserved: $\Pr[i \in R] = x_i$ for all i.
- The events $i \in R$ are negatively correlated
  $\longrightarrow$ concentration for linear functions $a(R)$.
- We also prove a Chernoff-type tail estimate
  for *monotone submodular* functions $f(R)$.

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.

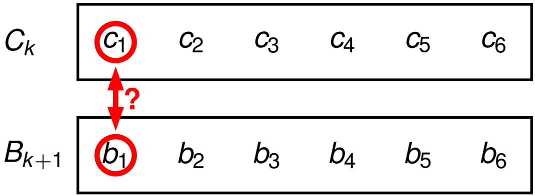
C_k	c_1	c_2	c_3	c_4	c_5	c_6
B_{k+1}	b_1	b_2	b_3	b_4	b_5	b_6

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n - 1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.



Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}$, $B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.

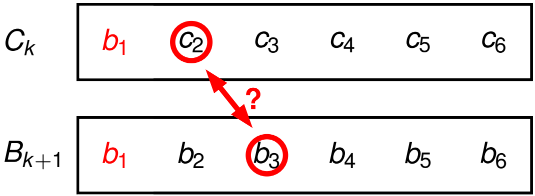
C_k	b_1	c_2	c_3	c_4	c_5	c_6
B_{k+1}	b_1	b_2	b_3	b_4	b_5	b_6

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}$, $B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.



Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}$, $B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.

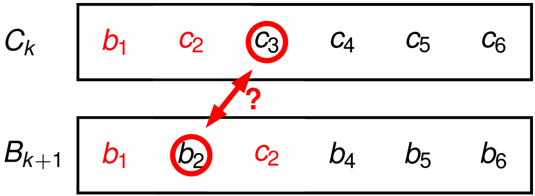
C_k	b_1	c_2	c_3	c_4	c_5	c_6
B_{k+1}	b_1	b_2	c_2	b_4	b_5	b_6

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.



Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.

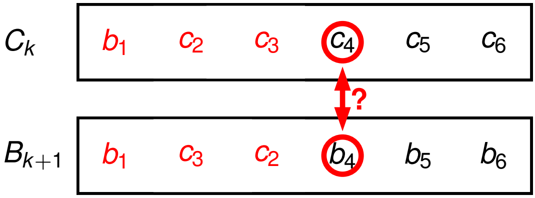
C_k	b_1	c_2	c_3	c_4	c_5	c_6
B_{k+1}	b_1	c_3	c_2	b_4	b_5	b_6

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n - 1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.



Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.

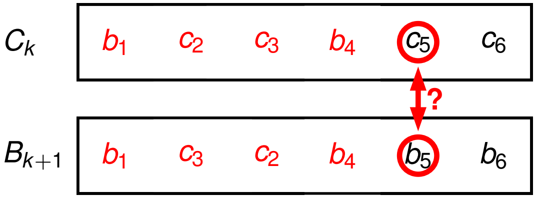
C_k	b_1	c_2	c_3	b_4	c_5	c_6
B_{k+1}	b_1	c_3	c_2	b_4	b_5	b_6

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.



Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.

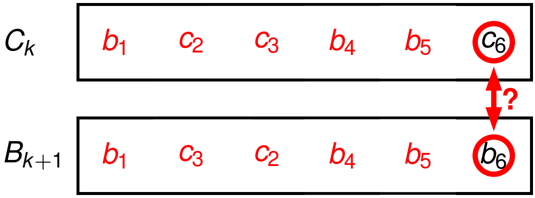
C_k	b_1	c_2	c_3	b_4	b_5	c_6
B_{k+1}	b_1	c_3	c_2	b_4	b_5	b_6

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.



Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n - 1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.

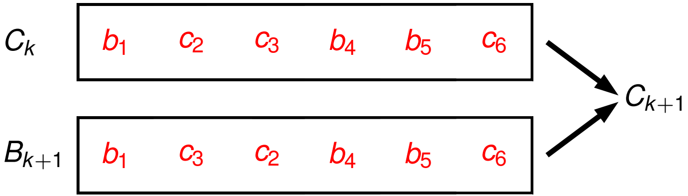
C_k	b_1	c_2	c_3	b_4	b_5	c_6
B_{k+1}	b_1	c_3	c_2	b_4	b_5	c_6

Randomized swap rounding

Randomized swap rounding:

1. Start with an arbitrary convex combination of bases $\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{1}_{B_i}, \quad B_i \in \mathcal{B}$.
2. In the first stage, merge B_1 and B_2 into $C_2 \in \mathcal{B}$.
3. In the k-th stage, merge C_k and B_{k+1} into $C_{k+1} \in \mathcal{B}$.
4. After $n-1$ stages, return $C_n \in \mathcal{B}$.

How do we merge two bases? randomly, using the strong exchange property: $\forall x \in C_k; \exists y \in B_{k+1}; C_k - x + y \in \mathcal{B}$ & $B_{k+1} - y + x \in \mathcal{B}$.



Properties of randomized swap rounding

Theorem (negative correlation)

Let X_i be the indicator variables for the events $i \in R$, where $R \in \mathcal{B}$ is obtained by randomized swap rounding from $x \in B(\mathcal{M})$. Then $\mathbb{E}[X_i] = x_i$ and for any $J \subseteq [n]$,

1. $\mathbb{E}[\prod_{j \in J} X_i] \leq \prod_{j \in J} x_i$.
2. $\mathbb{E}[\prod_{j \in J}(1 - X_i)] \leq \prod_{j \in J}(1 - x_i)$.

Properties of randomized swap rounding

Theorem (negative correlation)

Let X_i be the indicator variables for the events $i \in R$, where $R \in \mathcal{B}$ is obtained by randomized swap rounding from $x \in B(\mathcal{M})$. Then $\mathbb{E}[X_i] = x_i$ and for any $J \subseteq [n]$,

1. $\mathbb{E}[\prod_{j \in J} X_i] \leq \prod_{j \in J} x_i$.
2. $\mathbb{E}[\prod_{j \in J}(1 - X_i)] \leq \prod_{j \in J}(1 - x_i)$.

Using [Panconesi-Srinivasan '97], this implies Chernoff-type bounds.

Corollary

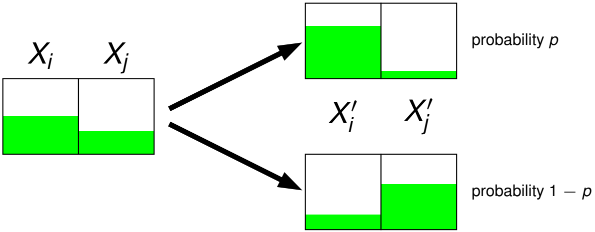
Let $a_1, \ldots, a_n \in [0,1]$ and $\mu = \sum a_i x_i$. Then for any $\delta > 0$

1. $\Pr[\sum a_i X_i < (1 - \delta)\mu] < e^{-\mu\delta^2/2}$.
2. $\Pr[\sum a_i X_i > (1 + \delta)\mu] < (e^\delta/(1+\delta)^{1+\delta})^\mu$.

Proof of negative correlation

Consider a step where X_i, X_j are modified into X_i', X_j'.

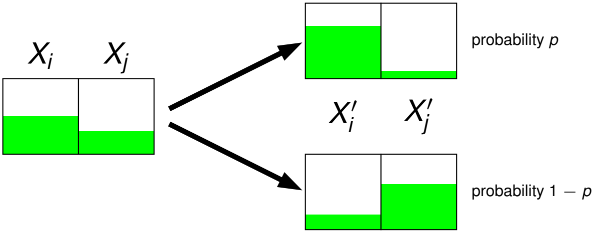
- $\mathbb{E}[X_i' \mid X_i] = X_i$.
- $X_i' + X_j' = X_i + X_j$.



Proof of negative correlation

Consider a step where X_i, X_j are modified into X_i', X_j'.

- $\mathbb{E}[X_i' \mid X_i] = X_i$.
- $X_i' + X_j' = X_i + X_j$.



$$\begin{aligned} \mathbb{E}[X_i' X_j' \mid X_i, X_j] &= \frac{1}{4}\mathbb{E}[(X_i' + X_j')^2 \mid X_i, X_j] - \frac{1}{4}\mathbb{E}[(X_i' - X_j')^2 \mid X_i, X_j] \\ &\leq \frac{1}{4}(X_i + X_j)^2 - \frac{1}{4}(X_i - X_j)^2 = X_i X_j. \end{aligned}$$

Application 1

Minimum stabbing spanning tree: Find a geometric spanning tree on $x_1, \ldots, x_n \in \mathbb{R}^d$ such that no hyperplane intersects too many edges ($d =$ constant).

Result: $O(\log n / \log \log n)$-approximation for minimizing the stabbing number $\sigma(T)$ over all spanning trees.

Application 1

Minimum stabbing spanning tree: Find a geometric spanning tree on $x_1, \ldots, x_n \in \mathbb{R}^d$ such that no hyperplane intersects too many edges ($d =$ constant).

Result: $O(\log n / \log\log n)$-approximation for minimizing the stabbing number $\sigma(T)$ over all spanning trees.

- This improves a recent $O(\log n)$-approximation [Har-Peled '08].
- We write an LP using the graphic matroid $\mathcal{M}$, including constraints for the $O(n^d)$ combinatorially different hyperplanes:

$$\min\{\sigma : \exists x \in B(\mathcal{M}), Ax \leq \sigma \mathbf{1}\}.$$

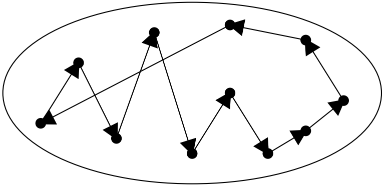
- A fractional solution is rounded using randomized swap rounding; Chernoff bounds imply that every hyperplane intersects at most $O(\frac{\log n}{\log\log n})\sigma$ edges.

Application 2

Asymmetric Traveling Salesman: an alternative (simpler) way to derive the $O(\log n / \log\log n)$-approximation of Asadpour et al.

Application 2

Asymmetric Traveling Salesman: an alternative (simpler) way to derive the $O(\log n / \log\log n)$-approximation of Asadpour et al.



- Solve the natural LP: the fractional solution x^* can be interpreted as a point in the spanning tree polytope.
- We apply randomized swap rounding to x^*, to obtain a random spanning tree T.
- Chernoff bounds imply that for any cut C, $|T \cap C| \leq \frac{\log n}{\log\log n} x^*(C)$.
- Such a spanning tree can be extended to a salesman tour of cost $O(\frac{\log n}{\log\log n})OPT$, using the techniques of Asadpour et al.

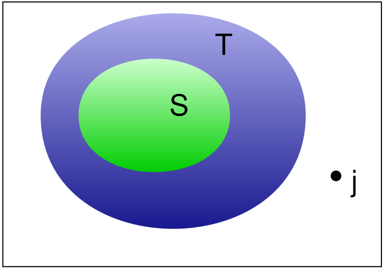
Submodular functions

- **Monotonicity:** f is monotone, if $f(S) \leq f(T)$ whenever $S \subseteq T$.

Submodular functions

- **Monotonicity:** f is monotone, if $f(S) \leq f(T)$ whenever $S \subseteq T$.
- **Submodularity:** Let the *marginal value* of element j be

$$f_S(j) = f(S+j) - f(S).$$



f is submodular, if
j adds more value to S than T:

$$f_T(j) \leq f_S(j)$$

Tail estimate for submodular functions

Theorem

Let $f : \{0,1\}^n \to \mathbb{R}_+$ be submodular with marginal values in $[0,1]$ and $X_1, \ldots, X_n \in \{0,1\}$ obtained by randomized swap rounding from $x \in B(\mathcal{M})$. Let $\delta \in [0,1]$ and $\mu = \mathbb{E}[F(x)]$ where F is the multilinear extension of f. Then $\mathbb{E}[f(X_1, \ldots, X_n)] \geq \mu$ and

- $\Pr[f(X_1, \ldots, X_n) < (1-\delta)\mu] < e^{-\mu\delta^2/8}$.

Tail estimate for submodular functions

Theorem

Let $f : \{0,1\}^n \to \mathbb{R}_+$ *be submodular with marginal values in* $[0,1]$ *and* $X_1, \ldots, X_n \in \{0,1\}$ *obtained by randomized swap rounding from* $x \in B(\mathcal{M})$. *Let* $\delta \in [0,1]$ *and* $\mu = \mathbb{E}[F(x)]$ *where* F *is the multilinear extension of* f. *Then* $\mathbb{E}[f(X_1, \ldots, X_n)] \geq \mu$ *and*

- $\Pr[f(X_1, \ldots, X_n) < (1-\delta)\mu] < e^{-\mu\delta^2/8}$.

Proof: bounding the exponential moment $\mathbb{E}[e^{\lambda(\mu - f(R))}]$ by careful aggregation of contributions from successive steps; don't know how to use negative correlation.

Note: weaker results follow directly from martingale concentration bounds; due to the length of the rounding process, it is crucial to remove dependence on n.

Application 3

Max-Min Submodular Allocation. Given n agents with submodular valuations $w_i : 2^N \to \mathbb{R}_+$, find an allocation $(S_1, \ldots, S_n)$ maximizing $\min_{1 \le i \le n} w_i(S_i)$.

Application 3

Max-Min Submodular Allocation. Given n agents with submodular valuations $w_i : 2^N \to \mathbb{R}_+$, find an allocation $(S_1, \ldots, S_n)$ maximizing $\min_{1 \le i \le n} w_i(S_i)$.

Result: A $(1 - 1/e - \epsilon)$-approximation for any constant $n \ge 2$ (or assuming that all singleton values are $O(OPT/\log n)$).

- We solve the multilinear relaxation using a multiobjective variant of the continuous greedy algorithm.
- We use randomized swap rounding to generate a random allocation.
- The lower-tail bound for submodular functions is crucial in getting a high-probability guarantee for all agents simultaneously.

Open problems

- Can similar tail estimates be obtained for non-monotone submodular functions?

Open problems

- Can similar tail estimates be obtained for non-monotone submodular functions?
- Can randomized rounding techniques with negative correlation properties be extended to other structures than matroids? [very recently: *matroid intersection*]

Open problems

- Can similar tail estimates be obtained for non-monotone submodular functions?
- Can randomized rounding techniques with negative correlation properties be extended to other structures than matroids? [very recently: *matroid intersection*]
- Can the rounding procedure be modified to be more sensitive to additional linear constraints?