Dependent Randomized Rounding via Exchange Properties of Combinatorial Structures

Rico Zenklusen
MIT

Joint work with Chandra Chekuri and Jan Vondrák
Outline

1 Introduction
   • Motivation

2 Randomized swap rounding: a new rounding framework
   • The general framework
   • Swap rounding in matroid polytopes
   • Swap rounding in the intersection of two matroids

3 Some consequences/applications

4 Conclusions
Outline

1 Introduction
   • Motivation

2 Randomized swap rounding: a new rounding framework
   • The general framework
   • Swap rounding in matroid polytopes
   • Swap rounding in the intersection of two matroids

3 Some consequences/applications

4 Conclusions
Randomized rounding
A technique to profit from relaxations of hard problems

A typical setting
\[
\begin{align*}
\text{max} / \text{min} & \quad f(x) \\
x & \in P \\
x & \in \mathcal{W} \\
x & \in \{0, 1\}^n
\end{align*}
\]

- \( P \subset [0, 1]^n \): integer polytope representing “hard” constraints.
- \( \mathcal{W} \): “weak” constraints.

The strategy
Randomly round a fractional solution \( x \) of the relaxation to \( X \in \{0, 1\}^n \) so that:

- \( X \) satisfies hard constraints: \( X \in P \),
- \( X \) is good in expectation: \( E[X] \approx x \),
- linear (and possibly other) functions \( g(X) \) concentrates around \( E[g(X)] \).
- Chernoff-type bounds \( \Rightarrow g(X) \approx g(x) \), and \( X \) is almost in \( \mathcal{W} \) whp.
Randomized rounding
A technique to profit from relaxations of hard problems

A typical setting
\[
\begin{align*}
\text{max} / \text{min} & \quad f(x) \\
 x & \in P \\
 x & \in W \\
 x & \in \{0, 1\}^n
\end{align*}
\]

- \( P \subset [0, 1]^n \): integer polytope representing “hard” constraints.
- \( W \): “weak” constraints.

The strategy
Randomly round a fractional solution \( x \) of the relaxation to \( X \in \{0, 1\}^n \) so that:
- \( X \) satisfies hard constraints: \( X \in P \),
- \( X \) is good in expectation: \( E[X] \approx x \),
- linear (and possibly other) functions \( g(X) \) concentrates around \( E[g(X)] \).
  \( \Rightarrow \) Chernoff-type bounds \( \Rightarrow g(X) \approx g(x) \), and \( X \) is almost in \( W \) whp.
Dependent rounding and negative correlations

Independent randomized rounding (Raghavan and Thompson [1987])

- \( \Pr[X_i = 1] = x_i \), (almost) independently for \( i \in [n] := 1, \ldots, n \).

- Linear functions \( g(X) \) satisfy Chernoff-type concentration bounds.
- Polytope \( P \) has to be very simple for this to work.

Dependent randomized rounding

- Typically, a rounding procedure tailored to \( P \) is needed to ensure feasibility.
  - Dependencies between different components of \( X \) are created.
- Still, Chernoff-type concentration bounds are desired.
  - They often follow from negative correlation.
Dependent rounding and negative correlations

Independent randomized rounding (Raghavan and Thompson [1987])

- $\Pr[X_i = 1] = x_i$, (almost) independently for $i \in [n] := 1, \ldots, n$.

- Linear functions $g(X)$ satisfy Chernoff-type concentration bounds.

- Polytope $P$ has to be very simple for this to work.

Dependent randomized rounding

- Typically, a rounding procedure tailored to $P$ is needed to ensure feasibility.
  - Dependencies between different components of $X$ are created.

- Still, Chernoff-type concentration bounds are desired.
  - They often follow from negative correlation.
Concentration through negative correlation

Obtaining Chernoff bounds without independence

**Theorem (Panconesi and Srinivasan [1997])**

Let $X \in \{0, 1\}^n$ be a random vector with $\mathbb{E}[X] = x$. If for any $S \subseteq [n]$

\[
\begin{align*}
\Pr[\bigwedge_{i \in S}(X_i = 1)] &\leq \prod_{i \in S} x_i, \\
\Pr[\bigwedge_{i \in S}(X_i = 0)] &\leq \prod_{i \in S} (1 - x_i),
\end{align*}
\]

then for $a \in [0, 1]^n$,

\[
\begin{align*}
\Pr [a^T X \geq \mu(1 + \delta)] &\leq \left( \frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^\mu \quad \text{for } \delta \geq 0, \mu \geq \mathbb{E}[a^T X] \\
\Pr [a^T X \leq \mu(1 - \delta)] &\leq e^{-\mu \delta^2 / 2} \quad \text{for } \delta \in [0, 1], \mu = \mathbb{E}[a^T X]
\end{align*}
\]

Recipe for creating dependent randomized rounding procedures

Round given point $x \in P$ to random integral vector $X \in P$ such that:

\[
\begin{align*}
\mathbb{E}[X] &= x, \\
\text{Coordinates of } X \text{ are negatively correlated.}
\end{align*}
\]
Concentration through negative correlation

Obtaining Chernoff bounds without independence

**Theorem (Panconesi and Srinivasan [1997])**

Let $X \in \{0, 1\}^n$ be a random vector with $E[X] = x$. If for any $S \subseteq [n]$

\[
\begin{align*}
\Pr[\bigwedge_{i \in S}(X_i = 1)] &\leq \prod_{i \in S} x_i, \\
\Pr[\bigwedge_{i \in S}(X_i = 0)] &\leq \prod_{i \in S} (1 - x_i),
\end{align*}
\]

then for $a \in [0, 1]^n$,

\[
\begin{align*}
\Pr [a^T X \geq \mu (1 + \delta)] &\leq \left( \frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^\mu & \text{for } \delta \geq 0, \mu \geq E[a^T X] \\
\Pr [a^T X \leq \mu (1 - \delta)] &\leq e^{-\mu \delta^2 / 2} & \text{for } \delta \in [0, 1], \mu = E[a^T X]
\end{align*}
\]

**Recipe for creating dependent randomized rounding procedures**

Round given point $x \in P$ to random integral vector $X \in P$ such that:

\[
\begin{align*}
\boldsymbol{E}[X] &= x, \\
\text{Coordinates of } X &\text{ are negatively correlated.}
\end{align*}
\]
Examples of this approach

Rounding procedures with $\mathbb{E}[X] = x$, and negative correlation

- $P = \{x \in [0, 1]^n \mid \sum_{i=1}^n x_i = k\}$ (Srinivasan [2001])
- Assignment polytope; negative correlation only for edges adjacent to any fixed vertex (Gandhi et al. [2006]).
- Spanning tree polytope (Asadpour et al. [2010])
  → get thin spanning tree $\Rightarrow O(\log n / \log \log n)$-approximation to ATSP
Motivating questions and main results

- Which polytopes admit negatively correlated rounding procedures?
- Unifying framework?
- Concentration for non-linear/submodular functions?

We suggest a new rounding technique (randomized swap rounding)

1. For matroid polytopes:
   - \( E[X] = x \), and negative correlation holds,
   - lower-tail concentration bound for monotone submodular functions (using martingale argument).

2. For the intersection of two matroids:
   - \( E[X] = x \), and negative correlation for “equivalent elements” (generalization of stated result on assignment polytope).

- Polytopes admitting negatively correlated rounding procedures are exactly axis-parallel projections of base polytopes of matroids.
Motivating questions and main results

- Which polytopes admit negatively correlated rounding procedures?
- Unifying framework?
- Concentration for non-linear/submodular functions?

We suggest a new rounding technique (randomized swap rounding)

1. For matroid polytopes:
   - $\mathbb{E}[X] = x$, and negative correlation holds,
   - lower-tail concentration bound for monotone submodular functions (using martingale argument).

2. For the intersection of two matroids:
   - $\mathbb{E}[X] = x$, and negative correlation for “equivalent elements” (generalization of stated result on assignment polytope).

- Polytopes admitting negatively correlated rounding procedures are exactly axis-parallel projections of base polytopes of matroids.
Outline

1 Introduction
   • Motivation

2 Randomized swap rounding: a new rounding framework
   • The general framework
   • Swap rounding in matroid polytopes
   • Swap rounding in the intersection of two matroids

3 Some consequences/applications

4 Conclusions
General rounding framework

Some terminology (to highlight underlying combinatorial problem):

- \( S \): finite ground set,
- \( \mathcal{I} \subseteq 2^S \): solution set \( \rightarrow P = \text{conv}(\{1_I \mid I \in \mathcal{I}\}) \)

1. Compute convex decomposition of \( x = \sum_{i=1}^{m} \beta_i 1_{I_i} \), with \( I_1, \ldots, I_m \in \mathcal{I} \).

2. We iteratively merge the sets \( I_1, \ldots, I_m \) to a single set \( R \in \mathcal{I} \).

\[
\begin{align*}
  x_1 &= \beta_1 1_{I_1} + \beta_2 1_{I_2} + \beta_3 1_{I_3} + \ldots + \beta_m 1_{I_m} \\
  x_2 &= (\beta_1 + \beta_2) 1_{I_{1:2}} + \beta_3 1_{I_3} + \ldots + \beta_m 1_{I_m} \\
  x_3 &= (\beta_1 + \beta_2 + \beta_3) 1_{I_{1:3}} + \ldots + \beta_m 1_{I_m} \\
  &\vdots \\
  x_m &= (\beta_1 + \cdots + \beta_m) 1_{I_{1:m}} = 1_{I_{1:m}}
\end{align*}
\]

- \( I_{1:2} = \text{Merge}(\beta_1, I_1, \beta_2, I_2) \)
- \( I_{1:3} = \text{Merge}(\beta_1 + \beta_2, I_{1:2}, \beta_3, I_3) \)
General rounding framework

Some terminology (to highlight underlying combinatorial problem):

- $S$: finite ground set, $\mathcal{I} \subseteq 2^S$: solution set $\rightarrow P = \text{conv}(\{1_I \mid I \in \mathcal{I}\})$

1. Compute convex decomposition of $x = \sum_{i=1}^{m} \beta_i 1_{l_i}$, with $l_1, \ldots, l_m \in \mathcal{I}$.
2. We iteratively merge the sets $l_1, \ldots, l_m$ to a single set $R \in \mathcal{I}$.

\[
\begin{align*}
x_1 &= \beta_1 1_{l_1} + \beta_2 1_{l_2} + \beta_3 1_{l_3} + \ldots + \beta_m 1_{l_m} \\
x_2 &= (\beta_1 + \beta_2) 1_{l_{1:2}} + \beta_3 1_{l_3} + \ldots + \beta_m 1_{l_m} \\
x_3 &= (\beta_1 + \beta_2 + \beta_3) 1_{l_{1:3}} + \ldots + \beta_m 1_{l_m} \\
&\vdots \\
x_m &= (\beta_1 + \cdots + \beta_m) 1_{l_{1:m}} = 1_{l_{1:m}}
\end{align*}
\]

$l_{1:2} = \text{Merge}(\beta_1, l_1, \beta_2, l_2)$

$l_{1:3} = \text{Merge}(\beta_1 + \beta_2, l_{1:2}, \beta_3, l_3)$
Matroids

Definition: matroid $M = (S, \mathcal{I})$

$S$: finite ground set, $\emptyset \subsetneq \mathcal{I} \subseteq 2^S$: \textit{independent sets} satisfying

- $\forall I \in \mathcal{I}, J \subseteq I \Rightarrow J \in \mathcal{I}$,
- $\forall I, J \in \mathcal{I}, |I| > |J| \Rightarrow \exists i \in I \setminus J$ with $J \cup \{i\} \in \mathcal{I}$.

The set of bases $B$ are all maximal independent sets.

Example: graphic matroid $M = (E, \mathcal{I})$
- $G = (V, E)$: undirected graph
- $\mathcal{I} = \{F \subseteq E \mid F$ is a forest$\}$

Example: laminar matroid $M = (S, \mathcal{I})$
- $\mathcal{I} = \{I \subseteq S \mid |I \cap L_i| \leq k_i \ \forall i \in [m]\}$, where $L_1, \ldots, L_m \subseteq S$ is laminar.

Strong exchange property

$\forall B_1, B_2 \in \mathcal{B}, i \in B_1 \Rightarrow \exists j \in B_2$ with $B_1 - i + j \in \mathcal{B}$ and $B_2 - j + i \in \mathcal{B}$.
Matroids

**Definition: matroid** $M = (S, \mathcal{I})$

$S$: finite ground set, $\emptyset \subsetneq \mathcal{I} \subseteq 2^S$: independent sets satisfying

- $\forall I \in \mathcal{I}, J \subseteq I \Rightarrow J \in \mathcal{I}$,
- $\forall I, J \in \mathcal{I}, |I| > |J| \Rightarrow \exists i \in I \setminus J$ with $J \cup \{i\} \in \mathcal{I}$.

The set of bases $\mathcal{B}$ are all maximal independent sets.

**Example: graphic matroid** $M = (E, \mathcal{I})$

- $G = (V, E)$: undirected graph
- $\mathcal{I} = \{F \subseteq E \mid F$ is a forest$\}$

**Example: laminar matroid** $M = (S, \mathcal{I})$

- $\mathcal{I} = \{I \subseteq S \mid |I \cap L_i| \leq k_i \forall i \in [m]\}$, where $L_1, \ldots, L_m \subseteq S$ is laminar.

**Strong exchange property**

$\forall B_1, B_2 \in \mathcal{B}, i \in B_1 \Rightarrow \exists j \in B_2$ with $B_1 - i + j \in \mathcal{B}$ and $B_2 - j + i \in \mathcal{B}$.
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do

Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that

$B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;

With probability $\beta_1/(\beta_1 + \beta_2)$, \{$B_2 \leftarrow B_2 - f + e$\};

Else \{$B_1 \leftarrow B_1 - e + f$\};

EndWhile

Output $B_1$. 
Merging for matroid polytopes

**Algorithm Merge**\((\beta_1, B_1, \beta_2, B_2)\)

While \((B_1 \neq B_2)\) do

Pick \(e \in B_1 \setminus B_2\) and find \(f \in B_2 \setminus B_1\) such that \(B_1 - e + f \in \mathcal{B}\) and \(B_2 - f + e \in \mathcal{B}\);

With probability \(\beta_1/(\beta_1 + \beta_2)\), \(\{B_2 \leftarrow B_2 - f + e\}\);

Else \(\{B_1 \leftarrow B_1 - e + f\}\);

EndWhile

Output \(B_1\).
Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do
    Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that
    $B_1 - e + f \in B$ and $B_2 - f + e \in B$;
    With probability $\beta_1/\left(\beta_1 + \beta_2\right)$, 
    $\{B_2 \leftarrow B_2 - f + e\}$;
    Else $\{B_1 \leftarrow B_1 - e + f\}$;

EndWhile

Output $B_1$. 

Merging for matroid polytopes

![Diagram showing the merging process](image-url)
Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do

Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that

$B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;

With probability $\beta_1 / (\beta_1 + \beta_2)$, 

\{ $B_2 \leftarrow B_2 - f + e$ \};

Else 

\{ $B_1 \leftarrow B_1 - e + f$ \};

EndWhile

Output $B_1$. 
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do
    Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that $B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;
    With probability $\beta_1/(\beta_1 + \beta_2)$, $\{B_2 \leftarrow B_2 - f + e\}$;
    Else $\{B_1 \leftarrow B_1 - e + f\}$;
EndWhile
Output $B_1$. 
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do

Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that $B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;

With probability $\beta_1/(\beta_1 + \beta_2)$, \{ $B_2 \leftarrow B_2 - f + e$ \};

Else \{ $B_1 \leftarrow B_1 - e + f$ \};

EndWhile

Output $B_1$. 

---

[Diagram: A geometric representation of the merging process, showing the sets $B_1$, $B_2$, and the elements $e$ and $f$.]
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do
  Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that
  $B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;
  With probability $\beta_1 / (\beta_1 + \beta_2)$, $\{B_2 \leftarrow B_2 - f + e\}$;
  Else $\{B_1 \leftarrow B_1 - e + f\}$;
EndWhile
Output $B_1$. 

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{algorithm_merge}
\caption{Illustration of Algorithm Merge}
\end{figure}
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do

Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that
$B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;

With probability $\frac{\beta_1}{\beta_1 + \beta_2}$, \{ $B_2 \leftarrow B_2 - f + e$ \};
Else \{ $B_1 \leftarrow B_1 - e + f$ \};

EndWhile

Output $B_1$. 

\begin{figure}
\centering
\includegraphics[width=\textwidth]{merging.png}
\caption{Algorithm for merging matroid polytopes.}
\end{figure}
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do
    Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that $B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;
    With probability $\beta_1/(\beta_1 + \beta_2)$, \{ $B_2 \leftarrow B_2 - f + e$ \};
    Else \{ $B_1 \leftarrow B_1 - e + f$ \};
EndWhile
Output $B_1$. 
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do

Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that $B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;

With probability $\beta_1/(\beta_1 + \beta_2)$, \{ $B_2 \leftarrow B_2 - f + e$ \};

Else \{ $B_1 \leftarrow B_1 - e + f$ \};

EndWhile

Output $B_1$. 

\begin{center}
\begin{tikzpicture}

% Diagram code here...
\end{tikzpicture}
\end{center}
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do

Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that

$B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;

With probability $\beta_1/(\beta_1 + \beta_2)$, \{ $B_2 \leftarrow B_2 - f + e$ \};

Else \{ $B_1 \leftarrow B_1 - e + f$ \};

EndWhile

Output $B_1$. 

[Diagram of merging of two matroid polytopes $B_1$ and $B_2$]
Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do
    Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that
    \[ B_1 - e + f \in \mathcal{B} \text{ and } B_2 - f + e \in \mathcal{B}; \]
    With probability $\beta_1/ (\beta_1 + \beta_2)$, \{ $B_2 \leftarrow B_2 - f + e$ \};
    Else \{ $B_1 \leftarrow B_1 - e + f$ \};

EndWhile

Output $B_1$. 
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do
    Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that
    $B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;
    With probability $\beta_1/(\beta_1 + \beta_2)$, 
    $\{ B_2 \leftarrow B_2 - f + e \}$;
    Else 
    $\{ B_1 \leftarrow B_1 - e + f \}$;
EndWhile
Output $B_1$. 

\[
\begin{array}{c}
\text{Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)} \\
\text{While ($B_1 \neq B_2$) do} \\
\text{Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that} \\
B_1 - e + f \in \mathcal{B} \text{ and } B_2 - f + e \in \mathcal{B}; \\
\text{With probability $\beta_1/(\beta_1 + \beta_2)$, } \\
\{ B_2 \leftarrow B_2 - f + e \}; \\
\text{Else } \\
\{ B_1 \leftarrow B_1 - e + f \}; \\
\text{EndWhile} \\
\text{Output $B_1$.}
\end{array}
\]
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do
  Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that $B_1 - e + f \in B$ and $B_2 - f + e \in B$;
  With probability $\beta_1/(\beta_1 + \beta_2)$, $\{B_2 \leftarrow B_2 - f + e\}$; Else $\{B_1 \leftarrow B_1 - e + f\}$;
EndWhile
Output $B_1$. 

\[\begin{align*}
B_1 & \quad \text{\footnotesize (Blue)} \\
B_2 & \quad \text{\footnotesize (Purple)} \\
\end{align*}\]
Merging for matroid polytopes

Algorithm Merge($\beta_1, B_1, \beta_2, B_2$)

While ($B_1 \neq B_2$) do
  Pick $e \in B_1 \setminus B_2$ and find $f \in B_2 \setminus B_1$ such that $B_1 - e + f \in \mathcal{B}$ and $B_2 - f + e \in \mathcal{B}$;
  With probability $\frac{\beta_1}{(\beta_1 + \beta_2)}$, 
    
    \{ $B_2 \leftarrow B_2 - f + e$ \};
  Else 
    \{ $B_1 \leftarrow B_1 - e + f$ \};

EndWhile

Output $B_1$. 

\[ B_1 \quad B_2 \]
**Concentration for linear functions**

**Lemma**
Let \( X_t = (X_{1,t}, \ldots, X_{n,t}) \) be a non-negative vector-valued random process with initial distribution given by \( X_0 = x \in \mathbb{R}^n \) with probability 1 and such that:

- \( \mathbb{E}[X_{t+1} | X_t] = X_t \),
- between \( X_t \) and \( X_{t+1} \) at most two components change,
- if two components change, one increases and the other one decreases.

Then for any \( t \), the components of \( X_t \) are negatively correlated.

- Above Lemma applies to swap rounding algorithm for matroids. 
  \( \Rightarrow \) Chernoff bounds hold for linear functions with coefficients in \([0, 1]\).

- We also get lower-tail concentration bounds for monotone submodular functions.
- This does not follow from negative correlation \( \Rightarrow \) martingale approach.
Concentration for linear functions

Lemma
Let $X_t = (X_{1,t}, \ldots, X_{n,t})$ be a non-negative vector-valued random process with initial distribution given by $X_0 = x \in \mathbb{R}^n$ with probability 1 and such that:

- $\mathbb{E}[X_{t+1} \mid X_t] = X_t$,
- between $X_t$ and $X_{t+1}$ at most two components change,
- if two components change, one increases and the other one decreases.

Then for any $t$, the components of $X_t$ are negatively correlated.

- Above Lemma applies to swap rounding algorithm for matroids.
  ⇒ Chernoff bounds hold for linear functions with coefficients in $[0, 1]$.

- We also get lower-tail concentration bounds for monotone submodular functions.

- This does not follow from negative correlation ⇒ martingale approach.
Concentration for linear functions

**Lemma**
Let $X_t = (X_{1,t}, \ldots, X_{n,t})$ be a non-negative vector-valued random process with initial distribution given by $X_0 = x \in \mathbb{R}^n$ with probability 1 and such that:

- $\mathbf{E}[X_{t+1} \mid X_t] = X_t$,
- between $X_t$ and $X_{t+1}$ at most two components change,
- if two components change, one increases and the other one decreases.

Then for any $t$, the components of $X_t$ are negatively correlated.

- Above Lemma applies to swap rounding algorithm for matroids.
  $\Rightarrow$ Chernoff bounds hold for linear functions with coefficients in $[0, 1]$.

- We also get **lower-tail concentration bounds for monotone submodular functions**.
- This does not follow from negative correlation $\Rightarrow$ **martingale approach**.
Submodular functions

Definition: submodular function
A function \( f : 2^S \rightarrow \mathbb{R} \) is submodular if it has the property of diminishing returns:
\[
f(A + i) - f(A) \geq f(B + i) - f(B) \quad \forall A \subseteq B \subseteq S, \ i \in S \setminus B.
\]
Furthermore, \( f \) is monotone if \( f(A) \leq f(B) \ \forall A \subseteq B \subseteq S. \)

Example I: coverage function
Let \( U \) be a finite ground set and \( W_i \subseteq U \) for \( i \in S. \)
\[
f(A) = \left| \bigcup_{i \in A} W_i \right| \quad \forall A \subseteq S
\]

Example II: cut function
Given is a graph \( G = (V, E) \).
\[
f(U) = |\delta(U)| = |E(U, V \setminus U)| \quad \forall U \subseteq V
\]

To work with submodular functions in relaxations, a continuous counterpart known as multilinear extension proved to be very useful.
**Submodular functions**

**Definition: submodular function**
A function \( f : 2^S \rightarrow \mathbb{R} \) is submodular if it has the property of diminishing returns:

\[
    f(A + i) - f(A) \geq f(B + i) - f(B) \quad \forall A \subseteq B \subseteq S, \ i \in S \setminus B.
\]

Furthermore, \( f \) is monotone if \( f(A) \leq f(B) \) \( \forall A \subseteq B \subseteq S \).

**Example I: coverage function**
Let \( U \) be a finite ground set and \( W_i \subseteq U \) for \( i \in S \).

\[
f(A) = \left| \bigcup_{i \in A} W_i \right| \quad \forall A \subseteq S
\]

**Example II: cut function**
Given is a graph \( G = (V, E) \).

\[
f(U) = |\delta(U)| = |E(U, V \setminus U)| \quad \forall U \subseteq V
\]

To work with submodular functions in relaxations, a continuous counterpart known as multilinear extension proved to be very useful.
Multilinear extension of submodular function

**Definition: multilinear extension**
The multilinear extension $F$ of a submodular function $f$ is defined by:

$$F(x) = \sum_{A \subseteq S} f(A) \prod_{i \in A} x_i \prod_{i \in S \setminus A} (1 - x_i) \quad \forall x \in [0, 1]^S.$$ 

▶ Hence, $F(x) = \mathbb{E}[f(R)]$ where $R$ is a random set containing each element $i \in S$ independently with probability $x_i$.

**Theorem (Vondrák [2008])**
There is a $(1 - 1/e)$-approximation for maximizing $F$ over any 0/1 polytope over which one can optimize efficiently linear functions.
Multilinear extension of submodular function

**Definition: multilinear extension**
The multilinear extension $F$ of a submodular function $f$ is defined by:

$$F(x) = \sum_{A \subseteq S} f(A) \prod_{i \in A} x_i \prod_{i \in S \setminus A} (1 - x_i) \quad \forall x \in [0, 1]^S.$$  

▶ Hence, $F(x) = \mathbb{E}[f(R)]$ where $R$ is a random set containing each element $i \in S$ independently with probability $x_i$.

**Theorem (Vondrák [2008])**
There is a $(1 - 1/e)$-approximation for maximizing $F$ over any 0/1 polytope over which one can optimize efficiently linear functions.
Concentration for submodular functions

Consider the following setting:

- $f : \{0, 1\}^n \rightarrow \mathbb{R}_+$: monotone submodular function with marginal values $\leq 1$,
- $F : [0, 1]^n \rightarrow \mathbb{R}_+$: multilinear extension of $f$,
- $x \in P$: point in matroid polytope $P$ to round,
- $X \in P \cap \{0, 1\}^n$: random point obtained by randomized swap rounding.

**Theorem**

$$\Pr[f(X) \leq (1 - \delta)F(x)] \leq e^{-F(x)\delta^2/8} \quad \forall \delta > 0.$$  

$\Rightarrow$ If $x$ approximately maximizes $F$ then $X$ approximately maximizes $f$.

**Remarks**

- A deterministic algorithm was already known for obtaining $X \in \{0, 1\}^n$ such that $f(X) \geq F(x)$ (Calinescu et al. [2007]).
- Advantage of randomized approach: handle additional weak linear/submodular constraints.
Concentration for submodular functions

Consider the following setting:

- \( f : \{0, 1\}^n \rightarrow \mathbb{R}_+ \): monotone submodular function with marginal values \( \leq 1 \),
- \( F : [0, 1]^n \rightarrow \mathbb{R}_+ \): multilinear extension of \( f \),
- \( x \in P \): point in matroid polytope \( P \) to round,
- \( X \in P \cap \{0, 1\}^n \): random point obtained by randomized swap rounding.

**Theorem**

\[
\Pr[f(X) \leq (1 - \delta)F(x)] \leq e^{-F(x)\delta^2/8} \quad \forall \delta > 0.
\]

\[\Rightarrow\] If \( x \) approximately maximizes \( F \) then \( X \) approximately maximizes \( f \).

**Remarks**

- A deterministic algorithm was already known for obtaining \( X \in \{0, 1\}^n \) such that \( f(X) \geq F(x) \) (Calinescu et al. [2007]).
- Advantage of randomized approach: handle additional weak linear/submodular constraints.
Concentration for submodular functions

Consider the following setting:

- $f : \{0, 1\}^n \rightarrow \mathbb{R}_+$: monotone submodular function with marginal values $\leq 1$,
- $F : [0, 1]^n \rightarrow \mathbb{R}_+$: multilinear extension of $f$,
- $x \in P$: point in matroid polytope $P$ to round,
- $X \in P \cap \{0, 1\}^n$: random point obtained by randomized swap rounding.

**Theorem**

$$\Pr[f(X) \leq (1 - \delta)F(x)] \leq e^{-F(x)\delta^2/8} \quad \forall \delta > 0.$$ 

$\Rightarrow$ If $x$ approximately maximizes $F$ then $X$ approximately maximizes $f$.

**Remarks**

- A deterministic algorithm was already known for obtaining $X \in \{0, 1\}^n$ such that $f(X) \geq F(x)$ (Calinescu et al. [2007]).
- Advantage of randomized approach: handle additional weak linear/submodular constraints.
Outline

1 Introduction
   • Motivation

2 Randomized swap rounding: a new rounding framework
   • The general framework
   • Swap rounding in matroid polytopes
   • Swap rounding in the intersection of two matroids

3 Some consequences/applications

4 Conclusions
Some consequences/applications

A congestion minization problem
Given:
- Matroid $M = (S, \mathcal{I})$,
- Matrix $A \in \mathbb{R}^{m \times S}$.

Task: \(\min\{\lambda | \exists \text{ base } B \text{ in } M \text{ with } A \cdot 1_B \leq \lambda 1\}\).

Theorem
There is an $O(\log m / \log \log m)$-approximation to the above problem.

Network routing: comparison to previous results
Consider congestion minimization in a network routing context: there are $m$ source-destination pairs $(s_i, t_i)$, for each of which a set of $s_i$-$t_i$ paths is given.

- If one path per commodity has to be chosen: $O(\log m / \log \log m)$-approximation by Raghavan and Thompson [1987].
- For commodity $i$, $k_i$ paths have to be chosen: $O(\log m / \log \log m)$-approximation by Srinivasan [2001].
- Using swap rounding, $O(\log m / \log \log m)$ approximation is obtained for any matroid constraint on the paths to choose.
Some consequences/applications

A congestion minization problem
Given: • Matroid $M = (S, \mathcal{I})$,
• Matrix $A \in \mathbb{R}^{m \times S}$.

Task: $\min \{ \lambda \mid \exists \text{ base } B \text{ in } M \text{ with } A \cdot 1_B \leq \lambda 1 \}$.

Theorem
There is an $O(\log m / \log \log m)$-approximation to the above problem.

Network routing: comparison to previous results
Consider congestion minimization in a network routing context: there are $m$
source-destination pairs $(s_i, t_i)$, for each of which a set of $s_i$-$t_i$ paths is given.

- If one path per commodity has to be chosen: $O(\log m / \log \log m)$-approximation
  by Raghavan and Thompson [1987].
- $k_i$ paths have to be chosen for commodity $i$: $O(\log m / \log \log m)$-approximation
  by Srinivasan [2001].
- Using swap rounding, $O(\log m / \log \log m)$ approximation is obtained for any
  matroid constraint on the paths to choose.
Some consequences/applications

**A congestion minization problem**

given:
- Matroid \( M = (S, \mathcal{I}) \),
- Matrix \( A \in \mathbb{R}^{m \times S} \).

Task: \( \min\{ \lambda \mid \exists \text{ base } B \text{ in } M \text{ with } A \cdot 1_B \leq \lambda 1 \} \).

**Theorem**

There is an \( O(\log m / \log \log m) \)-approximation to the above problem.

**Network routing: comparison to previous results**

Consider congestion minimization in a network routing context: there are \( m \) source-destination pairs \((s_i, t_i)\), for each of which a set of \( s_i-t_i \) paths is given.

- If **one path per commodity** has to be chosen: \( O(\log m / \log \log m) \)-approximation by Raghavan and Thompson [1987].
- \( k_i \) **paths** have to be chosen for commodity \( i \): \( O(\log m / \log \log m) \)-approximation by Srinivasan [2001].
- Using swap rounding, \( O(\log m / \log \log m) \) approximation is obtained for **any matroid constraint** on the paths to choose.
Some consequences/applications (II)

Max-min submodular allocation

Given:
- Constant number $k$ of agents interested in a set $N$ of items.
- Agent $i \in [k]$ has monotone submodular utility funct. $w_i : 2^N \to \mathbb{R}_+.$

Task: Find allocation of items to players, i.e., disjoint sets $S_1, \ldots, S_k \subseteq N$ maximizing $\min_{i \in [k]} w_i(S_i)$.

Theorem
There is a $(1 - 1/e - \epsilon)$-approximation to the above problem for any $\epsilon > 0$.

Sketch of algorithm

(i) Guess a constant number of items for each agent.

(ii) Get $(1 - 1/e)$-approx. to following relaxation using (variant of) continuous greedy: \[
\max \{ \min_{i \in [k]} F_i(x_{i1}, \ldots, x_{in}) \mid \sum_{i \in [k]} x_{ij} \leq 1 \ \forall j \in N, x_{ij} \geq 0 \},
\]
where $F_i$ is multilinear extension of $w_i$, and $n = |N|$.

(iii) Round obtained fractional solution.

Theorem (consequence of Mirrokni et al. [2008])
A $(1 - (1 - 1/|N|)|N| - \epsilon)$-approximation, requires exponentially many queries.
Outline

1. Introduction
   - Motivation

2. Randomized swap rounding: a new rounding framework
   - The general framework
   - Swap rounding in matroid polytopes
   - Swap rounding in the intersection of two matroids

3. Some consequences/applications

4. Conclusions
Conclusions

▶ Randomized swap rounding provides a unifying and simple framework for several known applications.

▶ Generality of matroids and matroid intersections allows us to easily handle richer sets of constraints.

▶ Lower-tail concentration bound for submodular functions, allows for approximate maximization of submodular functions under a variety of hard/weak constraints.

▶ Extension of the general swap rounding framework to other problems?
▶ Extension of martingale concentration argument to other settings?
▶ Derandomization?


