Submodular Function Maximization via the Multilinear Relaxation and Contention Resolution Schemes

Rico Zenklusen
MIT

Joint work with Chandra Chekuri and Jan Vondrák
Submodular functions

- Let $N$ be a finite ground set, $n := |N|$.

**Definition (submodular function)**

A set function $f : 2^N \rightarrow \mathbb{R}$ is submodular if it has diminishing returns:

$$f(A + i) - f(A) \geq f(B + i) - f(B) \quad \forall A \subseteq B \subseteq N, \forall i \in N \setminus B$$

where $A + i := A \cup \{i\}$

**Equivalent definition:**

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \quad \forall A, B \subseteq N.$$

→ Submodularity is a natural property of utility functions.

- $f$ is monotone $\iff f(A) \leq f(B) \quad \forall A \subseteq B$. 

Examples of submodular functions

Example I: coverage function
Let $U$ be a finite ground set and $W_i \subseteq U$ for $i \in \mathbb{N}$.

$$f(A) = \left| \bigcup_{i \in A} W_i \right| \quad \forall A \subseteq \mathbb{N}$$

Example II: cut function
Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$.

$$f(U) = w(\delta(U)) = w(E(U, V \setminus U)) \quad \forall U \subseteq V$$

Other examples
- Entropy function $H : 2^\mathbb{N} \to \mathbb{R}_+$ of random variables $\{X_i\}_{i \in \mathbb{N}}$:
  $$H(A) := H(\{X_i \mid i \in A\}) \quad \forall A \subseteq \mathbb{N}.$$  
- Reduction of connection costs in facility location problems.
- ...
Optimizing submodular functions

Access to $f$ by value oracle: can query $f(A)$ for $A \subseteq N$. 

Minimization vs. maximization

▶ Unconstrained minimization of submodular functions can be done efficiently.
▶ Unconstrained maximization of submodular functions is hard:
  • No $> 0.5$-approx without exponentially many calls to value oracle. (Feige et al., 2007)
  • Remains hard in many settings outside the value oracle model (Max-Cut, Max-k-Cover, . . .).
  • Currently best approximation ratio: 0.41. (Oveis Gharan and Vondr´ak, 2011)
Θ(1)-approximations often achievable under additional packing constraints.
Optimizing submodular functions

Access to $f$ by **value oracle**: can query $f(A)$ for $A \subseteq N$.

### Minimization vs. maximization

- Unconstrained minimization of submodular functions can be done efficiently.

- Unconstrained maximization of submodular functions is hard:
  - No $> 0.5$-approx without exponentially many calls to value oracle. (Feige et al., 2007)
  - Remains hard in many settings outside the value oracle model (Max-Cut, Max-k-Cover, ...).
  - Currently best approximation ratio: 0.41. (Oveis Gharan and Vondrák, 2011)

$\Theta(1)$-approximations often achievable under additional packing constraints.
Some previous results on SFM (subm. funct. max.)

Assume $f : 2^N \to \mathbb{R}_+$ (otherwise: no hope for good approximations).

Approaches for SFM are based either on

a) greedy approaches,

b) combinatorial local search procedures (replacing elements), or

c) relaxation and rounding techniques.

<table>
<thead>
<tr>
<th>Constraint type</th>
<th>Linear max.</th>
<th>Monotone subm. max.</th>
<th>Subm. max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(1)$ knapsacks</td>
<td>$1 - \epsilon$</td>
<td>$1 - 1/e - \epsilon$</td>
<td>$0.25 - \epsilon$</td>
</tr>
<tr>
<td>1 matroid</td>
<td>1</td>
<td>$1 - 1/e$</td>
<td>$0.325$</td>
</tr>
<tr>
<td>$k = O(1)$ matroids</td>
<td>$1/(k-1+\epsilon)$</td>
<td>$1/(k+\epsilon)$</td>
<td>$1/(k+1+\frac{1}{k-1}+\epsilon)$</td>
</tr>
</tbody>
</table>

---

1 Kulik et al. (2011)
2 Calinescu et al. (2011)
3 Oveis Gharan and Vondrák (2011)
4 Lee et al. (2009b)
Some previous results on SFM (subm. funct. max.)

- Assume \( f : 2^N \to \mathbb{R}_+ \) (otherwise: no hope for good approximations).

Approaches for SFM are based either on

a) greedy approaches,
b) combinatorial local search procedures (replacing elements), or
c) relaxation and rounding techniques.

<table>
<thead>
<tr>
<th>Constraint type</th>
<th>Linear max.</th>
<th>Monotone subm. max.</th>
<th>Subm. max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O(1) ) knapsacks</td>
<td>( 1 - \epsilon )</td>
<td>( 1 - 1/e - \epsilon ) (^1)</td>
<td>( 0.25 - \epsilon ) (^1)</td>
</tr>
<tr>
<td>1 matroid</td>
<td>1</td>
<td>( 1 - 1/e ) (^2)</td>
<td>0.325 (^3)</td>
</tr>
<tr>
<td>( k = O(1) ) matroids</td>
<td>( 1/(k - 1 + \epsilon) ) (^4)</td>
<td>( 1/(k + \epsilon) ) (^4)</td>
<td>( 1/(k + 1 + \frac{1}{k-1} + \epsilon) ) (^4)</td>
</tr>
</tbody>
</table>

**Issue with previous approaches**

Typically heavily tailored to the underlying constraints:

- unclear how to deal with combined constraints,
- no clear plan how to tackle new constraints.

---

\(^1\) Kulik et al. (2011)
\(^2\) Calinescu et al. (2011)
\(^3\) Oveis Gharan and Vondrák (2011)
\(^4\) Lee et al. (2009b)
Some previous results on SFM (subm. funct. max.)

- Assume $f : 2^N \rightarrow \mathbb{R}_+$ (otherwise: no hope for good approximations).

Approaches for SFM are based either on

- a) greedy approaches,
- b) combinatorial local search procedures (replacing elements), or
- c) relaxation and rounding techniques.

<table>
<thead>
<tr>
<th>Constraint type</th>
<th>Linear max.</th>
<th>Monotone subm. max.</th>
<th>Subm. max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(1)$ knapsacks</td>
<td>$1 - \epsilon$</td>
<td>$1 - 1/e - \epsilon^1$</td>
<td>$0.25 - \epsilon^1$</td>
</tr>
<tr>
<td>1 matroid</td>
<td>1</td>
<td>$1 - 1/e^2$</td>
<td>0.325</td>
</tr>
<tr>
<td>$k = O(1)$ matroids</td>
<td>$1/(k - 1 + \epsilon)^4$</td>
<td>$1/(k + \epsilon)^4$</td>
<td>$1/(k + 1 + \frac{1}{k-1} + \epsilon)^4$</td>
</tr>
</tbody>
</table>

### Issue with previous approaches

Typically heavily tailored to the underlying constraints:

- unclear how to deal with combined constraints,
- no clear plan how to tackle new constraints.

Is there some more versatile framework?

---

1 Kulik et al. (2011)
2 Calinescu et al. (2011)
3 Oveis Gharan and Vondrák (2011)
4 Lee et al. (2009b)
Our results

- We introduce a rather **general framework** based on relaxation-and-rounding that
  - allows for **combining constraints**, and
  - provides a **general recipe** for SFM with packing constraints.

<table>
<thead>
<tr>
<th>Constraint type</th>
<th>Linear max.</th>
<th>Monot. subm. max.</th>
<th>Subm. max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>knapsacks</td>
<td>$1 - \epsilon$</td>
<td>$1 - \frac{1}{e} - \epsilon$</td>
<td>0</td>
</tr>
<tr>
<td>matroids</td>
<td>$\frac{1}{(k-1 + \epsilon)}$</td>
<td>$\frac{1}{(k + \epsilon)}$ for $k \geq 2$</td>
<td>$\frac{1}{k + 1 + \frac{1}{k} + \epsilon}$</td>
</tr>
<tr>
<td>matr. &amp; ℓ-sparse PIP</td>
<td>$0.6/k$</td>
<td>$0.38/k$</td>
<td>$0.19/k$</td>
</tr>
<tr>
<td>UFP on paths and trees</td>
<td>$\Omega(1/(k+\ell))$</td>
<td>$\Omega(1/(k+\ell))$</td>
<td>$\Omega(1/(k+\ell))$</td>
</tr>
</tbody>
</table>
Our results

- We introduce a rather general framework based on relaxation-and-rounding that
  - allows for combining constraints, and
  - provides a general recipe for SFM with packing constraints.

(Some) new results due to our framework

<table>
<thead>
<tr>
<th>Constraint type</th>
<th>Linear max.</th>
<th>Monot. subm. max.</th>
<th>Subm. max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(1)$ knapsacks</td>
<td>$1 - \epsilon$</td>
<td>$1 - 1/e - \epsilon$</td>
<td>$0.325$ $0.25 - \epsilon$</td>
</tr>
<tr>
<td>$k = O(1)$ matroids</td>
<td>$1/(k - 1 + \epsilon)$</td>
<td>$1/(k + \epsilon)$ for $k \geq 2$</td>
<td>$1/(k + 1 + 1/k - 1 + \epsilon)$</td>
</tr>
<tr>
<td>$k$ matr. &amp; $\ell = O(1)$ knaps.</td>
<td>$0.6/k$</td>
<td>$0.38/k$</td>
<td>$0.19/k$</td>
</tr>
<tr>
<td>$k$-matchoid &amp; $\ell$-sparse PIP</td>
<td>$\Omega(1/(k + \ell))$</td>
<td>$\Omega(1/(k + \ell))$</td>
<td>$\Omega(1/(k + \ell))$</td>
</tr>
<tr>
<td>UFP on paths and trees</td>
<td>$\Omega(1)$</td>
<td>$\Omega(1)$</td>
<td>$\Omega(1)$</td>
</tr>
</tbody>
</table>

- new results
- previous results
General framework

1. Create relaxed problem
   i) Relax constraints:
      \[ \mathcal{F} \subseteq 2^N \leadsto \text{polytope } P \subseteq [0, 1]^N \]
   ii) Extend submodular function:
      \[ f \leadsto F : [0, 1]^N \to \mathbb{R}_+ \]

2. Maximize \( F \) over \( P \leadsto x \in P \)

3. Rounding: \( x \leadsto I(x) \in \mathcal{F} \)
   i) \( x \leadsto R(x) \subseteq N \) with
      \[ \Pr[i \in R(x)] = x_i \]
   ii) \( R(x) \leadsto I(x) \) \( \left\{ \subseteq R(x), \quad \in \mathcal{F}. \right\} \)
1. **Create relaxed problem**
   
i) Relax constraints: \( \mathcal{F} \subseteq 2^N \leadsto \text{polytope } P \subseteq [0, 1]^N \)
   
   ii) Extend submodular function: \( f \leadsto F : [0, 1]^N \to \mathbb{R}_+ \)

2. **Maximize** \( F \) **over** \( P \leadsto x \in P \)

3. **Rounding:** \( x \leadsto I(x) \in \mathcal{F} \)
   
i) \( x \leadsto R(x) \subseteq N \) with \( \Pr[i \in R(x)] = x_i \)
   
   ii) \( R(x) \leadsto I(x) \) \( \begin{cases} \subseteq R(x), \\ \in \mathcal{F} \end{cases} \)
General framework

1. Create relaxed problem
   i) Relax constraints:
      \[ \mathcal{F} \subseteq 2^N \leadsto \text{polytope } P \subseteq [0, 1]^N \]
   ii) Extend submodular function:
      \[ f \leadsto F : [0, 1]^N \to \mathbb{R}_+ \]

2. Maximize \( F \) over \( P \leadsto x \in P \)

3. Rounding: \( x \leadsto l(x) \in \mathcal{F} \)
   i) \( x \leadsto R(x) \subseteq N \) with
      \[ \Pr[i \in R(x)] = x_i \]
   ii) \( R(x) \leadsto l(x) \) \( \subseteq R(x) \), \( \in \mathcal{F} \).
General framework

1. Create relaxed problem
   i) Relax constraints:
      \( \mathcal{F} \subseteq 2^N \rightsquigarrow \text{polytope } P \subseteq [0, 1]^N \)
   ii) Extend submodular function:
      \( f \rightsquigarrow F : [0, 1]^N \rightarrow \mathbb{R}_+ \)

2. Maximize \( F \) over \( P \rightsquigarrow x \in P \)

3. Rounding: \( x \rightsquigarrow l(x) \in \mathcal{F} \)
   i) \( x \rightsquigarrow R(x) \subseteq N \) with
      \[ \Pr[i \in R(x)] = x_i \]
   ii) \( R(x) \rightsquigarrow l(x) \bigg\{ \begin{array}{l} \subseteq R(x), \\ \in \mathcal{F}. \end{array} \bigg\} \)
General framework

1. Create relaxed problem
   
   i) Relax constraints:
      \[ \mathcal{F} \subseteq 2^N \leadsto \text{polytope } P \subseteq [0, 1]^N \]
   
   ii) Extend submodular function:
      \[ f \leadsto F : [0, 1]^N \to \mathbb{R}_+ \]

2. Maximize \( F \) over \( P \leadsto x \in P \)

3. Rounding:
   \( x \leadsto I(x) \in \mathcal{F} \)
   
   i) \( x \leadsto R(x) \subseteq N \) with
      \[ \Pr[i \in R(x)] = x_i \]
   
   ii) \( R(x) \leadsto I(x) \begin{cases} \subseteq R(x), \\ \in \mathcal{F}. \end{cases} \)
General framework

1. Create relaxed problem
   i) Relax constraints:
      \[ \mathcal{F} \subseteq 2^N \leadsto \text{polytope } P \subseteq [0, 1]^N \]
   ii) Extend submodular function:
      \[ f \leadsto F : [0, 1]^N \to \mathbb{R}_+ \]

2. Maximize \( F \) over \( P \leadsto x \in P \)

3. Rounding: \( x \leadsto I(x) \in \mathcal{F} \)
   i) \( x \leadsto R(x) \subseteq N \) with
      \[ \Pr[i \in R(x)] = x_i \]
   ii) \( R(x) \leadsto I(x) \begin{cases} \subseteq R(x), \\ \in \mathcal{F}. \end{cases} \)
The multilinear extension

Definition: multilinear extension $F$

$$F(x) := \sum_{S \subseteq N} f(S) \prod_{i \in S} x_i \prod_{i \in N \setminus S} (1 - x_i) = E[f(R(x))],$$

where $R(x) \subseteq N$: random set with $\Pr[i \in R(x)] = x_i$ independently for $i \in N$.

- Behaves nicely w.r.t. indep. rounding (would lead to constraint violations).
- Efficient approximate evaluation possible through Monte-Carlo sampling.
Improvements to the relax.-and-rounding framework

Maximization of $F$ over a polytope $P$

- It is well understood how to maximize $F$ for monotone submodular functions,
  - $(1 - \frac{1}{e})$-approx for down-monot. & solvable $P$ by Vondrák (2008).

- For non-monotone submodular functions, not so much is known:
  - $(\frac{1}{4} - \epsilon)$-approx for $O(1)$ knapsack constraints by Lee et al. (2009a),
  - 0.325-approx for 1 matroid constraint by Oveis Gharan and Vondrák (2011).

- Our contribution: Constant-factor (0.325) approximations for maximizing $F$ over any down-monotone & solvable polytope $P$. 
Improvements to the relax.-and-rounding framework

Maximization of $F$ over a polytope $P$

- It is well understood how to maximize $F$ for monotone submodular functions,
  - $(1 - \frac{1}{e})$-approx for down-monot. & solvable $P$ by Vondrák (2008).

- For non-monotone submodular functions, not so much is known:
  - $(\frac{1}{4} - \epsilon)$-approx for $O(1)$ knapsack constraints by Lee et al. (2009a),
  - 0.325-approx for 1 matroid constraint by Oveis Gharan and Vondrák (2011).

- Our contribution: Constant-factor (0.325) approximations for maximizing $F$ over any down-monotone & solvable polytope $P$.

The rounding step

- Only little was known about how to round a point $x \in P$.
  - e.g. integrality gap of $F$ over $P$ unknown so far even if $P$ is intersection of 2 matroids, and underlying submodular function is monotone.

- Our contribution: A rather general rounding framework based on contention resolution (CR) schemes.
Improvements to the relax.-and-rounding framework

Maximization of $F$ over a polytope $P$

- It is well understood how to maximize $F$ for monotone submodular functions,
  - $(1 - \frac{1}{e})$-approx for down-monot. & solvable $P$ by Vondrák (2008).

- For non-monotone submodular functions, not so much is known:
  - $(\frac{1}{4} - \epsilon)$-approx for $O(1)$ knapsack constraints by Lee et al. (2009a),
  - 0.325-approx for 1 matroid constraint by Oveis Gharan and Vondrák (2011).

- Our contribution: Constant-factor (0.325) approximations for maximizing $F$
  over any down-monotone & solvable polytope $P$.

The rounding step

- Only little was known about how to round a point $x \in P$.
  - e.g. integrality gap of $F$ over $P$ unknown so far even if $P$ is intersection of
    2 matroids, and underlying submodular function is monotone.

- Our contribution: A rather general rounding framework based on contention
  resolution (CR) schemes.
**Contention resolution (CR) schemes**

A balanced CR scheme for $P$ is a (random) procedure parametrized by $x \in P$, that selects a set $I = I(x) \in \mathcal{F}$, $I \subseteq R(x)$ with $\Pr[i \in I] \geq c \cdot x_i \iff \Pr[i \in I | i \in R(x)] \geq c \ \forall i \in \mathcal{R}$. The scheme is called monotone if $\Pr[i \in I | R(x) = R_1] \geq \Pr[i \in I | R(x) = R_2] \ \forall i \in R_1 \subseteq R_2$.

Theorem (follows from Bansal et al. (2010))

Let $x \in P$ and $I(x)$ be the output of a monotone $c$-balanced CR scheme (that satisfies $\Pr[i \in I | i \in R(x)] = c$). Then $E[f(I(x))] \geq c \cdot F(x)$.
Contention resolution (CR) schemes

A \( c \)-balanced CR scheme for \( P \) is a (random) procedure parametrized by \( x \in P \), that selects a set \( I = I(x) \in F \), \( I \subseteq R(x) \) with \( \Pr[i \in I] \geq c \cdot x_i \) \( \forall i \in N \).

The scheme is called \( \triangleright \) monotone if \( \Pr[i \in I | R(x) = R_1] \geq \Pr[i \in I | R(x) = R_2] \) \( \forall i \in R_1 \subseteq R_2 \); Theorem (follows from Bansal et al. (2010))

Let \( x \in P \) and \( I(x) \) be the output of a monotone \( c \)-balanced CR scheme (that satisfies \( \Pr[i \in I | i \in R(x)] = c \)). Then \( \mathbb{E}[f(I(x))] \geq c \cdot F(x) \).
Definition: balanced CR scheme

A *c*-balanced CR scheme for $P$ is a (random) procedure parametrized by $x \in P$, that selects a set $I = I(x) \in \mathcal{F}$, $I \subseteq R(x)$ with

$$\Pr[i \in I] \geq c \cdot x_i$$

$$\Pr[i \in I | i \in R(x)] \geq c \quad \forall i \in N.$$

The scheme is called

- monotone if $\Pr[i \in I | R(x) = R_1] \geq \Pr[i \in I | R(x) = R_2] \quad \forall i \in R_1 \subseteq R_2$.
Contestation resolution (CR) schemes

\[ \begin{array}{c}
\text{x} \in P \quad \xrightarrow{\text{indep. rounding}} \quad R(x) \quad \xrightarrow{\text{CR scheme}} \quad I \left\{ \subseteq R(x) \atop \in \mathcal{F} \right\} \\
\Pr[i \in R(x)] = x_i 
\end{array} \]

\[ \Pr[i \in I] \geq c \cdot x_i \]

**Definition: balanced CR scheme**

A \( c \)-balanced CR scheme for \( P \) is a (random) procedure parametrized by \( x \in P \), that selects a set \( I = I(x) \in \mathcal{F} \), \( I \subseteq R(x) \) with

\[ \Pr[i \in I] \geq c \cdot x_i \quad \iff \quad \Pr[i \in I \mid i \in R(x)] \geq c \quad \forall i \in N. \]

The scheme is called

- **monotone** if \( \Pr[i \in I \mid R(x) = R_1] \geq \Pr[i \in I \mid R(x) = R_2] \quad \forall i \in R_1 \subseteq R_2; \)

**Theorem (follows from Bansal et al. (2010))**

Let \( x \in P \) and \( I(x) \) be the output of a monotone \( c \)-balanced CR scheme (that satisfies \( \Pr[i \in I \mid i \in R(x)] = c \)). Then

\[ \mathbb{E}[f(I(x))] \geq c \cdot F(x). \]
A simple $\frac{1}{8}$-balanced CR scheme for matchings

1. Remove each edge $e \in R(x)$ independently with probability $\frac{1}{2} \rightarrow R(x)'$. 

2. Remove all edges that are not isolated and return remaining edge $I$. 

$\Pr[e \in I | e \in R(x)'] \geq \frac{1}{4}$, since $e = \{u, v\}$ is only edge adjacent to $u$ (or $v$) in $R(x)'$ with prob. $\geq \frac{1}{2}$.

$\Rightarrow \Pr[e \in I | e \in R(x)'] = \Pr[e \in I | e \in R(x)] \geq \frac{1}{4}$.

$\Pr[e \in R(x) | e \in R(x)] = \frac{1}{2} \geq \frac{1}{8}$.

$\Rightarrow$ This CR scheme is indeed monotone.
A simple $\frac{1}{8}$-balanced CR scheme for matchings

1. Remove each edge $e \in R(x)$ independently with probability $1/2 \rightarrow R'(x)$.

This CR scheme is indeed monotone.
A simple $\frac{1}{8}$-balanced CR scheme for matchings

1. Remove each edge $e \in R(x)$ independently with probability $1/2 \rightarrow R'(x) \sim R(\frac{1}{2}x)$. 

$\text{Pr}[e \in I | e \in R(x)] = \text{Pr}[e \in I | e \in R'(x)] \geq \frac{1}{4}$, since $e = \{u, v\}$ is only edge adjacent to $u$ (or $v$) in $R'(x)$ with prob. $\geq 1/2$.
A simple $\frac{1}{8}$-balanced CR scheme for matchings

1. Remove each edge $e \in R(x)$ independently with probability $1/2 \rightarrow R'(x) \sim R(\frac{1}{2}x)$.

2. Remove all edges that are not isolated and return remaining edge $I$. 

This CR scheme is indeed monotone.
A simple \(\frac{1}{8}\)-balanced CR scheme for matchings

1. Remove each edge \(e \in R(x)\) independently with probability \(1/2\) \(\rightarrow R'(x) \sim R(\frac{1}{2}x)\).

2. Remove all edges that are not isolated and return remaining edge \(I\).

\[\Pr[e \in I \mid e \in R'(x)] \geq \frac{1}{4},\] since \(e = \{u, v\}\) is only edge adjacent to \(u\) (or \(v\)) in \(R'(x)\) with prob. \(\geq 1/2\).

\[\Rightarrow \Pr[e \in I \mid e \in R(x)] = \Pr[e \in I \mid e \in R'(x)] \Pr[e \in R'(x) \mid e \in R(x)] \geq \frac{1}{8}.\]

\[\geq 1/4 \quad = 1/2\]

\[\] This CR scheme is indeed monotone.
Combining CR schemes

Often, \( F \) is composed of simpler constraints: \( F = F_1 \cap F_2 \implies P = P_1 \cap P_2 \).
Combining CR schemes

Often, $\mathcal{F}$ is composed of simpler constraints: $\mathcal{F} = \mathcal{F}_1 \cap \mathcal{F}_2 \Rightarrow P = P_1 \cap P_2$.

A simple approach:

\[ I = I_1 \cap I_2 \begin{cases} \subseteq R(x) \\ \in \mathcal{F} \end{cases} \]

- $c_1$-balanced CR scheme
- $c_2$-balanced CR scheme

$x$\rightarrow\text{indep. rounding}\rightarrow R(x)\rightarrow I_1 \begin{cases} \subseteq R(x) \\ \in \mathcal{F}_1 \end{cases}$

\[ I_1 = I_1 \cap I_2 \begin{cases} \subseteq R(x) \\ \in \mathcal{F}_2 \end{cases} \]
Combining CR schemes

Often, $\mathcal{F}$ is composed of simpler constraints: $\mathcal{F} = \mathcal{F}_1 \cap \mathcal{F}_2 \Rightarrow P = P_1 \cap P_2$.

A simple approach:

- Resulting CR scheme is $c_1c_2$-balanced (follows from FKG inequality).
- Monotonicity is preserved.
Combining CR schemes

Often, $\mathcal{F}$ is composed of simpler constraints: $\mathcal{F} = \mathcal{F}_1 \cap \mathcal{F}_2 \Rightarrow P = P_1 \cap P_2$.

A simple approach:

Resulting CR scheme is $c_1c_2$-balanced (follows from FKG inequality).

Monotonicity is preserved.

Often, stronger combined schemes can be obtained by first scaling down $x$ a bit, $x \sim b \cdot x$, and working with $R(bx)$ instead of $R(x) \rightarrow$ obtain higher values for $c_1$, $c_2$. 

Pr[$i \in R(bx)$] = $bx_i$  \quad Pr[$i \in I \mid i \in R(bx)$] $\geq c$
Combining CR schemes (II)

A stronger approach:

\[ I = I_1 \cap I_2 \]

\[ I_1 = \{ \subseteq R(bx), \in \mathcal{F}_1 \} \]

\[ I_2 = \{ \subseteq R(bx), \in \mathcal{F}_2 \} \]
Combining CR schemes (II)

A stronger approach:

Resulting scheme is $bc_1c_2$-balanced.

This approach is stronger in the parallel part.
Existence of strong CR scheme

Results on CR schemes

- $(b, \frac{1-e^{-b}}{b})$-balanced, monotone and strict CR scheme for matroid constraint, for $b \in (0, 1]$. This scheme is optimal.

- For any fixed $\epsilon > 0$: $(1 - \epsilon, 1 - \epsilon)$-balanced monot. and strict CR scheme for knapsack constraint.

- $(b, 1 - \Omega(b))$-balanced, monotone and strict CR scheme for UFP on trees.

- $(b, 1 - 2kb)$-balanced, monotone and strict CR scheme for $k$-sparse PIP.
Existence of strong CR scheme

Results on CR schemes

- $(b, \frac{1-e^{-b}}{b})$-balanced, monotone and strict CR scheme for matroid constraint, for $b \in (0, 1]$. This scheme is optimal.
- For any fixed $\epsilon > 0$: $(1-\epsilon, 1-\epsilon)$-balanced monot. and strict CR scheme for knapsack constraint.
- $(b, 1-\Omega(b))$-balanced, monotone and strict CR scheme for UFP on trees.
- $(b, 1-2kb)$-balanced, monotone and strict CR scheme for $k$-sparse PIP.

Putting the pieces together to obtain the claimed results

E.g. to optimize over $k$ matroid constraints and a $\ell = \Omega(1)$ knapsacks, a $c$-balanced CR scheme can be obtained for

$$c = b \cdot \left( \frac{1-e^{-b}}{b} \right)^k \cdot (1-\epsilon) \ell \quad \overset{b=1/k}{\Rightarrow} \quad \Omega(1/k).$$

$$\Rightarrow \alpha \cdot \Omega(1/k) = \Omega(1/k)\text{-approx} \text{ to maximize } f \text{ over those constraints, where } \alpha = 0.325 \text{ is the approximation ratio for maximizing } F \text{ over } P.$$
Conclusions

- The multilinear extension can be maximized up to a constant factor on any down-closed and solvable polytope.

- Contention resolution schemes provide a modular way for rounding a fractional point in the context of SFM.

- What is the best possible approximation ratio for maximizing $F$ over $P$?

- Extend techniques to find optimal/good CR scheme?

- What about other extensions than the multilinear one?

- Derandomization?
Conclusions

- The multilinear extension can be maximized up to a constant factor on any down-closed and solvable polytope.
- Contention resolution schemes provide a modular way for rounding a fractional point in the context of SFM.

- What is the best possible approximation ratio for maximizing $F$ over $P$?
- Extend techniques to find optimal/good CR scheme?
- What about other extensions than the multilinear one?
- Derandomization?

Thank you!
References I


