Submodular Function Maximization via the Multilinear Relaxation and Contention Resolution Schemes

Rico Zenklusen

MIT

Joint work with Chandra Chekuri and Jan Vondrák

Outline

Outline

Submodular functions

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

Definition (submodular function)

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

$$f(\underbrace{A+i}_{:=A\cup\{i\}}) - f(A) \geq f(B+i) - f(B) \qquad \forall A \subseteq B \subseteq N, \forall i \in N \setminus B$$

Equivalent definition: $f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \qquad \forall A, B \subseteq N.$

$\longrightarrow$ Submodularity is a natural property of utility functions.

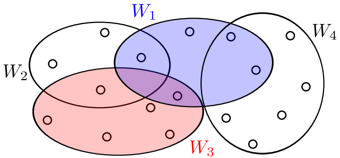
- f is monotone $\Leftrightarrow f(A) \leq f(B) \quad \forall A \subseteq B$.

Examples of subm. funct. beyond utility functions

Example I: coverage function

Let U be a finite ground set and $W_i \subseteq U$ for $i \in N$.

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

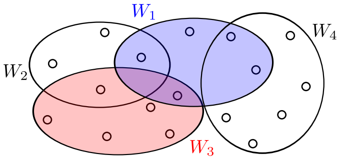


Examples of subm. funct. beyond utility functions

Example I: coverage function

Let U be a finite ground set and $W_i \subseteq U$ for $i \in N$.

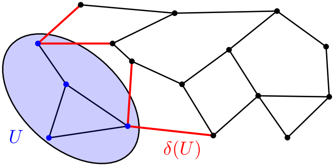
$$f(A) = \left| \bigcup_{i \in A} W_i \right| \quad \forall A \subseteq N$$



Example II: cut function

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

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

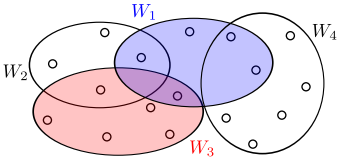


Examples of subm. funct. beyond utility functions

Example I: coverage function

Let U be a finite ground set and $W_i \subseteq U$ for $i \in N$.

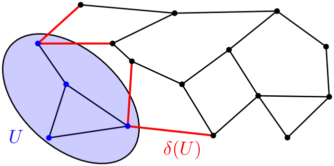
$$f(A) = \left| \bigcup_{i \in A} W_i \right| \quad \forall A \subseteq N$$



Example II: cut function

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

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



Other examples

- Entropy function $H : 2^N \to \mathbb{R}_+$ of random variables $\{X_i\}_{i \in N}$:

$$H(A) := H(\{X_i \mid i \in A\}) \quad \forall A \subseteq N.$$

- Reduction of connection costs in facility location problems.
- ...

Optimizing submodular functions

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

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:
  - Currently best approximation ratio: 0.41. (Oveis Gharan and Vondrák, 2011)
  - 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, ...).
- $\Theta(1)$-approximations often achievable under additional 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:
  - Currently best approximation ratio: 0.41. (Oveis Gharan and Vondrák, 2011)
  - 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, $\ldots$).
- $\Theta(1)$-approximations often achievable under additional constraints.

Under which constraints is it possible to approximately maximize submodular functions?

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

Constraint type	Linear max.	Monotone subm. max.	Subm. max.
$O(1)$ knapsacks	$1-\epsilon$	$1-1/e-\epsilon$ [1]	$0.25-\epsilon$ [1]
1 matroid	1	$1-1/e$ [2]	0.325 [3]
$k=O(1)$ matroids	$1/(k-1+\epsilon)$ [4]	$1/(k+\epsilon)$ [4]	$1/(k+1+\frac{1}{k-1}+\epsilon)$ [4]

[1] Kulik et al. (2011)
[2] Calinescu et al. (2011)
[3] Oveis Gharan and Vondrák (2011)
[4] Lee et al. (2009)

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) combinatorial local search procedures (replacing elements), or

b) relaxation and rounding techniques.

Constraint type	Linear max.	Monotone subm. max.	Subm. max.
$O(1)$ knapsacks	$1-\epsilon$	$1-1/e-\epsilon$ [1]	$0.25-\epsilon$ [1]
1 matroid	1	$1-1/e$ [2]	0.325 [3]
$k=O(1)$ matroids	$1/(k-1+\epsilon)$ [4]	$1/(k+\epsilon)$ [4]	$1/(k+1+\frac{1}{k-1}+\epsilon)$ [4]

Issue with previous approaches

Typically heavily tailored to the underlying constraints.

$\to$ e.g., despite progress on knapsack and matroid constraints, not much was known about a combination of those constraints.

[1] Kulik et al. (2011)

[2] Calinescu et al. (2011)

[3] Oveis Gharan and Vondrák (2011)

[4] Lee et al. (2009)

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

Constraint type	Linear max.	Monotone subm. max.	Subm. max.
$O(1)$ knapsacks	$1-\epsilon$	$1-1/e-\epsilon$ [1]	$0.25-\epsilon$ [1]
1 matroid	1	$1-1/e$ [2]	0.325 [3]
$k=O(1)$ matroids	$1/(k-1+\epsilon)$ [4]	$1/(k+\epsilon)$ [4]	$1/(k+1+\frac{1}{k-1}+\epsilon)$ [4]

Issue with previous approaches

Typically heavily tailored to the underlying constraints.
$\to$ e.g., despite progress on knapsack and matroid constraints, not much was known about a combination of those 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. (2009)

Our results

We introduce a rather general relaxation-and-rounding framework that allows for combining constraints (at the price of a slightly weaker approximation quality).

(Some) new results due to our framework

Constraint type	Linear max.	Monot. subm. max.	Subm. max.
$O(1)$ knapsacks	$1-\epsilon$	$1-1/e-\epsilon$	0.325 $0.25-\epsilon$
$k=O(1)$ matroids	$1/(k-1+\epsilon)$	$1/(k+\epsilon)$ for $k\geq 2$	$1/(k+1+\frac{1}{k-1}+\epsilon)$
k matr. & $\ell=O(1)$ knaps.	$0.6/k$	$0.38/k$	$0.19/k$
k-matchoid & ℓ-sparse PIP	$\Omega(1/(k+\ell))$	$\Omega(1/(k+\ell))$	$\Omega(1/(k+\ell))$
UFP on paths and trees	$\Omega(1)$	$\Omega(1)$	$\Omega(1)$

- new results
- previous results

Our results

We introduce a rather general relaxation-and-rounding framework that allows for combining constraints (at the price of a slightly weaker approximation quality).

(Some) new results due to our framework

Constraint type	Linear max.	Monot. subm. max.	Subm. max.
$O(1)$ knapsacks	$1-\epsilon$	$1-1/e-\epsilon$	0.325 $0.25-\epsilon$
$k = O(1)$ matroids	$1/(k-1+\epsilon)$	$1/(k+\epsilon)$ for $k \geq 2$	$1/(k+1+\frac{1}{k-1}+\epsilon)$
k matr. & $\ell = O(1)$ knaps.	$0.6/k$	$0.38/k$	$0.19/k$
k-matchoid & ℓ-sparse PIP	$\Omega(1/(k+\ell))$	$\Omega(1/(k+\ell))$	$\Omega(1/(k+\ell))$
UFP on paths and trees	$\Omega(1)$	$\Omega(1)$	$\Omega(1)$

- new results
- previous results

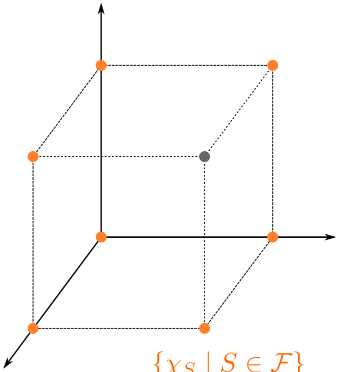
Remark

The constraints $\mathcal{F} \subseteq 2^N$ we consider are all closed under inclusion, i.e.,

$$A \in \mathcal{F}, B \subseteq A \Rightarrow B \in \mathcal{F}.$$

Outline

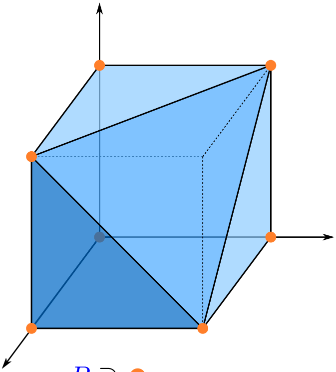
General framework



$\{\chi_S \mid S \in \mathcal{F}\}$

1. **Create relaxed problem**
   - **i)** Relax constraints:
     $\mathcal{F} \subseteq 2^N \rightsquigarrow$ polytope $P \subseteq [0,1]^N$
   - **ii)** Extend submodular function:
     $f \rightsquigarrow F : [0,1]^N \to \mathbb{R}_+$
     $(F(\mathbf{1}_S) = f(S)\ \forall S \subseteq N)$.
2. **Maximize** F **over** $P \rightsquigarrow x \in P$
3. **Rounding:** $x \rightsquigarrow I(x) \in \mathcal{F}$
   - **i)** $x \rightsquigarrow R(x) \subseteq N$ with
     $$\Pr[i \in R(x)] = x_i$$
   - **ii)** $R(x) \rightsquigarrow I(x) \in \mathcal{F}$, with
     $I(x) \subseteq R(x)$ and
     $$\mathbf{E}[f(I(x))] \geq cF(x)$$
     (this *randomized* step depends on x)

General framework

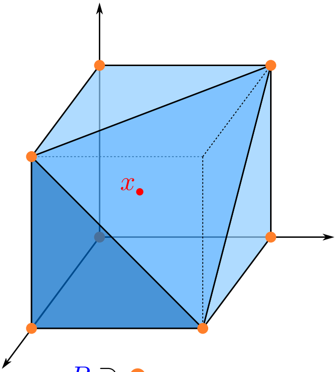


1. **Create relaxed problem**
   - **i)** Relax constraints:
     $\mathcal{F} \subseteq 2^N \rightsquigarrow$ polytope $P \subseteq [0,1]^N$
   - **ii)** Extend submodular function:
     $f \rightsquigarrow F : [0,1]^N \to \mathbb{R}_+$
     ($F(\mathbf{1}_S) = f(S)\ \forall S \subseteq N$).

2. **Maximize** F **over** $P \rightsquigarrow x \in P$

3. **Rounding:** $x \rightsquigarrow I(x) \in \mathcal{F}$
   - **i)** $x \rightsquigarrow R(x) \subseteq N$ with
     $$\Pr[i \in R(x)] = x_i$$
   - **ii)** $R(x) \rightsquigarrow I(x) \in \mathcal{F}$, with
     $I(x) \subseteq R(x)$ and
     $$\mathbf{E}[f(I(x))] \geq cF(x)$$
     (this *randomized* step depends on x)

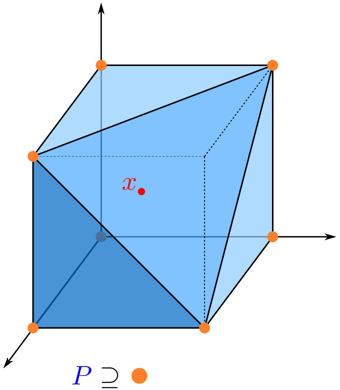
General framework



$P \supseteq$ ●

1. **Create relaxed problem**
   - **i)** Relax constraints:
     $\mathcal{F} \subseteq 2^N \rightsquigarrow$ polytope $P \subseteq [0,1]^N$
   - **ii)** Extend submodular function:
     $f \rightsquigarrow F : [0,1]^N \to \mathbb{R}_+$
     ($F(\mathbf{1}_S) = f(S)\ \forall S \subseteq N$).
2. **Maximize F over $P \rightsquigarrow x \in P$**
3. **Rounding:** $x \rightsquigarrow I(x) \in \mathcal{F}$
   - **i)** $x \rightsquigarrow R(x) \subseteq N$ with
     $$\Pr[i \in R(x)] = x_i$$
   - **ii)** $R(x) \rightsquigarrow I(x) \in \mathcal{F}$, with
     $I(x) \subseteq R(x)$ and
     $$\mathbf{E}[f(I(x))] \geq cF(x)$$
     (this *randomized* step depends on x)

General framework



1. **Create relaxed problem**
   - **i)** Relax constraints:
     $\mathcal{F} \subseteq 2^N \rightsquigarrow$ polytope $P \subseteq [0,1]^N$
   - **ii)** Extend submodular function:
     $f \rightsquigarrow F : [0,1]^N \to \mathbb{R}_+$
     ($F(\mathbf{1}_S) = f(S)\ \forall S \subseteq N$).
2. **Maximize F over $P \rightsquigarrow x \in P$**
3. **Rounding: $x \rightsquigarrow I(x) \in \mathcal{F}$**
   - **i)** $x \rightsquigarrow R(x) \subseteq N$ with

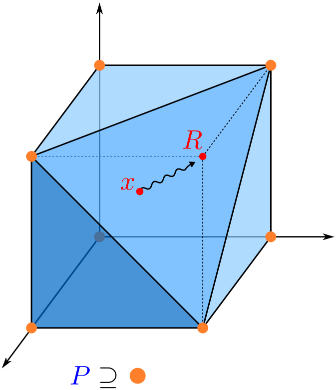
     $$\Pr[i \in R(x)] = x_i$$

   - **ii)** $R(x) \rightsquigarrow I(x) \in \mathcal{F}$, with
     $I(x) \subseteq R(x)$ and

     $$\mathbf{E}[f(I(x))] \geq cF(x)$$

     (this *randomized* step depends on x)

General framework



1. Create relaxed problem

i) Relax constraints:
$\mathcal{F} \subseteq 2^N \rightsquigarrow$ polytope $P \subseteq [0,1]^N$

ii) Extend submodular function:
$f \rightsquigarrow F : [0,1]^N \to \mathbb{R}_+$
$(F(\mathbf{1}_S) = f(S) \; \forall S \subseteq N)$.

2. Maximize F over $P \rightsquigarrow x \in P$

3. Rounding: $x \rightsquigarrow I(x) \in \mathcal{F}$

i) $x \rightsquigarrow R(x) \subseteq N$ with

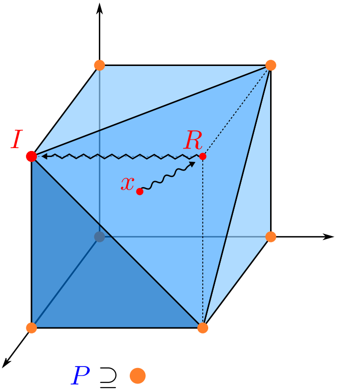
$$\Pr[i \in R(x)] = x_i$$

ii) $R(x) \rightsquigarrow I(x) \in \mathcal{F}$, with
$I(x) \subseteq R(x)$ and

$$\mathbf{E}[f(I(x))] \geq cF(x)$$

(this *randomized* step depends on x)

General framework



1. **Create relaxed problem**
   - **i)** Relax constraints:
     $\mathcal{F} \subseteq 2^N \rightsquigarrow$ polytope $P \subseteq [0,1]^N$
   - **ii)** Extend submodular function:
     $f \rightsquigarrow F : [0,1]^N \to \mathbb{R}_+$
     ($F(\mathbf{1}_S) = f(S)\ \forall S \subseteq N$).
2. **Maximize F over $P \rightsquigarrow x \in P$**
3. **Rounding: $x \rightsquigarrow I(x) \in \mathcal{F}$**
   - **i)** $x \rightsquigarrow R(x) \subseteq N$ with
     $$\Pr[i \in R(x)] = x_i$$
   - **ii)** $R(x) \rightsquigarrow I(x) \in \mathcal{F}$, with
     $I(x) \subseteq R(x)$ and
     $$\mathbf{E}[f(I(x))] \geq cF(x)$$
     (this *randomized* step depends on x)

Shooting for a good extension

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

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

  - Easy to approximately evaluate through Monte-Carlo sampling.
  - Behaves nicely w.r.t. indep. rounding (would lead to constraint violations).

Shooting for a good extension

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

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

  - Easy to approximately evaluate through Monte-Carlo sampling.
  - Behaves nicely w.r.t. indep. rounding (would lead to constraint violations).

- Lovász extension: $f^L(x) := \min \left\{ \sum_{S \subseteq N} \alpha_S f(S) \,\middle|\, \sum_{S \subseteq N, i \in S} \alpha_S = x_i, \sum_{S \subseteq N} \alpha_S = 1, \alpha_S \geq 0 \right\}$

  - Convex
  - Easy to evaluate
  - Hard to maximize

- Concave closure: $f^+(x) := \max \left\{ \sum_{S \subseteq N} \alpha_S f(S) \,\middle|\, \sum_{S \subseteq N, i \in S} \alpha_S = x_i, \sum_{S \subseteq N} \alpha_S = 1, \alpha_S \geq 0 \right\}$

  - Concave
  - Hard to evaluate

Outline

1. Introduction
2. General framework
3. **Maximizing the multilinear extension**
4. Rounding through contention resolution schemes
5. An optimal CR-scheme for matroids
6. Conclusions

Maximizing F over solvable down-closed polytopes P

Definitions

- P is down-closed (or down-monotone) if $x \in P, y \leq x \Rightarrow y \in P$.
- P is solvable if linear functions can be optimizing efficiently over P.

Our main results here

- We can find $y \in P$ with $F(y) \geq 0.25 \cdot \max\{F(x) \mid x \in P\}$.
- We can find $y \in P$ with $F(y) \geq 0.325 \cdot \max\{F(x) \mid x \in P \cap \{0,1\}^N\}$.

- Next slides: very short sketch of the 0.25-approx due to its simplicity.
- To get some intuition let's first consider a related 1/3-approx for unconstrained SFM (which is a variation of an algo of Feige et al. (2007)).

Getting some intuition

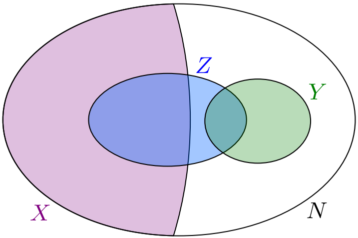
A related 1/3-approx for unconstrained SFM

$\frac{1}{3}$-approx for unconstrained SFM

1. Find a local opt $X \subseteq N$: $\quad f(X \pm i) \leq f(X) \quad \forall i \in N$.
2. Find a local opt $Y \subseteq N \setminus X$: $\quad f(Y \pm i) \leq f(Y) \quad \forall i \in N \setminus X$.
3. Return the better of X and Y.

Proof.

- Let Z be a global opt.
- X local opt:
  - $f(X) \geq f(X \cup Z)$,
  - $f(X) \geq f(X \cap Z)$.
- Y local opt:
  - $f(Y) \geq f(Y \cup (Z \setminus X))$.



$$2f(X) + f(Y) \geq f(X \cap Z) + \underbrace{f(X \cup Z) + f(Y \cup (Z \setminus X))}_{\geq f(Z \setminus X)} \geq f(Z)$$

□

Sketch of the 0.25-approx for down-closed P

0.25-approx

1. Find an (approximate) local opt x of F over P, i.e.,
$$\nabla F(x) \cdot (v - x) \leq 0 \quad \forall v \in P.$$
2. Find an (approximate) local opt y of F over $Q = \{v \in P \mid v \leq 1 - x\}$,
$$\nabla F(y) \cdot (v - y) \leq 0 \quad \forall v \in Q.$$
3. Return the better of x and y.

Outline

Contention resolution (CR) schemes

- $F(x) = \mathbf{E}[f(R(x))] \Rightarrow$ independent rounding preserves value in expectation but is likely to violate constraints.

Contention resolution (CR) schemes

- $F(x) = \mathbf{E}[f(R(x))] \Rightarrow$ independent rounding preserves value in expectation but is likely to violate constraints.

Plan:
- accept lower value for expectation to obtain feasibility,
- maintain "sufficient" independence in rounding process to get good expectation.

Contention resolution (CR) schemes

- $F(x) = \mathbf{E}[f(R(x))] \Rightarrow$ independent rounding preserves value in expectation but is likely to violate constraints.

Plan:
- accept lower value for expectation to obtain feasibility,
- maintain "sufficient" independence in rounding process to get good expectation.

Definition: balanced CR scheme

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

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

Furthermore, the scheme is called

- monotone if
$$\Pr[i \in I \mid R(x) = R_1] \geq \Pr[i \in I \mid R(x) = R_2] \qquad \forall i \in R_1 \subseteq R_2 \subseteq N,$$
- and strict if
$$\Pr[i \in I \mid i \in R(x)] = c \qquad \forall i \in N.$$

Rounding guarantees

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

Let $x \in P$, and let $I(x)$ be the output of a monotone and strict c-balanced CR scheme. Then

$$\mathbf{E}[f(I(x))] \geq c \cdot F(x).$$

Rounding guarantees

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

Let $x \in P$, and let $I(x)$ be the output of a monotone and strict c-balanced CR scheme. Then

$$\mathbf{E}[f(I(x))] \geq c \cdot F(x).$$

Remarks

- Strictness is only needed for non-monotone f, and can be avoided by a simple post-processing of I.
- The rounding procedure is oblivious to f.

Proof of rounding guarantee (I)

- We number the elements $N = [n] := \{1, \ldots, n\}$.
- For $A \subseteq N, i \in N$, let $f_A(i) = f(A + i) - f(A)$.

$$\mathbf{E}[f(I)] = f(\emptyset) + \sum_{i=1}^{n} \mathbf{E}\left[f(I \cap [i]) - f(I \cap [i-1])\right].$$

We want to show: $\boxed{\mathbf{E}[f(I \cap [i]) - f(I \cap [i-1])] \geq \Pr[i \in R] \cdot c \cdot \mathbf{E}[f_{R \cap [i-1]}(i)]}$

This then implies

$$\begin{aligned} f(\emptyset) + \sum_{i=1}^{n} \mathbf{E}[f(I \cap [i]) - f(I \cap [i-1])] &\geq c \left[f(\emptyset) + \sum_{i=1}^{n} \Pr[i \in R] \mathbf{E}[f_{R \cap [i-1]}(i)] \right] \\ &= c \left[f(\emptyset) + \sum_{i=1}^{n} \mathbf{E}[f(R \cap [i]) - f(R \cap [i-1])] \right] \\ &= c \cdot F(x). \end{aligned}$$

Proof of rounding guarantee (II)

To show: $\boxed{\mathbf{E}[f(I \cap [i]) - f(I \cap [i-1])] \geq \Pr[i \in R] \cdot c \cdot \mathbf{E}[f_{R\cap[i-1]}(i)]}$

$$\begin{aligned}
\mathbf{E}[f(I \cap [i]) - f(I \cap [i-1])] &= \mathbf{E}[\mathbf{1}_{i\in I} f_{I\cap[i-1]}(i)] \\
&\geq \mathbf{E}[\mathbf{1}_{i\in I} f_{R\cap[i-1]}(i)] \\
&\geq \mathbf{E}_R[\mathbf{E}_I[\mathbf{1}_{i\in I} f_{R\cap[i-1]}(i) \mid R]] \\
&= \mathbf{E}_R[\mathbf{E}_I[\mathbf{1}_{i\in I} \mid R]\, f_{R\cap[i-1]}(i)] \\
&\geq \Pr[i \in R] \cdot \mathbf{E}[\Pr[i \in I \mid R] f_{R\cap[i-1]}(i) \mid i \in R]
\end{aligned}$$

On the product space associated with distribution of R conditioned on $i \in R$:

- $\Pr[i \in I \mid R]$ is non-decreasing $\Leftarrow$ monotonicity of CR scheme,
- $f_{R\cap[i-1]}(i)$ is non-decreasing $\Leftarrow$ submodularity of f.

$\Rightarrow$ we can apply FKG.

$$\begin{aligned}
&\mathbf{E}_R[\Pr[i \in I \mid R] f_{R\cap[i-1]}(i) \mid i \in R] \\
&\qquad \overset{FKG}{\geq} \mathbf{E}_R[\Pr[i \in I \mid R] \mid i \in R] \cdot \mathbf{E}_R[f_{R\cap[i-1]}(i) \mid i \in R] \\
&\qquad = \Pr[i \in I \mid i \in R] \cdot \mathbf{E}[f_{R\cap[i-1]}(i)] \\
&\qquad \overset{strictness}{=} c \cdot \mathbf{E}[f_{R\cap[i-1](i)}].
\end{aligned}$$

□

Combining CR schemes

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

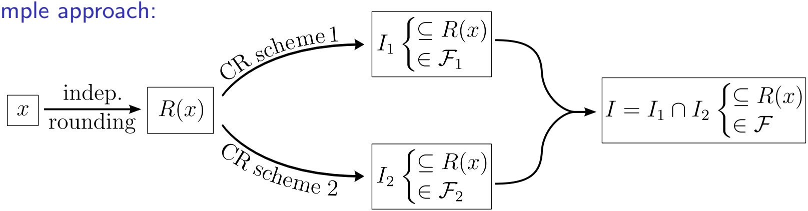
Goal: combine monotone (and strict) c_1-balanced CR scheme for P_1 with monotone (and strict) c_2-balanced CR scheme for P_2 to obtain one for P.

Combining CR schemes

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

Goal: combine monotone (and strict) c_1-balanced CR scheme for P_1 with monotone (and strict) c_2-balanced CR scheme for P_2 to obtain one for P.

A simple approach:

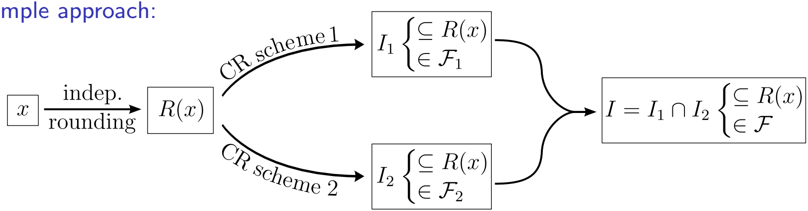


Combining CR schemes

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

Goal: combine monotone (and strict) c_1-balanced CR scheme for P_1 with monotone (and strict) c_2-balanced CR scheme for P_2 to obtain one for P.

A simple approach:



- Monotonicity is preserved.
- Resulting CR scheme is $c_1 c_2$-balanced:

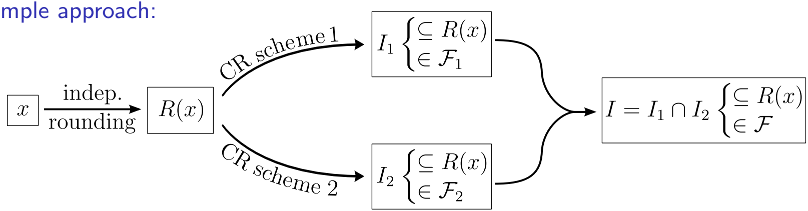
$$\Pr[i \in I \mid i \in R(x)] = \mathbf{E}[\mathbf{1}_{i \in I_1} \mathbf{1}_{i \in I_2} \mid i \in R(x)] \overset{FKG}{\geq} \underbrace{\mathbf{E}[\mathbf{1}_{i \in I_1} \mid i \in R(x)]}_{=c_1} \underbrace{\mathbf{E}[\mathbf{1}_{i \in I_2} \mid i \in R(x)]}_{=c_2}.$$

Combining CR schemes

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

Goal: combine monotone (and strict) c_1-balanced CR scheme for P_1 with monotone (and strict) c_2-balanced CR scheme for P_2 to obtain one for P.

A simple approach:



- Monotonicity is preserved.
- Resulting CR scheme is $c_1 c_2$-balanced:

$$\Pr[i \in I \mid i \in R(x)] = \mathbf{E}[\mathbf{1}_{i\in I_1}\mathbf{1}_{i \in I_2} \mid i \in R(x)] \overset{FKG}{\geq} \underbrace{\mathbf{E}[\mathbf{1}_{i\in I_1} \mid i \in R(x)]}_{=c_1} \underbrace{\mathbf{E}[\mathbf{1}_{i\in I_2} \mid i \in R(x)]}_{=c_2}.$$

- Combining k schemes being c-balanced $\rightarrow$ c^k-balanced scheme.
- Our goal: obtain $\Omega(1/k)$-balanced CR scheme.

Combining CR schemes (II)

Definition: (b, c)-balanced CR scheme ($b, c \in (0, 1]$)

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

$$\Pr[i \in I \mid i \in R(b \cdot x)] \geq c \qquad \forall i \in N.$$

Combining CR schemes (II)

Definition: (b, c)-balanced CR scheme ($b, c \in (0, 1]$)

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

$$\Pr[i \in I \mid i \in R(b \cdot x)] \geq c \qquad \forall i \in N.$$

- A (b, c)-balanced CR scheme is a special case of a bc-balanced CR scheme.

Combining CR schemes (II)

Definition: (b, c)-balanced CR scheme ($b, c \in (0,1]$)

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

$$\Pr[i \in I \mid i \in R(b \cdot x)] \geq c \qquad \forall i \in N.$$

- A (b, c)-balanced CR scheme is a special case of a bc-balanced CR scheme.

Goal: Create families of monotone (b, c)-balanced CR schemes s.t. $c \to 1$ (at the possible cost of $b \to 0$).

Combining CR schemes (II)

Definition: (b, c)-balanced CR scheme ($b, c \in (0, 1]$)

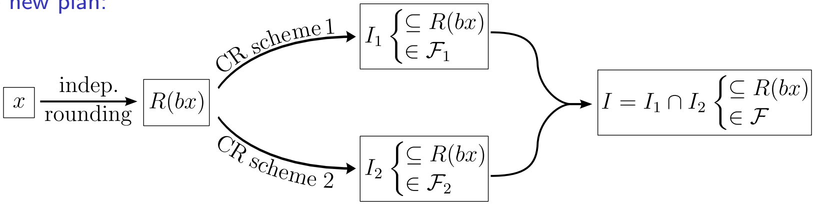
A (b, c)-balanced CR scheme for P is a (random) procedure parametrized by $x \in P$, that selects a set $I \in \mathcal{F}$, $I \subseteq R(b \cdot x)$ with

$$\Pr[i \in I \mid i \in R(b \cdot x)] \geq c \qquad \forall i \in N.$$

- A (b, c)-balanced CR scheme is a special case of a bc-balanced CR scheme.

Goal: Create families of monotone (b, c)-balanced CR schemes s.t. $c \to 1$ (at the possible cost of $b \to 0$).

The new plan:



Combining CR schemes (II)

Definition: (b, c)-balanced CR scheme ($b, c \in (0, 1]$)

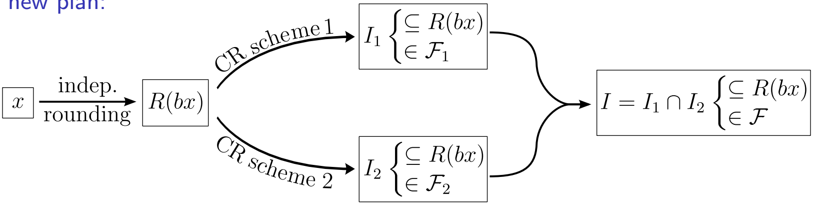
A (b, c)-balanced CR scheme for P is a (random) procedure parametrized by $x \in P$, that selects a set $I \in \mathcal{F}$, $I \subseteq R(b \cdot x)$ with

$$\Pr[i \in I \mid i \in R(b \cdot x)] \geq c \qquad \forall i \in N.$$

- A (b, c)-balanced CR scheme is a special case of a bc-balanced CR scheme.

Goal: Create families of monotone (b, c)-balanced CR schemes s.t. $c \to 1$ (at the possible cost of $b \to 0$).

The new plan:



- This approach is stronger in the parallel part.
- Resulting scheme is $(b, c_1 c_2)$-balanced.

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.
- $(b, 1-2kb)$-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.
- $(b, 1-2kb)$-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 obtaind for

$$c = b \cdot \underbrace{\left(\frac{1-e^{-b}}{b}\right)^k}_{\text{matroids}} \cdot \underbrace{(1-\epsilon)^\ell}_{\text{knapsacks}} \overset{b=1/k}{=} \Omega(1/k).$$

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

Outline

Very short introduction to matroids I

Definition: Matroid

A matroid $M = (N, \mathcal{F})$ consists of a finite ground set N and a non-empty family $\mathcal{F} \subseteq 2^N$ of subsets of N such that:

i) If $I \in \mathcal{F}$ and $J \subseteq I$, then $J \in \mathcal{F}$.

ii) If $I, J \in \mathcal{F}$ and $|I| > |J|$, then $\exists i \in I \setminus J$ with $J \cup \{i\} \in \mathcal{F}$.

- The sets in $\mathcal{F}$ are called independent sets and are typically described by an independence oracle.
- Maximal independent sets are called bases.
  $\rightarrow$ Because of ii) all bases of a matroid have the same cardinality.

Example: graphic matroid

Let $G = (V, E)$ be an undirected graph. The graphic matroid of G is defined to be $M = (E, \mathcal{F})$, where $\mathcal{F}$ is the set of all forests of G.

- Greedy algorithm finds a maximum weight independent set.

Very short introduction to matroids I

Definition: Matroid

A matroid $M = (N, \mathcal{F})$ consists of a finite ground set N and a non-empty family $\mathcal{F} \subseteq 2^N$ of subsets of N such that:

i) If $I \in \mathcal{F}$ and $J \subseteq I$, then $J \in \mathcal{F}$.

ii) If $I, J \in \mathcal{F}$ and $|I| > |J|$, then $\exists i \in I \setminus J$ with $J \cup \{i\} \in \mathcal{F}$.

- The sets in $\mathcal{F}$ are called independent sets and are typically described by an independence oracle.
- Maximal independent sets are called bases.
  $\rightarrow$ Because of *ii*) all bases of a matroid have the same cardinality.

Example: graphic matroid

Let $G = (V, E)$ be an undirected graph. The graphic matroid of G is defined to be $M = (E, \mathcal{F})$, where $\mathcal{F}$ is the set of all forests of G.

- Greedy algorithm finds a maximum weight independent set.

Very short introduction to matroids I

Definition: Matroid

A matroid $M = (N, \mathcal{F})$ consists of a finite ground set N and a non-empty family $\mathcal{F} \subseteq 2^N$ of subsets of N such that:

i) If $I \in \mathcal{F}$ and $J \subseteq I$, then $J \in \mathcal{F}$.

ii) If $I, J \in \mathcal{F}$ and $|I| > |J|$, then $\exists i \in I \setminus J$ with $J \cup \{i\} \in \mathcal{F}$.

- The sets in $\mathcal{F}$ are called independent sets and are typically described by an independence oracle.
- Maximal independent sets are called bases.
  $\rightarrow$ Because of *ii*) all bases of a matroid have the same cardinality.

Example: graphic matroid

Let $G = (V, E)$ be an undirected graph. The graphic matroid of G is defined to be $M = (E, \mathcal{F})$, where $\mathcal{F}$ is the set of all forests of G.

- Greedy algorithm finds a maximum weight independent set.

Very short introduction to matroids II

The rank function $r : 2^N \to \mathbb{Z}_+$ of a matroid $M = (N, \mathcal{F})$ is defined by:

$$r(A) = \max\{|I| \mid I \subseteq A, I \in \mathcal{F}\}$$

(BTW, this function is also submodular)

An optimal CR scheme for matroids (I)

We show how to obtain a monotone $(1 - e^{-1})$-balanced CR scheme. (getting a $(b, \frac{1-e^{-b}}{b})$-balanced CR scheme is analogous)

An optimal CR scheme for matroids (I)

We show how to obtain a monotone $(1 - e^{-1})$-balanced CR scheme. (getting a $(b, \frac{1-e^{-b}}{b})$-balanced CR scheme is analogous)

- Let Π be the set of all monotone deterministic CR scheme $\pi : 2^N \to \mathcal{F}$.
- We create the best CR scheme being a convex combination of monotone deterministic schemes $\to$ leads to a monotone CR scheme.

An optimal CR scheme for matroids (I)

We show how to obtain a monotone $(1-e^{-1})$-balanced CR scheme. (getting a $(b, \frac{1-e^{-b}}{b})$-balanced CR scheme is analogous)

- Let Π be the set of all monotone deterministic CR scheme $\pi : 2^N \to \mathcal{F}$.
- We create the best CR scheme being a convex combination of monotone deterministic schemes $\to$ leads to a monotone CR scheme.

Let $q_{i,\pi} := \Pr_R[i \in \pi(R(x))]$.

$$
\text{(LP1)} \quad \begin{array}{lrcll}
\max & c & & & \\
s.t. & \sum_{\pi \in \Pi} q_{i,\pi} \lambda_\pi & \geq & x_i c & \forall i \in N \\
 & \sum_{\pi \in \Pi} \lambda_\pi & = & 1 & \\
 & \lambda_\pi & \geq & 0 & \forall \pi \in \Pi
\end{array}
$$

An optimal CR scheme for matroids (I)

We show how to obtain a monotone $(1-e^{-1})$-balanced CR scheme. (getting a $(b, \frac{1-e^{-b}}{b})$-balanced CR scheme is analogous)

- Let Π be the set of all monotone deterministic CR scheme $\pi : 2^N \to \mathcal{F}$.
- We create the best CR scheme being a convex combination of monotone deterministic schemes $\to$ leads to a monotone CR scheme.

Let $q_{i,\pi} := \Pr_R[i \in \pi(R(x))]$.

$$
\boxed{
(\text{LP1}) \quad
\begin{array}{llcll}
\max & c & & & \\
s.t. & \sum_{\pi\in\Pi} q_{i,\pi}\lambda_\pi & \geq & x_i c & \forall i \in N \\
 & \sum_{\pi\in\Pi} \lambda_\pi & = & 1 & \\
 & \lambda_\pi & \geq & 0 & \forall \pi \in \Pi
\end{array}
}
$$

$$
\boxed{
(\text{DP1}) \quad
\begin{array}{llcll}
\min & \mu & & & \\
\text{s.t.} & \sum_{i\in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\
 & \sum_{i\in N} x_i y_i & = & 1 & \\
 & y_i & \geq & 0 & \forall i \in N
\end{array}
}
$$

An optimal CR scheme for matroids (II)

$$
(\mathrm{DP1}) \quad \begin{array}{rrcll}
\min & \mu & & & \\
\text{s.t.} & \sum_{i \in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\
& \sum_{i \in N} x_i y_i & = & 1 & \\
& y_i & \geq & 0 & \forall i \in N
\end{array}
$$

We can (approximately) separate over the dual

- Goal: find $\pi \in \Pi$ maximizing

$$
\sum_{i \in N} q_{i,\pi} y_i = \sum_{i \in N} \Pr[i \in \pi(R(x))] \cdot y_i = \mathbf{E}\left[\sum_{i \in N} \mathbf{1}_{i \in \pi(R(x))} \cdot y_i\right] = \mathbf{E}\left[\sum_{i \in \pi(R(x))} y_i\right].
$$

An optimal CR scheme for matroids (II)

$$\begin{array}{llrcll} & \min & & \mu & & \\ (\text{DP1}) & \text{s.t.} & \sum_{i \in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\ & & \sum_{i \in N} x_i y_i & = & 1 & \\ & & y_i & \geq & 0 & \forall i \in N \end{array}$$

We can (approximately) separate over the dual

- Goal: find $\pi \in \Pi$ maximizing

$$\sum_{i \in N} q_{i,\pi} y_i = \sum_{i \in N} \Pr[i \in \pi(R(x))] \cdot y_i = \mathbf{E}\left[\sum_{i \in N} \mathbf{1}_{i \in \pi(R(x))} \cdot y_i\right] = \mathbf{E}\left[\sum_{i \in \pi(R(x))} y_i\right].$$

- $\sum_{i \in \pi(R(x))} y_i$ has to be maximized $\forall R(x) \subseteq N$.

An optimal CR scheme for matroids (II)

$$
\text{(DP1)} \quad \begin{array}{llcll}
\min & \mu & & & \\
\text{s.t.} & \sum_{i \in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\
& \sum_{i \in N} x_i y_i & = & 1 & \\
& y_i & \geq & 0 & \forall i \in N
\end{array}
$$

We can (approximately) separate over the dual

- Goal: find $\pi \in \Pi$ maximizing

$$
\sum_{i \in N} q_{i,\pi} y_i = \sum_{i \in N} \Pr[i \in \pi(R(x))] \cdot y_i = \mathbf{E}\left[\sum_{i \in N} \mathbf{1}_{i \in \pi(R(x))} \cdot y_i\right] = \mathbf{E}\left[\sum_{i \in \pi(R(x))} y_i\right].
$$

- $\sum_{i \in \pi(R(x))} y_i$ has to be maximized $\forall R(x) \subseteq N$.
- π =greedy algorithm w.r.t. weights y maximizes it even among all CR schemes, including non-deterministic ones.

An optimal CR scheme for matroids (II)

$$
\text{(DP1)} \quad \begin{array}{rrcll} \min & \mu & & & \\ \text{s.t.} & \sum_{i\in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\ & \sum_{i\in N} x_i y_i & = & 1 & \\ & y_i & \geq & 0 & \forall i \in N \end{array}
$$

We can (approximately) separate over the dual

- Goal: find $\pi \in \Pi$ maximizing

$$
\sum_{i\in N} q_{i,\pi} y_i = \sum_{i\in N} \Pr[i \in \pi(R(x))] \cdot y_i = \mathbf{E}\left[\sum_{i\in N} \mathbf{1}_{i\in\pi(R(x))} \cdot y_i\right] = \mathbf{E}\left[\sum_{i\in\pi(R(x))} y_i\right].
$$

- $\sum_{i\in\pi(R(x))} y_i$ has to be maximized $\forall R(x) \subseteq N$.
- π =greedy algorithm w.r.t. weights y maximizes it even among all CR schemes, including non-deterministic ones.
- Greedy algorithms are indeed deterministic and monotone.

An optimal CR scheme for matroids (II)

$$
(\mathrm{DP1}) \quad \begin{array}{rrcll} \min & \mu & & & \\ \text{s.t.} & \sum_{i\in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\ & \sum_{i\in N} x_i y_i & = & 1 & \\ & y_i & \geq & 0 & \forall i \in N \end{array}
$$

We can (approximately) separate over the dual

- Goal: find $\pi \in \Pi$ maximizing

$$
\sum_{i\in N} q_{i,\pi} y_i = \sum_{i\in N} \Pr[i \in \pi(R(x))] \cdot y_i = \mathbf{E}\left[\sum_{i\in N} \mathbf{1}_{i\in\pi(R(x))} \cdot y_i\right] = \mathbf{E}\left[\sum_{i\in\pi(R(x))} y_i\right].
$$

- $\sum_{i\in\pi(R(x))} y_i$ has to be maximized $\forall R(x) \subseteq N$.
- π =greedy algorithm w.r.t. weights y maximizes it even among all CR schemes, including non-deterministic ones.
- Greedy algorithms are indeed deterministic and monotone.

$\Rightarrow$ The best monotone CR scheme for matroids is a convex combination of greedy CR schemes, and we can find it (approximately).

An optimal CR scheme for matroids (II)

$$
\text{(DP1)} \quad \begin{array}{llcll}
\min & \mu & & & \\
\text{s.t.} & \sum_{i\in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\
& \sum_{i\in N} x_i y_i & = & 1 & \\
& y_i & \geq & 0 & \forall i \in N
\end{array}
$$

We can (approximately) separate over the dual

- Goal: find $\pi \in \Pi$ maximizing

$$\sum_{i\in N} q_{i,\pi} y_i = \sum_{i\in N} \Pr[i \in \pi(R(x))] \cdot y_i = \mathbf{E}\left[\sum_{i\in N} \mathbf{1}_{i\in\pi(R(x))} \cdot y_i\right] = \mathbf{E}\left[\sum_{i\in\pi(R(x))} y_i\right].$$

- $\displaystyle\sum_{i\in\pi(R(x))} y_i$ has to be maximized $\forall R(x) \subseteq N$.
- π =greedy algorithm w.r.t. weights y maximizes it even among all CR schemes, including non-deterministic ones.
- Greedy algorithms are indeed deterministic and monotone.

$\Rightarrow$ The best monotone CR scheme for matroids is a convex combination of greedy CR schemes, and we can find it (approximately).

How good is it?

An optimal CR scheme for matroids (III)

$$
\text{(DP1)} \quad \begin{array}{lrcll} \min & \mu & & & \\ \text{s.t.} & \sum_{i \in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\ & \sum_{i \in N} x_i y_i & = & 1 & \\ & y_i & \geq & 0 & \forall i \in N \end{array}
$$

Goal

- We want to show that optimal dual value is $\geq 1 - e^{-1}$.
- This is optimal: easy to find examples showing that $\nexists$ $(1 - e^{-1} - \epsilon)$-balanced CR scheme. (e.g. uniform matroid of rank one with $x_i = 1/n$ for $i \in N$)

An optimal CR scheme for matroids (III)

$$
\text{(DP1)} \quad
\begin{array}{rrcll}
\min & \mu & & & \\
\text{s.t.} & \sum_{i \in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\
 & \sum_{i \in N} x_i y_i & = & 1 & \\
 & y_i & \geq & 0 & \forall i \in N
\end{array}
$$

Goal

- We want to show that optimal dual value is $\geq 1 - e^{-1}$.
- This is optimal: easy to find examples showing that $\nexists$ $(1 - e^{-1} - \epsilon)$-balanced CR scheme. (e.g. uniform matroid of rank one with $x_i = 1/n$ for $i \in N$)

Proof procedure

We show that for any dual-feasible $y \in [0,1]^N$, $\exists \pi \in \Pi$ with $\sum_{i \in N} q_{i,\pi} y_i \geq 1 - e^{-1}$.

An optimal CR scheme for matroids (III)

$$\text{(DP1)} \quad \begin{array}{llcll} \min & \mu & & & \\ \text{s.t.} & \sum_{i\in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\ & \sum_{i\in N} x_i y_i & = & 1 & \\ & y_i & \geq & 0 & \forall i \in N \end{array}$$

Goal

- We want to show that optimal dual value is $\geq 1 - e^{-1}$.
- This is optimal: easy to find examples showing that $\nexists$ $(1 - e^{-1} - \epsilon)$-balanced CR scheme. (e.g. uniform matroid of rank one with $x_i = 1/n$ for $i \in N$)

Proof procedure

We show that for any dual-feasible $y \in [0,1]^N$, $\exists \pi \in \Pi$ with $\sum_{i\in N} q_{i,\pi} y_i \geq 1 - e^{-1}$.

- Let $y \in [0,1]^N$ be dual-feasible, we choose $\pi \in \Pi$ to be the greedy algorithm w.r.t. the weights y.

$$\sum_{i\in N} q_{i,\pi} y_i = \mathbf{E}\left[\sum_{i\in\pi(R(x))} y_i\right] = \mathbf{E}[r_y(R(x))],$$

where r_y is the y-weighted rank function of the underlying matroid.

An optimal CR scheme for matroids (IV)

$$
\boxed{
\begin{array}{lllcll}
 & \min & \mu & & & \\
(\text{DP1}) & \text{s.t.} & \sum_{i \in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\
 & & \sum_{i \in N} x_i y_i & = & 1 & \\
 & & y_i & \geq & 0 & \forall i \in N
\end{array}
}
$$

$$
\sum_{i \in N} q_{i,\pi} y_i = \mathbf{E}\left[\sum_{i \in \pi(R(x))} y_i \right] = \mathbf{E}[r_y(R(x))]
$$

An optimal CR scheme for matroids (IV)

$$
\text{(DP1)} \quad \begin{array}{llcll}
\min & \mu & & & \\
\text{s.t.} & \sum_{i \in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\
& \sum_{i \in N} x_i y_i & = & 1 & \\
& y_i & \geq & 0 & \forall i \in N
\end{array}
$$

$$
\sum_{i \in N} q_{i,\pi} y_i = \mathbf{E}\left[\sum_{i \in \pi(R(x))} y_i\right] = \mathbf{E}[r_y(R(x))] \geq (1 - e^{-1}) \sum_{i \in N} x_i y_i
$$

Theorem (Calinescu et al., 2007; Vondrák, 2007)

Let $r_w : 2^N \to \mathbb{R}_+$ be the weighted rank function of a matroid $M = (N, \mathcal{I})$, with weights $w : N \to \mathbb{R}_+$, and let $v \in P_M$ be a point in the matroid polytope. Then

$$
\mathbf{E}[r_w(R(v))] \geq (1 - e^{-1}) \sum_{i \in N} v_i w_i .
$$

An optimal CR scheme for matroids (IV)

$$
\text{(DP1)} \quad \begin{array}{llcll}
\min & \mu & & & \\
\text{s.t.} & \sum_{i \in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\
& \sum_{i \in N} x_i y_i & = & 1 & \\
& y_i & \geq & 0 & \forall i \in N
\end{array}
$$

$$\sum_{i \in N} q_{i,\pi} y_i = \mathbf{E}\left[\sum_{i \in \pi(R(x))} y_i \right] = \mathbf{E}[r_y(R(x))] \geq (1 - e^{-1}) \sum_{i \in N} x_i y_i = 1 - e^{-1}.$$

Theorem (Calinescu et al., 2007; Vondrák, 2007)

Let $r_w : 2^N \to \mathbb{R}_+$ be the weighted rank function of a matroid $M = (N, \mathcal{I})$, with weights $w : N \to \mathbb{R}_+$, and let $v \in P_M$ be a point in the matroid polytope. Then

$$\mathbf{E}[r_w(R(v))] \geq (1 - e^{-1}) \sum_{i \in N} v_i w_i.$$

An optimal CR scheme for matroids (IV)

$$
\text{(DP1)} \quad \begin{array}{lrcll}
\min & \mu & & & \\
\text{s.t.} & \sum_{i \in N} q_{i,\pi} y_i & \leq & \mu & \forall \pi \in \Pi \\
& \sum_{i \in N} x_i y_i & = & 1 & \\
& y_i & \geq & 0 & \forall i \in N
\end{array}
$$

$$\sum_{i \in N} q_{i,\pi} y_i = \mathbf{E}\left[\sum_{i \in \pi(R(x))} y_i \right] = \mathbf{E}[r_y(R(x))] \geq (1 - e^{-1}) \sum_{i \in N} x_i y_i = 1 - e^{-1}.$$

Theorem (Calinescu et al., 2007; Vondrák, 2007)

Let $r_w : 2^N \to \mathbb{R}_+$ be the weighted rank function of a matroid $M = (N, \mathcal{I})$, with weights $w : N \to \mathbb{R}_+$, and let $v \in P_M$ be a point in the matroid polytope. Then

$$\mathbf{E}[r_w(R(v))] \geq (1 - e^{-1}) \sum_{i \in N} v_i w_i.$$

- Hence, the optimal dual value is at least $1 - e^{-1}$.
- $\Rightarrow \exists$ a $(1 - e^{-1})$-balanced and monotone CR-scheme for matroids.

Outline

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?
- Convex combinations of monotone deterministic CR schemes are in general not as powerful as randomized CR schemes. How much do we lose?
- What about other extensions than the multilinear one?
- Derandomization?

Thank you!

References I

Bansal, N., Korula, N., Nagarajan, V., and Srinivasan, A. (2010). On k-column sparse packing programs. In Eisenbrand, F. and Shepherd, F., editors, *Integer Programming and Combinatorial Optimization*, volume 6080 of *Lecture Notes in Computer Science*, pages 369–382. Springer Berlin / Heidelberg. 10.1007/978-3-642-13036-6_28.

Calinescu, G., Chekuri, C., Pál, M., and Vondrák, J. (2007). Maximizing a submodular set function subject to a matroid constraint. In *IPCO '07: Proceedings of the 12th international conference on Integer Programming and Combinatorial Optimization*, pages 182–196, Berlin, Heidelberg. Springer-Verlag.

Calinescu, G., Chekuri, C., Pál, M., and Vondrák, J. (2011). Maximizing a monotone submodular function subject to a matroid constraint. To appear in SIAM Journal on Computing.

Feige, U., Mirrokni, V. S., and Vondrák, J. (2007). Maximizing non-monotone submodular functions. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 461–471, Washington, DC, USA. IEEE Computer Society.

References II

Kulik, A., Shachnai, H., and Tamir, T. (2011). Approximations for monotone and non-monotone submodular maximization with knapsack constraints. http://arxiv.org/abs/1101.2940/v1.

Lee, J., Sviridenko, M., and Vondrák, J. (2009). Submodular maximization over multiple matroids via generalized exchange properties. In *APPROX '09 / RANDOM '09: Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 244–257, Berlin, Heidelberg. Springer-Verlag.

Oveis Gharan, S. and Vondrák, J. (2011). Submodular function maximization by simulated annealing. In *SODA '11: Proceedings of the Twenty-First Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 1098–1116.

Vondrák, J. (2007). *Submodularity in Combinatorial Optimization*. PhD thesis, Charles University.