Minimizing submodular functions on diamonds via generalized fractional matroid matchings

Tamás Király
joint work with Satoru Fujishige, Kazuhisa Makino, Kenjiro Takazawa, and Shin-ichi Tanigawa

RIMS, Kyoto University
MTA-ELTE Egerváry Research Group, Budapest

Shonan Meeting on Combinatorial Optimization, April 2016

1. First part
   - Submodular minimization and generalizations
   - Submodular functions on diamonds
   - Polyhedra
   - Fractional matroid matching
2. Second part (optional)
   - Pseudopolynomial algorithm for optimization on $P^{=}(f)$
   - Polynomial algorithm via scaling
3. Open questions

1. First part
   - Submodular minimization and generalizations
   - Submodular functions on diamonds
   - Polyhedra
   - Fractional matroid matching

2. Second part (optional)
   - Pseudopolynomial algorithm for optimization on $P^{=}(f)$
   - Polynomial algorithm via scaling

3. Open questions

Definition

A set function $f : 2^V \to \mathbb{Z}$ with $f(\emptyset) = 0$ is submodular if $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$ for every $X, Y \subseteq V$.

- Evaluation oracle: given $X \subseteq V$, returns $f(X)$
- $n = |V|$
- M is a known upper bound on $|f(X)|$

Submodular function minimization problem

Find $\min_{X \subseteq V} f(X)$ in time polynomial in $\max\{n, \log M\}$.

Algorithm using the ellipsoid method: Grötschel, Lovász, Schrijver, 1981

- $P(f) = \{x \in \mathbb{R}^V : x(Z) \leq f(Z)\ \forall Z \subseteq V\}$
- Optimization on $P(f)$ by greedy algorithm
- Equivalence of separation and optimization

Algorithm using the ellipsoid method: Grötschel, Lovász, Schrijver, 1981

- $P(f) = \{x \in \mathbb{R}^V : x(Z) \leq f(Z)\ \forall Z \subseteq V\}$
- Optimization on $P(f)$ by greedy algorithm
- Equivalence of separation and optimization

More efficient algorithm based on the Lovász extension

Huge recent improvement on running time: $O(n^3 \log^{O(1)} nM)$ [Lee, Sidford, Wong 2015]

Algorithm using the ellipsoid method: Grötschel, Lovász, Schrijver, 1981

- $P(f) = \{x \in \mathbb{R}^V : x(Z) \leq f(Z)\ \forall Z \subseteq V\}$
- Optimization on $P(f)$ by greedy algorithm
- Equivalence of separation and optimization

More efficient algorithm based on the Lovász extension

Huge recent improvement on running time: $O(n^3 \log^{O(1)} nM)$ [Lee, Sidford, Wong 2015]

Combinatorial algorithms, 1999

- Schrijver and independently Iwata, Fleischer, Fujishige
- Both based on the characterization $\min_{Z \subseteq V} f(Z) = \max\{x^-(V) : x \in P(f)\}$

Submodular function on the product of lattices

Let $\mathcal{L}$ be a finite lattice.

Definition

A function $f : \mathcal{L} \to \mathbb{Z}$ with $f(0_{\mathcal{L}}) = 0$ is submodular if $f(a) + f(b) \geq f(a \wedge b) + f(a \vee b)$ for every $a, b \in \mathcal{L}$.

The submodular minimization problem for an explicitly given lattice is trivial: we can evaluate on all elements

Submodular function on the product of lattices

Let $\mathcal{L}$ be a finite lattice.

Definition

A function $f : \mathcal{L} \to \mathbb{Z}$ with $f(0_{\mathcal{L}}) = 0$ is submodular if $f(a) + f(b) \geq f(a \wedge b) + f(a \vee b)$ for every $a, b \in \mathcal{L}$.

The submodular minimization problem for an explicitly given lattice is trivial: we can evaluate on all elements

Submodular minimization over product of lattices

- Lattices $\mathcal{L}_1, \ldots, \mathcal{L}_n$ with ground sets $U_1, \ldots, U_n$
- $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$
- f: submodular function on $\mathcal{L}$ given by an evaluation oracle
- Find $\min_{a \in \mathcal{L}} f(a)$ in time polynomial in $\max\{\sum |U_i|, \log M\}$

Submodular function on the product of lattices

Let $\mathcal{L}$ be a finite lattice.

Definition

A function $f : \mathcal{L} \to \mathbb{Z}$ with $f(0_{\mathcal{L}}) = 0$ is submodular if $f(a) + f(b) \geq f(a \wedge b) + f(a \vee b)$ for every $a, b \in \mathcal{L}$.

The submodular minimization problem for an explicitly given lattice is trivial: we can evaluate on all elements

Submodular minimization over product of lattices

- Lattices $\mathcal{L}_1, \ldots, \mathcal{L}_n$ with ground sets $U_1, \ldots, U_n$
- $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$
- f: submodular function on $\mathcal{L}$ given by an evaluation oracle
- Find $\min_{a \in \mathcal{L}} f(a)$ in time polynomial in $\max\{\sum |U_i|, \log M\}$

Standard submodular minimization: $\mathcal{L}_i = \{0_i, 1_i\}$ ($i \in [n]$)

Solvable if

- every $\mathcal{L}_i$ is distributive
- every $\mathcal{L}_i$ is distributive or the pentagon [Krokhin, Larose 2008]
- f is the sum of submodular functions with constant arity (constant number of components affect the function value) [Thapper, Živný 2012]

Latter is also true for modular semi-lattices [Hirai 2012], which includes k-submodular functions

Previous results

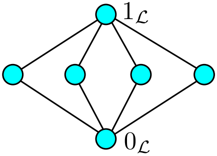
Solvable if

- every $\mathcal{L}_i$ is distributive
- every $\mathcal{L}_i$ is distributive or the pentagon [Krokhin, Larose 2008]
- f is the sum of submodular functions with constant arity (constant number of components affect the function value) [Thapper, Živný 2012]

Latter is also true for modular semi-lattices [Hirai 2012], which includes k-submodular functions

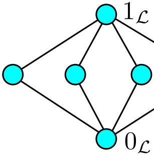
- Bisubmodular minimization is polynomial-time solvable [Qi 1988, Fujishige, Iwata 2006]

Diamond: finite complemented modular lattice of rank 2



middle elements

Diamond: finite complemented modular lattice of rank 2

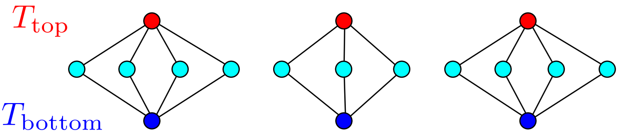


middle elements

Previous results for diamonds

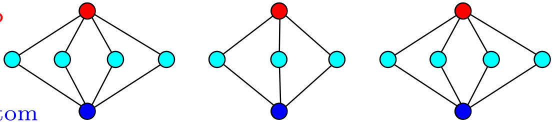
- If there are two middle elements: distributive lattice
- Polynomial-time minimization for sum of constant arity functions [Krokhin, Larose 2008]
- Kuivinen 2011: pseudo-polynomial algorithm (polynomial in $\max\{\sum |U_i|, M\}$) using the ellipsoid method
- Kuivinen 2011: good characterization based on a polyhedral characterization of the optimum

The elements of the product lattice are transversals of the family $\{U_i\}_{i=1}^{n}$. Transversals corresponding to $0_{\mathcal{L}}$ and $1_{\mathcal{L}}$:



The elements of the product lattice are transversals of the family $\{U_i\}_{i=1}^{n}$. Transversals corresponding to $0_{\mathcal{L}}$ and $1_{\mathcal{L}}$:

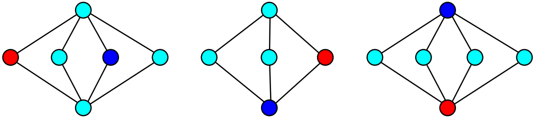
T_{top}



T_{bottom}

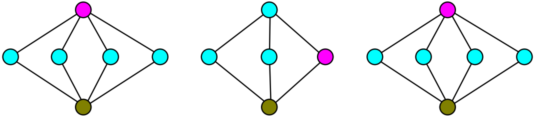
Meet, join:

T_1, T_2



$T_1 \vee T_2$

$T_1 \wedge T_2$



Rank vector of transversals:

$$a(T)_i = \begin{cases} 0 \text{ if } T \cap U_i = 0_i \\ 2 \text{ if } T \cap U_i = 1_i \\ 1 \text{ otherwise} \end{cases}$$

$$P(f) = \{x \in \mathbb{R}^n : a(T)x \leq f(T)\ \forall T \in \mathcal{T}\}$$

$$P^{=}(f) = \{x \in \mathbb{R}^n : a(T)x \leq f(T)\ \forall T \in \mathcal{T},\ 2\sum x_i = f(T_{\text{top}})\}$$

Lemma

The linear systems of $P(f)$ and $P^{=}(f)$ are half-TDI.

Our goal: optimization on $P^{=}(f)$

Why is optimization over $P^{=}(f)$ useful?

- Optimization over $P^{=}(f) \rightsquigarrow$ optimization over $P(f)$ by binary search
  (f remains submodular if we decrease $f(T_{\text{top}})$)

Why is optimization over $P^{=}(f)$ useful?

- Optimization over $P^{=}(f) \rightsquigarrow$ optimization over $P(f)$ by binary search
  (f remains submodular if we decrease $f(T_{\text{top}})$)
- Optimization over $P(f) \rightsquigarrow$ separation over $P(f)$
  (ellipsoid method)

Why is optimization over $P^{=}(f)$ useful?

- Optimization over $P^{=}(f)$ $\rightsquigarrow$ optimization over $P(f)$ by binary search
  (f remains submodular if we decrease $f(T_{\text{top}})$)
- Optimization over $P(f)$ $\rightsquigarrow$ separation over $P(f)$
  (ellipsoid method)
- Separation over $P(f)$ $\rightsquigarrow$ minimization of f by binary search
  (f remains submodular if increased by a constant everywhere except for T_{bottom})

Fractional matroid matching problem (Vande Vate)

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

$L_1, \ldots, L_n$: disjoint subsets of rank 2

$a(Z)_i$: rank of $Z \cap L_i$

Definition

$x \in \mathbb{R}_+^n$ is a fractional matroid matching if

$$a(Z)x \leq r(Z) \text{ for every } Z \subseteq S.$$

Fractional matroid matching problem (Vande Vate)

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

$L_1, \ldots, L_n$: disjoint subsets of rank 2

$a(Z)_i$: rank of $Z \cap L_i$

Definition

$x \in \mathbb{R}^n_+$ is a fractional matroid matching if

$$a(Z)x \leq r(Z) \text{ for every } Z \subseteq S.$$

- an integer solution corresponds to a matroid matching
- Chang, LLewellyn, Vande Vate 2001: maximum size fractional matroid matching in polynomial time
- Gijswijt, Pap, 2008: maximum weight fractional matroid matching in polynomial time

Fractional matroid matching and diamonds

$(\mathcal{M}; L_1, \ldots, L_n)$: an instance of fractional matroid matching

- $\mathcal{L}_i$: diamond with middle elements L_i

Fractional matroid matching and diamonds

$(\mathcal{M}; L_1, \ldots, L_n)$: an instance of fractional matroid matching

- $\mathcal{L}_i$: diamond with middle elements L_i
- For a transversal T, let

$$Z_T^i = \begin{cases} L_i & \text{if } 1_i \in T \\ T \cap L_i & \text{otherwise} \end{cases}$$

$$Z_T = \cup_{i=1}^n Z_T^i.$$

Fractional matroid matching and diamonds

$(\mathcal{M}; L_1, \ldots, L_n)$: an instance of fractional matroid matching

- $\mathcal{L}_i$: diamond with middle elements L_i
- For a transversal T, let

$$Z_T^i = \begin{cases} L_i & \text{if } 1_i \in T \\ T \cap L_i & \text{otherwise} \end{cases}$$

$$Z_T = \cup_{i=1}^n Z_T^i.$$

- Let f be defined as $f(T) = r(Z_T)$

Fractional matroid matching and diamonds

$(\mathcal{M}; L_1, \ldots, L_n)$: an instance of fractional matroid matching

- $\mathcal{L}_i$: diamond with middle elements L_i
- For a transversal T, let

$$Z_T^i = \begin{cases} L_i & \text{if } 1_i \in T \\ T \cap L_i & \text{otherwise} \end{cases}$$

$$Z_T = \cup_{i=1}^n Z_T^i.$$

- Let f be defined as $f(T) = r(Z_T)$

Proposition

f is submodular on $\mathcal{L}_1 \times \cdots \times \mathcal{L}_n$, and the nonnegative vectors in $P(f)$ are the fractional matroid matchings.

Our aim is to solve the LP

$$\max\{cx : x \in P^{=}(f)\}.$$

Dual problem:

$$\begin{aligned} \min \quad & \sum_{T \in \mathcal{T}} f(T) y_T & \\ s.t. \quad & y_T \geq 0 & \forall T \in \mathcal{T} \setminus \{T_{\text{top}}\} \\ & \sum_{T \in \mathcal{T}} a(T)_i y_T = c_i & \forall i \in [n] \end{aligned}$$

Lemma

The dual problem has a half-integral optimal solution with chain support. If c has distinct values, then the chain is "dense"

Polyhedron associated to a chain

y: dual solution with chain support $\mathcal{C} = \{T_1, \ldots, T_k = T_{\text{top}}\}$

$$P(f,\mathcal{C}) = \{x \in \mathbb{R}^n : (a(T) - a(T_{j-1}))x \leq f(T) - f(T_{j-1}) \quad \forall T \in [T_{j-1}, T_j] \; \forall j \in [k]\}.$$

- System for $P(f,\mathcal{C})$ is half-TDI
- If $\mathcal{C}$ is dense, then we can separate over $P(f,\mathcal{C})$ in polynomial time
- $P(f,\mathcal{C}) \subseteq P(f)$ because of submodularity
- If $x \in P(f,\mathcal{C}) \cap P^{=}(f)$, then $a(T_j)x = f(T_j)$ for every $j \in [k]$, thus x and y are optimal.

Pseudo-polynomial algorithm (?)

- Start with a dual solution y with dense chain support $\mathcal{C}$
- Maximize $\sum x_i$ over $P(f, \mathcal{C})$
- If the maximum is $f(T_{\text{top}})/2$, then we are done
- Otherwise the optimal x gives an improving direction for y

Problem: It seems hard to prove a polynomial bound on the number of dual improvements.

Pseudo-polynomial algorithm (?)

- Start with a dual solution y with dense chain support $\mathcal{C}$
- Maximize $\sum x_i$ over $P(f, \mathcal{C})$
- If the maximum is $f(T_{\mathrm{top}})/2$, then we are done
- Otherwise the optimal x gives an improving direction for y

Problem: It seems hard to prove a polynomial bound on the number of dual improvements.

Solution (based on the ideas of Gijswijt and Pap)

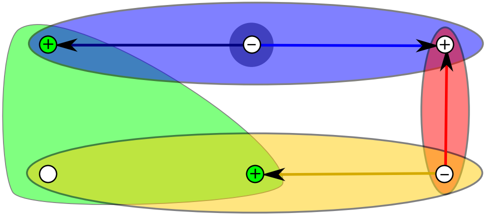
Instead of bounding the number of dual improvements, we show that $\max\{\sum x_i : x \in P(f, \mathcal{C})\}$ increases after $O(n^3)$ iterations.

Combinatorial algorithm for $\max\{\sum x_i : x \in P(f, \mathcal{C})\}$

A dense chain $\mathcal{C}$ defines an almost 2-regular hypergraph on ground set $V = [n]$

Augmenting walk: $x(V)$ can be increased by alternately increasing and decreasing along the walk

$H = (V, \mathcal{E})$

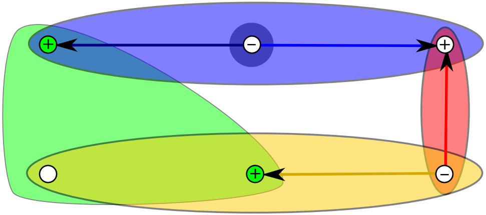


Combinatorial algorithm for $\max\{\sum x_i : x \in P(f, \mathcal{C})\}$

A dense chain $\mathcal{C}$ defines an almost 2-regular hypergraph on ground set $V = [n]$

Augmenting walk: $x(V)$ can be increased by alternately increasing and decreasing along the walk

$H = (V, \mathcal{E})$



- This is a generalization of matroid intersection
- Although there is a half-integral solution, we cannot guarantee half-integral improvement
- Lexicographic shortest walk: polynomial running time

Polynomial algorithm via scaling

We use the scaled functions

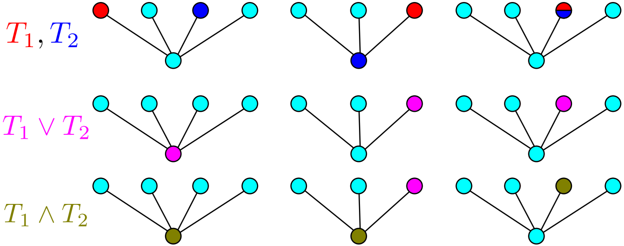
$$f^k(T) = \left\lceil \frac{f(T)}{2^k} \right\rceil - (a(T)\mathbf{1})^2 \qquad \text{for } k = 0, 1 \ldots, \lceil \log M \rceil$$

- f^k is submodular on the diamonds
- if $P^{=}(f^k) = \emptyset$, then $P^{=}(f) = \emptyset$
- if $x \in P(f^k, \mathcal{C})$, then $2x \in P(f^{k-1}, \mathcal{C})$
- if $x \in P(f^0, \mathcal{C})$, then $x \in P(f, \mathcal{C})$

We can sucessively compute $\max\{cx : x \in P^{=}(f^k)\}$ for $\lceil \log M \rceil, \lceil \log M \rceil - 1, \ldots, 0$, and then for f.

Open questions

- Strongly polynomial algorithm for $\max\{cx : x \in P(f)\}$
- Combinatorial algorithm for the minimization of f
- Generalization to other modular lattices
- Polynomial algorithm for k-submodular functions



Thank you