CrossMark

Set Function Optimization

Wei-Li Wu[1] · Zhao Zhang[2] · Ding-Zhu Du[1]

Abstract
This article is an introduction to recent development of optimization theory on set functions, the nonsubmodular optimization, which contains two interesting results, DS (difference of submodular) functions decomposition and sandwich theorem, together with iterated sandwich method and data-dependent approximation. Some potential research problems will be mentioned.

1 Introduction

In recent development of computer technology, such as wireless networks [1,2], cloud computing [3,4], sentiment analysis [5–8], and machine learning [9], many nonlinear optimization problems come out with discrete structure. They form a large group of new problems, which belong to a research area, nonlinear combinatorial optimization. The nonlinear combinatorial optimization has been studied for a long time, but recently becomes very active. One of the important fields in this area is the set function optimization. Its development can be roughly divided into three periods.

✉ Ding-Zhu Du
   dzdu@utdallas.edu

   Wei-Li Wu
   weiliwu@utdallas.edu

   Zhao Zhang
   hxhzz@sina.com

1   Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA

2   Department of Computer Science, Zhejiang Normal University, Jinhua 321004, Zhejiang, China

The first period is before 2000. The research works came mainly from researchers in operations research. Those works are mainly on submodular function optimization, often with monotone nondecreasing property. For any set function $f : 2^E \to \mathbb{R}$, f is submodular if

$$f(A) + f(B) \geqslant f(A \cup B) + f(A \cap B).$$

f is monotone nondecreasing if

$$A \subset B \text{ implies } f(A) \leqslant f(B).$$

In this period, major results include following:

- Unconstrained submodular minimization can be solved in polynomial time [10–12].
- For constrained monotone nondecreasing submodular maximization, it has $(1 - 1/e)$-approximation with size-constraint [13] or a knapsack constraints [14,15].
- For nonlinear-constrained linear optimization, the linear maximization with k matroid constraints has $(1/(k + 1))$-approximation [16,17], and the linear minimization with submodular cover constraint, called the submodular cover problem, has $(1 + \ln \gamma)$-approximation where γ is a number determined by the submodular function defining the constraint [18].

The second period is from 2007 to 2012, the research activity occurs mainly in the theoretical computer science. The major results are about nonmonotone submodular optimization, including submodular maximization with knapsack constraints and matroid constraints [19–21] and submodular minimization with size-constraint [22]. Most of them were published in theoretical computer science conferences, such as ACM Symposium on Theory of Computing, IEEE Symposium on Foundations of Computer Science, ACM-SIAM Symposium on Discrete Algorithms, and journals, such as SIAM Journal on Computing.

The third period is starting from 2014. The research is in application-driving. The main focus is on nonsubmodular optimization. In the study of nonsubmodular optimization, we may find four clusters of research efforts.

- supermodular degree [23,24].
- Difference of Submodular (DS) functions [25–27].
- Discrete Difference of Convex (DC) functions [28,29].
- Nonlinear integer programming.

In this article, we discuss research works on DS functions, especially introduce two surprising results, the DS decomposition and the sandwich theorem together with the iterated sandwich method.

2 DS Decomposition

The first one is as follows.

Theorem 2.1 [26] *Every set function* $f : 2^X \to \mathbb{R}$ *can be expressed as the difference of two monotone nondecreasing submodular functions g and h, i.e.,* $f = g - h$, *where X is a finite set.*

To prove this theorem, we first show two lemmas.

Lemma 2.1 [25] *Every set function* $f : 2^X \to \mathbb{R}$ *can be expressed as the difference of two submodular functions g and h, i.e.,* $f = g - h$.

Proof Define

$$\alpha(f) = \min_{A \subset B, x \in X \setminus B} (\Delta_x f(A) - \Delta_x f(B)),$$

where $\Delta_x f(A) = f(A \cup \{x\}) - f(A)$. It is well-known that f is submodular if and only if $\alpha(f) \geqslant 0$ (see [26]). If f is submodular, then set $g = f$ and $h = 0$, which meets lemma's requirement. Thus, we may assume $\alpha(f) < 0$. Choose a submodular function h such that $\alpha(h) > 0$. For example, set $h(A) = \sqrt{|A|}$. Then, we have

$$\begin{aligned}
\alpha(h) &= \min_{A \subset B \subseteq X \setminus x} \left(\sqrt{|A|+1} - \sqrt{|A|} - \sqrt{|B|+1} + \sqrt{|B|} \right) \\
&= \min_{A \subset X \setminus x} \left(\sqrt{|A|+1} - \sqrt{|A|} - \sqrt{|A|+2} + \sqrt{|A|+1} \right) \\
&= 2\sqrt{n+1} - \sqrt{n} - \sqrt{n+2} > 0,
\end{aligned}$$

where $n = |X|$. Set $g(A) = f(A) + \frac{|\alpha(f)|}{\alpha(h)} h(A)$. Then, $\alpha(g) \geqslant 0$, i.e., g is submodular and moreover, $f = g - h$.

Lemma 2.2 [30] *Every submodular function g can be expressed as* $g = p + m$ *where p is a polymatroid function (i.e., a monotone nondecreasing submodular function with* $p(\varnothing) = 0$) *and m is a modular function (i.e., for any two sets A and B,* $m(A) + m(B) = m(A \cup B) + m(A \cap B)$).

Proof Define $m(A) = f(\varnothing) - \sum_{x \in A} \Delta_x f(X \setminus x)$ and $p = f - m$. It is easy to verify that m is a modular function. Thus, p is a submodular function. Moreover, $p(\varnothing) = f(\varnothing) - m(\varnothing) = 0$ and for any set A and $x \in X \setminus A$, $\Delta_x p(A) = \Delta_x f(A) - \Delta_x f(X \setminus x) \geqslant 0$, i.e., p is monotone nondecreasing. Therefore, p is a polymatroid function.

Now, we are ready to prove Theorem 2.1.

Proof of Theorem 2.1 By Lemma 2.1, f can be expressed as $f = g - h$ where g and h are submodular functions. By Lemma 2.2, g and h can be expressed as $g = p_g + m_g$ and $h = p_h + m_h$ where p_g and p_h are polymatroid functions, and m_g and m_h are modular functions. Therefore, $f = (p_g + m_g(\varnothing)) - (p_h + m_h(\varnothing)) + m$ where $m = m_g - m_g(\varnothing) - m_h + m_h(\varnothing)$ which is a modular function with $m(\varnothing) = 0$. Define $m^+(x) = \max(0, m(x))$ for $x \in X$ and $m^+(A) = \sum_{x \in A} m^+(x)$ for any set A. Define $m^-(x) = -\min(0, m(x))$ for $x \in X$ and $m^-(A) = \sum_{x \in A} m^-(x)$ for any set

A. Then, $m = m^+ - m^-$ and both m^+ and m^- are monotone nondecreasing modular functions. Set $g' = p_g + m_g(\varnothing) + m^+$ and $h' = p_h + m_h(\varnothing) + m^-$. Then, g' and h' are monotone nondecreasing submodular functions such that $f = g' - h'$.

The following is an example of DS functions.

***Example* 2.1** (Profit Maximization [31]) The profit maximization is a problem in social computing. Consider a social network which is a directed graph $G = (V, E)$ with an information diffusion model m. Usually, an information diffusion process consists of discrete steps. Consider every node has two states, active and inactive. Initially, every node is inactive. The process starts to activate a subset of nodes, called seeds. After seeds become active, they can activate their neighbors based on certain rules of model m. The process ends when no node newly becomes active. Let S be the set of seeds and $I(S)$ the set of active nodes at the end of process. Then, maximization of $|I(S)|$ (or $E(|I(S)|)$ when m is a probabilistic model), called the influence spread, is an important problem, called the influence maximization, in many applications of social networks. However, in viral marketing, seeds are often free samples or coupons for a certain product, i.e., distribution of seeds needs cost. Therefore, the objective function of maximization should be the difference of the influence spread and the seed cost, called the profit. When the seed cost is a submodular function with respect to seed set S, the profit becomes a DS function.

3 Sandwich Theorem

The second surprising result is the following sandwich theorem.

Theorem 3.1 (Sandwich Theorem) *For any set function $f : 2^X \to \mathbb{R}$ and any set $Y \subseteq X$, there are two modular functions $m_u : 2^X \to \mathbb{R}$ and $m_l : 2^X \to \mathbb{R}$ such that $m_u \geqslant f \geqslant m_l$ and $m_u(Y) = f(Y) = m_l(Y)$.*

Why is this result surprising? To explain this, let us look at a property of modular functions.

Lemma 3.1 *For any modular function $m : 2^X \to \mathbb{R}$,*

$$m(A) = m(\varnothing) + \sum_{x \in A}(m(x) - m(\varnothing))$$

for any set $A \subseteq X$.

Proof This lemma can be proved by induction on $|A|$. For $|A| = 1$, it is trivial. For $|A| \geqslant 2$, suppose $y \in A$. Then,

$$m(y) + m(A \backslash y) = m(A) + m(\varnothing).$$

Therefore,

$$m(A) = m(A \backslash y) + (m(y) - m(\varnothing))$$
$$= m(\varnothing) + \sum_{x \in A \backslash y} (m(x) - m(\varnothing)) + (m(y) - m(\varnothing))$$
$$= m(\varnothing) + \sum_{x \in A} (m(x) - m(\varnothing)).$$

This lemma indicates that the modular function is a linear set function. Theorem 3.1 contains two different modular functions passing through the same set and one is always smaller than or equal to the other. This phenomenon cannot occur for continuous linear functions. A continuous linear function with n variables can be expressed as an n-dimensional plane in the $(n + 1)$-dimensional space. A pair of different n-dimensional planes with a point in common cannot have a coordinate along which one is always smaller than or equal to the other. Therefore, this theorem states a special property of the set function.

To prove the sandwich theorem, we show two lemmas.

Lemma 3.2 [26] *For any submodular function $f : 2^X \to \mathbb{R}$ and any set $Y \subseteq X$, there exists a modular function $m_u : 2^X \to \mathbb{R}$ such that $m_u \geqslant f$ and $m_u(Y) = f(Y)$.*

Proof Define

$$m_u(A) = f(Y) + \sum_{j \in A \backslash Y} \Delta_j f(\varnothing) - \sum_{j \in Y \backslash A} \Delta_j f(Y \backslash j).$$

Clearly, m_l is modular and $m_u(Y) = f(Y)$. Next, we show that $m_l \geqslant f$.

Assume $A \backslash Y = \{j_1, \cdots, j_k\}$. Then,

$$f(A) - f(A \cap Y) = \Delta_{j_1} f(A \cap Y) + \Delta_{j_2} f((A \cap Y) \cup \{j_1\})$$
$$+ \cdots + \Delta_{j_k} f((A \cap Y) \cup \{j_1, \cdots, j_{k-1}\})$$
$$\leqslant \sum_{j \in A \backslash Y} \Delta_j f(A \cap Y).$$

Assume $Y \backslash A = \{i_1, .., i_k\}$. Then,

$$f(Y) - f(A \cap Y) = \Delta_{i_1} f(A \cap Y) + \Delta_{i_2} f((A \cap Y) \cup \{i_1\})$$
$$+ \cdots + \Delta_{i_k} f((A \cap Y) \cup \{i_1, \cdots, i_{k-1}\})$$
$$\geqslant \sum_{j \in A \backslash Y} \Delta_j f(Y \backslash j).$$

Therefore,

$$f(A) \leqslant f(Y) + \sum_{j \in A \backslash Y} \Delta_j f(A \cap Y) - \sum_{j \in A \backslash Y} \Delta_j f(Y \backslash j)$$
$$\leqslant f(Y) + \sum_{j \in A \backslash Y} \Delta_j f(\varnothing) - \sum_{j \in A \backslash Y} \Delta_j f(Y \backslash j)$$
$$= m_u(A).$$

Lemma 3.3 [26] *For any submodular function $f : 2^X \to \mathbb{R}$ and any set $Y \subseteq X$, there exists a modular function $m_l : 2^X \to \mathbb{R}$ such that $f \geqslant m_l$ and $f(Y) = m_l(Y)$.*

Proof Put all elements of X into an ordering $X = \{x_1, x_2, \cdots, x_n\}$ such that $Y = \{x_1, x_2, \cdots, x_{|Y|}\}$. Denote $S_i = \{x_1, x_2, \cdots, x_i\}$. Define $m_l(\varnothing) = f(\varnothing)$ and for $\varnothing \neq A \subseteq X$, define

$$m_l(A) = f(\varnothing) + \sum_{x_i \in A} (f(S_i) - f(S_{i-1})).$$

Clearly m_l is modular and

$$m_l(Y) = f(\varnothing) + \sum_{x_i \in Y} (f(S_i) - f(S_{i-1})) = f(Y).$$

Moreover, for any set $A \subseteq X$ with $A \neq \varnothing$, suppose $A = \{x_{i_1}, x_{i_2}, \cdots, x_{i_k}\}$ and then we have

$$
\begin{aligned}
m_l(A) &= f(\varnothing) + (f(S_{i_1}) - f(S_{i_1-1})) + (f(S_{i_2}) - f(S_{i_2-1})) \\
&\quad + \cdots + (f(S_{i_k}) - f(S_{i_k-1})) \\
&\leqslant f(\varnothing) + (f(\{x_{i_1}\}) - f(\varnothing)) + (f(\{x_{i_1}, x_{i_2}\}) - f(\{x_1\})) \\
&\quad + \cdots + (f(A) - f(\{x_{i_1}, \cdots x_{i_{k-1}}\})) \\
&= f(A).
\end{aligned}
$$

Now, we are ready to prove Theorem 3.1.

Proof of Theorem 3.1 By Theorem 2.1, there exist submodular functions g and h such that $f = g - h$. By Lemmas 3.2 and 3.3, there exist modular functions $m_{gu}, m_{gl}, m_{hu}, m_{hl}$ such that $m_{gu} \geqslant g \geqslant m_{gl}, m_{gu}(Y) = g(Y) = m_{gl}(Y), m_{hu} \geqslant g \geqslant m_{hl}$ and $m_{hu}(Y) = h(Y) = m_{hl}(Y)$. Set $m_u = m_{gu} - m_{hl}$ and $m_l = m_{gl} - m_{hu}$. Then, $m_u \geqslant f \geqslant m_l$ and $m_u(Y) = g(Y) - h(Y) = f(Y) = g(Y) - h(Y) = m_l(Y)$.

4 Iterated Sandwich Method

Based on the sandwich theorem, we can design following algorithm for $\min_{A \in 2^X} f(A)$:

Iterated Sandwich Method:

- **Input** a set function $f : 2^X \to \mathbb{R}$.
- Initially, compute a DS decomposition $f = g - h$ and choose an arbitrary set $A \subseteq X$.
- At each iteration, carry out following

  - Compute a modular upper bound m_{gu} and a modular lower bound m_{gl} for g such that $g(A) = m_{gu}(A) = m_{gl}(A)$.

  – Compute a modular upper bound m_{hu} and a modular lower bound m_{hl} for h
    such that $h(A) = m_{hu}(A) = m_{hl}(A)$.
  – Compute $m_u = m_{gu} - m_{hl}$, $m_l = m_{gl} - m_{hu}$ and $m_o = m_{gl} - m_{hl}$.
  – Compute an optimal solution A_u for m_u, an optimal solution A_l for m_l and an
    optimal solution A_o for m_o.
  – Set $A^+ = \mathrm{argmin}(f(A_u), f(A_l), f(A_o))$.
  – If $f(A^+) = f(A)$, then stop iteration and go to **output**; else set $A \leftarrow A^+$ and
    start a new iteration.

• **Output** A.

A similar one can be designed for $\min_{A \in 2^X} f(A)$. What can we say about the
solution obtained by this algorithm? Is it a local optimal solution? Oh, let us first
explain what is a local optimal solution for the set function optimization.

For a submodular set function $f : 2^X \to \mathbb{R}$, the subgradient at set A consists
of all linear functions $c : X \to \mathbb{R}$ satisfying $f(Y) \geqslant f(A) + c(Y) - c(A)$ where
$c(Y) = \sum_{y \in Y} c(y)$. Each linear function c can also be seen as a vector in $\mathbb{R}^X$, i.e., a
vector c with components labeled by elements in X. The characteristic vector of each
subset Y of X is a vector in $\{0, 1\}^X$ such that the component with label $x \in X$ is equal
to 1 if and only if $x \in Y$. For simplicity of notation, we may use the same notation Y
to represent the set Y and its characteristic vector. Then, the subgradient of f at set A
can be described as

$$\partial f(A) = \{c \in \mathbb{R}^X \mid f(Y) \geqslant f(A) + <c, Y - A>\}.$$

If $c, d \in \partial f(A)$, then for any $0 \leqslant \lambda \leqslant 1$, $\lambda c + (1 - \lambda)d \in \partial f(A)$, that is, $\partial f(A)$
is a convex set in $\mathbb{R}^X$. The extreme point of this convex set can be characterized as
follows.

Theorem 4.1 [32] *A point $c \in \mathbb{R}^X$ is an extreme point of $\partial f(A)$ if and only if there
is a permutation σ for elements in X, i.e., $X = \{\sigma(1), \sigma(2), \cdots, \sigma(|X|)\}$, such that
$A = \{\sigma(1), \sigma(2), \cdots, \sigma(|A|)\}$ and $c(\{\sigma(i)\}) = f(S_i) - f(S_{i-1})$ for $1 \leqslant i \leqslant |X|$,
where $S_0 = \varnothing$ and $S_i = \{\sigma(1), \sigma(2), \cdots, \sigma(i)\}$.*

Consider a DS function $f = g - h$ where g and h are submodular functions. A set
A is a local minimum (the first type) for f if

$$\partial h(A) \subseteq \partial g(A). \tag{4.1}$$

Actually, (4.1) is a necessary condition for set A to be a minimum solution.

Theorem 4.2 *Let $f = g - h$ be a set function and g and h submodular functions on
subsets of X. If set A is a minimum solution (the first type) for $\min_{Y \subseteq X} f(Y)$, then
$\partial h(A) \subseteq \partial g(A)$.*

Proof Since A is a minimum solution, we have $f(A) \leqslant f(Y)$ and hence $g(Y) -
g(A) \geqslant h(Y) - h(A)$ for any $Y \subseteq X$. Therefore, for any $c \in \partial h(A)$,

$$g(Y) - g(A) \geqslant h(Y) - h(A) \geqslant c(Y) - c(A).$$

This means that (4.1) holds.

Condition (4.1) is also sufficient for certain minimality.

Theorem 4.3 *Suppose A satisfies condition* (4.1). *Then, for any* $Y \in \mathcal{U}$, $f(A) \leqslant f(Y)$, *where*

$$\mathcal{U} = \{Y \mid \partial h(Y) \cap \partial g(A) \neq \varnothing\}.$$

Proof Choose $c \in \partial h(Y) \cap \partial g(A)$. Then,

$$h(A) \geqslant h(Y) + (c(A) - c(Y)) \text{ and } g(Y) \geqslant g(A) + (c(Y) - c(A)).$$

Hence $h(Y) - h(A) \leqslant c(Y) - c(A) \leqslant g(Y) - g(A)$. Therefore, $f(Y) \geqslant f(A)$.

Now, we come back to the iterated sandwich method. Could the method produce a solution satisfying condition (4.1) surely? It is a problem for further research. However, if we look at a local minimum (the second type) as a set for which adding or removing an element would not decrease the objective function value. A positive answer would be reached with an approach given by [25,26] with a little modification as follows.

Theorem 4.4 *In the iterated sandwich method, compute* m_{gl} *and* m_{hl} *by using the same permutation of elements of X. At each iteration, try at most n permutations* $\sigma_1, \cdots, \sigma_k$ *such that* $A = \{\sigma_1(|A|-1), \cdots, \sigma_k(|A|-1)\}$ *and* $X \backslash A = \{\sigma_1(|A|+1), \cdots, \sigma_k(|A|+1)\}$. *Then, the iterated sandwich method would stop at a local minimum* (*the second type*).

What can we say about the approximation performance for the iterated sandwich method? At least, it may produce a solution comparable with the data-dependent approximation described in the next section.

5 Data-Dependent Approximation

The sandwich method has been used quite often for solving several nonsubmodular optimization problems in the literature [33–36]. It runs as follows.
Sandwich Method:

- **Input** a set function $f : 2^X \to \mathbb{R}$.
- Initially, find two submodular functions u and l such that $u(A) \geqslant f(A) \geqslant l(A)$ for $A \in \Omega$ where Ω is a collection of subsets of X. Then carry out following

  – Compute an α-approximation solution S_u for $\min_{A \in O\text{mega}} u(A)$ and a β-approximation solution S_l for $\min_{A \in \Omega} l(A)$.
  – Compute a solution S_o for f.
  – Set $S = \text{argmin}(f(S_u), f(S_o), f(S_l))$.

- **Output** S.

This method is called a data-dependent approximation algorithm with following guaranteed performance.

Theorem 5.1 [33] *The solution S produced by the sandwich method satisfies the following*:

$$f(S) \leqslant \min \left\{ \frac{f(S_1)}{l(S_l)} \cdot \beta, \frac{\mathrm{opt}_u}{\mathrm{opt}_f} \cdot \alpha \right\} \cdot \mathrm{opt}_f,$$

where opt_f *(*opt_u*) is the objective function value of the minimum solution for* $\min_{A \in \Omega} f(A)$ *(*$\min_{A \in \Omega} u(A)$*).*

Proof Since S_l is a β-approximation solution for $\min_{A \in \Omega} l(A)$, we have

$$f(S_l) = \frac{f(S_l)}{l(S_l)} \cdot l(S_l) \leqslant \frac{f(S_l)}{l(S_l)} \cdot \beta \cdot \mathrm{opt}_l \leqslant \frac{f(S_l)}{l(S_l)} \cdot \beta \cdot l(\mathrm{OPT}_f) \leqslant \frac{f(S_l)}{l(S_l)} \cdot \beta \cdot \mathrm{opt}_f,$$

where OPT_f is an optimal solution for $\min_{A \in \Omega} f(A)$. Since S_u is an α-approximation solution for $\min_{A \in \Omega} u(A)$, we have

$$f(S_u) \leqslant u(S_u) \leqslant \alpha \cdot \mathrm{opt}_u = \alpha \cdot \frac{\mathrm{opt}_u}{\mathrm{opt}_f} \cdot \mathrm{opt}_f.$$

Therefore, the theorem holds.

From theoretical point of view, the sandwich method is always applicable since we have following.

Theorem 5.2 *For any set function f on 2^X, there exist two monotone nondecreasing submodular functions u and l such that $u(A) \geqslant f(A) \geqslant l(A)$ for every $A \in 2^X$.*

Proof By the DS decomposition theorem, there exist two monotone nondecreasing submodular functions g and h such that $f = g - h$. Note that for every $A \in 2^X$, $h(\varnothing) \leqslant h(A) \leqslant h(X)$. Set $u(A) = g(A) - h(\varnothing)$ and $l(A) = g(A) - h(X)$ for any $A \in 2^X$. Then, u and l meet our requirement.

However, in practice, it is often quite hard to find such an upper bound u and a lower bound l which are easily computable. Therefore, more efforts are required to construct them for specific real-world problems.

References

[1] Kirousis, L.M., Kranakis, E., Krizanc, D., Pelc, A.: Power consumption in packer radio networks. Theor. Comput. Sci. **243**, 289–305 (2000)

 [2] Chen, W.T., Huang, N.F.: The strongly connection problem on multihop packet radio networks. IEEE Trans. Commun. **37**(3), 293–295 (1989)

 [3] Pruhs, K.: Speed scaling. In: Encyclopedia of Algorithms, pp. 2045–2047 (2016)

 [4] Edmonds, J., Pruhs, K.: Scalably scheduling processes with arbitrary speedup curves. ACM Trans. Algorithms **8**(3), 28 (2012)

 [5] Lin, H., Bilmes, J.: Optimal selection of limited vocabulary speech corpora. In: Interspeech (2011)

 [6] Iyer, R.K., Jegelka, S., Bilmes, J.A.: Fast semidifferential-based submodular function optimization. ICML **3**, 855–863 (2013)

 [7] Nagano, K., Kawahara, Y., Aihara, K.: Size-constrained submodular minimization through minimum norm base. In: Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA (2011)

 [8] Barhan, A., Shakhomirov, A.: Methods for sentiment analysis of Twitter messages. In: Proceeding of the 12th Conference of Fruct Association, pp. 216–222 (2012)

 [9] Wu, B., Lyu, S., Ghanem, B.: Constrained submodular minimization for missing labels and class imbalance in multi-label learning. In: The Thirtieth AAAI Conference on Artificial Intelligence, pp. 2229–2236 (2016)

[10] Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization, 2nd edn. Springer, Berlin (1988)

[11] Orlin, J.B.: A faster strongly polynomial time algorithm for submodular function minimization. Math. Program. **118**, 237–251 (2009)

[12] Schrijver, A.: A combinatorial algorithm minimizing submodular functions in strong polynomial time. J. Comb. Theory (B) **80**, 346–355 (2000)

[13] Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—I. Math. Program. **14**(1), 265–294 (1978)

[14] Sviridenko, M.: A note on maximizing a submodular set function subject to knapsack constraint. Oper. Res. Lett. **32**, 41–43 (2004)

[15] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: KDD'07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA, pp. 420–429. ACM (2007)

[16] Calinescu, G., Chekuri, C., Pl, M., Vondrk, J.: Maximizing a submodular set function subject to a matroid constraint. SIAM J. Comput. **40**(6), 1740–1766 (2011)

[17] Fisher, M.L., Nemhauser, G.L., Wolsey, L.A.: An analysis of approximations for maximizing submodular set functions—II. In: Polyhedral Combinatorics. Springer, Berlin, Heidelberg (1978)

[18] Wolsey, A.L.: An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica **2**(4), 385–393 (1982)

[19] Feige, U., Mirrokni, V., Vondrak, J.: Maximizing nonmonotone submodular functions. In: Proceedings of the IEEE Foundations of Computer Science, pp. 461–471 (2007)

[20] Lee, J., Mirrokni, V., Nagarajan, V., Sviridenko, M.: Nonmonotone submodular maximization under matroid and knapsack constraints. In: Proceedings of the ACM Symposium on Theory of Computing, pp. 323–332 (2009)

[21] Feldman, M., Naor, J., Schwartz, R.: A unified continuous greedy algorithm for submodular maximization. In: IEEE FOCS, pp. 570–579 (2011)

[22] Svitkina, Z., Fleischer, L.: Submodular approximation: sampling-based algorithms and lower bounds. SIAM J. Comput. **40**, 1715–1737 (2011)

[23] Feige, U., Izsak, R.: Welfare maximization and the supermodular degree. In: Conference on Innovations in Theoretical Computer Science. ACM, pp. 247–256 (2013)

[24] Feldman, M., Izsak, R.: Constrained monotone function maximization and the supermodular degree. (2014). arXiv:1407.6328

[25] Narasimhan, M., Bilmes, J.: A submodular-supermodular procedure with applications to discriminative structure learning. In: The Twenty-First Conference on Uncertainty in Artificial Intelligence (2005)

[26] Iyer, R., Bilmes, J.: Algorithms for approximate minimization of the difference between submodular functions. In: Twenty-eighth Conference on Uncertainty in Artificial Intelligence (2012)

[27] Iyer, R., Bilmes, J.: Submodular optimization subject to submodular cover and submodular knapsack constraints. In: Advances of NIPS (2013). arXiv:1311.2106

[28] Wu, C., Wang, Y., Lu, Z., Pardalos, P.M., Xu, D., Zhang, Z., Du, D.-Z.: Solving the degree-concentrated fault-tolerant spanning subgraph problem by DC programming. Math. Program. **169**(1), 255–275 (2018)

[29] Maehara, T., Murota, K.: A framework of discrete DC programming by discrete convex analysis. Math. Program. **152**(1–2), 435–466 (2015)

[30] Cunningham, W.H.: Decomposition of submodular functions. Combinatorica **3**(1), 53C68 (1983)

[31] Zhu, Y., Lu, Z., Bi, Y., Wu, W., Jiang, Y., Li, D.: Influence and profit: two sides of the coin. In: ICDM, pp. 1301–1306 (2013)

[32] Fujishige, S.: Submodular Functions and Optimization, vol. 58. Elsevier Science, Amsterdam (2005)

[33] Wei, L., Chen, W., Lakshmanan, L.V.S.: From competition to complementarity: comparative influence diffusion and maximization. Proc. VLDB Endow. **9**(2), 60–71 (2015)

[34] Chen, W., Lin, T., Tan, Z., Zhao, M., Zhou, X.: Robus Influence Maximization, KDD'16. San Francisco, CA, USA (2016)

[35] Wang, Z., Yang, Y., Pei, J., Chen, E.: Activity maximization by effective information diffusion in social networks (2016). arXiv:1610.07754v1 [cs.SI]

[36] Zhu, J., Zhu, J., Ghosh, S., Wu, W., Yuan, J.: Social influence maximization in hypergraph in social networks. IEEE Trans. Netw. Sci. Eng. (2018). https://doi.org/10.1109/TNSE.2018.2873759