劣モジュラ最適化の最近の進展
Recent Progress in Submodular Optimization

岩田　覚
Satoru Iwata

京都大学数理解析研究所
Research Institute for Mathematical Sciences
Kyoto University

Kyoto 606-8502, Japan
iwata@kurims.kyoto-u.ac.jp

Abstract

Submodular functions often arise in various fields of operations research including discrete optimization, game theory, queueing theory and information theory. In this survey paper, we give overview on the fundamental properties of submodular functions and recent algorithmic devolopments of their minimization.

Keywords: submodular function, combinatorial algorithm, discrete convexity

1 Introduction

Let V be a finite set. A set function $f : 2^V \to \mathbf{R}$ is said to be submodular if it satisfies

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y), \qquad \forall X, Y \subseteq V.$$

Submodular function minimization is to compute the minimum value as well as a minimizer of a submodular function f, provided that an oracle for evaluating the function value $f(X)$ for $X \subseteq V$ is available. A set fucntion f is supermodular if $-f$ is submodular. A set function that is both submodular and supermodular is called a modular function.

Submodular functions arise in discrete optimization [30, 53] and various other fields of operations research such as constraint satisfaction [6, 37], game theory [57], information theory [23], and queueing theory [14, 56]. Examples include the cut capacity functions of networks, the rank functions of matroids, and the entropy functions of multiple information sources. Submodular functions play important roles in statistical physics as well [1, 2].

Submodular functions are discrete analogue of convex functions. This analogy was exhibited by the discrete separation theorem of Frank [20] and the Fenchel-type duality theorem of Fujishige [24]. A more direct connection was established by Lovász [41], who clarified that the submodularity of a set function can be characterized by the convexity of a continuous function obtained by extending the set function in an appropriate manner. This observation together with valuated matroids invented by Dress and Wenzel [11] motivated Murota [45, 46, 47] to develop the theory of discrete convex analysis.

The first polynomial algorithm for submodular function minimization is due to Grötschel, Lovász, and Schrijver [28]. A strongly polynomial version was also presented by Grötschel, Lovász, and Schrijver [29]. These algorithms employ the ellipsoid method, which was used

by Khachiyan [39] to develop the first polynomial-time algorithm for linear programming. In spite of its polynomial time complexity, the ellipsoid method is not so efficient in practice.

Cunningham [9] developed a combinatorial strongly polynomial algorithm for solving the membership problem for matroid polyhedra, which is a special case of submodular function minimization. Then Cunningham [10] extended this method to compute the minimum value of a general submodular function in pseudopolynomial time.

Recently, combinatorial strongly polynomial algorithms have been developed by Iwata, Fleischer, and Fujishige (IFF) [35] and by Schrijver [54]. Both of these algorithms build on works of Cunningham [9, 10]. The IFF algorithm employs a scaling scheme developed in capacity scaling algorithms for the submodular flow problem [19, 32, 36]. In contrast, Schrijver [54] directly achieves a strongly polynomial bound by introducing a novel subroutine in framework of lexicographic augmentation. Subsequently, Fleischer and Iwata [17, 18] have described a push/relabel algorithm using Schrijver's subroutine to improve the running time bound. Then Vygen [59] refined the complexity analysis of Schrijver's algorithm to show that it is as fast as the push/relabel algorithm. Combining the scaling scheme with the push/relabel technique yields a faster combinatorial algorithm [34], which currently achieves the best running time bound for general submodular function minimization.

All of these combinatorial algorithms perform multiplications and divisions, although the problem of submodular function minimization does not involve these arithmetic operations. Schrijver [54] has asked if one can minimize a submodular function in strongly polynomial time using only additions, subtractions, comparisons, and the oracle calls for function values. It turns out that the IFF strongly polynomial algorithm can be converted to such a fully combinatorial algorithm [33].

This paper describes recent progress on submodular function minimization. Section 2 exhibits examples of submodular functions and related minimization problems. Section 3 is an introduction to the polyhedral approach to submodular functions. It describes the greedy algorithm and the connection between submodularity and convexity. In Section 4, we expound a general framework that are commonly used by the combinatorial algorithms for submodular function minimization. In Section 5, we describe the faster scaling algorithm developed in [34]. Then Section 6 is devoted to the strongly polynomial version. Finally, Section 7 provides some open problems.

Other surveys on submodular function minimization have been given by Fleischer [16], Fujishige [26], and McCormick [42]. The readers are also referred to related chapters of Fujishige [27], Korte and Vygen [40], Murota [47], and Schrijver [55].

Throughout this paper, let $\mathbf{R}^V$ denote the set of all the real valued functions $x : V \to \mathbf{R}$, which forms a linear space of dimension $n = |V|$. We identify a vector $x \in \mathbf{R}^V$ with a modular function defined by $x(Y) = \sum_{v \in Y} x(v)$.

2 Examples of Submodular Functions

In this section, we describe four examples of submodular functions. The first two come from discrete mathematics, while the others are taken from queueing theory and information theory.

Matroids

The concept of matroids was introduced by Whitney [60] as a combinatorial abstraction of linear independence. Let V be a finite set and $\mathcal{I}$ be a family of subsets of V. A pair $(V, \mathcal{I})$ is a matroid if it satisfies a certain system of axioms. The rank function ρ of a matroid is defined by $\rho(X) = \max\{|J| \mid J \subseteq X, J \in \mathcal{I}\}$. Then ρ is a monotone nondecreasing submodular function that satisfies $\rho(\emptyset) = 0$ and $\rho(X) \leq |X|$ for $X \subseteq V$. Conversely, such a set function defines a matroid by $\mathcal{I} = \{J \mid J \subseteq V, \rho(J) = |J|\}$.

The convex hull of the characteristic vectors of the independent sets in $\mathbf{R}^V$ coincides with

$$\mathrm{MP}(\rho) = \{z \mid z \in {\mathbf{R}_+}^V, \forall X \subseteq V, z(X) \leq \rho(X)\},$$

which is called the matroid polyhedron. Testing if a given vector $z \in {\mathbf{R}_+}^V$ is in $\mathrm{MP}(\rho)$ can be reduced to minimizing the submodular function $f(X) = \rho(X) - z(X)$. Cunningham [9] presented a combinatorial strongly polynomial algorithm for this special type of submodular function minimization.

Connected Detachment

Let $G = (V, E)$ be a connected graph with vertex set V and edge E. Consider a function $b : V \to \mathbf{Z}_+$. A b-detachment of G is a new graph $G' = (W, E)$ obtained by splitting each vertex $v \in V$ into $b(v)$ vertices. Each edge $e \in E$ incident to $v \in V$ in G should be incident in G' to one of the $b(v)$ vertices that come from v. For any $X \subseteq V$, let $e(X)$ denote the number of edges incident to X. We also denote by $c(G \setminus X)$ the number of connected components in the graph obtained from G by deleting the vertices in X. Nash-Williams [49] found the following theorem on the existence of a connected b-detachment.

Theorem 1 (Nash-Williams [49]) *There exists a connected b-detachment of $G = (V, E)$ if and only if*

$$b(X) \leq e(X) - c(G \setminus X) + 1 \tag{1}$$

holds for any $X \subseteq V$.

Let $f(X)$ denote the right-hand side of (1). Then we have $f(\emptyset) = 0$ and $f(V) = |E| + 1$. Furthermore, it can be shown that f is a submodular function. Theorem 1 suggests that one can check the existence of a connected b-detachment by minimizing the submodular function $f(X) - b(X)$.

The original proof was based on the matroid intersection theorem. Simple alternative proofs have been given to this theorem [50, 51, 52]. The submodularity of f plays a crucial role in the one that uses orientations [51].

Detachments with higher edge-connectivity requirements have recently been investigated by Fleiner [15] and by Jordán and Szigeti [38]. See Frank [22, 21] for other interesting applications of submodular functions in graph theory.

Multiclass Queueing Systems

Consider a queueing system which deals with various types of jobs. Each job of different classes waits in different queues and the server chooses the job to serve the next by a control policy. One of the most fundamental models of this type is the so-called preemptive

M/M/1, where the arrival interval and service time of each class of jobs follow exponential distributions and the premption is allowed in its control policy.

If the average arrival rates and the average service rates of the job classes are given, the performance of the system depends only on the control policy. Let V be the set of job classes. The region of performance-measuring vectors in $\mathbf{R}^V$ achieved by all control policies is called the achievable region. The performance of a multiclass M/M/1 is often measured by the average staying time vector $s \in \mathbf{R}^V$. If the preemption is allowed, the performance region of the staying time vector is explicitly given as follows.

Theorem 2 (Coffman and Mitrani [5]) *For each job $j \in V$, let λ_j and μ_j be the average arrival rates and the average service rates, respectively. Suppose that the utilization $\rho_j = \lambda_j/\mu_j$ satisfies $\sum_{j\in V} \rho_j < 1$. Then the achievable region of the average staying time vector is the set of vectors $s \in \mathbf{R}^V$ that satisfy*

$$\sum_{j\in X} \rho_j s_j \geq \frac{\sum_{j\in X} \frac{\rho_j}{\mu_j}}{1 - \sum_{j\in X} \rho_j} \tag{2}$$

for every $X \subseteq V$.

The right hand side of (2) can be written as $f(X) = y(X)h(x(X))$, where $x(j) := \rho_j$, $y(j) := \dfrac{\rho_j}{\mu_j}$ and $h(x) = \dfrac{1}{1-x}$. If we assign $z(j) := \rho_j s_j$, the problem of checking the achievability of a given vector s is reduced to minimizing a set function f defined by

$$f(X) = z(X) - y(X)h(x(X)).$$

Since h is a monotone nondecreasing convex function, one can verify that f is a submodular function.

A recent paper [31] presents an efficient algorithm for minimizing this type of submodular functions in $O(n^2)$ time. The algorithm utilizes the topological sweeping method of Edelsbrunner and Guibas [12] for line arrangements in the plane.

Apart from the multiclass preemptive M/M/1, submodular functions often arise in the analysis of achievable regions of various types of multiclass queueing systems [3, 14, 56].

Entropy Functions

Let V be a set of discrete memoryless information sources (random variables). For each nonempty subset X of V, let $h(X)$ denote the Shannon entropy of the corresponding joint distrubution. In addition, we assign $h(\emptyset) = 0$. Then the set function h is a submodular function, which follows from the nonnegativity of conditional mutual information.

Consider the situation that we encode data generated by this set of sources. Each source has its encoder, which compresses each data and transmits the code to the central decoder, which decodes all the codes it receives. We call the rate vector $R \in \mathbf{R}^V$ achievable if there exists a coding method of rate R with arbitraily small error probability. The following theorem of Slepian and Wolf [58] suggests that one can exploit the correlation among the sources to reduce the total rate required for the transmission. See also Cover [7] and Cover and Thomas [8, §14.4].

Theorem 3 (Slepian and Wolf [58]) *The rate vector R is achievable if and only if*

$$R(X) > h(V) - h(V \setminus X) \tag{3}$$

holds for any nonempty $X \subseteq V$.

Note that the right-hand side of (3) is a supermodular function. Theorem 3 implies that one can check if a specified rate vector R is achievable by minimizing a submodular function $R(X) - h(V) + h(V \setminus X)$. The rate vector R is achievable if and only if the empty set is the only minimizer. The only known method to do this is to apply an algorithm for general submodular function minimization.

Let K be a positive definite symmetric matrix whose row/column set is indexed by V. For each $X \subseteq V$, let $K[X]$ denote the principal submatrix of K indexed by X. The set function f defined by $f(\emptyset) = 0$ and $f(X) = \log\det K[X]$ for nonempty X is a submodular function. The submodularity of this function f, known as Ky Fan's inequality, is a refinement of Hadamard's inequality. It can be interpreted as the submodularity of the entropy function of a multivariate normal distribution with covariance matrix K.

3 Greedy Algorithm and Discrete Convexity

For a submodular function f with $f(\emptyset) = 0$, we consider the submodular polyheron $\mathrm{P}(f)$ and the base polyhedron $\mathrm{B}(f)$ defined by

$$\begin{aligned} \mathrm{P}(f) &= \{x \mid x \in \mathbf{R}^V, \forall Y \subseteq V, x(Y) \leq f(Y)\}, \\ \mathrm{B}(f) &= \{x \mid x \in \mathrm{P}(f), x(V) = f(V)\}. \end{aligned}$$

A vector in $\mathrm{B}(f)$ is called a base. In particular, an extreme point of $\mathrm{B}(f)$ is called an extreme base. The base polyhedron $\mathrm{B}(f)$ is the set of maximal vectors in $\mathrm{P}(f)$.

An extreme base can be obtained by the greedy algorithm of Edmonds [13] and Shapley [57] as follows.

Let $L = (v_1, \cdots, v_n)$ be a linear ordering of V. For any $v_j \in V$, we denote $L(v_j) = \{v_1, \cdots, v_j\}$. The greedy algorithm with respect to L generates an extreme base $y \in \mathrm{B}(f)$ by

$$y(u) := f(L(u)) - f(L(u) \setminus \{u\}). \tag{4}$$

Conversely, any extreme base can be obtained in this way with an appropriate linear ordering.

Given a nonnegative vector $p \in {\mathbf{R}_+}^V$, consider a linear ordering $L = (v_1, \cdots, v_n)$ such that $p(v_1) \geq p(v_2) \geq \cdots \geq p(v_n)$. The greedy algorithm with respect to L yields an optimal solution to the problem of maximizing the inner product $\langle p, x\rangle = \sum_{v \in V} p(v)x(v)$ in $\mathrm{B}(f)$.

Let $p_1 > p_2 > \cdots > p_k$ be the distinct values of p. For $j = 1, \ldots, k$, we denote $U_j = \{v \mid p(v) \geq p_j\}$. Then p can be expressed as

$$p = \sum_{j=1}^{k} q_j \chi_{U_j},$$

with $q_j = p_j - p_{j+1}$ for $j = 1, \ldots, k-1$ and $q_k = p_k \geq 0$. We now define $\hat{f}(p)$ by

$$\hat{f}(p) = \sum_{j=1}^{k} q_j f(U_j).$$

Then the function $\hat{f}$ satisfies

$$\hat{f}(p) = \max\{\langle p, x\rangle \mid x \in \mathrm{B}(f)\}, \tag{5}$$

which follows from the validity of the greedy algorithm.

Note that the above definition of $\hat{f}$ is free from the submodularity of f. For a set function f in general, we define $\hat{f}$ in the same way. Then $\hat{f}(\chi_X) = f(X)$ holds for any $X \subseteq V$. Hence we may regard $\hat{f}$ as an extension of f.

The restriction of $\hat{f}$ to the hypercube $[0,1]^V$ can be interpreted as follows. A linear ordering L corresponds to the simplex whose extreme points are given by the characteristic vectors of $L(v)$ for $v \in V$ and the empty set. Since there are $n!$ linear orderings of V, the hypercube $[0,1]^V$ can be partitioned into $n!$ congruent simplices obtained by this way. Determine the function values of $\hat{f}$ in each simplex by the linear interpolation of the values at the extreme points. The resulting function $\hat{f}$ is a continuous function on the hypercube.

The following theorem provides a connection between submodularity and convexity.

Theorem 4 (Lovász [41]) *A set function f is submodular if and only if $\hat{f}$ is convex.*

Proof. If f is a submodular function, then it follows from (5) that $\hat{f}$ is a convex function. Conversely, if $\hat{f}$ is convex, then we have

$$\hat{f}(\chi_X + \chi_Y) \leq \hat{f}(\chi_X) + \hat{f}(\chi_Y) = f(X) + f(Y).$$

On the other hand, it follows from the definition of $\hat{f}$ that

$$\hat{f}(\chi_X + \chi_Y) = \hat{f}(\chi_{X\cap Y}) + \hat{f}(\chi_{X\cup Y}) = f(X \cap Y) + f(X \cup Y)$$

holds for any $X, Y \subseteq V$. Thus f is a submodular function. ∎

4 Min-Max Theorem

For any vector $x \in \mathbf{R}^V$, we denote $x^-(v) := \min\{x(v), 0\}$. The following min-max theorem plays a central role in combinatorial algorithms for submodular function minimization. We describe a proof based on the discrete convexity of submodular functions.

Theorem 5 (Edmonds [13]) *For a submodular function f with $f(\emptyset) = 0$, we have*

$$\min_{X\subseteq V} f(X) = \max\{x^-(V) \mid x \in \mathrm{B}(f)\}.$$

Proof. The minimum value of $\hat{f}$ in the cube $[0,1]^V$ is attained at the extreme points. Then it follows from (5) and the linear programming duality that

$$\begin{aligned}\min_{X\subseteq V} f(X) = \min_{p\in[0,1]^V} \hat{f}(p) &= \min_{p\in[0,1]^V} \max_{x\in \mathrm{B}(f)} \langle p, x\rangle \\ &= \max_{x\in \mathrm{B}(f)} \min_{p\in[0,1]^V} \langle p, x\rangle = \max_{x\in \mathrm{B}(f)} x^-(V)\end{aligned}$$

holds. ∎

Theorem 5 seems to provide a good characterization of the minimum value of f. In fact, if we have a pair of $W \subseteq V$ and $x \in \mathrm{B}(f)$ with $f(W) = x^-(V)$, then it follows from Theorem 5 that W attains the minimum value of f. This suggests a natural way to find the

minimum by moving $x \in \mathrm{B}(f)$ so that $x^-(V)$ increases. However, it is not easy to verify that the vector x in our hand stays in $\mathrm{B}(f)$. A direct way to check this by the definition requires an exponential number of steps. On the other hand, an extreme base y of $\mathrm{B}(f)$ can be verified by a linear ordering of V generating y. According to Caratheodory's theorem, an arbitrary point in a bounded polyhedron can be expessed as a convex combination of its extreme points. Keeping $x \in \mathrm{B}(f)$ as a convex combination $x = \sum_{i \in I} \lambda_i y_i$ of extreme bases y_i, we are able to verify $x \in \mathrm{B}(f)$ efficiently, provided that I is not too large. A base $x \in \mathrm{B}(f)$ expressed by this way provides a compact certificate of $f(W)$ being the minimum value if $x^-(V) = f(W)$ holds.

This approach was introduced by Cunningham [9] in the separation problem for matroid polyhedra. Bixby, Cunningham, and Topkis [4] employed this approach to develop a combinatorial algorithm for minimizing a submodular function by a finite number of steps. Furthermore, Cunningham [10] improved this algorithm to the first combinatorial pseudopolynomial algorithm for computing the minimum value of an integer valued submodular function. In general, a pseudopolynomial algorithm runs in time polynomial in the number of inputs and the maximum absolute value of the inputs. The running time bound of Cunningham's algorithm is $O(n^6 \gamma M \log nM)$, where γ is the time required for computing the function value and M is the maximum absolute value of f.

Since the dimension of a base polyhedron is at most $n-1$, it follows from Caratheodory's theorem that any base $x \in \mathrm{B}(f)$ can be expressed as a convex combination of at most n extreme bases. When the set I becomes large, we are able to reduce $|I|$ to at most n as follows. Consider a $V \times I$ matrix that consists of extreme bases y_i for $i \in I$. Let H be a matrix obtained by attaching a row with all components being one to this matrix. Applying the Gaussian elimination by row transformations to H, detect a linear dependence $\sum_{i \in I} \mu_i y_i = 0$, $\sum_{i \in I} \mu_i = 0$. Compute $\theta = \max\{\lambda_i / \mu_i \mid \mu_i > 0\}$ and update $\lambda_i := \lambda_i - \theta \mu_i$ for each $i \in I$. At least one index $i \in I$ will satisfy $\lambda_i = 0$, and then delete such i from I. Repeat this process until H becomes lenearly independent. This process will be referred to as $\mathsf{Reduce}(x, I)$.

5 A Faster Scaling Algorithm

This section is devoted to a faster scaling algorithm developed in for minimizing an integer-valued submodular function. This algorithm achieves the currently best running time bound among combinatorial algorithms for submodular function minimization [34].

The algorithm consists of scaling phases with a scale parameter $\delta > 0$. It starts with an arbitrary linear ordering L and an extreme base $x \in \mathrm{B}(f)$ generated by L. The initial value of δ is given by $\delta := \min\{|x^-(V)|, x^+(V)\}/n^2$. In each scaling phase, the algorithm cuts the value of δ in half. Finally, the algorithm terminates when $\delta < 1/n^2$. Since the initial value of δ satisfies $\delta < M/n^2$, the algorithm performs $O(\log M)$ scaling phases.

The algorithm keeps a set of linear orderings $\{L_i \mid i \in I\}$ of the vertices in V. We denote $v \prec_i u$ if v precedes u in L_i. Each linear ordering L_i generates an extreme base $y_i \in \mathrm{B}(f)$ by the greedy algorithm. The algorithm also keeps a base $x \in \mathrm{B}(f)$ as a convex combination $x = \sum_{i \in I} \lambda_i y_i$ of the extreme bases. The initial setting is $I = \{0\}$, $y_0 = x$, $L_0 = L$, $\lambda_0 = 1$.

Furthermore, the algorithm works with a flow in the complete directed graph on the vertex set V. The flow is represented as a skew-symmetric function $\varphi : V \times V \to \mathbf{R}$, which satisfies $\varphi(u, v) + \varphi(v, u) = 0$ for each (u, v). The arc capacity is δ for each arc. Hence the cut capacity function is given by $\kappa_\delta(X) = \delta\,|X| \cdot |V \setminus X|$. A flow φ is said to be feasible if

$-\delta \le \varphi(u,v) \le \delta$ holds for each arc (u,v). The boundary $\partial\varphi$ of φ is defined by

$$\partial\varphi(u) = \sum_{v \in V} \varphi(u,v).$$

Then we have $\partial\varphi \in \mathrm{B}(\kappa_\delta)$. Initially, we set $\varphi(u,v) = 0$ for any $u, v \in V$.

Each scaling phase aims at increasing $z^-(V)$ for $z = x + \partial\varphi$. Given a flow φ, the algorithm constructs an auxiliary directed graph $G_\varphi = (V, A_\varphi)$ with arc set $A_\varphi = \{(u,v) \mid u \neq v,\ \varphi(u,v) \le 0\}$. Let $S = \{v \mid z(v) \le -\delta\}$ and $T = \{v \mid z(v) \ge \delta\}$. A directed path in G_φ from S to T is called an augmenting path.

Each scaling phase also keeps a valid labeling d. A labeling $d : V \to \mathbf{Z}$ is valid if $d(u) = 0$ for $u \in S$ and $v \preceq_i u$ implies $d(v) \le d(u) + 1$. A valid labeling $d(v)$ serves as a lower bound on the number of arcs from S to v in the directed graph $G_I = (V, A_I)$ with the arc set $A_I = \{(u,v) \mid \exists i \in I,\ v \preceq_i u\}$.

If there exists an augmenting path P, the algorithm augments the flow φ through P by δ, namely $\varphi(u,v) := \varphi(u,v) + \delta$ and $\varphi(v,u) := \varphi(v,u) - \delta$ for each $(u,v) \in P$. This procedure is reffered to as $\mathsf{Augment}(\varphi, P)$. As a result of $\mathsf{Augment}(\varphi, P)$, the initial vertex s of P may get rid of S and no new vertex joins S. Thus $\mathsf{Augment}(\varphi, P)$ increases $z^-(V)$ by δ without violating the validity of d.

Suppose that there is no augmenting path in $G_\varphi = (V, E_\varphi)$. Let W be the set of vertices reachable from S in G_φ. Let Z be the set of vertices that attains the minimum labeling in $V \setminus W$. A triple (i,u,v) is called active if v is the first vertex of Z in L_i and u is the last vertex in L_i with $v \preceq_i u$ and $d(v) = d(u) + 1$. The procedure $\mathsf{Multiple\text{-}Exchange}(i,u,v)$ is applicable to an active triple (i,u,v).

For an active triple (i,u,v), the set of vertices from v to u in L_i is called an *active interval*. The active interval is divided into $Q = \{w \mid w \in W,\ v \prec_i w \preceq_i u\}$ and $R = \{w \mid w \in V \setminus W,\ v \preceq_i w \prec_i u\}$.

The procedure $\mathsf{Multiple\text{-}Exchange}(i,u,v)$ moves the vertices in R to the place immediately after u in L_i, without changing the ordering in Q and in R. Then it computes an extreme base y_i generated by the new L_i. This results in $y_i(q) \ge y_i^\circ(q)$ for $q \in Q$ and $y_i(r) \le y_i^\circ(r)$ for $r \in R$, where y_i° denotes the previous y_i.

Consider a complete bipartite graph with the vertex sets Q and R. The algorithm finds a flow $\xi : Q \times R \to \mathbf{R}_+$ such that $\sum_{r \in R} \xi(q,r) = y_i(q) - y_i^\circ(q)$ for each $q \in Q$ and $\sum_{q \in Q} \xi(q,r) = y_i^\circ(r) - y_i(r)$ for each $r \in R$. Such a flow can be obtained easily by the so-called northwest corner rule. Then the procedure computes $\alpha = \min\{\lambda_i, \delta/\beta\}$ with $\beta = \max\{\xi(q,r) \mid q \in Q, r \in R\}$, and moves x to $x := x + \alpha(y_i - y_i^\circ)$. In order to keep z invariant, the procedure adjusts the flow φ by $\varphi(q,r) := \varphi(q,r) - \alpha\xi(q,r)$ and $\varphi(r,q) := \varphi(r,q) + \alpha\xi(q,r)$ for every $(q,r) \in Q \times R$. The resulting φ satisfies the capacity constraints due to the choice of α, and the vertices in W remain reachable from S in G_φ.

If $\alpha < \lambda_i$, a new index k is added to I. The associated linear ordering L_k is the previous L_i. The coefficient λ_k is determined by $\lambda_k := \lambda_i - \alpha$, and then λ_i is replaced by $\lambda_i := \alpha$. Thus the algorithm continues to keep x as a convex combination $x = \sum_{i \in I} \lambda_i y_i$.

Each scaling phase begins with setting $d(v) = 0$ for each $v \in V$. If there exists an augmenting path P in G_φ, then the algorithm performs $\mathsf{Augment}(x, P)$ and $\mathsf{Reduce}(x, I)$. Otherwise, the algorithm computes $\ell = \min\{d(v) \mid v \in V \setminus W\}$. If $\ell < n$, then the algorithm applies $\mathsf{Multiple\text{-}Exchange}(i,u,v)$ to an active triple (i,u,v). If there is no active triple, it applies $\mathsf{Relabel}(v)$, which increases $d(v)$ by one, to each $v \in Z$. Finally, if $\ell = n$, there is no directed path from S to $V \setminus W$ in G_I. Then the set X of vertices reachable from S in G_I

satisfies $x(X) = f(X)$, which gives the end of the current scaling phase. The algorithm goes to the next scaling phase by cutting the value of δ in half.

The resulting scaling algorithm is now described as follows.

Step 0: Let L_0 be an arbitrary linear ordering. Compute an extreme base y_0 by the greedy algorithm with respect to L_0. Put $x := y_0$, $\lambda_0 := 1$, $I := \{0\}$, and $\delta := |x^-(V)|/n^2$.

Step 1: Put $d(v) := 0$ for $v \in V$, and $\varphi(u, v) := 0$ for $u, v \in V$.

Step 2: Put $S := \{v \mid z(v) \leq -\delta\}$ and $T := \{v \mid z(v) \geq \delta\}$, where $z = x + \partial\varphi$. Let W be the set of vertices reachable from S in G_φ.

Step 3: If there is an augmenting path P, then do the following.

(3-1) Apply Augment(φ, P).

(3-2) Apply Reduce(x, I).

(3-3) Go to Step 2.

Step 4: Compute $\ell := \min\{d(v) \mid v \in V \setminus W\}$ and put $Z := \{v \in v \in V \setminus W, d(v) = \ell\}$. If $\ell < n$, then do the following.

(4-1) If there is an active triple (i, u, v), then apply Multiple-Exchange(i, u, v).

(4-2) Otherwise, apply Relabel(v) for each $v \in Z$.

(4-3) Go to Step 2.

Step 5: Determine the set X of vertices reacheble from S in G_I. If $\delta \geq 1/n^2$, then apply Reduce(x, I), $\delta := \delta/2$, and go to Step 1.

The number of applications of Augment and Relabel are both $O(n^2)$ in each scaling phase. The total number of function evaluations is $O(n^2)$ in consecutive applications of Multiple-Exchange between Relabel or Augment. Thus each scaling phase takes $O(n^4\gamma + n^5)$ time. Since the algorithm performs $O(\log M)$ scaling phases, the total running time bound is $O((n^4\gamma + n^5) \log M)$.

Finally, at the end of the last scaling phase with $\delta < 1/n^2$, we have $x^-(V) \geq f(X) - n^2\delta > f(X) - 1$ for the subset X obtained in Step 5. Since $x^-(V) \leq f(Y)$ for any $Y \subseteq V$, it follows from the integrality of f that X is a minimizer of f.

6 A Strongly Polynomial Scaling Algorithm

This section describes a strongly polynomial scaling algorithm for minimizing a real-valued submodular function developed in [35]. The algorithm keeps a directed acyclic graph $D = (U, F)$ and a subset $Z \subseteq V$. It starts with $U = V$, $F = \emptyset$, $Z = \emptyset$. The set Z represents the set of elements that turns out to be contained in any minimizer of f. The vertex set U of the directed acyclic graph D corresponds to the partition of $V \setminus Z$. For a subset $Y \subseteq U$, we denote by $\Gamma(Y)$ the union of the subsets of V represented by the vertices in U. An edge (u, v) reflects an implication that any minimizer that includes $\Gamma(\{u\})$ must include $\Gamma(\{v\})$ as well.

A submodular function $\tilde{f} : 2^U \to \mathbf{R}$ is defined by

$$\tilde{f}(Y) = \begin{cases} f(\Gamma(Y) \cup Z) - \min\{f(V), f(Z)\} & (\emptyset \neq Y \subset U) \\ 0 & (Y = \emptyset, U) \end{cases}$$

Then a minimizer of X of f can be represented as $X = \Gamma(Y) \cup Z$ by a minimizer Y of $\tilde{f}$.

For each vertex $u \in U$, we denote by $R(u)$ the set of vertices reachable from u in D. At the start of each iteration, the algorithm computes

$$\eta = \max\{\tilde{f}(R(u)) - \tilde{f}(R(u) \setminus \{u\}) \mid u \in U\} \tag{6}$$

If $\eta \leq 0$, then it turns out that either V or Z is a minimizer of f.

On the other hand, if $\eta > 0$, then the vertex $u \in U$ that attains the maximum in the right-hand side must satisfy either $\tilde{f}(R(u) \setminus \{u\}) \leq -\eta/2$ or $\tilde{f}(R(u)) > \eta/2$. In the former case, apply $\mathsf{Fix}(\tilde{f}, \eta)$ described below to detect a vertex $w \in R(u)$ that is contained in every minimizer of $\tilde{f}$. Then the algorithm deletes w from U and adds $\Gamma(\{w\})$ to Z. In the latter case, define a submodular function $\tilde{f}_u$ by

$$\tilde{f}_u(Y) = \tilde{f}(Y \cup R(u)) - \tilde{f}(R(u)) \qquad (Y \subseteq U \setminus R(u))$$

and apply $\mathsf{Fix}(\tilde{f}_u, \eta)$ to find a vertex $w \in U \setminus R(u)$ contained in every minimizer of $\tilde{f}_u$. Then the algorithm adds (u, w) to F. If the resulting graph contains a directed cycle, then the algorithm shrinks it to a new vertex.

The procedure $\mathsf{Fix}(\tilde{f}, \eta)$ is applicable to a submodular function $\tilde{f}$ such that $\tilde{f}(Y) \leq -\eta/2$ for some Y. It performs the scaling algorithm with the arc set of G_φ replaced by $E_\varphi \cup F$. If $x(w) < -m^2\eta$ holds for $m = |U|$ at the end of a scaling phase, w must be contained in every minimizer of $\tilde{f}$. The existence of Y with $\tilde{f}(Y) \leq -\eta/2$ ensures that such a vertex w must be found within $O(\log n)$ scaling phases.

Empolying the faster scaling algorithm in Fix, this algorithm runs in $O((n^6\gamma + n^7)\log n)$ time, which is currently the best strongly polynomial bound among combinatorial algorithms. Applying the technique in [33], one can implement this algorithm in a fully combinatorial manner. A straightforward implementation would result in an $O((n^8\gamma + n^9)\log^2 n)$ algorithm. McCormick [42] has suggested a more careful implementation to achieve an $O(n^8\gamma \log^2 n)$ bound.

An advantage of a fully combinatorial algorithm from theoretical point of view is not only aesthetic. Suppose we are given a vector $z \in \mathrm{P}(f)$ and a direction vector $a \in \mathbf{R}^V$. Then what is the maximum $t \in \mathbf{R}$ such that $ta + z \in \mathrm{P}(f)$? A recent paper of Nagano [48] presents the first strongly polynomial algorithm for solving this problem. The algorithm is based on the parametric search technique of Megiddo [43, 44], which requires a fully combinatorial subroutine for submodular function minimization.

7 Conclusion

We now conclude this paper by mentioning some open problems concerning submodular function minimization.

1. An obvious one is of course to improve theoretical efficiency of submodular function minimization. In particular, the current stronly polynomial bound is far from being satisfactory.

2. Fujishige [25] showed a connection between submodular funtion minimization and the minimum Euclidean norm point in the base polyhedron. A practical algorithm to solve the minimization problem based on this result is presented in Fujishige [27, §7.1 (a)]. It remains open to analyse the complexity of this algorithm.

3. What is the lower bound on the number of oracle calls for function evaluation required before determining the minimum value?

References

[1] J.-C. Anglès d'Auriac: Computing the Potts free energy and submodular functions. In A. K. Hartmann and H. Rieger (eds.), *New Optimization Algorithms in Physics* (Wiley, 2004), 101–117.

[2] J.-C. Anglès d'Auriac, F. Iglói, M. Preissmann, and A. Sebő: Optimal cooperation and submodularity for computing Potts' partition functions with a large number of states, *Journal of Physics*, Ser. A, **35** (2002), 6973–6983.

[3] D. Bertsimas and J. Niño-Mora: Conservation laws, extended polymatroids and multi-armed bandit problems; a polyhedral approach to indexable systems, *Mathematics of Operations Research*, **21** (1996), 257–306.

[4] R. E. Bixby, W. H. Cunningham, and D. M. Topkis: Partial order of a polymatroid extreme point, *Mathematics of Operations Research*, **10** (1985), 367–378.

[5] E. G. Coffman, Jr., and I. Mitrani: A characterization of waiting time performance realizable by single-server queues, *Operations Research*, **28** (1980), 810–821.

[6] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin: Supermodular functions and the complexity of Max CSP, *Discrete Applied Mathematics*, 149 (2005), 53–72.

[7] T. M. Cover: A proof of the data compression theorem of Slepian and Wolf for ergodic sources, *IEEE Transactions on Information Theory*, **IT21** (1975), 226–228.

[8] T. M. Cover and J. A. Thomas: *Elements of Information Theory* (Wiley, 1991).

[9] W. H. Cunningham: Testing membership in matroid polyhedra, *Journal of Combinatorial Theory*, Ser. B, **36** (1984), 161–188.

[10] W. H. Cunningham: On submodular function minimization, *Combinatorica*, **5** (1985), 185–192.

[11] A. W. M. Dress and W. Wenzel: Valuated matroids, *Advances in Mathematics*, **93** (1992), 214–250.

[12] H. Edelsbrunner and L. J. Guibas: Topologically sweeping an arrangement, *Journal of Computer and System Sciences*, 38 (1989), 165–194.

[13] J. Edmonds: Submodular functions, matroids, and certain polyhedra. In R. Guy, H. Hanani, N. Sauer, and J. Schönheim (eds.), *Combinatorial Structures and Their Applications*, (Gordon and Breach, 1970), 69–87.

[14] A. Federgruen and H. Groenevelt: Characterization and optimization of achievable performance in general queueing systems, *Operations Research*, 36 (1988), 733–741.

[15] B. Fleiner: Detachment of vertices of graphs preserving edge-connectivity, *SIAM Journal on Discrete Mathematics*, **18** (2005), 581–591.

[16] L. Fleischer: Recent progress in submodular function minimization, *OPTIMA*, **64** (2000), 1–11.

[17] L. Fleischer and S. Iwata: Improved algorithms for submodular function minimization and submodular flow, *Proceedings of the 32nd ACM Symposium on Theory of Computing* (2000), 107–116.

[18] L. Fleischer and S. Iwata: A push-relabel framework for submodular function minimization and applications to parametric optimization, *Discrete Applied Mathematics*, **131** (2003), 311–322.

[19] L. Fleischer, S. Iwata, and S. T. McCormick: A faster capacity scaling algorithm for minimum cost submodular flow, *Math. Programming*, **92** (2002), 119–139.

[20] A. Frank: An algorithm for submodular functions on graphs, *Annals of Discrete Mathematics*, **16** (1982), 97–120.

[21] A. Frank: Submodular functions in graph theory, *Discrete Mathematics*, **111** (1993), 231–241.

[22] A. Frank: Applications of submodular functions. In K. Walker (ed.), *Surveys in Combinatorics* (Cambridge University Press, 1993), 85–136.

[23] S. Fujishige: Polymatroidal dependence structure of a set of random variables, *Information and Control*, **39** (1978), 55–72.

[24] S. Fujishige: Theory of submodular programs — A Fenchel-type min-max theorem and subgradients of submodular functions, *Mathematical Programming*, **29** (1984), 142–155.

[25] S. Fujishige: Submodular systems and related topics, *Mathematical Programming Study*, **22** (1984), 113–131.

[26] S. Fujishige: Submodular function minimization and related topics, *Optimization Methods and Software*, **18** (2003), 169–180.

[27] S. Fujishige: *Submodular Functions and Optimization* (North-Holland, 2005).

[28] M. Grötschel, L. Lovász, and A. Schrijver: The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica*, **1** (1981), 169–197.

[29] Grötschel, M., L. Lovász, and A. Schrijver: *Geometric Algorithms and Combinatorial Optimization* (Springer-Verlag, 1988).

[30] B. Hoppe and É. Tardos: The quickest transshipment problem, *Mathematics of Operations Research*, **25** (2000), 36–62.

[31] T. Itoko and S. Iwata: Computational geometric approach to submodular function minimization for multiclass queueing systems. Technical Report METR 2005-29, University of Tokyo, October 2005.

[32] S. Iwata: A capacity scaling algorithm for convex cost submodular flows, *Mathematical Programming*, **76** (1997), 299–308.

[33] S. Iwata: A fully combinatorial algorithm for submodular function minimization. *Journal of Combinatorial Theory*, Ser. B, **84** (2002), 203–212.

[34] S. Iwata: A faster scaling algorithm for minimizing submodular functions. *SIAM Journal on Computing*, **32** (2003), 833–840.

[35] S. Iwata, L. Fleischer, and S. Fujishige: A combinatorial strongly polynomial algorithm for minimizing submodular functions, *Journal of the ACM*, **48** (2001), 761–777.

[36] S. Iwata, S. T. McCormick, and M. Shigeno: A strongly polynomial cut canceling algorithm for minimum cost submodular flow, *SIAM Journal on Discrete Mathematics*, **19** (2005), 304–320.

[37] P. Jonsson, M. Klasson, and A. Krokhin: The approximability of three-valued Max CSP, *SIAM Journal on Computing*, **35** (2006), pp. 1329–1349.

[38] T. Jordán and Z. Szigeti: Detachments preserving local edge-connectivity of graphs, *SIAM Journal on Discrete Mathematics*, **17** (2003), 72–87.

[39] L. G. Khachiyan: A polynomail algorithm in linear programming, *Soviet Mathamatics Doklady*, **20** (1979), 191–194.

[40] B. Korte and J. Vygen: *Combinatorial Optimization — Theory and Algorithms* (Springer-Verlag, 2000).

[41] L. Lovász: Submodular functions and convexity. In A. Bachem, M. Grötschel and B. Korte (eds.), *Mathematical Programming — The State of the Art* (Springer-Verlag, 1983), 235–257.

[42] S. T. McCormick: Submodular function minimization. In K. Aardal, G. Nemhauser, and R. Weismantel (eds.), *Discrete Optimization* (Handbooks in Operations Research, Vol. 12, Elsevier, 2005).

[43] N. Megiddo: Combinatorial optimization with rational objective functions, *Mathematics of Operations Research*, **4** (1979), 414–424.

[44] N. Megiddo: Applying parallel computation algorithms in the design of serial algorithms, *Journal of the ACM*, **30** (1983), 852–865.

[45] K. Murota: Convexity and Steinitz's exchange property, *Advances in Mathematics*, **124** (1996), 272–311.

[46] K. Murota: Discrete convex analysis, *Mathematical Programming*, **83** (1998), 313–371.

[47] K. Murota: *Discrete Convex Analysis* (SIAM, 2003).

[48] K. Nagano: A strongly polynomial algorithm for line search in submodular polyhedra. Technical Report METR 2004-33, University of Tokyo, June 2004.

[49] Connected detachments of graphs and generalized Euler trails, *Journal of the London Mathematical Society*, **31** (1985), 17–29.

[50] C. St. J. A. Nash-Williams: Another proof of a theorem concerning detachments of graphs, *European Journal of Combinatorics*, **12** (1991), 245–247

[51] C. St. J. A. Nash-Williams: Strongly connected mixed graphs and connected detachments of graphs, *Journal of Combinatorial Mathematics and Combinatorial Computing*, **19** (1995), pp. 33–47.

[52] C. St. J. A. Nash-Williams: A direct proof of a theorem on detachments of finite graphs, *Journal of Combinatorial Mathematics and Combinatorial Computing*, **19** (1995), pp. 314–318.

[53] M. Queyranne: Structure of a simple scheduling polyhedra, *Mathematical Programming*, **58** (1993), 263–285.

[54] A. Schrijver: A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory*, Ser. B, **80** (2000), 346–355.

[55] A. Schrijver: *Combinatorial Optimization — Polyhedra and Efficiency* (Springer-Verlag, 2003).

[56] J. G. Shanthikumar and D. D. Yao: Multiclass queueing systems: polymatroidal structure and optimal scheduling control. *Operations Research*, **40** (1992), S293–S299.

[57] L. S. Shapley: Cores of convex games, *International Journal of Game Theory*, **1** (1971), 11–26.

[58] D. Slepian and J. K. Wolf: Noiseless coding with of correlated information sources, *IEEE Transactions on Information Theory*, **IT19** (1973), 471–480.

[59] J. Vygen: A note on Schrijver's submodular function minimization algorithm, *Journal of Combinatorial Theory*, Ser. B, **88** (2003), 399–402.

[60] H. Whitney: On the abstract properties of linear dependence, *American Journal of Mathematics*, **57** (1935), 509–533.