The following comment is for the published paper appeared on the preceding page.

A Combinatorial Strongly Polynomial Algorithm for Minimizing Submodular Functions

FUJISHIGE Satoru

(Graduate School of Engineering Science)

1. Introduction

Submodular functions play fundamental roles in combinatorial optimization (see [7]) and submodular functions are discrete analogue of convex functions. There exist a lot of practical (efficient) minimization algorithms for ordinary convex functions, while for submodular functions there had been only one 'strongly polynomial' algorithm, due to Grötschel, Lovász, and Schrijver ([9], [10]), that utilizes the so-called ellipsoid method for linear programming, where 'strongly polynomial' and 'polynomial' are synonyms of 'efficient' in Computer Science. The ellipsoid method is far from efficient and is not a combinatorial one, so that the Grötschel-Lovász-Schrijver algorithm is a quite unsatisfactory one both theoretically and practically. Since 1981 it had been a long-standing open problem to devise a combinatorial efficient (polynomial-time) algorithm for minimizing submodular functions. Iwata-Fleischer-Fujishige [14] and Schrijver [16] independently and simultaneously succeeded in solving the open problem in July, 1999 by presenting combinatorial strongly polynomial algorithms for minimizing submodular functions.

We describe how submodular functions are related to convexity and sketch our algorithm [14] in the sequel.

2. Submodular functions and convexity

Let V be a nonempty finite set and 2^V be the power set of V, i.e., $2^V = \{X \mid X \subseteq V\}$. Also let $\mathbf{R}$ be the set of reals and $\mathbf{R}_+$ the set of nonnegative reals. We call a function $f : 2^V \to \mathbf{R}$ a *submodular function* if it satisfies

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \quad (X, Y \subseteq V) \tag{1}$$

(see Fig.1). Without loss of generality we assume that $f(\emptyset) = 0$, where

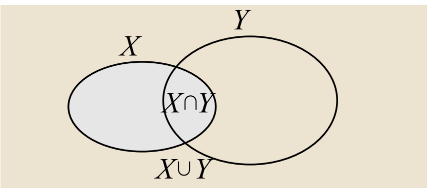

Fig 1.
Four sets X, Y, $X \cup Y$, and $X \cap Y$ associated with submodularity inequality.

if necessary, we may consider the function f' defined by $f'(X) = f(X) - f(\emptyset)$ $(X \subseteq V)$. Then, we have

$$f(X) + f(Y) \geq f(X \cup Y) \quad (X, Y \subseteq V, X \cap Y = \emptyset). \tag{2}$$

A function f satisfying (2) is called a *subadditive function*. Hence theclass of submodular functions is a special class of subadditive functios. However, subadditive functions do not have as nice combinatorial structure as submodular functions.

Inequalities (1) can be rewritten as

$$f(X) - f(X \cap Y) \geq f(X \cup Y) - f(Y) \quad (X, Y \subseteq V). \tag{3}$$

This would remind us of concavity. In fact, if we are given a concave function $\psi : \mathbf{R} \to \mathbf{R}$, then we get a submodular function $f : 2^V \to \mathbf{R}$ defined by

$$f(X) = \psi(|X|) \tag{4}$$

for any $X \subseteq V$, where $|X|$ denotes the cardinality of X. Any submodular function of symmetric type can be represented in such a way.

However, submodular functions are more closely related to convexity through the so-called Lovász extension of submodular functions. For any set function $f : 2^V \to \mathbf{R}$ with $f(\emptyset) = 0$ define a function $\hat{f} : \mathbf{R}_+^V \to \mathbf{R}$ as follows. For a non-zero vector $x \in \mathbf{R}_+^V$ there uniquely exist a sequence

$$C : S_1 \subset S_2 \subset \cdots \subset S_k \tag{5}$$

of nonempty subsets of V and positive scalars λ_i $(i = 1, 2, \cdots, k)$ such that

$$x = \sum_{i=1}^{k} \lambda_i \chi_{S_i}, \tag{6}$$

where χ_{S_i} is the characteristic vector of set S_i. By means of the unique representation (6) of x define

$$\hat{f}(x) = \sum_{i=1}^{k} \lambda_i f(S_i). \tag{7}$$

We also define $\hat{f}(\mathbf{0}) = 0$. The function $\hat{f}$ is called the *Lovász extension* of f.

Theorem 1 (Lovász[15]): *A set function $f : 2^V \to \mathbf{R}$ with $f(\emptyset) = 0$ is a submodular function if and only if the Lovász extension $\hat{f}$ of f is a convex function.* □

Associated with a submodular function $f : 2^V \to \mathbf{R}$, we define convex polyhedra $\mathrm{P}(f)$ and $\mathrm{B}(f)$, respectivey, called the *submodular polyhedron* and the *base polyhedron* as

$$\mathrm{P}(f) = \{x \mid x \in \mathbf{R}^V, \forall X \subseteq V : x(X) \leq f(X)\}, \tag{8}$$

$$\mathrm{B}(f) = \{x \mid x \in \mathrm{P}(f), x(V) = f(V)\}, \tag{9}$$

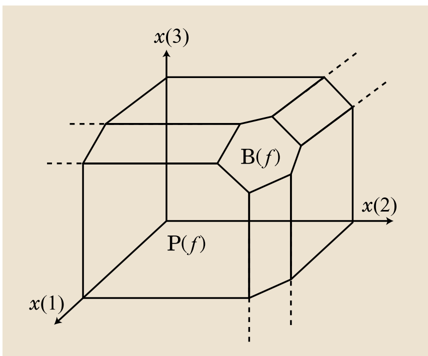

Fig 2.
Examples of a submodular polyhedron and a base polyhedron in $\mathbf{R}^3$.

where $x(X) = \sum_{v \in X} x(v)$ (see [7]). A vector in $\mathrm{B}(f)$ is called a *base*. Both $\mathrm{P}(f)$ and $\mathrm{B}(f)$ uniquely determine f (see Fig.2). The submodular polyhedron $\mathrm{P}(f)$ and the base polyhedron $\mathrm{B}(f)$ are also related to the Lovász extension of f as follows. For a convex function $h : \mathbf{R}^V \to \mathbf{R} \cup \{+\infty\}$ and a vector x with $h(x) < +\infty$ the subdifferential $\partial h(x)$ of h at x is defined by $\partial h(x) = \{z \mid z \in \mathbf{R}^V, \forall y \in \mathbf{R}^V : h(x) + \langle z, y - x \rangle \leq h(y)\}$, where $\langle \cdot, \cdot \rangle$ is the canonical inner product.

Theorem 2 (Fujishige[6]): *We have*

$$\mathrm{P}(f) = \partial\hat{f}(\mathbf{0}), \quad \mathrm{B}(f) = \partial\hat{f}(\mathbf{1}), \tag{10}$$

where $\mathbf{0} = \chi_\emptyset$ *and* $\mathbf{1} = \chi_V$ *in* $\mathbf{R}^V$ *and we assume* $\hat{f}(x) = +\infty$ *for any* $x \in \mathbf{R}^V \setminus \mathbf{R}^V_+$. □

Examples of a submodular function are the following.

(a) Cut functions for capacitated networks: Consider a capaciated network $N = (G = (V, A), c)$, where $G = (V, A)$ is the underlying graph with vertex set V and arc set A and $c : A \to \mathbf{R}_+$ is a non-negative capacity function. For each vertex subset $U \subseteq V$ let $\kappa_c(U)$ be the sum of the capacities of arcs leaving U. Then $\kappa_c : 2^V \to \mathbf{R}$ is a submodular function (see Fig.3).

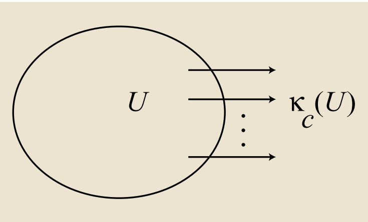

Fig 3. Cut function κ_c.

(b) Matroid rank functions, matrix rank functions, graph rank functions: Consider a (real) matrix M with a set E of columns. For any subset $X \subseteq E$ define $\rho_M(X)$ to be the rank of the submatrix M^X formed by columns in X. Then $\rho_M : 2^E \to \mathbf{R}$ is a submodular function. For a graph $G = (V, A)$ let M be the incidence matrix of G. Then ρ_M is the graph rank function.

(c) Multi-terminal flow-value functions: Consider a single source multiple-sink flow network $N = (G = (V, A), c, s^+, S^-)$, where $G = (V, A)$ is the underlying graph, $c : A \to \mathbf{R}_+$ is a capacity function, s^+ is a source, and S^- is the set of sinks. For any $X \subseteq S^-$ let $f(X)$ be the maximum flow value from s^+ to X in N. Then $f : 2^{S^-} \to \mathbf{R}$ is a submodular function.

(d) Entropy functions: Let $x_1, x_2, \cdots, x_n$ be random variables taking on values from a finite set. Put $E = \{x_1, x_2, \cdots, x_n\}$. For any $X \subseteq E$ define $h(X)$ to be the entropy of X in Shannon's sense. Then $h: 2^E \to \mathbf{R}$ is submodular.

One of the most general models of combinatorial optimization problems that can efficiently be solvable is the so-called *submodular flow problem* proposed by Edmonds-Giles [4] in 1977. The submodular flow problem includes a lot of graph and network optimization problems and the matroid-intersection problem. However, algorithms to solve the general submodular flow problem require an oracle (an efficient procedure) for submodular function minimization(see [8]). Combining our algorithm for submodular function minimization and the existing algorithms for submodular flows yields the first combinatorial strongly polynomial algorithms for the submodular flow problem.

It should also be noted that some combinatorial optimization problems such as a dynamic flow problem [11] and a location problem on trees [18] can be solved in strongly polynomial time only by using the general submodular function minimization algorithm.

3. Submodular function minimization

Our (strongly) polynomial algorithm [14] relies on the following min-max theorem. For any $x \in \mathbf{R}^V$ define $x^- \in \mathbf{R}^V$ as $x^-(v) = \min\{x(v), 0\}$ $(v \in V)$.

Theorem 3 ([3], [7, Cor. 3.5]) : *For any submodular function* $f : 2^V \to \mathbf{R}$ *with* $f(\emptyset) = 0$,

$$\max\{x^-(V) \mid x \in \mathrm{B}(f)\} = \min\{f(X) \mid X \subseteq V\}. \tag{11}$$

Moreover, if f *is integer-valued, an integral* x *attains the maximum in the left-hand side of* (11). □

Our algorithm tries to find an approximate maximizer of the left-hand side of (11). When f is integer-valued, if we get a (not necessarily integral) base $x \in \mathrm{B}(f)$ and a set $X \subseteq V$ such that $x^-(V) > f(X) - 1$, then we see that X is a minimizer of f.

First, we suppose that f is integer-valued. We express a base x as a convex combination of extreme bases (extreme points of $\mathrm{B}(f)$) y_i $(i \in I)$ as $x = \sum_{i \in I} \lambda_i y_i$, where $\lambda_i \geq 0$ $(i \in I)$ and $\sum_{i \in I} \lambda_i = 1$. This is an approach taken by Cunningham ([1], [2]). By the greedy algorithm of Edmonds-Shapley ([3], [17]), each extreme base y_i is given in terms of a linear ordering $L_i = (v_1, v_2, \cdots, v_n)$ of V as

$$y_i(v_l) = f(L_i(v_l)) - f(L_i(v_{l-1})) \quad (l = 1, 2, \cdots, n), \tag{12}$$

where $L_i(v_l) = \{v_1, v_2, \cdots, v_l\}$ and $n = |V|$. We increase $x^-(V)$ by transforming y_i's to their adjacent extreme bases. This would provide us with a pseudo-polynomial algorithm of Cunningham [2].

We then employ the capacity scaling algorithm for submodular flows, due to Iwata [12]. We perturb the given submodular function f by means of the cut function $\kappa_\delta : 2^V \to \mathbf{R}$ as

$$f(X) + \kappa_\delta(X) \quad (X \subseteq V), \tag{13}$$

where $\kappa_\delta(X) = \delta\,|X|\,|V \setminus V|$ $(X \subseteq V)$ and κ_δ is the cut function of the complete directed network N_δ with a parameter $\delta > 0$ and uniform capacities $c(u, v) = \delta$ for arcs (u, v). A flow φ in N_δ is called δ-*feasible* if $0 \leq \varphi(u, v) \leq \delta$ $(u, v \in V)$ and $\varphi(u, v) > 0$ implies $\varphi(v, u) = 0$. The boundary $\partial\varphi : V \to \mathbf{R}$ is defined by $\partial\varphi(u) = \sum_{v \in V}\varphi(u, v) - \sum_{v \in V}\varphi(v, u)$ $(u \in V)$. Denote by $\partial\Phi_\delta$ the set of all the boundaries of feasible flows in N_δ. The polyhedron $\partial\Phi_\delta$ is a base polyhedron and so is the Minkowski sum $\mathrm{B}(f) + \partial\Phi_\delta$. Instead of the original min-max relation (11) we then consider

$$\begin{aligned} &\max\{(x + z)^- \mid x \in \mathrm{B}(f),\ z \in \partial\Phi_\delta\} \\ &= \min\{f(X) + \kappa_\delta(X) \mid X \subseteq V\} \end{aligned} \tag{14}$$

and try to approximately maximize the left-hand side.

We call the procedure, given below, for a fixed $\delta > 0$ the δ*-scaling phase*. The basic idea for the δ-scaling phase is as follows. For a current base $x = \sum_{i \in I} \lambda_i y_i$ and a current δ-feasible flow φ we incease $\partial\varphi(s)$ for some $s \in S \equiv \{v \mid x(v) + \partial\varphi(v) \leq -\delta\}$ by δ and simultaneously decrease $\partial\varphi(s)$ for some $t \in T \equiv \{v \mid x(v) + \partial\varphi(v) \geq \delta\}$ by δ while keeping other $\partial\varphi(v)$ invariant. We call this operation δ-*augmentation*. The δ-augmentation can be carried out by finding a directed path in the so-called residual network associated with the current δ-feasible flow φ. We repeat the δ-augmentation as far as possible.

If the δ-augmentation becomes impossible, then let W be the set of vertices that can be reached from S in the residual network. If we have $y_i(W) = f(W)$ for all $i \in I$, then $x(W) = f(W)$ and we finish the δ-scaling phase (if necessary, put $\delta \leftarrow \frac{1}{2}\delta$ and $\varphi \leftarrow \frac{1}{2}\varphi$ and perform the δ-scaling phase for the new δ). It should be noted that if W is the set of elements in an initial segment of L_i, then we have $y_i(W) = f(W)$. While $W \cap T = \emptyset$ and $y_i(W) \neq f(W)$ for some $i \in I$, we modify $x = \sum_{i \in I} \lambda_i y_i$ and φ, keeping $x + \partial\varphi$ invariant. Though we omit the details of our algorithm (this part is the most complicated to describe and an idea from [5] is employed), in $O(n^3)$ time we obtain a set $W \subseteq V$ reachable from S and a base $x = \sum_{i \in I} \lambda_i y_i$ with a new set of extreme bases y_i ($i \in I$) such that $W \cap T \neq \emptyset$ or $y_i(W) = f(W)$ for each $i \in I$, where $|I| \leq 2n-1$, assuming $|I| \leq n$ at the beginning of the δ-scaling phase. When $W \cap T \neq \emptyset$, we can carry out a δ-augmentation. After the δ-augmentation we compute a new expression $x = \sum_{i \in I} \lambda_i y_i$ with $|I| \leq n$ by using affinely independent y_i's, which requires $O(n^3)$ time. We can show the following

Lemma 4: *At the end of the δ-scaling phase, $(x + \partial\varphi)^-(V) \geq f(W) - n\delta$ and hence $x^-(V) \geq f(W) - n^2\delta$.* □

The latter inequality shows that the difference between $f(W)$ and the minimum of f is at most $n^2\delta$, so that if $\delta < 1/n^2$, then W is a minimizer of f, due to Theorem 3.

Since $x + \partial\varphi \in B(f) + \partial\Phi_\delta$, we have $(x + \partial\varphi)(W) \leq f(W) + n^2\delta/4$. Therefore, it follows from Lemma 4 that after putting $\delta \leftarrow \delta/2$ and $\varphi \leftarrow \varphi/2$, we have

$$f(W) - 2n\delta - n^2\delta/4 \leq (x + \partial\varphi)^-(V) \leq f(W) + n^2\delta/4 \qquad (15)$$

at the beginning of the next δ-scaling phase. Hence there are $O(n^2)$ δ-augmentations in the next δ-scaling phase. If we choose any extreme base as the initial x and put $\varphi \leftarrow \mathbf{0}$ and $\delta \leftarrow \min\{|x^-(V)|, x^+(V)\}/n^2$ where $x^+(v) = \max\{x(v), 0\}$, then there are $O(n^2)$ δ-augmentations in the initial δ-scaling phase as well.

For an initial base x and a set $X = \{v \mid x(v) > 0\}$ we have $\min\{|x^-(V)|, x^+(V)\} \leq x^+(V) = x(X) \leq f(X)$. It follows that defining $M = \max\{|f(X)| \mid X \subseteq V\}$, we perform $O(\log M)$ scaling phases from the initial δ till $\delta < 1/n^2$.

Consequently,

(1) there are $O(\log M)$ scaling phases,

(2) there are $O(n^2)$ δ-augmentations in each δ-scaling phase.

(3) each δ-augmentation requires $O(n^3)$ time.

Hence the algorithm desribed above finds a minimizer of the integer-valued submodular function f in $O(n^5 \log M)$ time. This is a combinatorial, weakly polynomial algorithm. We utilize the weakly polynomial algorithm to devise a strongly polynomial algorithm.

Using the weakly polynomial algorithm, we can achieve one of the following four:

(i) We find that a base $x \leq \mathbf{0}$ exists, and V is a minimizer. Here, V may be modified by operations (ii) ~ (iv) given below.

(ii) After performing $O(\log n)$ scaling phases We find an element that does not belong to any minimizer of f. We delete such an element from the underlying set V.

(iii)After performing $O(\log n)$ scaling phases we find an element that belongs to any minimizer of f. We contract such an element.

(iv)After performing $O(\log n)$ scaling phases we find a pair of elements (u,w) such that any minimizer of f containing u contains w. If we have a directed cycle formed by arcs represented by such pairs (u,w), then we shrink elements lying on the cycle to a single element.

(ii) and (iii) are repeated $O(n)$ times and (iv) is $O(n^2)$ times.

Each scaling phase requires $O(n^5)$ time and there are $O(\log n)$ scaling phases. Hence the total running time is $O(n^7 \log n)$. We thus get a strongly polynomial algorithm.

4. Concluding remarks

We have solved the long-standing open problem but as Schrijver[16] pointed out, both Schrijver's and our algorithms employ multiplications and/or divisions. It is desirable to construct a fully combinatorial polynomial algorithm for submodular function minimization that reqires only additions and subtractions. Very recently Iwata [13] solved this problem.

Further research is extensively being made to improve the time complexity of the proposed algorithms and to simplify them.

References

1. Cunningham, W. H.: Testing membership in matroid polyhedra, *Journal of Combinatorial Theory*, **B36** (1984), 161-188.
2. Cunningham, W. H.: On submodular function minimization, *Combinatorica*, **5** (1985), 185-192
3. Edmonds, J.: Submodular functions, matroids, and certain polyhedra, *Combinatorial Structures and Their Applications*, R. Guy, H. Hanani N. Sauer, and J. Schönheim, eds., Gordon and Breach, 1970, 69-87.
4. Edmonds, J., and Giles, R.: A min-max relation for submodular function on graphs, *Annals of Discrete Mathematics*, **1** (1977),185-204.
5. Fleischer, L., Iwata, S., and McCormick, S. T.: A faster capacity scaling algorithm for submodular flow, July 1999.
6. Fujishige, S.: Theory of submodular programs: A Fenchel-type min-max theorem and subgradients of submodular functions, *Mathematical Programming*, **29** (1984), 142--155.
7. Fujishige, S.: *Submodular Functions and Optimization*, North-Holland, 1991.
8. Fujishige, S., and Iwata, S.: Algorithms for submodular flows. *IEICE Transactions on Information and Systems* **E83- D** (2000) 322-329.
9. Grötschel, M., Lovász, L., and Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica*, **1** (1981), 169-197.
10. Grötschel, M., Lovász, L., and Schrijver, A.: *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, 1988.
11. Hoppe, B., and Tardos, É.: The quickest transshipment problem, *Mathematics of Operations Research*, **25** (2000), 36-62.
12. Iwata, S.: A capacity scaling algorithm for convex cost submodular flows, *Mathematical Programming*, **76** (1997), 299-308.
13. Iwata, S.: A fully combinatorial algorithm for submodular function minimization, *J. Combinatorial Theory*, Ser.B (to appear).
14. Iwata, S., Fleischer, L., and Fujishige, S.: A combinatorial strongly polynomial algorithm for minimizing submodular functions.*Proceedings of the 32nd Annual ACM Symposium on Theory of Computing* (2000), 97-106; *Journal of ACM* (to appear).
15. Lovász, L.: Submodular functions and convexity. *Mathematical Programming — The State of the Art*, A.Bachem, M.Grötschel, and B.Korte, eds., Springer-Verlag, 1983, 235-257.
16. Schrijver, A.: A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *Journal of Combinatorial Theory*, Ser.B **80** (2000), 346-355.
17. Shapley, L. S.: Cores of convex games, *International Journal of Game Theory*, **1** (1971), 11-26.
18. Tamir, A.: A unifying location model on tree graphs based on submodularity properties, *Discrete Applied Mathematics*, **47** (1993), 275-283.