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 \( \mathbb{R} \) be the set of reals and \( \mathbb{R}^+ \) the set of non-negative reals. We call a function \( f : 2^V \to \mathbb{R} \) 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 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 the class of submodular functions is a special class of subadditive functions. 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 \( \square : \mathbb{R} \to \mathbb{R} \), then we get a submodular function \( f : 2^V \to \mathbb{R} \) defined by

\[
 f(X) = \square (|X|) \tag{4}
\]

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 \( f : 2^V \to \mathbb{R} \) with \( f(\emptyset) = 0 \) define a function \( \hat{f} : \mathbb{R}^V \to \mathbb{R} \) as follows. For a non-zero vector \( x \in \mathbb{R}^V \) 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 \( \lambda_i (i = 1, 2, \cdots, k) \) such that

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

where \( \chi_S \) is the characteristic vector of set \( S \). 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 \( \hat{f}(0) = 0 \). The function \( \hat{f} \) is called the Lovász extension of \( f \).

Theorem 1 (Lovász[15]): A set function \( f : 2^V \to \mathbb{R} \) 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 \( f : 2^V \to \mathbb{R} \), we define convex polyhedra \( P(f) \) and \( B(f) \), respectively, called the submodular polyhedron and the base polyhedron as

\[
P(f) = \{x \mid x \not\subseteq \mathbb{R}^V, \ \forall X \subseteq V : x(X) \leq f(X)\}, \tag{8}
\]

\[
B(f) = \{x \mid x \not\subseteq \mathbb{R}^V, x(V) = f(V)\}. \tag{9}
\]

where \( x(X) = \sum_{Y \subseteq X} x(Y) \) (see [7]). A vector in \( B(f) \) is called a base. Both \( P(f) \) and \( B(f) \) uniquely determine \( f \) (see Fig.2). The submodular polyhedron \( P(f) \) and the base polyhedron \( B(f) \) are also related to the Lovász extension of \( f \) as follows. For a convex function \( h : \mathbb{R}^V \to \mathbb{R} \cup \{+\infty\} \) and a vector \( x \) with \( h(x) \leq +\infty \) the subdifferential \( \partial h(x) \) of \( h \) at \( x \) is defined by \( \partial h(x) = \{z \mid z \in \mathbb{R}^V, \forall y \in \mathbb{R}^V : h(y) \in (\langle z, y-x \rangle \leq h(y)) \} \), where \( \langle \cdot, \cdot \rangle \) is the canonical inner product.
Theorem 2 (Fujishige[6]): We have
\[ P(f) = \partial \tilde{f}(0), \quad B(f) = \partial \tilde{f}(1), \]  
where \( 0 = \partial_b \) and \( 1 = \partial_v \) in \( R^V \) and we assume \( \tilde{f}(x) = +\infty \) for any \( x \in R^V \setminus R^V_+ \).
\[
\square
\]

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

(b) Matroid rank functions, matrix rank functions, graph rank functions: Consider a (real) matrix \( M \) to be the rank of the submatrix \( M^X \) formed by columns in \( X \). Then \( \partial(M) : 2^E \to R \) is a submodular function. For a graph \( G = (V, A) \) let \( M \) be the incidence matrix of \( G \). Then \( \partial(M) \) is the graph rank function.
(c) Multi-terminal flow-value functions: Consider a single source \( (V, A, c) \), \( A \subseteq X \) is a submodular function, \( x \) is the maximum flow value from \( u \) to \( v \). We have
\[ \partial(\) \subseteq V \) \( ) = min { f(X) \mid X \subseteq V \} \] 

(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 \( \partial(M)(X) \) to be the rank of the submatrix \( M^X \) formed by columns in \( X \). Then \( \partial(M) : 2^E \to R \) is a submodular function. For a graph \( G = (V, A) \) let \( M \) be the incidence matrix of \( G \). Then \( \partial(M) \) is the graph rank function.
(c) Multi-terminal flow-value functions: Consider a single source \( (V, A, c) \), \( A \subseteq X \) is a submodular function, \( x \) is the maximum flow value from \( u \) to \( v \). We have
\[ \partial(\) \subseteq V \) \( ) = min { f(X) \mid X \subseteq V \} \] 

(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 \( \partial(M)(X) \) to be the rank of the submatrix \( M^X \) formed by columns in \( X \). Then \( \partial(M) : 2^E \to R \) is a submodular function. For a graph \( G = (V, A) \) let \( M \) be the incidence matrix of \( G \). Then \( \partial(M) \) is the graph rank function.
(c) Multi-terminal flow-value functions: Consider a single source \( (V, A, c) \), \( A \subseteq X \) is a submodular function, \( x \) is the maximum flow value from \( u \) to \( v \). We have
\[ \partial(\) \subseteq V \) \( ) = min { f(X) \mid X \subseteq V \} \] 

(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 \( \partial(M)(X) \) to be the rank of the submatrix \( M^X \) formed by columns in \( X \). Then \( \partial(M) : 2^E \to R \) is a submodular function. For a graph \( G = (V, A) \) let \( M \) be the incidence matrix of \( G \). Then \( \partial(M) \) is the graph rank function.
(c) Multi-terminal flow-value functions: Consider a single source \( (V, A, c) \), \( A \subseteq X \) is a submodular function, \( x \) is the maximum flow value from \( u \) to \( v \). We have
\[ \partial(\) \subseteq V \) \( ) = min { f(X) \mid X \subseteq V \} \] 

(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 \( \partial(M)(X) \) to be the rank of the submatrix \( M^X \) formed by columns in \( X \). Then \( \partial(M) : 2^E \to R \) is a submodular function. For a graph \( G = (V, A) \) let \( M \) be the incidence matrix of \( G \). Then \( \partial(M) \) is the graph rank function.
(c) Multi-terminal flow-value functions: Consider a single source \( (V, A, c) \), \( A \subseteq X \) is a submodular function, \( x \) is the maximum flow value from \( u \) to \( v \). We have
\[ \partial(\) \subseteq V \) \( ) = min { f(X) \mid X \subseteq V \} \]
set of elements in an initial segment of \( I_i \), then we have \( y_j(W) = f(W) \). While \( W \cap T = \emptyset \) and \( y_j(W) \neq f(W) \) for some \( i \in I \), we modify \( x = \sum_{i \in I} y_i \) and \( \phi \), keeping \( x + \partial \phi \) 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 \( x = \sum_{i \in I} y_i \) with a new set of extreme bases \( y(i) \) such that \( W \cap T = \emptyset \) or \( y(W) = f(W) \) for each \( i \in I \), where \( |I| \leq 2n-1 \), assuming \( |I| \leq n \) at the beginning of the \( \diamond \)-scaling phase. When \( W \cap T \neq \emptyset \), we can carry out a \( \diamond \)-augmentation. After the \( \diamond \)-augmentation we compute a new expression \( x = \sum_{i \in I} y_i \) with \( |I| \leq n \) by using affinely independent \( y_i \)'s, which requires \( O(n^2 \log n) \) time. We thus get a strongly polynomial algorithm.

**Lemma 4:** At the end of the \( \square \)-scaling phase, \((x + \partial \phi)^{-1}(V) \geq f(W) + n^2 \square /4 \) and hence \( x(V) \geq f(W) - n^2 \square \). \( \square \)

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

Since \( x + \partial \phi \diamond B(f) + \partial \phi \diamond \), we have \((x + \partial \phi)^{-1}(W) \leq f(W) + n^2 \square /4 \). Therefore, it follows from Lemma 4 that after putting \( \square \leftarrow \square /2 \) and \( \phi \leftarrow \phi /2 \), we have

\[
(f(W) - 2n \square - n^2 \square /4 \leq (x + \partial \phi)^{-1}(V) \leq f(W) + n^2 \square /4)
\]

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

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

Consequently,

1. there are \( O(\log M) \) scaling phases,
2. there are \( O(n^2 \diamond) \) \( \diamond \)-augmentations in each \( \diamond \)-scaling phase.
3. each \( \diamond \)-augmentation requires \( O(n^3) \) time.

Hence the algorithm described above finds a minimizer of the integer-valued submodular function \( f \) in \( O(n^2 \log M) \) 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 \( x \leq 0 \) exists, and \( V \) is a minimizer. Here, \( V \) may be modified by operations (ii) \sim (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 \( O(n^2 \log n) \) times.

Each scaling phase requires \( O(n^2) \) time and there are \( O(\log n) \) scaling phases. Hence the total running time is \( O(n^2 \log n) \). 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 requires 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*, **I** (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*, **I** (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.Grótschel, 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. Shepley, 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.