

劣モジュラ関数の最小化

01012384 東京大学 岩田 覚 IWATA Satoru

有限集合 V 上の集合関数 $f: 2^V \rightarrow \mathbf{R}$ が任意の $X, Y \subseteq V$ に対して

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \quad (1)$$

を満たすとき、 f を劣モジュラ関数という。関数値を計算する手続きが存在するものと仮定して、劣モジュラ関数の最小値を求める問題を考える。

一般性を失うことなく、 $f(\emptyset) = 0$ と仮定する。このとき、 \mathbf{R}^V 上の劣モジュラ多面体 $P(f)$ と基多面体 $B(f)$ が

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

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

によって定義される。ただし、 $x(X) = \sum_{v \in X} x(v)$ 。

数理計画法における劣モジュラ関数の重要性を最初に指摘したのは、Edmonds [3] である。劣モジュラ多面体 $P(f)$ 上で非負ベクトル $c \in \mathbf{R}^V$ を係数ベクトルとする線形関数の最大化問題を考える。各要素を $c(v)$ の大きい順にならべ、 v 以前に位置する要素の集合を $L(v)$ と表す。このとき、各 $v \in V$ に対して

$$y(v) = f(L(v)) - f(L(v) \setminus \{v\})$$

として得られるベクトル $y \in \mathbf{R}^V$ は、 $P(f)$ の端点となり、線形目的関数の最大値を達成する。このアルゴリズムは貪欲アルゴリズムと呼ばれている。

劣モジュラ関数の最小値を求める最初の多項式時間アルゴリズムは、Grötschel, Lovász, Schrijver [8] によって示された。彼らは、楕円体法を用いて、より一般に多面体上の最適化問題と分離問題とが多項式時間可解性の意味で等価であることを示した。分離問題とは、与えられた点が多面体に含まれるかどうかを判定し、含まれない場合には、両者を分離する不等式を求める問題である。劣モジュラ多面体においては、貪欲アルゴリズムによって最適化問題が多項式時間で解けるのであるから、分離問題も多項式時間で解けることになる。原点に関する分離問題を解

くと、劣モジュラ関数が非負かどうかの判定ができる。したがって、二分法によって、劣モジュラ関数の最小値が多項式時間で計算できる。

劣モジュラ関数の最小値を計算する方法としては、Lovász [13] によって明らかにされた凸性との関連を利用する手もある。集合関数 f が与えられた際に、単位立方体 $Q = \{p \mid p \in \mathbf{R}^V, \forall v \in V: 0 \leq p(v) \leq 1\}$ 中の p に対して、 p に関する貪欲アルゴリズムを適用した結果の目的関数値を $\hat{f}(p)$ とする。こうして得られた $\hat{f}: Q \rightarrow \mathbf{R}$ は、 f が劣モジュラ関数のとき、かつそのときに限り凸関数となる。また、定義から明らかのように、 \hat{f} の最小値は、 Q の端点で達成される。したがって、劣モジュラ関数 f の最小値を計算するには、凸関数 \hat{f} の最小値を計算すればよい。Grötschel, Lovász, Schrijver [9] は、凸関数最小化の部分に楕円体法を用いることによって、劣モジュラ関数の最小値を求める強多項式時間アルゴリズム、すなわち、計算量が関数値の大きさとは無関係に、 $n = |V|$ のみの多項式オーダーとなるアルゴリズムの存在を示している。

楕円体法を用いたこれらの方法は、計算量理論の観点からは“効率的なアルゴリズム”に分類されるものの、実際上は極めて非効率的であることが、予想される。実用的な方法として、Fujishige [6] は、基多面体における最小ノルム点、すなわち原点からの Euclid 距離が最小となる点を求めることによって、劣モジュラ関数を最小化するアルゴリズムを示している。この方法は、多項式性の保証はないものの、実際には速く、いわば線形計画法における単体法に相当する。

実際に高速な多項式時間アルゴリズムを開発するには、組合せ的なアルゴリズムの設計が有効であろうと期待される。この方向では、Cunningham [2] が、整数値劣モジュラ関数の最小値を擬多項式時間、すなわち $n = |V|$ と $M = \max |f(X)|$ との多項式時間で計算する組合せ的なアルゴリズムを開発した。この

アルゴリズムの出発点は、次の最大最小定理である。

定理 1 ([3]) 任意の劣モジュラ関数 f に関して、 $\min\{f(X) \mid X \subseteq V\} = \max\{x(V) \mid x \in P(f)\}$. ■

任意の $x \in \mathbf{R}^V$ に対して、 $x^-(v) = \min\{0, x(v)\}$ で $x^- \in \mathbf{R}^V$ を定めるものとする、定理 1 の右辺は、 $\max\{x^-(V) \mid x \in B(f)\}$ と書き換えられる。この形から、 $x \in B(f)$ を維持しながら、 $x^-(V)$ の増大を図るというアルゴリズムの原型が思い浮かぶ。しかし、 $x \in B(f)$ を維持すること自体が、決して自明でない。Cunningham [2] は、貪欲アルゴリズムによって得られる端点の凸結合として $x \in B(f)$ を維持することを提案した。

Cunningham のアルゴリズムを骨格とし、劣モジュラ流問題の容量スケールリング法 [10, 5] で導入された新たな手法を用いて、我々は、整数値劣モジュラ関数の最小値を求める組合せ的な多項式時間アルゴリズムを開発した [12]。このアルゴリズムの計算量は、 $O(n^5 \gamma \log M)$ である。ただし、 γ は関数値の計算に要する時間を表す。さらに、このアルゴリズムを繰り返し呼び出すことによって、 $O(n^7 \gamma \log n)$ 時間で劣モジュラ関数を最小化する強多項式時間アルゴリズムも示した。

一方、Schrijver [14] も、我々の研究とは全く独立に、Cunningham [1, 2] の枠組を踏襲した組合せ的な強多項式時間アルゴリズムを発表している。アルゴリズムの計算量は $O(n^8 \gamma + n^9)$ であるが、若干の変更を施すことによって、 $O(n^7 \gamma + n^8)$ に改善されている [4]。

これらのアルゴリズムは、組合せ的とはいうものの、凸結合を維持するために乗除算を用いている。そこで、Schrijver [14] は、加減算と大小比較のみを用いるという意味で完全に組合せ的な強多項式時間アルゴリズムの存在を未解決問題として提起した。この問題は、[12] のアルゴリズムを变形することで、肯定的に解決された [11]。

今後の研究課題としては、計算機実験による新たなアルゴリズムの実用性の検証、より高速なアルゴリズムの開発が挙げられる。さらには、楕円体法によって多項式時間アルゴリズムの存在が示されている種々の最適化問題に対する組合せ的アルゴリズム

の開発が興味深い。この第一歩として、双劣モジュラ関数最小化を行う多項式時間アルゴリズムが得られている [7]。

参考文献

- [1] W. H. Cunningham: Testing membership in matroid polyhedra, *J. Combin. Theory*, **B36** (1984), 161–188.
- [2] W. H. Cunningham: On submodular function minimization, *Combinatorica*, **5** (1985), 185–192.
- [3] J. Edmonds: Submodular functions, matroids, and certain polyhedra, *Combinatorial Structures and Their Applications*, Gordon and Breach, 69–87, 1970.
- [4] L. Fleischer and S. Iwata: Improved algorithms for submodular function minimization and submodular flow, *Proc. 32nd STOC* (2000), 107–116.
- [5] L. Fleischer, S. Iwata, and S. T. McCormick: A faster capacity scaling algorithm for minimum cost submodular flow, *Math. Programming*, to appear.
- [6] S. Fujishige: *Submodular Functions and Optimization*, North-Holland, 1991.
- [7] S. Fujishige and S. Iwata: Bisubmodular function minimization, *Proc. IPCO VIII* (2001), 160–169.
- [8] M. Grötschel, L. Lovász, and A. Schrijver: The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica*, **1** (1981), 169–197.
- [9] M. Grötschel, L. Lovász, and A. Schrijver: *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, 1988.
- [10] S. Iwata: A capacity scaling algorithm for convex cost submodular flows, *Math. Programming*, **76** (1997), 299–308.
- [11] S. Iwata: A fully combinatorial algorithm for submodular function minimization, *J. Combin. Theory*, **B**, to appear.
- [12] S. Iwata, L. Fleischer, and S. Fujishige: A combinatorial strongly polynomial algorithm for minimizing submodular functions, *J. ACM*, to appear.
- [13] L. Lovász: Submodular functions and convexity. *Mathematical Programming — The State of the Art*, Springer-Verlag, 1983, 235–257.
- [14] A. Schrijver: A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *J. Combin. Theory*, **B80** (2000), 346–355.