組合せ最適化のアルゴリズムと離散凸解析

塩浦　昭義

本稿では，離散凸関数の概念である M 凸関数と L 凸関数に対し，それらの最小化問題を考える．これらの問題は離散凸解析における基本的な最適化問題であり，一般的な組合せ最適化問題の枠組みを与える．本稿ではまず，最小木問題や最短路問題などの組合せ最適化問題が M 凸関数最小化や L 凸関数最小化の特殊な場合と見なせることを示す．次に，一般の M 凸関数と L 凸関数の最小化問題に対する貪欲アルゴリズムを説明する．この貪欲アルゴリズムは最小木問題や最短路問題にも適用可能であるが，これらの問題にアルゴリズムを特殊化すると，クラスカル法やダイクストラ法のような有名なアルゴリズムが導出されることを示す．

キーワード：組合せ最適化，離散凸解析，離散凸関数最小化，貪欲アルゴリズム

1. はじめに

組合せ最適化の分野にはさまざまな問題が存在するが，解きやすい（つまり，効率的に解ける）問題もあれば解きにくい（つまり，効率的に解くことが難しい）問題もある．解きやすい問題の代表格としては最小木問題（例 2.1 参照）や最短路問題（例 2.3 参照）が挙げられる．この問題は組合せ最適化の教科書には必ず載っている問題であるが，教科書をみると，これらの問題は数学的に良い性質を持っていて，クラスカル法やダイクストラ法というアルゴリズムにより解くことができる，などと書かれている．このような事実を，離散凸解析の視点からあらためて理解しよう，というのが本稿の目的である．

離散凸解析とは解きやすい組合せ最適化問題に対する統一的な枠組みであり（詳しくは本特集の室田氏の記事を参照），離散凸性という視点から，組合せ最適化問題の解きやすさを理解することが離散凸解析の大きな目的である．離散凸解析では M 凸関数と L 凸関数と呼ばれる離散凸関数が中心的な役割を果たしており，これまでの研究により，M 凸関数と L 凸関数が数学的に良い性質を持っていて，離散凸関数としてふさわしい概念であることが知られている．

本稿では，離散凸解析において最も基本的な最適化問題である M 凸関数と L 凸関数の最小化問題について考える．この問題は一般的な組合せ最適化問題の枠組みを与えるが，実際に最小木問題や最短路問題などの組合せ最適化問題が，M 凸関数と L 凸関数の最小化の特殊な場合とみなせることを第 2 節で示す．このことは，最小木問題や最短路問題という問題が解きやすいという既知の事実を，離散凸解析の立場から裏づけると同時に，これらの問題に対するより良い理解を与えるものである．

次に，M 凸関数と L 凸関数の最小化問題が実際に解きやすい問題であることを示すために，最急降下法のような貪欲アプローチにより解けることを第 3 節および第 4 節で説明する．ここでは，貪欲アルゴリズムの出力として最小解が得られるだけではなく，アルゴリズムの各反復において「単調に」最小解に近づいていくように解が更新されることを示す．

先に述べたように，最小木問題や最短路問題は M 凸関数最小化や L 凸関数最小化の特殊ケースなので，これらの問題に対して上記で述べた貪欲アルゴリズムが適用可能である．第 5 節では，離散凸関数最小化の貪欲アルゴリズムを最小木問題や最短路問題などの問題に対して特殊化すると，クラスカル法やダイクストラ法のような有名なアルゴリズムに一致することを示す．このことは，離散凸関数最小化の貪欲アルゴリズムが（特殊ケースとして）自然な形で既に存在していたことを示すとともに，既存のアルゴリズムに対するより良い理解と新たな見方を与える．

2. 離散凸関数の最小化

M 凸関数と L 凸関数の最小化問題を説明する．M 凸関数と L 凸関数の定義など，本稿で定義していない用語や記号については，本特集の室田氏の記事を参照されたい．

2.1 M 凸関数の最小化とその例

まず，M 凸関数 $f : \mathbb{Z}^n \to \overline{\mathbb{R}}$ が与えられたとき，そ

しおうら　あきよし
東北大学大学院情報科学研究科
〒 980-8579 宮城県仙台市青葉区荒巻字青葉 6–3–09

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

れを実効定義域 $\mathrm{dom}\, f$ 上で最小化する問題 (Mmin)

$$\text{最小化}\quad f(x) \qquad \text{条件}\quad x \in \mathrm{dom}\, f$$

を考える．なお，M 凸関数と $\mathrm{M}^\natural$ 凸関数の概念は等価であるので，以下では主に M 凸関数を扱う．この問題に対するアルゴリズムを第 4.1 節で示す．

また，成分和一定という制約条件の下で $\mathrm{M}^\natural$ 凸関数 $f : \mathbb{Z}^n \to \overline{\mathbb{R}}$ を最小化する問題 ($\mathrm{M}^\natural\mathrm{min}(r)$)

$$\text{最小化}\ \ f(x) \quad \text{条件}\ \ \sum_{i=1}^{n} x(i) = r,\ \ x \in \mathrm{dom}\, f$$

も扱う．問題 ($\mathrm{M}^\natural\mathrm{min}(r)$) は，M 凸関数や $\mathrm{M}^\natural$ 凸関数の最小化問題として書き直せる．関数 $f_1 : \mathbb{Z}^n \to \overline{\mathbb{R}}$ を

$$f_1(x) = \begin{cases} f(x) & \left(\sum_{i=1}^{n} x(i) = r \text{ のとき}\right), \\ +\infty & (\text{それ以外}) \end{cases} \quad (1)$$

と定義すると，f_1 は M 凸関数であり，f_1 の最小解集合は問題 ($\mathrm{M}^\natural\mathrm{min}(r)$) の最適解集合と一致する．また，十分大きな正の実数 Γ を用いて関数

$$f_2(x) = f(x) + \Gamma \left| \sum_{i=1}^{n} x(i) - r \right| \quad (x \in \mathbb{Z}^n) \quad (2)$$

を定義すると，$f_2 : \mathbb{Z}^n \to \overline{\mathbb{R}}$ は $\mathrm{dom}\, f_2 = \mathrm{dom}\, f$ を満たす $\mathrm{M}^\natural$ 凸関数であり，f_2 の最小解集合は問題 ($\mathrm{M}^\natural\mathrm{min}(r)$) の最適解集合と一致する．ここで，$f_2$ の最小解 x_* は必ず $\sum_{i=1}^{n} x_*(i) = r$ を満たすことに注意する．

したがって，問題 ($\mathrm{M}^\natural\mathrm{min}(r)$) を解くには，式 (1)，式 (2) の関数 f_1，f_2 に M 凸関数，$\mathrm{M}^\natural$ 凸関数の最小化アルゴリズムを適用すればよいが，第 4.2 節では，問題 ($\mathrm{M}^\natural\mathrm{min}(r)$) の特殊性を利用することで，直観的にわかりやすく，かつ効率的なアルゴリズムが得られることを示す．

以下では，組合せ最適化問題から生じる M 凸関数最小化の例を示す．最も基本的な組合せ最適化問題の一つである最小木問題は，問題 (Mmin) や ($\mathrm{M}^\natural\mathrm{min}(r)$) の特殊ケースと見ることができる．

例 2.1 (最小木問題) 無向グラフ $G = (V, E)$ および枝長 $d(e)$ $(e \in E)$ が与えられたとき，総枝長が最小の全域木（最小木）を求める最小木問題を考える（図 1 参照）．集合 $S_1, S_2 \subseteq \{0,1\}^E$ を

$$S_1 = \{e_X \mid X \text{ は全域木の枝集合}\},$$
$$S_2 = \{e_X \mid X \text{ は閉路を含まない枝集合}\}$$

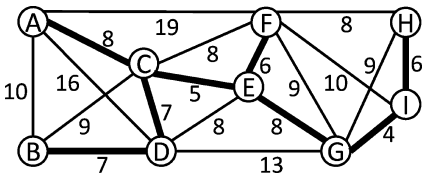

図 1 最小木問題の例（太線は最小木を表す）

とおき[1]，関数 $f_{\mathrm{MST1}} : \mathbb{Z}^E \to \overline{\mathbb{R}}$ を

$$f_{\mathrm{MST1}}(x) = \begin{cases} \sum_{e \in X} d(e) & (x = e_X \in S_1), \\ +\infty & (\text{それ以外}) \end{cases}$$

によって定義すると，関数 f_{MST1} は S_1 を実効定義域とする M 凸関数である．つまり，最小木問題は M 凸関数 f_{MST1} の最小化問題として書き換えられる．

また，上の関数の定義において S_1 を S_2 に置き換えて得られる関数を f_{MST2} とおくと，これは $\mathrm{M}^\natural$ 凸関数である．全域木は，閉路を含まず，枝数が $|V| - 1$ である枝集合なので，最小木問題は f_{MST2} に関する問題 ($\mathrm{M}^\natural\mathrm{min}(r)$) において $r = |V| - 1$ とおいたものと等価である． ■

また，OR の典型的な問題である資源配分問題も問題 (Mmin) や ($\mathrm{M}^\natural\mathrm{min}(r)$) の特殊ケースである．

例 2.2 (資源配分問題) 制約 $\sum_{i=1}^{n} x(i) = r$ （r は非負整数）の下で分離凸関数 $\sum_{i=1}^{n} \varphi_i(x(i))$ （各 $\varphi_i : \mathbb{Z} \to \mathbb{R}$ は一変数離散凸関数）を最小化する非負整数ベクトル x を求める（単純）資源配分問題を考える．資源配分問題が与えられたとき，関数 $f_{\mathrm{RA1}} : \mathbb{Z}^n \to \overline{\mathbb{R}}$ を

$$f_{\mathrm{RA1}}(x) = \begin{cases} \sum_{i=1}^{n} \varphi_i(x(i)) & \left(x \geq 0 \text{ かつ} \sum_{i=1}^{n} x(i) = r\right), \\ +\infty & (\text{それ以外}) \end{cases}$$

によって定義すると，関数 f_{RA1} は M 凸関数であり，資源配分問題は M 凸関数 f_{RA1} の最小化問題と書き直せる．

一方，関数 f_{RA1} の定義における等式 $\sum_{i=1}^{n} x(i) = r$ を不等式 $\sum_{i=1}^{n} x(i) \leq r$ に置き換えて得られる関数 f_{RA2} は $\mathrm{M}^\natural$ 凸関数であり，資源配分問題を問題 ($\mathrm{M}^\natural\mathrm{min}(r)$) の形に書き直すことができる． ■

[1] $e_X \in \{0,1\}^E$ は集合 $X \subseteq E$ の特性ベクトルであり，$i \in X$ のとき $e_X(i) = 1$，$i \notin X$ のとき $e_X(i) = 0$ である．

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

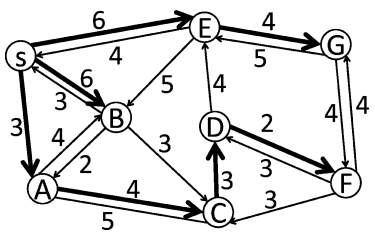

図 2　最短路問題の例（太線は最短路を表す）

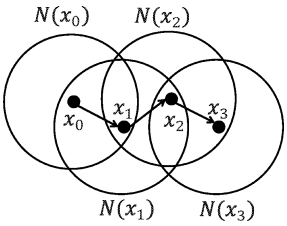

図 3　貪欲アプローチのイメージ（黒い点は各反復の解 x，大きな円は近傍 $N(x)$ を，それぞれ表す）

2.2　L 凸関数の最小化とその例

$\mathrm{L}^\natural$ 凸関数 $g : \mathbb{Z}^n \to \overline{\mathbb{R}}$ が与えられたとき，それを実効定義域 $\operatorname{dom} g$ 上で最小化する問題 ($\mathrm{L}^\natural$min)

$$\text{最小化}\quad g(p) \qquad \text{条件}\quad p \in \operatorname{dom} g$$

を考える．L 凸関数の最小化問題は $\mathrm{L}^\natural$ 凸関数の最小化問題の特殊ケースなので，以下では主に $\mathrm{L}^\natural$ 凸関数を扱う．

次の例で示すように，最短路問題や最小凸費用流問題の双対問題は，$\mathrm{L}^\natural$ 凸（L 凸）関数最小化問題の特殊ケースと見ることができる．

例 2.3 (最短路問題)　有向グラフ $G = (V, E)$ および非負整数値の枝長 $\ell(e)$ $(e \in E)$ が与えられたとき，頂点 s からすべての頂点への最短路を同時に求める問題を考える（図 2 参照）．集合

$$S = \{p \in \mathbb{Z}^V \mid p(s) = 0, \\ p(v) - p(u) \le \ell(u, v)\ (\forall (u, v) \in E)\}$$

は原点 $\mathbf{0}$ を含むので非空であるが，これを用いて関数 $g_{\mathrm{SP}} : \mathbb{Z}^V \to \mathbb{Z} \cup \{-\infty\}$ を

$$g_{\mathrm{SP}}(p) = \begin{cases} \displaystyle\sum_{v \in V} p(v) & (p \in S), \\ -\infty & (\text{それ以外}) \end{cases}$$

によって定義する．すると，g_{SP} は S を実効定義域とする $\mathrm{L}^\natural$ 凹関数である．なお，関数 g_{SP} を最大化する $p_* \in S$ は一意に定まり，値 $p_*(v)$ は頂点 s から頂点 v への最短路長に一致する．よって，$\mathrm{L}^\natural$ 凹関数 g_{SP} の最大化問題を解くことによって，頂点 s から各頂点への最短路長を求めることができる．■

例 2.4 (最小凸費用流問題の双対)　グラフ $G = (V, E)$ と，$\sum_{u \in V} b(u) = 0$ を満たすベクトル $b \in \mathbb{Z}^V$，および各枝 e の容量 $c(e) \in \mathbb{Z}_+$ と費用 $k(e) \in \mathbb{Z}$ が与えられたとき，最小費用流問題は次のように定式化される：

$$\begin{aligned} \text{最小化}\quad & \sum_{e \in E} k(e)x(e) \\ \text{制約条件}\quad & \partial x(u) = b(u) \quad (u \in V), \\ & 0 \le x(e) \le c(e) \quad (e \in E). \end{aligned}$$

この問題は線形計画問題であり，その双対問題は

$$g_{\mathrm{D}}(p) = \sum_{u \in V} b(u)p(u) \\ + \sum_{(u,v) \in E} c(u, v) \min\{0, -p(u) + p(v) + k(u, v)\}$$

と与えられる関数 g_{D} の最大化問題として書ける．入力データが整数値であるという仮定より，この問題は整数最適解を持つ．この事実を踏まえ，関数 g_{D} を整数ベクトル $p \in \mathbb{Z}^V$ に関する関数とみなすと，g_{D} は L 凹関数である．つまり，最小費用流問題の双対問題は L 凹関数の最大化問題の特殊ケースである．■

3.　貪欲アプローチ

離散凸関数の最小化に対する基本的な手法である，貪欲アプローチを説明する．このアプローチでは，アルゴリズムの各反復において現在の解 x の近傍 $N(x)$ を調べ，近傍の中で関数値が最小の解（もしくは関数値がより小さい解）に移動することを繰り返す（図 3 参照）．近傍 $N(x)$ は前もって適切に定めることになる．この手法は，局所探索法や最急降下法とも呼ばれる．より具体的には，以下のように記述される．

手順 0：初期解 $x \in \operatorname{dom} f$ を適切に選ぶ．

手順 1：$f(x) = \min\{f(y) \mid y \in N(x)\}$ ならば x を出力して終了．

手順 2：$f(y_*) = \min\{f(y) \mid y \in N(x)\}$ なる $y_* \in N(x)$ を選び，$x := y_*$ とおき，手順 1 に戻る．

貪欲アプローチでは，各反復において関数値が減少することから，実効定義域 $\operatorname{dom} f$ が有界ならばアルゴリズムは有限回の反復で終了する．重要なポイントは，

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

近傍 $N(x)$ の選び方である．出力される解は近傍 $N(x)$ に関する局所的最小解で，大域的に最小とは限らないが，離散凸関数のクラスによっては，近傍を適切に選ぶことにより「局所的最小＝大域的最小」を保証できる場合もある．また，近傍の大きさによって，近傍内での局所的最小解を求めるための計算時間とアルゴリズムの反復回数が大きく変わりうる．

3.1 一変数離散凸関数の場合

一変数の離散凸関数 $f:\mathbb{Z}\to\overline{\mathbb{R}}$ は

$$2f(x)\le f(x-1)+f(x+1)\qquad(\forall x\in\mathbb{Z})$$

という性質によって定義される．一変数離散凸関数 $f:\mathbb{Z}\to\overline{\mathbb{R}}$ の場合，大域的な最小性は局所的な条件により特徴づけられる．

命題 3.1 $x\in\mathbb{Z}$ は f の最小解 $\iff$ $f(x)\le\min\{f(x-1),f(x+1)\}$. ■

したがって，解 x の近傍は

$$N(x)=\{x-1,x,x+1\}$$

と定めるのが自然であろう．一変数離散凸関数に対する貪欲アプローチでは，条件

$$f(x)\le\min\{f(x-1),f(x+1)\}$$

が成り立てば終了し，現在の解を最小解として出力し，そうでなければ，解 $x-1$ または $x+1$ のうち，関数値の小さいほうに移動する，という手順を繰り返す．

近傍は 3 個の点からなるので，各反復では f の関数値を定数回評価するとともに，そのほかの基本的な演算（四則演算，比較，代入など）を定数回行うことになる．また，最小解が見つかるまでの間，各反復において x の値が単調に増加，または単調に減少，のどちらかになることが証明できるので，初期解を x_0，出力される最小解を x_* とおくと，反復回数は $|x_0-x_*|$ である．

3.2 多変数離散凸関数の場合

貪欲アプローチは，最小化する関数が多変数関数になっても，自然に拡張が可能である．しかし，多変数関数の場合は近傍の選び方に自由度があり，近傍をどう設定するかによって，得られる解の良さと計算時間に大きな違いが出てくる．

多変数関数の場合の近傍としては，L_∞ 距離が 1 以下のベクトルの集合

$$N_\infty(x)=\{y\in\mathbb{Z}^n\mid\|y-x\|_\infty\le 1\}$$

が（一つの候補として）考えられる．実際，ある種の離散凸関数においては，この近傍に関して「局所的最小＝大域的最小」を保証できることが示されている．一方で，この近傍に含まれるベクトルは指数個（3^n 個）となるため，近傍内で関数値最小のベクトルを求めることが一般には難しくなる．

近傍の別の候補としては，L_1 距離が 1 以下のベクトルの集合

$$N_1(x)=\{y\in\mathbb{Z}^n\mid\|y-x\|_1\le 1\}$$

が考えられる．この近傍は

$$N_1(x)=\{x\}\cup\{x+e_i\mid 1\le i\le n\}\cup\{x-e_i\mid 1\le i\le n\}$$

とも書けるので，そこに含まれるベクトルの数は $2n+1$ 個となり，近傍内で関数値最小のベクトルを求めることは容易である．一方で，近傍が小さいことから，局所的最小解が大域的最小解になるとは限らない．また，1 回の反復で変更されるベクトルの成分はただ一つなので，アルゴリズムの反復回数が大きくなる可能性が高い．

このように，近傍の設定方法によってアルゴリズムの性能は大きく変わる．扱う離散凸関数のクラスに応じて適切な近傍設定を行う必要がある．

4. 離散凸関数の貪欲アルゴリズム

4.1 M 凸関数に対する貪欲アルゴリズム

M 凸関数 $f:\mathbb{Z}^n\to\overline{\mathbb{R}}$ の最小化問題 (Mmin) に対し，第 3 節の貪欲アプローチを適用する．M 凸関数の最小解は次のような局所的な性質で特徴づけられる．以下では $N=\{1,2,\dots,n\}$ とおく．

定理 4.1 $x\in\operatorname{dom}f$ は f の最小解 $\iff$ 任意の $i,j\in N$ に対して $f(x)\le f(x+e_i-e_j)$. ■

したがって，M 凸関数最小化においては，ベクトル x の近傍 $N(x)$ を

$$N(x)=\{x+e_i-e_j\mid i,j\in N\}$$

とおくのが自然である．この近傍に含まれる（x 以外の）ベクトルの個数は $n(n-1)$ なので，近傍内での最小化は容易である．アルゴリズムは次のようになる．

M 凸関数最小化の貪欲アルゴリズム

手順 0：初期解 $x\in\operatorname{dom}f$ を選ぶ．

手順 1：任意の $i,j\in N$ に対して $f(x)\le f(x+e_i-e_j)$ ならば，x を出力して終了．

手順 2：$f(x+e_{i_*}-e_{j_*})$ を最小にする $i_*,j_*\in N$ を選び，$x:=x+e_{i_*}-e_{j_*}$ とおき，手順 1 に戻る．

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

定理 4.1 より，このアルゴリズムは M 凸関数 f の最小解を出力する．関数値は単調に減少するので，有限回の反復で終了するが，さらに反復回数を（関数値ではなくて）定義域の大きさで評価することができる．その証明には「最小解を含むベクトル集合から（最小解でない）x をカットできる」という M 凸関数特有の性質が有用である．まず，この性質を示そう．

定理 4.2 $x \in \mathrm{dom}\, f$ は f の最小解でないとする．$i_*, j_* \in N$ が

$$f(x + e_{i_*} - e_{j_*}) = \min\{f(x + e_i - e_j) \mid i, j \in N\}$$

を満たすとき，

$$x_*(i_*) \geq x(i_*) + 1, \quad x_*(j_*) \leq x(j_*) - 1$$

を満たす f の最小解 x_* が存在する． ■

関数 f の最小解が一意に定まるとき，貪欲アルゴリズムの反復回数は初期解 x_0 と出力された最小解 x_* の L_1 距離の半分の値 $\|x_* - x_0\|_1/2$ に等しいことが，定理 4.2 よりわかる．

一方，f の最小解が複数存在する場合にも，貪欲アルゴリズムを少し修正することで，反復回数を $\|x_* - x_0\|_1/2$ 以下に抑えることができる．一つの方法は，元の関数 f の代わりに

$$f_\varepsilon(x) = f(x) + \sum_{i=1}^{n} \varepsilon^i (x(i) - x_0(i))^2$$

（ε は十分に小さい正の実数）を最小化するというものである．置き換えた関数 f_ε もまた M 凸関数であるが，その最小解 x_* は唯一に定まり，かつ x_* は f の最小解である．

上記の貪欲アルゴリズムでは，各反復において $n(n-1)$ 個のベクトルを調べている．実は，各反復で n 個のベクトルを調べるだけでも，$\|x_* - x_0\|_1$ 以下の反復回数で最小解が得られる．これを示すためには，定理 4.2 より強い形の定理が必要である．

定理 4.3 $x \in \mathrm{dom}\, f$ は f の最小解でないとする．任意に選んだ $j \in N$ に対し，$i_* \in N$ が

$$f(x + e_{i_*} - e_j) = \min\{f(x + e_i - e_j) \mid i \in N\}$$

を満たすとき，

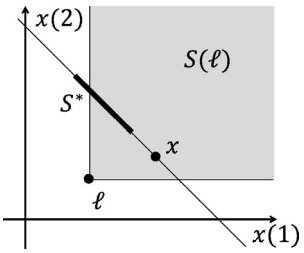

図 4 修正版貪欲アルゴリズムの説明（f が 2 変数関数の場合）

$$x_*(i_*) \geq \begin{cases} x(i_*) + 1 & (i_* \neq j \text{ のとき}), \\ x(i_*) & (i_* = j \text{ のとき}) \end{cases}$$

を満たす f の最小解 x_* が存在する． ■

関数 f の最小解集合を S_* とおく．以下に示すアルゴリズムでは，ある最小解の下界を与える整数ベクトル ℓ を用いる．つまり，

$$S(\ell) = \{y \in \mathrm{dom}\, f \mid y \geq \ell\}$$

とおいたとき，

$$S_* \cap S(\ell) \neq \emptyset \tag{3}$$

が常に成り立つようにする（図 4 参照）．アルゴリズムの各反復では，条件 (3) を満たしつつ，定理 4.3 を使って最小解の下界 ℓ を改良する．なお，各反復の x は常に領域 $S(\ell)$ に含まれる．M 凸関数の定義域に含まれるベクトルの成分和は常に一定の値をとるので，領域 $S(\ell)$ は常に有界である．

M 凸関数最小化の修正版貪欲アルゴリズム 1

手順 0：初期解 $x \in \mathrm{dom}\, f$ を選ぶ．
　　$\ell(i) := \min\{y(i) \mid y \in \mathrm{dom}\, f\}$ $(i \in N)$ とおく．
手順 1：$x = \ell$ ならば x を出力して終了．
手順 2：$x(j) > \ell(j)$ を満たす $j \in N$ を任意に選ぶ．
手順 3：$f(x + e_{i_*} - e_j)$ を最小にする $i_* \in N$ を選ぶ．
手順 4：$i_* \neq j$ ならば $\ell(i_*) := x(i_*) + 1$，$i_* = j$ ならば $\ell(i_*) := x(i_*)$ とおき，$x := x + e_{i_*} - e_j$ として，手順 1 に戻る．

各反復において，f を領域 $S(\ell)$ 上に制限した関数は M 凸関数であるので，定理 4.3 が適用可能である．このことから，各反復において領域 $S(\ell)$ が f の最小解を含むことが示される．とくに，アルゴリズム終了時には $x = \ell$ となるが，$S(\ell) = \{x\}$ が成り立つので，条

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

件 (3) より出力 x は最小解となる．

なお，$g(x) = f(-x)$ $(x \in \mathbb{Z}^n)$ で定義される M 凸関数 g に修正版貪欲アルゴリズムを適用することにより，次の（f に関する）最小化アルゴリズムを得る．上の貪欲アルゴリズムでは減らす方向 j を任意に選ぶのに対し，下のアルゴリズムでは増やす方向 i を任意に選ぶことに注意されたい．

M 凸関数最小化の修正版貪欲アルゴリズム 2

手順 0：初期解 $x \in \operatorname{dom} f$ を選ぶ．$u(i) := \max\{y(i) \mid y \in \operatorname{dom} f\}$ $(i \in N)$ とおく．

手順 1：$x = u$ ならば x を出力して終了．

手順 2：$x(i) < u(i)$ を満たす $i \in N$ を任意に選ぶ．

手順 3：$f(x + e_i - e_{j_*})$ を最小にする $j_* \in N$ を選ぶ．

手順 4：$j_* \neq i$ ならば $u(j_*) := x(j_*) - 1$，$j_* = i$ ならば $u(j_*) := x(j_*)$ とおき，$x := x + e_i - e_{j_*}$ として，手順 1 に戻る．

この貪欲アルゴリズムと最小木問題のアルゴリズムの関係は第 5 節で示される．

4.2 M$^\natural$ 凸関数の成分和制約付き最小化に対する逐次追加型の貪欲アルゴリズム

次に，M$^\natural$ 凸関数の成分和制約付き最小化問題 (M$^\natural$min(r)) に対する貪欲アルゴリズムを説明する．ここでは，原点 $\mathbf{0}$ が $\operatorname{dom} f$ に含まれ，かつ $r > 0$ と仮定する．問題 (M$^\natural$min(r)) は式 (2) で定義される M$^\natural$ 凸関数の最小化問題と等価であるという事実より，修正版貪欲アルゴリズム 1（の M$^\natural$ 凸関数版）が適用可能である．問題 (M$^\natural$min(r)) の特殊性を用いて，このアルゴリズムを簡略化すると，次のような逐次追加型の貪欲アルゴリズムが得られる．これと最小木問題や資源配分問題のアルゴリズムとの関係は第 5 節で示される．

手順 0：初期解を $x := \mathbf{0}$ とする．

手順 1：$\sum_{i=1}^n x(i) = r$ ならば x を出力して終了．

手順 2：$f(x + e_{i_*})$ を最小にする $i_* \in N$ を選び，$x := x + e_{i_*}$ とおき，手順 1 に戻る．

4.3 L 凸関数に対する貪欲アルゴリズム

問題 (L$^\natural$min) に対し，第 3 節の貪欲アプローチを適用する．L$^\natural$ 凸関数の最小解は以下のように特徴づけられる．

定理 4.4 $p \in \operatorname{dom} g$ は g の最小解 $\iff$ 任意の $X \subseteq N$ に対して $g(p) \leq \min\{g(p + e_X), g(p - e_X)\}$. ■

したがって，L$^\natural$ 凸関数最小化に対する貪欲アプローチにおいては，ベクトル p の近傍 $N(p)$ を

$$N(p) = \{p + e_X \mid X \subseteq N\} \cup \{p - e_X \mid X \subseteq N\}$$

とおくのが自然である．この近傍には指数個のベクトルが含まれるが，局所最小解の計算は

$$\rho_p^+(X) = g(p + e_X) - g(p) \quad (X \subseteq N),$$
$$\rho_p^-(X) = g(p - e_X) - g(p) \quad (X \subseteq N)$$

により定義される劣モジュラ集合関数 ρ_p^+, ρ_p^- の最小化に帰着できるので，多項式時間で実行可能である．アルゴリズムは次のようになる．

L$^\natural$ 凸関数最小化の貪欲アルゴリズム

手順 0：初期解 $p_0 \in \operatorname{dom} g$ を選び，$p := p_0$ とおく．

手順 1：任意の $X \subseteq N$ および $\varepsilon \in \{+1, -1\}$ に対して $g(p) \leq g(p + \varepsilon e_X)$ ならば p を出力して終了．

手順 2：$g(p + \varepsilon_* e_{X_*})$ を最小にする $X_* \subseteq N$ と $\varepsilon_* \in \{+1, -1\}$ を選び，$p := p + \varepsilon_* e_{X_*}$ として手順 1 に戻る．

定理 4.4 より，このアルゴリズムは L$^\natural$ 凸関数 g の最小解を出力する．関数値は単調に減少するので，有限回の反復で終了するが，さらに反復回数を（関数値ではなくて）定義域の大きさで評価することができる．実際，アルゴリズムにより出力される最小解を p_* とおくと，反復回数が高々$2\|p_* - p_0\|_\infty$ であることを示せる．

なお，関数 g が L 凸関数の場合には，貪欲アルゴリズムの手順 1 において常に $\varepsilon = +1$ とすることができて，ベクトル p の各成分は単調非減少である．

5. 組合せ最適化問題への適用

本節では，第 2 節で説明した組合せ最適化問題に対して M 凸関数と L 凸関数の貪欲アルゴリズムを特殊化すると，有名なアルゴリズムが得られることを示す．

例 5.1 (最小木問題) 最小木問題を M 凸関数 f_{MST1} の最小化と見なして（例 2.1 参照），修正版貪欲アルゴリズム 1 を適用すると，次のアルゴリズムが得られる．

手順 0：適当に見つけた全域木を T とおく．残りの枝 $E \setminus T$ に適当な順番で番号 $g_1, g_2, \ldots, g_{m-n+1}$ をつける．$k := 1$ とおく．

手順 1：全域木 T と枝 g_k から得られる閉路において最も長い枝 e_k に対し，$T := T - e_k + g_k$ とおく．

手順 2：$k = m - n + 1$ ならば枝集合 T を出力し終了．そうでなければ，$k := k + 1$ として手順 1 に戻る．

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

これはカラバのアルゴリズム (Kalaba, 1960) として知られる解法である．このアルゴリズムの詳細については文献 [5, 第 50 章] を参照されたい．

一方，最小木問題を $\mathrm{M}^\natural$ 凸関数 f_{MST2} の成分和制約付き最小化とみなして（例 2.1 参照），逐次追加型の貪欲アルゴリズムを適用すると，有名なクラスカルのアルゴリズム (Kruskal, 1956) が得られる．

手順 0：枝を長さの短い順に並べ，$E = \{e_1, e_2, \ldots, e_m\}$ とする．$T := \emptyset$，$k := 1$ とおく．

手順 1：$T \cup \{e_k\}$ が閉路を含まなければ $T := T \cup \{e_k\}$ とおく．

手順 2：$k = m$ ならば枝集合 T を出力して終了．$k < m$ ならば，$k := k + 1$ とおき，手順 1 に戻る．■

例 5.2 資源配分問題を $\mathrm{M}^\natural$ 凸関数 f_{RA2} の成分和制約付き最小化とみなして（例 2.2 参照），逐次追加型の貪欲アルゴリズムを適用すると，次のアルゴリズムを得る．これはグロスのアルゴリズム (Gross, 1956) として知られる．詳しくは文献 [2, 第 4 章] を参照されたい．

手順 0：初期解を $x := \mathbf{0}$ とする．

手順 1：$\sum_{i=1}^{n} x(i) = r$ ならば x を出力して終了．

手順 2：$\varphi_{i_*}(x(i_*) + 1) - \varphi_{i_*}(x(i_*))$ を最小にする $i_* \in N$ を選び，$x := x + e_{i_*}$ とおき，手順 1 に戻る．■

例 5.3 (最短路問題) 例 2.3 の最短路問題を $\mathrm{L}^\natural$ 凹関数 g_{SP} の最大化とみなして，関数 $-g_{\mathrm{SP}}$ に対して貪欲アルゴリズムを適用すると，次のアルゴリズムを得る．

手順 0：初期ベクトルを $p := \mathbf{0}$ とおく．

手順 1：任意の非空な $X \subseteq V$ に対して $p + e_X \notin S$ ならば，現在のベクトル p を出力し，終了する．

手順 2：$p + e_X \in S$ を満たす要素数最大の $X \subseteq V$ を求める．$\lambda := \max\{\lambda' \mid p + \lambda' e_X \in S\}$ として，p を $p := p + \lambda e_X$ により更新し，手順 1 に戻る．

このアルゴリズムでは手順 2 においてステップ方向 e_X とステップサイズ λ の計算が必要となるが，ここで補助変数を使うと計算が容易になる．補助変数を用いてアルゴリズムを適切に書き換えると，ダイクストラのアルゴリズム (Dijkstra, 1959) に一致することが示せる．詳細については [3] を参照されたい．■

例 5.4 (最小費用流問題の双対) 最小費用流問題の双対問題を L 凹関数 g_{D} の最大化とみなして（例 2.4 参照），関数 $-g_{\mathrm{D}}$ に対して貪欲アルゴリズムを適用すると，次のアルゴリズムを得る．ここで，ベクトル $p \in \mathbb{Z}^V$ と $S \subseteq V$ に対し，$I(p, S) = g(p + e_S) - g(p)$ とおく．$I(p, S)$ は次のように書き換えられる：

$$I(p,S) = \sum_{v \in S} b(v) + \sum_{(u,v) \in E_1} c(u,v) - \sum_{(u,v) \in E_2} c(u,v),$$

$$E_1 = \{(u,v) \in E \mid p(u) - p(v) > k(u,v), \ u \in V \setminus S, v \in S\},$$

$$E_2 = \{(u,v) \in E \mid p(u) - p(v) \geq k(u,v), \ u \in S, v \in V \setminus S\}.$$

手順 0：初期ベクトルを $p := \mathbf{0}$ とおく．

手順 1：$I(p, S) \leq 0$ $(\forall S \subseteq V)$ ならば，現在のベクトル p を出力し，終了する．

手順 2：$I(p, S)$ を最大にする $S \subseteq V$ を求め，p を $p := p + e_S$ により更新し，手順 1 に戻る．

これはハッシン (Hassin) のアルゴリズム [1] と本質的に同じアルゴリズムである．■

6. おわりに

本稿では，M 凸関数と L 凸関数の最小化問題に対する貪欲アルゴリズムを説明した．これ以外の最小化アルゴリズムについては，本特集の土村氏の記事にも簡単な説明がある．より詳しくは，文献 [4] を参照されたい．また，幾つかの組合せ最適化問題と離散凸解析の関係についても説明したが，本稿で扱っていないマッチング問題やネットワークフロー問題については文献 [4] で詳しく議論されているので，こちらも参照されたい．

参考文献

[1] R. Hassin, The minimum cost flow problem: a unifying approach to dual algorithms and a new tree-search algorithm, *Mathematical Programming*, **25**, 228–239, 1983.

[2] T. Ibaraki, and N. Katoh, *Resource Allocation Problems: Algorithmic Approaches*, MIT Press, 1988.

[3] K. Murota, and A. Shioura, Dijkstra's algorithm and L-concave function maximization, *Mathematical Programming*, to appear.

[4] 室田一雄，塩浦昭義，離散凸解析と最適化アルゴリズム，朝倉書店，2013.

[5] A. Schrijver, *Combinatorial Optimization—Polyhedra and Efficiency*, Springer, 2003.

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.