M凸関数最小化に対する高速スケーリング算法
Fast Scaling Algorithms for M-convex Function Minimization

平成14年3月23日
統計数理研究所 研究集会「最適化：モデリングとアルゴリズム」

塩浦 昭義 （東北大学大学院 情報科学研究科）

発表の流れ

- 研究の背景
- M凸関数の定義
- M凸関数最小化に対する貪欲算法
- M凸関数のスケーリング
- 提案するスケーリング算法1
- 提案するスケーリング算法2
- 資源配分問題への応用

○ ● ○ ● 研究の背景 (その1) ● ○ ● ○

(非線形) 組合せ最適化問題の解きやすさ

$\Longrightarrow$ 離散凸関数の研究 ── $f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ に対する凸性

M凸関数 (室田96) ── 離散凸関数としてふさわしい性質を数多く満たす

- 局所的に最小 = 大域的に最小
- 普通の凸関数への拡張可能性
- Fenchel型双対定理, 離散分離定理

M凸関数最小化 ── もっとも基本的な問題のひとつ

Minimize $f(x)$ subject to $x \in \mathbf{Z}^n$

○ ● ○ ●　研究の背景 (その2)　● ○ ● ○

既存の算法　(n: 次元, L: 実行可能領域の大きさ)

- 「局所最小 = 大域的最小」$\Longrightarrow$ 貪欲算法 (降下法) が適用可, しかし擬多項式 $O(n^2L)$

- 塩浦98 — 任意の解と最小解は「簡単」に分離可 (minimizer cut prop.)
  $\Longrightarrow$ 最初の多項式時間算法 $O(n^4(\log L)^2)$

- 森口–室田–塩浦02 — 近似最小解は最小解の近くに存在 (proximity thm)
  $\Longrightarrow$ 貪欲算法 + スケーリング, あるクラスのM凸関数に対して $O(n^3 \log_2 \frac{L}{n})$

- 田村02 — スケーリング版 minimizer cut prop.
  $\Longrightarrow$ 森口–室田–塩浦02 の精緻化, 巧妙な解析により $O(n^3 \log_2 \frac{L}{n})$

提案する算法

- 森口–室田–塩浦02 を改良, 解析が簡単
  算法1 — スケーリング版 minimizer cut prop. を利用, $O(n^3 \log_2 \frac{L}{n})$
  算法2 — minimizer cut prop. を利用
  $O((n^3 + n^2 \log_2 \frac{L}{n}) \log_n \frac{L}{n})$　($< O(n^3 \log_2 \frac{L}{n})$ if $L \leq n^n$)

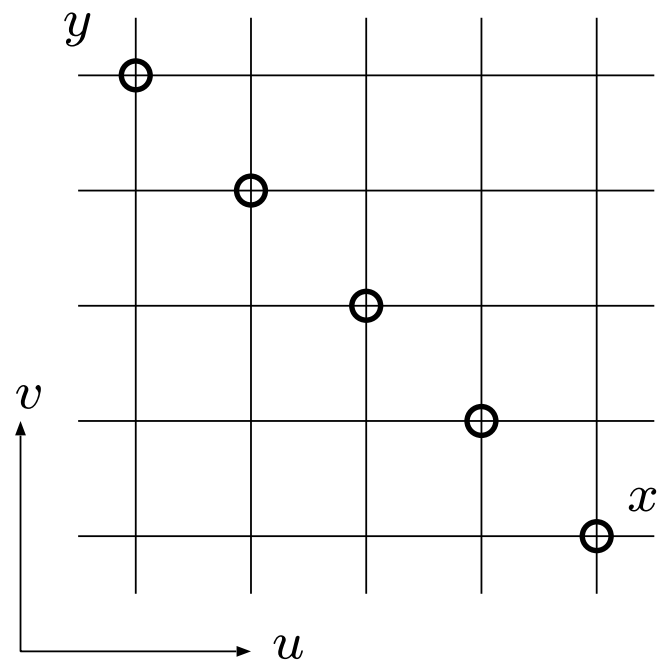
○ ● ○ ● M凸関数の定義 (その1) ● ○ ● ○

$f : \mathbf{Z}^n \to \mathbf{Z} \cup \{+\infty\}$, 定義域 $\mathrm{dom}\, f = \{x \in \mathbf{Z}^n \mid f(x) < +\infty\}$

f はM凸関数 $\overset{\text{def}}{\Longleftrightarrow}$ (M-EXC) を満たす

(M-EXC) $\forall x, y \in \mathrm{dom}\, f$, $\forall i \in \mathrm{supp}^+(x-y)$, $\exists j \in \mathrm{supp}^-(x-y)$:

$$f(x) + f(y) \geq f(x-\chi_i+\chi_j) + f(y+\chi_i-\chi_j)$$



$\mathrm{supp}^+(x-y) = \{i \mid x_i > y_i\}$
$\mathrm{supp}^-(x-y) = \{i \mid x_i < y_i\}$

$$\chi_i = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \leftarrow \text{第 } i \text{ 成分}$$

M凸関数の定義 (その2)

M凸関数の例：資源配分問題

$$f(x) = \begin{cases} \sum_{i=1}^{n} f_i(x_i) & \text{if } \sum_{i=1}^{n} x_i = B,\ 0 \le x_i \le U_i, \\ +\infty & \text{otherwise.} \end{cases}$$

(各 $f_i : \mathbf{Z} \to \mathbf{R} \cup \{+\infty\}$ は凸関数)

性質

f がM凸 $\Longrightarrow \exists B \in \mathbf{Z}$ s.t. $\sum_{i=1}^{n} x_i = B$ $(\forall x \in \mathrm{dom}\, f)$

M凸関数最小化に対する貪欲算法 (その1)

$f : \mathbf{Z}^n \to \mathbf{Z} \cup \{+\infty\}$ はM凸関数, $N = \{1, 2, \ldots, n\}$, arg min f : 最小解集合

定理 [局所最小 = 大域的最小 (室田96)]

$$f(x) \le f(x+\chi_i-\chi_j)\ (\forall i,j \in N) \quad \Longleftrightarrow \quad f(x) \le f(y)\ (\forall y \in \mathbf{Z}^n)$$

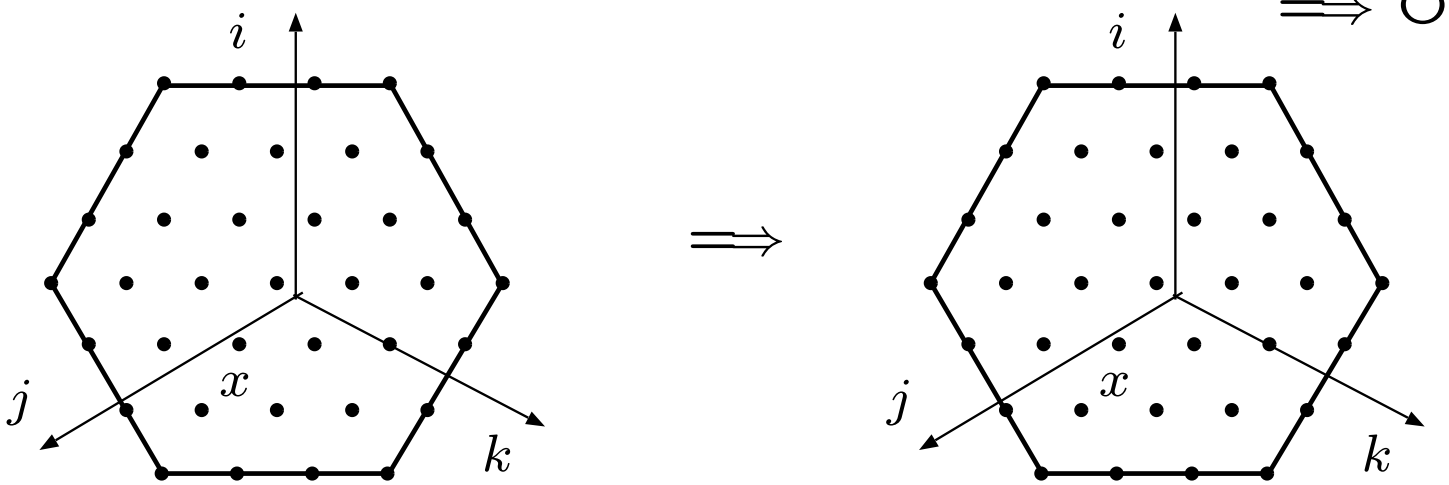
$\Longrightarrow O(n^2)$ で最小解のチェック

定理 [minimizer cut property(塩浦98)]

$x \in \mathrm{dom}\, f$, $j \in N$, $i \in N : f(x+\chi_i-\chi_j) = \min\{f(x+\chi_k-\chi_j) \mid k \in N\}$

$$\Longrightarrow \exists x^* \in \arg\min f \text{ with } x_i^* \ge \begin{cases} x_i + 1 & (i \ne j), \\ x_i & (i = j). \end{cases}$$

$\Longrightarrow O(n)$ で探索領域をカット



アルゴリズム

最小解を含む領域 $S_l \equiv \{y \mid \sum_i y_i = B,\ y_i \geq l_i\ (\forall i)\}$ を $l \in \mathbf{Z}^n$ で表現

minimizer cut prop. により繰り返し S_l を縮小

現在の点 $x \in \mathrm{dom}\, f$ は常に S_l の中, $x = l$ のとき x は最小解

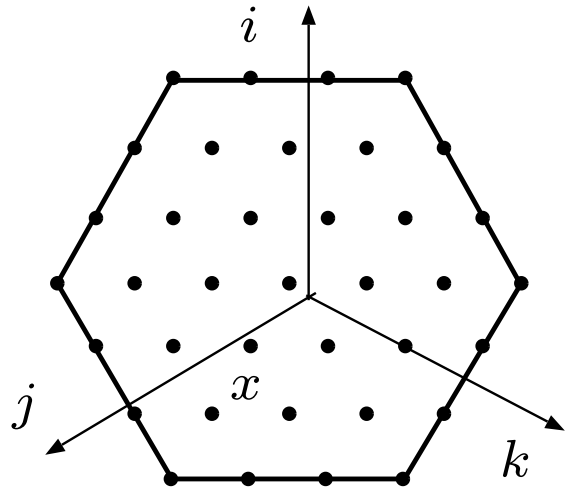
手順**0:** x, l の初期化

手順**1:** $x = l$ ならば終了

手順**2:** $x_j > l_j$ なる $j \in N$ を選ぶ

手順**3:** $f(x + \chi_i - \chi_j) = \min_k f(x + \chi_k - \chi_j)$
を満たす $i \in N$ を求める

手順**4:** $l_i := \begin{cases} x_i + 1 & (i \neq j), \\ x_i & (i = j), \end{cases}$ $x := x + \chi_i - \chi_j$.



$\sum_{i=1}^{n}(x_i - l_i)$ は単調減少, $0 \leq \sum_{i=1}^{n}(x_i - l_i) \leq nL$ $\left(L = \max_{x,y \in \mathrm{dom}\, f} ||x - y||_\infty\right)$

$\Longrightarrow$ 反復回数 $nL \times O(n) = O(n^2 L)$ (擬多項式時間)

○ ● ○ ●　M凸関数のスケーリング (その1)　● ○ ● ○

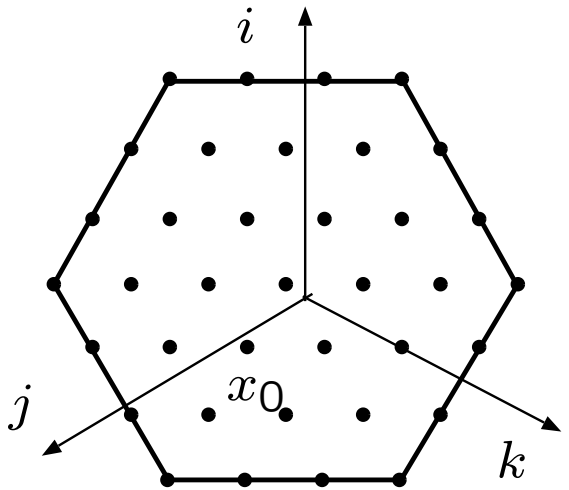
$f : \mathbf{Z}^n \to \mathbf{Z} \cup \{+\infty\}$ はM凸, $\alpha > 0$, $x_0 \in \mathrm{dom}\, f$

$f_\alpha : \mathbf{Z}^n \to \mathbf{Z} \cup \{+\infty\}$ を　$f_\alpha(y) = f(x_0 + \alpha y)$ $(y \in \mathbf{Z}^n)$　により定義
(f のスケーリング)

f_α の最小解の近くに f の最小解が存在

定理 [proximity thm(森口–室田–塩浦02)]
y は f_α の最小解, $x \equiv x_0 + \alpha y$
$\Longrightarrow \exists x^* \in \arg\min f$ with $|x_i^* - x_i| < n\alpha$ $(\forall i \in N)$

$\therefore$ $\forall \alpha > 0$, f_α がM凸
$\Longrightarrow$ 貪欲算法 + スケーリングで多項式算法
(森口–室田–塩浦02)



○ ● ○ ●　M凸関数のスケーリング (その2)　● ○ ● ○

一般に f_α はM凸ではない $\longrightarrow$ f_α 最小化に貪欲算法は適用不可

しかし, minimizer cut property の緩和版が成立

定理 [スケーリング版 minimizer cut prop.(田村02)]
$x \in \mathrm{dom}\, f$, $j \in N$, $\alpha > 0$
$i \in N : f(x + \alpha(\chi_i - \chi_j)) = \min\{f(x + \alpha(\chi_k - \chi_j)) \mid k \in N\}$

$$\Longrightarrow \exists x^* \in \arg\min f \text{ with } x_i^* \geq \begin{cases} x_i + \alpha - (n-1)(\alpha - 1) & (i \neq j), \\ x_i - (n-1)(\alpha - 1) & (i = j). \end{cases}$$

※ $\alpha = 1$ のとき, minimizer cut prop. に一致

f_α に貪欲算法を適用 $\longrightarrow$ f の最小解を含む領域を縮小可能

○ ● ○ ●　提案するスケーリング算法1 (その1)　● ○ ● ○

$S_l \equiv \{y \mid \sum_i y_i = B,\ y_i \geq l_i\ (\forall i)\}$ — f の最小解を含む領域
$x \in S_l$ に対し「$x = l \Rightarrow x \in \arg\min f$」$\longrightarrow$ x と l の「近さ」が近似精度を表現

手続き **ScaledGreedy1**(α, x, l)
— スケーリング版 minimizer cut prop. により l を更新

入力: $\alpha > 0$, $x \in \mathrm{dom}\, f$, $l \in \mathbf{Z}^n$　　s.t.　$x \geq l$, $S_l \cap \arg\min f \neq \emptyset$

出力: $x' \in \mathrm{dom}\, f$, $l' \in \mathbf{Z}^n$
s.t.　$l_j \leq x_j \leq l_j + (n-1)(\alpha - 1)\ (j \in N)$, $S_{l'} \cap \arg\min f \neq \emptyset$

手順**1:** $l_j \leq x_j \leq l_j + (n-1)(\alpha-1)\ (\forall j \in N)$ ならば終了

手順**2:** $x_j > l_j + (n-1)(\alpha-1)$ なる $j \in N$ を選ぶ

手順**3:** $f(x + \alpha(\chi_i - \chi_j)) = \min_k f(x + \alpha(\chi_k - \chi_j))$ を満たす $i \in N$ を求める

手順**4:**
$$l_i := \begin{cases} \max\{l_i, x_i + \alpha - (n-1)(\alpha-1)\} & (i \neq j), \\ \max\{l_i, x_i - (n-1)(\alpha-1)\} & (i = j), \end{cases}$$
$$x := x + \alpha(\chi_i - \chi_j).$$

※ $\alpha = 1$ のとき, 貪欲算法に一致

○ ● ○ ● 提案するスケーリング算法1 (その2) ● ○ ● ○

ScaledGreedy1(α, x, l) の計算時間

$\sum_{i=1}^{n}(x_i - l_i)$ は非負, 各反復でα 以上減る $\Longrightarrow$ 反復回数 $\sum_{i=1}^{n}\frac{x_i - l_i}{\alpha} \times O(n)$

アルゴリズム Scaling1

手順0: $x \in \mathrm{dom}\, f$, l: 自明な下界, $\alpha := 2^{\lceil \log_2(L/2n) \rceil}$.

手順1: $\alpha < 1$ ならば終了. x は最小解.

手順2: 手続き ScaledGreedy1(α, x, l) により x', l' を得る.

手順3: $x := x'$, $l := l'$, $\alpha := \alpha/2$.

手順2で $x'_i - l'_i \leq (n-1)(\alpha - 1)$ $(i \in N)$ 成立

- $\alpha = 1 \longrightarrow x' = l' \longrightarrow x' \in \arg\min f$
- 各反復の最初で $x_i - l_i < 2n\alpha$ $(i \in N)$ 成立

  $\longrightarrow$ 一反復 $O(n^3)$ × $O(\log_2 \frac{L}{n})$ 回反復 = $O(n^3 \log_2 \frac{L}{n})$

提案するスケーリング算法２ (その1)

２つの算法の異なる点

- l の更新方法:
  - 算法１ — スケーリング版minimizer cut prop. を利用
  - 算法２ — minimizer cut prop. を利用

- $x + \alpha(\chi_i - \chi_j) \notin \mathrm{dom}\, f$ のとき, x に対し:
  - 算法１ — $\chi_i - \chi_j$ 方向への移動を許さない
  - 算法２ — $x + \beta(\chi_i - \chi_j)$ への移動を許す
    $(\beta = \max\{\alpha' \mid x + \alpha'(\chi_i - \chi_j) \in \mathrm{dom}\, f\})$

- 手続きの出力:
  - 算法１ — $l'_j \leq x'_j \leq l'_j + (n-1)(\alpha - 1)$
  - 算法２ — $l'_j \leq x'_j \leq l'_j + (\alpha - 1)$

○ ● ○ ● 提案するスケーリング算法２ (その2) ● ○ ● ○

手続き **ScaledGreedy2**(α, x, l)

入力: $\alpha > 0$, $x \in \mathrm{dom}\, f$, $l \in \mathbf{Z}^n$ s.t. $x \geq l$, $S_l \cap \arg\min f \neq \emptyset$

出力: $x' \in \mathrm{dom}\, f$, $l' \in \mathbf{Z}^n$

s.t. $l_j \leq x_j \leq l_j + (\alpha - 1)$ $(j \in N)$, $S_{l'} \cap \arg\min f \neq \emptyset$

手順1: $l_j \leq x_j \leq l_j + (\alpha - 1)$ $(\forall j \in N)$ ならば終了

手順2: $x_j > l_j + (\alpha - 1)$ なる $j \in N$ を選ぶ

手順3: $f(x + \chi_i - \chi_j) = \min_k f(x + \chi_k - \chi_j)$ を満たす $i \in N$ を求める

手順4:

$$l_i := \begin{cases} x_i + 1 & (i \neq j), \\ x_i & (i = j), \end{cases}$$

$$x := \begin{cases} x + \alpha(\chi_i - \chi_j) & (x + \alpha(\chi_i - \chi_j) \in \mathrm{dom}\, f) \text{ — [full step]} \\ x + \beta(\chi_i - \chi_j) & (x + \alpha(\chi_i - \chi_j) \notin \mathrm{dom}\, f) \text{ — [partial step]} \end{cases}$$

$$(\beta = \max\{\alpha' \mid x + \alpha'(\chi_i - \chi_j) \in \mathrm{dom}\, f\})$$

※ $\alpha = 1$ のとき, 貪欲算法に一致

提案するスケーリング算法２ (その3)

ScaledGreedy2(α, x, l) の計算時間

- full step: ScaledGreedy1(α, x, l) と同じ解析

  １反復の時間 — $O(n)$ 時間,　　反復回数 — $\displaystyle\sum_{i=1}^{n} \frac{x_i - l_i}{\alpha}$ 回

- partial step:
  １反復の時間 — ２分探索を使って $O(n + \log_2 \alpha)$ 時間
  反復回数 — 一般に算定は困難, 手順２ での j の選び方を工夫 $\rightarrow n^2$ 回

アルゴリズム **Scaling2**

手順**0:** $x \in \mathrm{dom}\, f$, l: 自明な下界, $\alpha := 2^{\lceil \log_n (L/n) \rceil}$.

手順**1:** $\alpha < 1$ ならば終了. x は最小解.

手順**2:** 手続き ScaledGreedy2(α, x, l) により x', l' を得る.

手順**3:** $x := x'$, $l := l'$, $\alpha := \alpha/n$.

手順２で $x'_i - l'_i \leq \alpha - 1$ $(i \in N)$ 成立

- $\alpha = 1 \longrightarrow x' = l' \longrightarrow x' \in \arg\min f$
- 各反復の最初で $x_i - l_i < n\alpha$ $(i \in N)$ 成立

  $\longrightarrow$ 一反復 $O(n^3 + n^2 \log_2 \frac{L}{n})$ × $O(\log_n \frac{L}{n})$ 回 $= O((n^3 + n^2 \log_2 \frac{L}{n}) \log_n \frac{L}{n})$

資源配分問題への応用

劣モジュラ制約のもとでの資源配分問題── M凸関数最小化の特殊ケース

given:

凸関数 $f_i : \mathbf{Z} \to \mathbf{R} \cup \{+\infty\}$ $(i \in N)$

正整数 B

劣モジュラ関数 $\rho : 2^N \to \mathbf{Z}$

$\rho(S) + \rho(T) \geq \rho(S \cap T) + \rho(S \cup T)$

$$
\begin{array}{ll}
\text{Minimize} & \sum_{i=1}^{n} f_i(x_i) \\
\text{subject to} & \sum_{i=1}^{n} x_i = B \\
& \sum_{i \in S} x_i \leq \rho(S) \quad (S \subseteq N) \\
& x_i \geq 0 \quad (i \in N)
\end{array}
$$

2つの算法を資源配分問題に特化 (F: x の実行可能性判定の時間)

算法1 $\longrightarrow$ $O(n^2(\log_2 n + F)\log_2 \frac{B}{n})$

── 新たなスケーリングアルゴリズム, きわめてシンプル

算法2 $\longrightarrow$ $O(n(\log_2 n + F\log_2 \frac{B}{n})\log_2 \frac{B}{n})$

── 森口–塩浦02 による Hochbaum のアルゴリズムの修正版(最速)に一致