M凸関数最小化に対する高速スケーリング算法と資源配分問題への応用

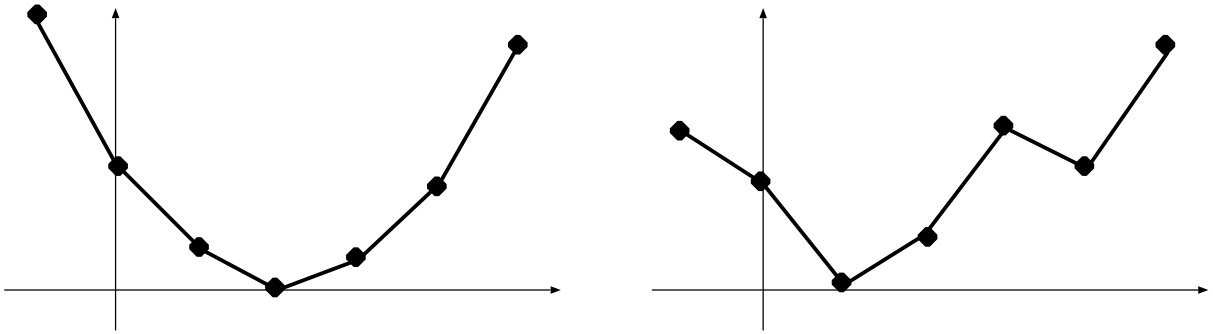
東北大学 大学院 情報科学研究科
塩浦 昭義

発表の概要

- M凸関数最小化と既存の結果
- 本研究の結果
- 資源配分問題に特化したアルゴリズム(2種類)
- M凸関数最小化に対するアルゴリズム(2種類)

(非線形) 組合せ最適化問題の解きやすさ

$\Longrightarrow$ 離散凸関数 — $f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ 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

- 局所最小 = 大域的最小
- 普通の凸関数へ拡張可能
- **Fenchel** 型双対定理, 分離定理

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

○ ● ○ ●　M凸関数最小化の例　● ○ ● ○

- 分離凸資源配分問題

Minimize $\sum_{i=1}^{n} f_i(x_i)$ **subject to** $\sum_{i=1}^{n} x_i = L,\ 0 \le x_i \le U_i, x \in \mathbf{Z}^n$

(各 $f_i : \mathbf{Z} \to \mathbf{R}$ は凸)

- ある種の整数2次計画

Minimize $x^\top A x$ **subject to** $\sum_{i=1}^{n} x_i = L,\ 0 \le x_i \le U_i, x \in \mathbf{Z}^n$

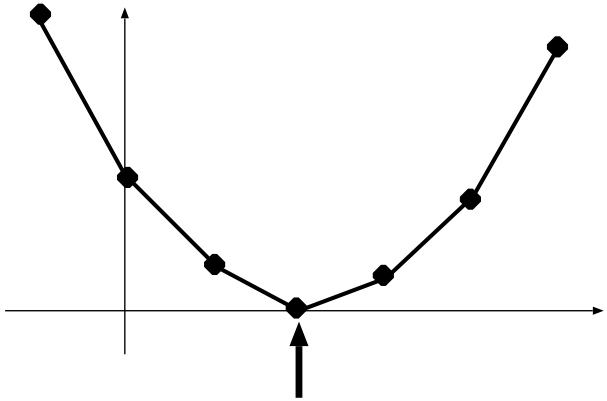
$$A = \begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_n \\ a_2 & a_2 & a_3 & \cdots & a_n \\ a_3 & a_3 & a_3 & \cdots & a_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_n & a_n & a_n & \cdots & a_n \end{bmatrix}, a_1 \ge a_2 \ge a_3 \ge \cdots \ge a_n$$

○ ● ○ ● M凸関数最小化アルゴリズム ● ○ ● ○

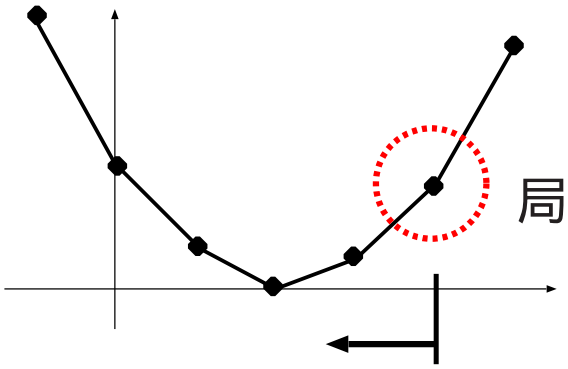
既存の算法	n: 次元, L: 実行可能領域の大きさ
貪欲算法	擬多項式 $O(n^2L)$
塩浦**98**	$O(n^4(\log L)^2)$
森口–室田–塩浦**02**	$O(n^3 \log L)$ **(限定されたクラスのみ)**
田村**02**	$O(n^3 \log L)$

○ ● ○ ● 有用な4つの性質 (1/2) ● ○ ● ○

- 局所最小 = 大域的最小

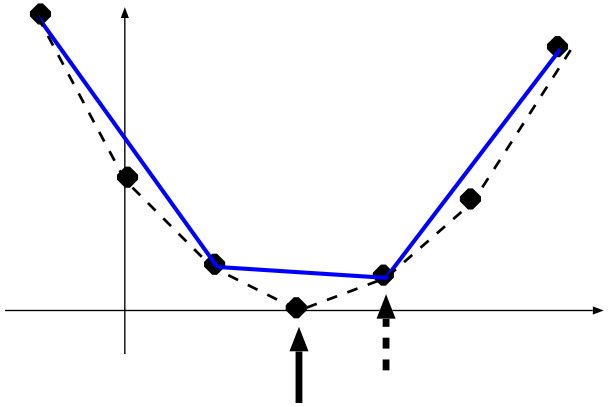


- 最小解カット [塩浦 98]



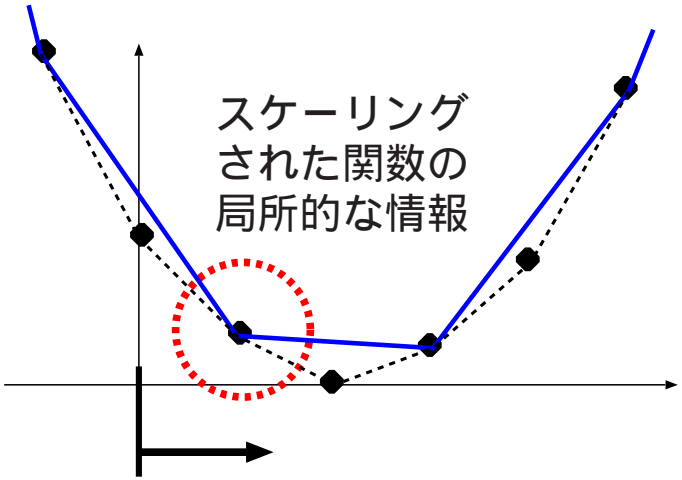
有用な4つの性質 (2/2)

- 近接定理 [森口–室田–塩浦 02]



スケーリングされた関数の最小解
$\simeq$ 本当の最小解

- スケーリング版最小解カット [田村 02]



最小解の存在する範囲

○ ● ○ ●　提案するアルゴリズム　● ○ ● ○

提案する算法	貪欲算法 + スケーリング技法
その**1**	$O((n^3 + n^2 \log L)\frac{\log L}{\log n})$　— 最小解カット — L がそれほど大きくない($L \leq n^n$)ときは最速
その**2**	$O(n^3 \log L)$　— スケーリング版最小解カット — シンプルなスケーリング算法, 田村**02**と同じ計算量

資源配分問題への応用

- 劣モジュラ制約分離凸資源配分問題

既存の最速算法	**Hochbaum 94**	$O(n(\log n + F)\log L)$ ← 誤りあり
		n = ベクトルの次元
		L = 配分する資源量
		F = 実行可能性判定の時間
提案する算法	その**1**	$O(n(\log n + F\log L)\log L)$, 現時点で最速
	その**2**	$O(n^2(\log n + F)\log L)$

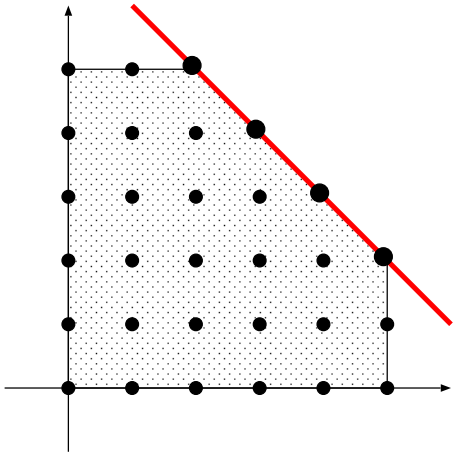
以下, 説明の簡略化のため資源配分問題に特化

○ ● ○ ● 分離凸資源配分問題 (1/2) ● ○ ● ○

- (単純な)分離凸資源配分問題 $(N = \{1, 2, \ldots, n\})$

Minimize $f(x) = \sum_{i=1}^{n} f_i(x_i)$ — 各 $f_i : \mathbf{Z} \to \mathbf{R}$ は凸

subject to $\sum_{i=1}^{n} x_i = L,\ x \in \mathbf{Z}_+^n$



- 劣モジュラ制約　劣モジュラ関数 $\rho : 2^N \to \mathbf{Z}$ に対して

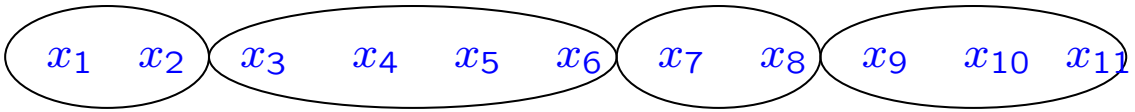
$x(S) \leq \rho(S)\ (\forall S \subseteq N)$　$(x(S) \equiv \sum_{i \in S} x_i)$

$$\rho(X) + \rho(Y) \geq \rho(X \cap Y) + \rho(X \cup Y)\ (\forall X, Y \subseteq N)$$

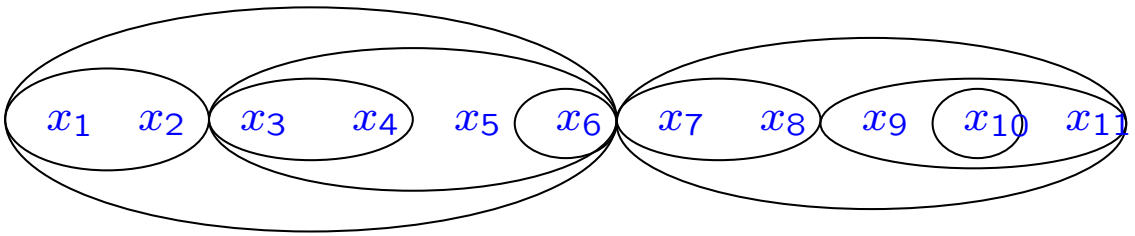
分離凸資源配分問題 (2/2)

劣モジュラ制約の特殊な場合

- 上限制約　$x_i \leq u_i \ (\forall i = 1, \ldots, n)$
- 一般化上限制約　N の分割 $\{S_j\}_{j=1}^k$ に対して　$x(S_j) \leq u_{S_j}$



- ツリー制約　層族 $\{S_j\}_{j=1}^k$ に対して　$x(S_j) \leq u_{S_j}$



貪欲算法

$x \in \mathbf{Z}_+^n$: 実行可能 $\overset{\text{def}}{\Longleftrightarrow}$ 劣モジュラ制約を満たす

手順 **0:** $x = (0, 0, \ldots, 0)$

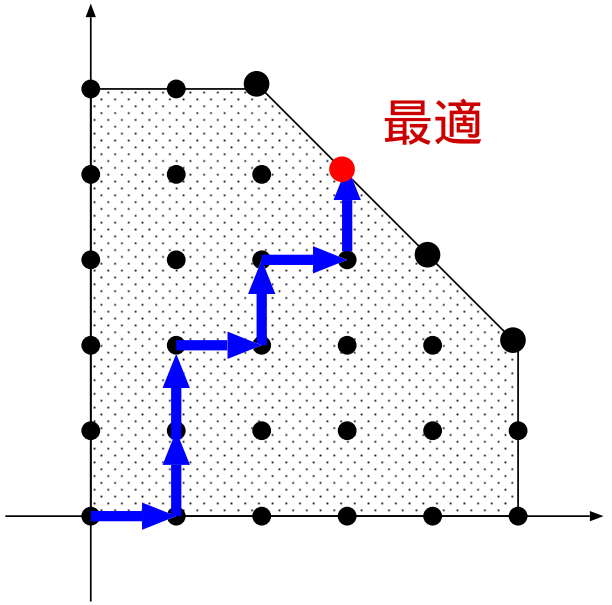
手順 **1:** $\sum_{i=1}^{n} x_i = L$ ならば終了

手順 **2:** 「貪欲」な方向 i を求める:

$f(x + \chi_i) = \min\{f(x + \chi_j) \mid x + \chi_j : \text{実行可能}\}$

手順 **3:** $x := x + \chi_i$, 手順**1**へ

(χ_i: 第 i 単位ベクトル)



定理 [Glebov 73 ?]

貪欲算法は必ず最小解を求める

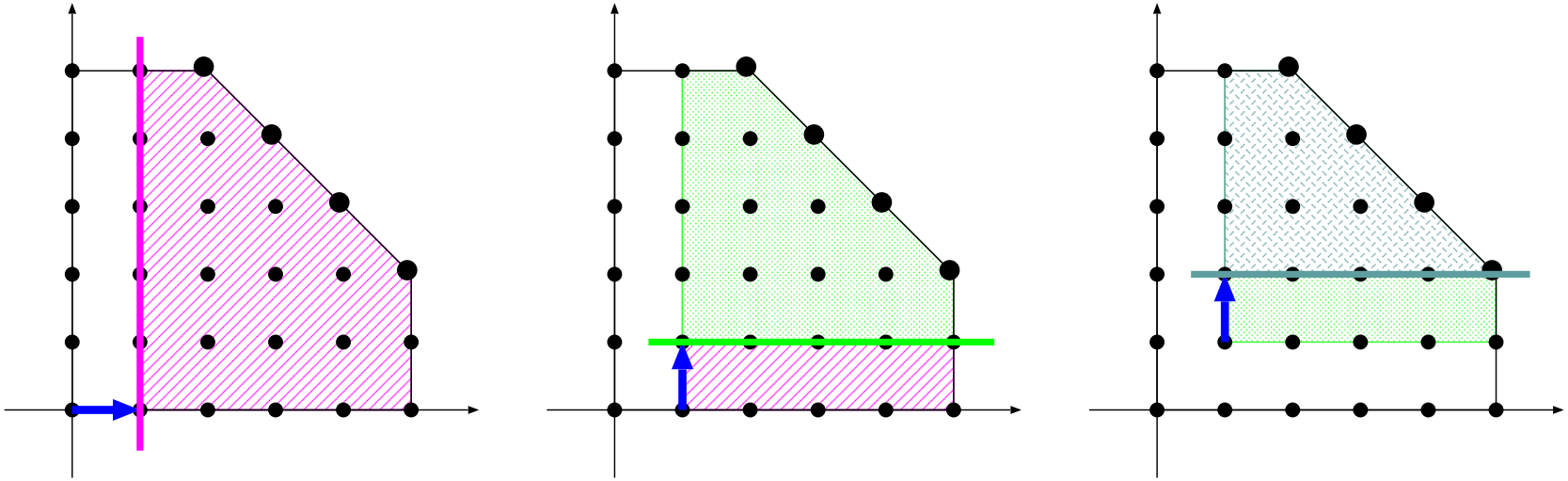
貪欲算法の正当性

定理 (最小解カット) [Girlich–Kovalev–Zaporozhets 96]

x: 実行可能解

i: 貪欲方向　$f(x+\chi_i) = \min\{f(x+\chi_j) \mid x+\chi_j$：実行可能$\}$

$\Longrightarrow \exists x^*$: 最小解 **s.t.** $x^*(i) > x(i)$



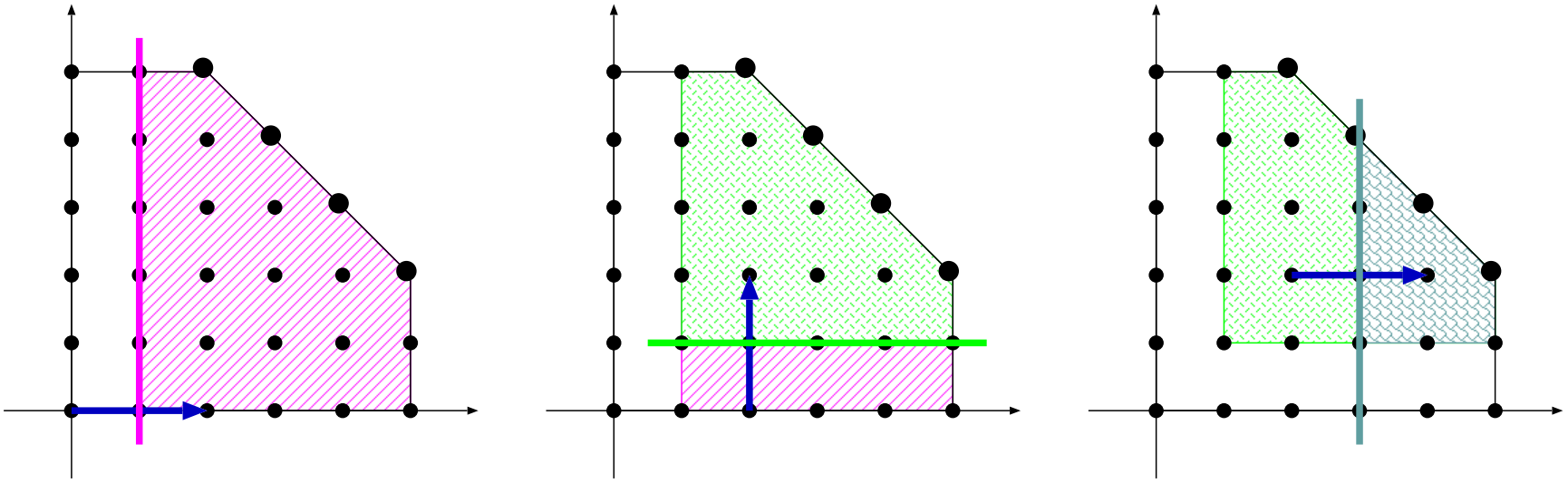
最小解を含む領域を狭める $\Longrightarrow$ 最小解が見つかる

提案するアルゴリズム その1 (1/2)

アイディア: 最小解カット + スケーリング技法

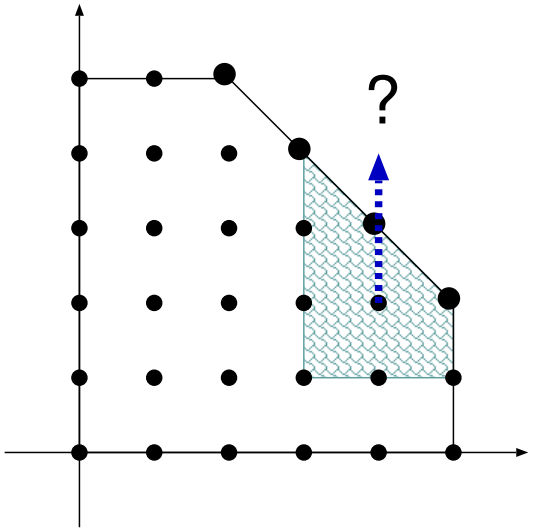
- スケーリングフェイズの各反復:
  貪欲方向 i に α ステップ進む (α：正整数)
  $f(x+\chi_i) = \min\{f(x+\chi_j) \mid x+\chi_j$：実行可能$\}$
  $\Longrightarrow x := x + \alpha\chi_i$

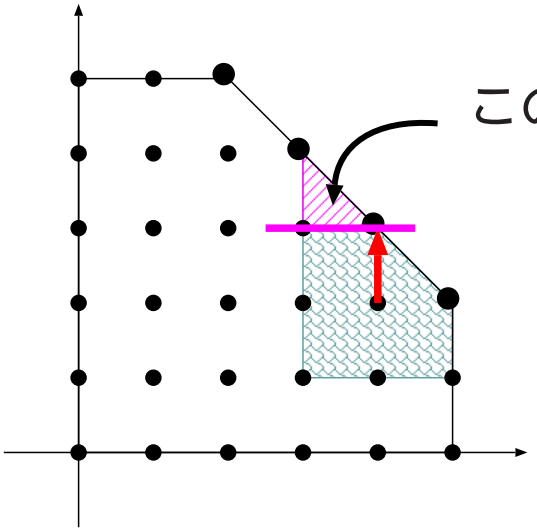
$\alpha = 2$ の場合:



○ ● ○ ● 提案するアルゴリズム その1 (2/2) ● ○ ● ○

- α ステップ進めない
  $\Longrightarrow$ 出来る限り進む





定理 x^α**:** スケーリングフェイズで得られた解
$\Longrightarrow \exists x^*$**:** 最小解 **s.t.** $x^\alpha(i) - x^*(i) < \alpha\ (\forall i)$

スケーリングフェイズ $O(n(\log n + F \log L))$ 時間× フェイズ数 $\log L$

= $O(n(\log n + F \log L) \log L)$ 時間

(L = 配分する資源量, F = 実行可能性判定の時間)

○ ● ○ ●　提案するアルゴリズム その2 (1/3)　● ○ ● ○

アイディア: 近似的な貪欲方向を利用

手順 **0:** $x = (0, 0, \ldots, 0)$

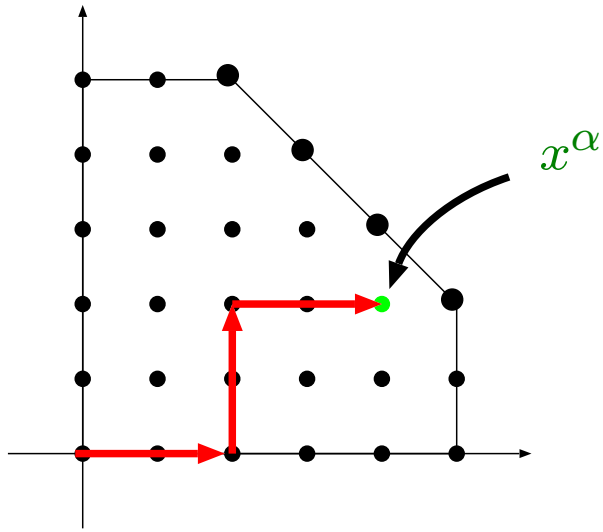
手順 **1:** $\sum_{i=1}^{n} x_i + \alpha > L$ ならば終了. $x^\alpha := x$ を出力

手順 **2:** 「近似的に貪欲」な方向 i を求める:

$f(x + \alpha\chi_i) = \min\{f(x + \alpha\chi_j) \mid x + \alpha\chi_j$: 実行可能$\}$

手順 **3:** $x := x + \alpha\chi_i$, 手順**1**へ

※ $\alpha = 1 \Longrightarrow$ 貪欲算法に一致



? 命題 ?　x^{α}: スケーリングフェイズで得られた解

$\Longrightarrow \exists x^*$: 最小解 **s.t.**　$x^{\alpha}(i) - x^*(i) < \alpha \ (\forall i)$

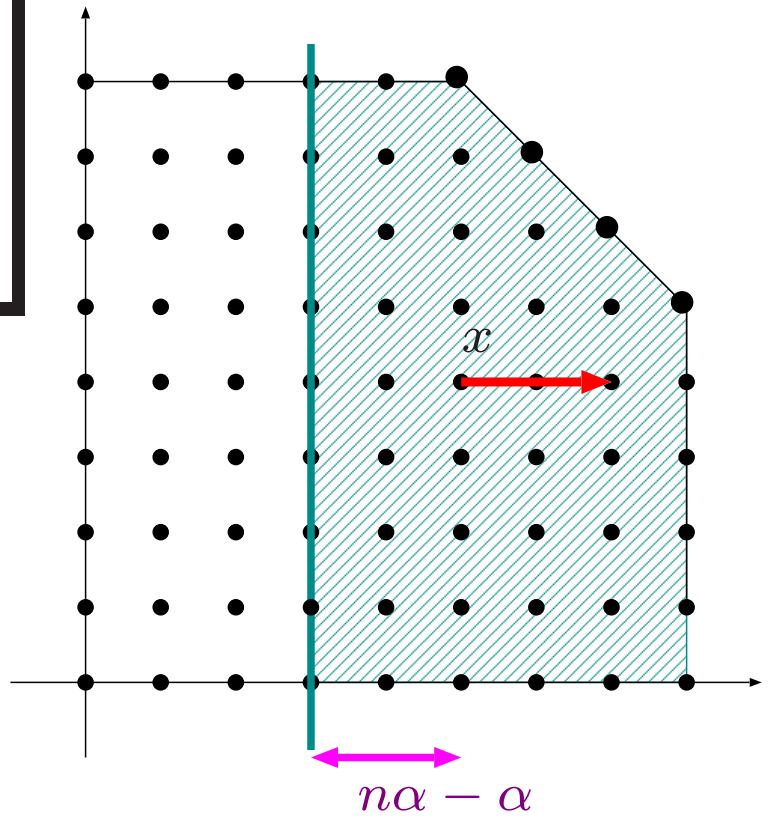
※ この命題は成り立たず

定理 (スケーリング版最小解カット) [田村 02]

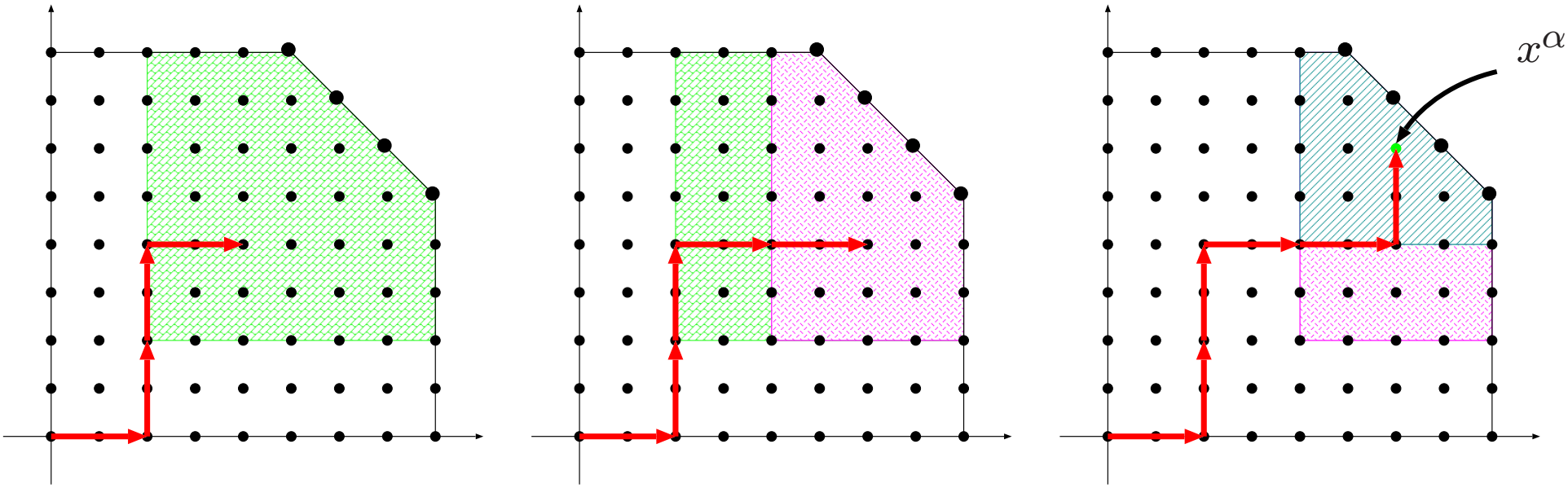
x: 実行可能解, i: 近似的に貪欲な方向

$f(x+\alpha\chi_i) = \min\{f(x+\alpha\chi_j) \mid x+\alpha\chi_j$: 実行可能$\}$

$\Longrightarrow \exists x^*$: 最小解 **s.t.** $x^*(i) > x(i) - (n\alpha - \alpha)$



提案するアルゴリズム その2 (3/3)



定理　x^α: スケーリングフェイズで得られた解

$\Longrightarrow \exists x^*$: 最小解 **s.t.**　$x^\alpha(i) - x^*(i) < n\alpha \ (\forall i)$

スケーリングフェイズ $O(n^2(\log n + F))$ × フェイズ数 $\log L$

= $O(n^2(\log n + F)\log L)$ 時間

※ アルゴリズムその1: $O(n(\log n + F\log L)\log L)$

M凸関数の定義 (1/2)

資源配分問題 $\rightarrow$ 関数 $f : \mathbf{Z}^n \rightarrow \mathbf{R} \cup \{+\infty\}$

$$f(x) = \begin{cases} \sum_{i=1}^{n} f_i(x_i) & \textbf{if } \sum_{i=1}^{n} x_i = L,\ x(S) \leq \rho(S)\ (\forall S \subseteq N), x \geq 0 \\ +\infty & \text{その他.} \end{cases}$$

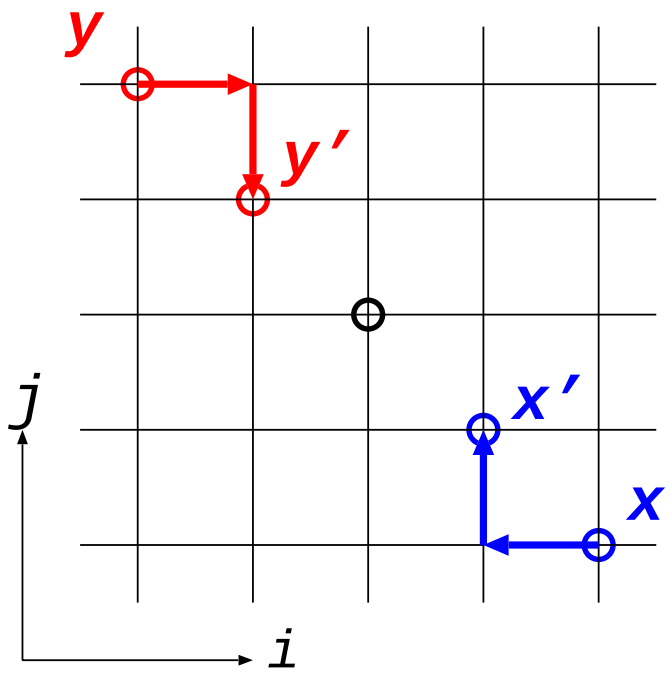
f の組合せ構造を抽象化 $\rightarrow$ M凸関数

M凸関数の定義 (2/2)

$f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ はM凸関数 $\overset{\text{def}}{\Longleftrightarrow}$

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

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



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

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

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

χ_i: 第 i 単位ベクトル

○ ● ○ ● 本発表の結果 (1/2) ● ○ ● ○

M凸関数最小化に対するアルゴリズム

提案する算法 — 資源配分問題に対する算法の拡張
(スケーリング版) 最小解カットを利用

拡張の難しさ — 解の更新方法

- 資源配分問題 — x の各成分は単調増加

- M凸関数最小化 — 成分和 $\sum_{i=1}^{n} x(i)$ は一定値
  $\Longrightarrow$ x のある成分 $\nearrow$, 別の成分 $\searrow$
  $\Longrightarrow$ 特定の成分 $x(i)$ のみ $\searrow$, 他の成分を単調増加

○ ● ○ ● 本発表の結果 (2/2) ● ○ ● ○

提案する算法

その1　$O((n^3 + n^2 \log L)\frac{\log L}{\log n})$, 最小解カット

L がそれほど大きくない($L \leq n^n$)ときは最速

その2　$O(n^3 \log L)$, スケーリング版最小解カット

シンプルなスケーリングアルゴリズム

L が大きい($L > n^n$)ときは最速

[資源配分問題　その1　$O(n(\log n + F \log L) \log L)$

その2　$O(n^2(\log n + F) \log L)$]

アルゴリズムの枠組みに自由度 $\Longrightarrow$ 更なる改良の可能性