

ジャンプシステム上の 最適化問題に対する アルゴリズム

塩浦昭義

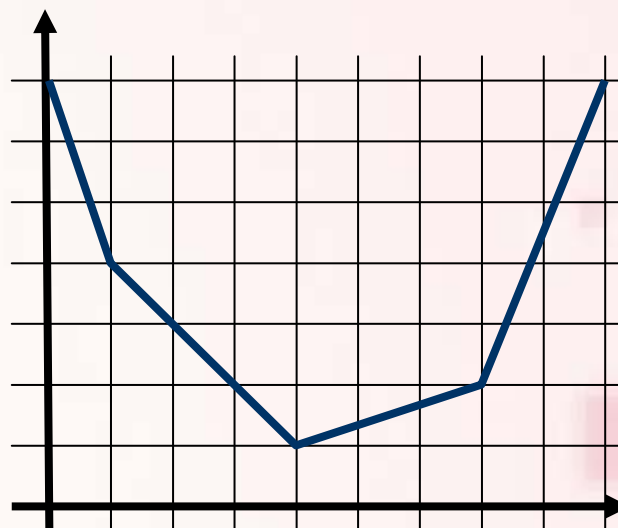
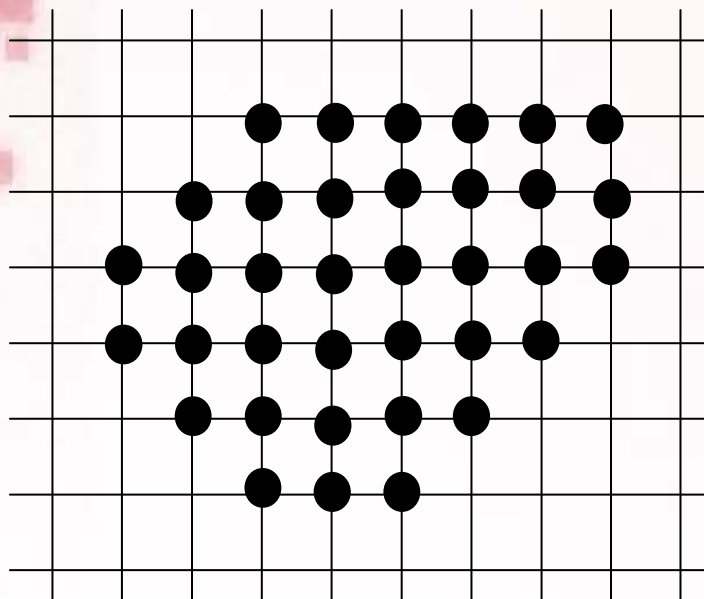
東北大学大学院情報科学研究科

（田中健一郎氏（元東大）との共同研究）

本講演の概要

- 離散凸集合 S の上での離散凸関数 $f(x)$ の最小化
 - 離散凸集合 = 組合せ的に良い性質をもつ整数ベクトル集合
 - 離散凸関数 = 組合せ的に良い性質をもつ関数

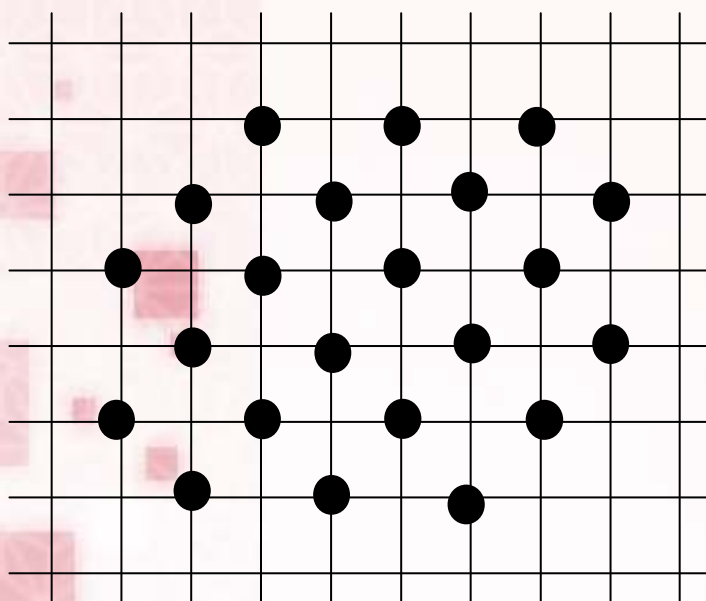
Minimize $f(x)$ subject to $x \in S$



本講演の概要

● やりたいこと:

ジャンプシステムSの上での離散凸関数 $f(x)$ の最小化



■ f --- 分離凸関数 $f(x) = \sum_{i=1}^n f_i(x_i)$

安藤-藤重-内藤(1995)の
擬多項式時間アルゴリズム

■ f --- M 凸関数

室田-田中(2006)の
擬多項式時間アルゴリズム

講演の内容:

これらの問題に対する多項式時間アルゴリズムの紹介

発表の流れ

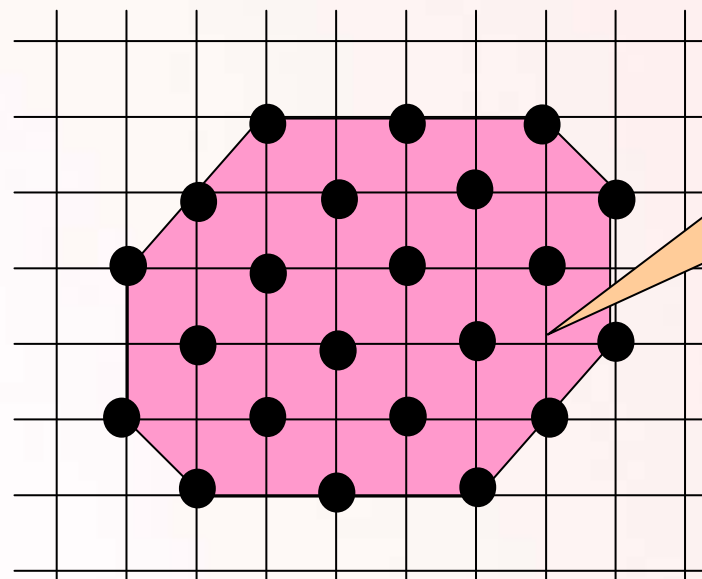
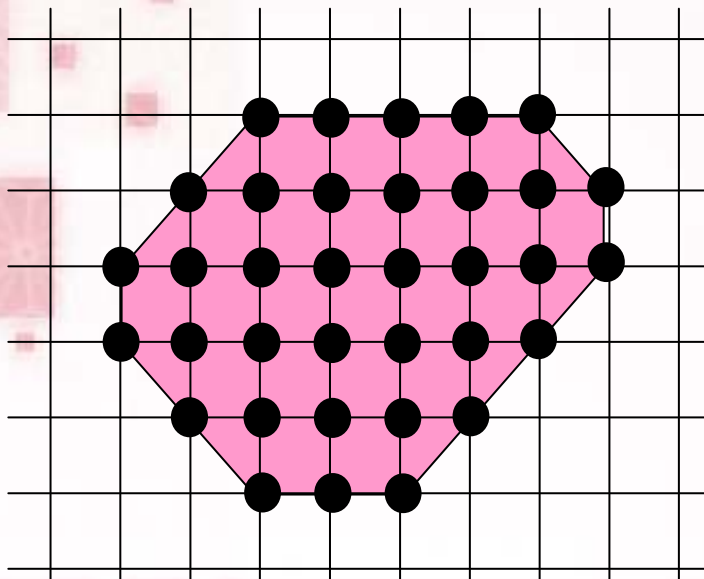
- ジャンプシステムとは？
- 離散凸関数の最小化手法
- ジャンプシステムに対する多項式時間アルゴリズム

発表の流れ

- ジャンプシステムとは？
 - 整数基集合との関係
 - 整数基集合, ジャンプシステムの例
 - 貪欲アルゴリズム
 - 定義
- 離散凸関数の最小化手法
- ジャンプシステムに対する多項式時間アルゴリズム

ジャンプシステムとは？

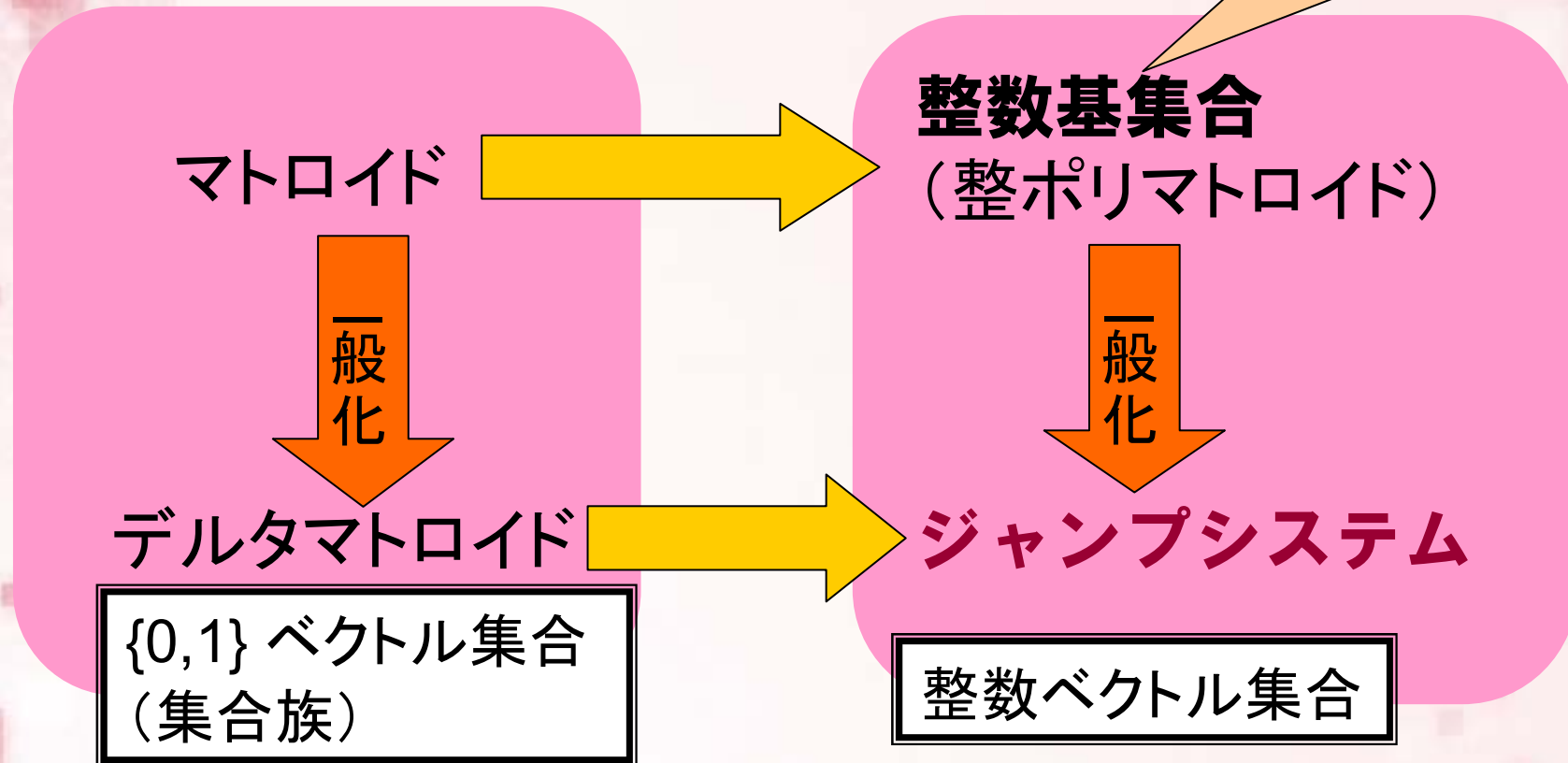
- Bouchet-Cunningham (1995) が提案した概念
- 組合せ的に良い性質をもつ整数ベクトル集合
- マトロイド, 整数基集合 (整ポリマトロイド), デルタマトロイドなどの共通の一般化
- 貪欲アルゴリズムにより線形関数最小化が可能



「穴」が
存在する

各種組合せ構造の関係

整基多面体の
整数点集合



- マトロイド = $\{0,1\}$ ベクトルからなる整数基集合
- デルタマトロイド = $\{0,1\}$ ベクトルからなるジャンプシステム

各種組合せ構造の関係

➤ 部分集合の $\{0,1\}$ ベクトルによる表現

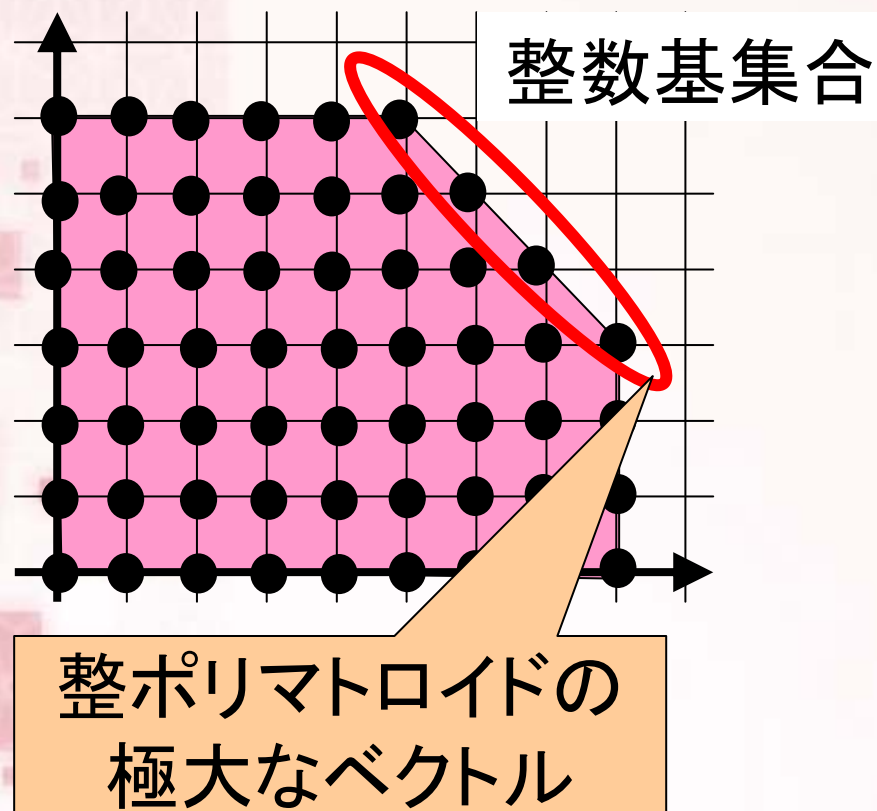
$$X \subseteq E \iff \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \in \{0,1\}^E$$
$$\begin{array}{ll} e \in X & \iff x_e = 1 \\ e \in E \setminus X & \iff x_e = 0 \end{array}$$

例: $E = \{a, b, c\}$

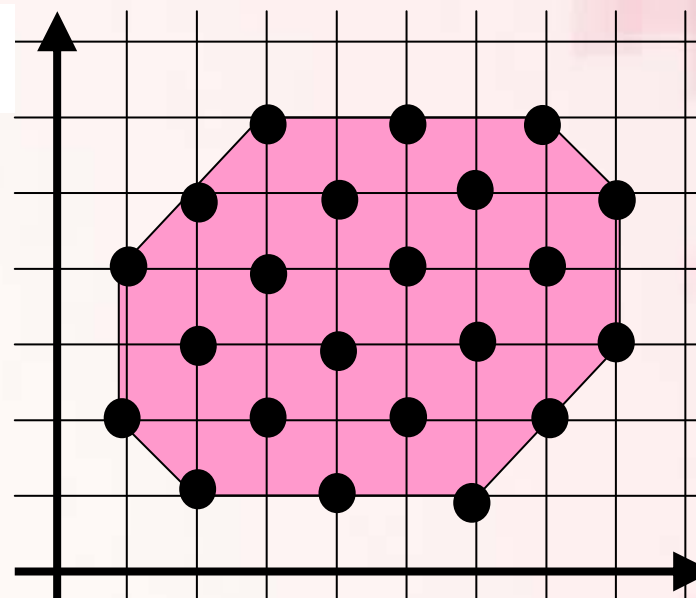
$$X = \{a, c\} \implies \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \quad X = \{b\} \implies \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

各種組合せ構造の関係

整ポリマトロイド



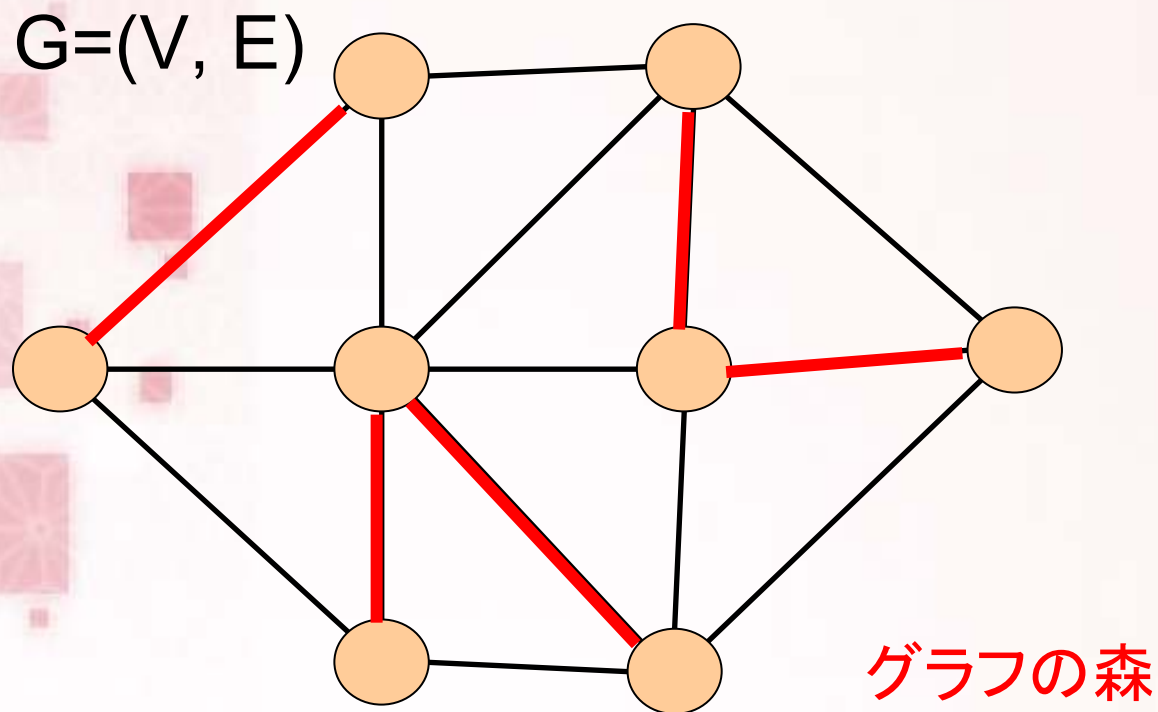
ジャンプシステム



すべての象限に
「整ポリマトロイド」が存在

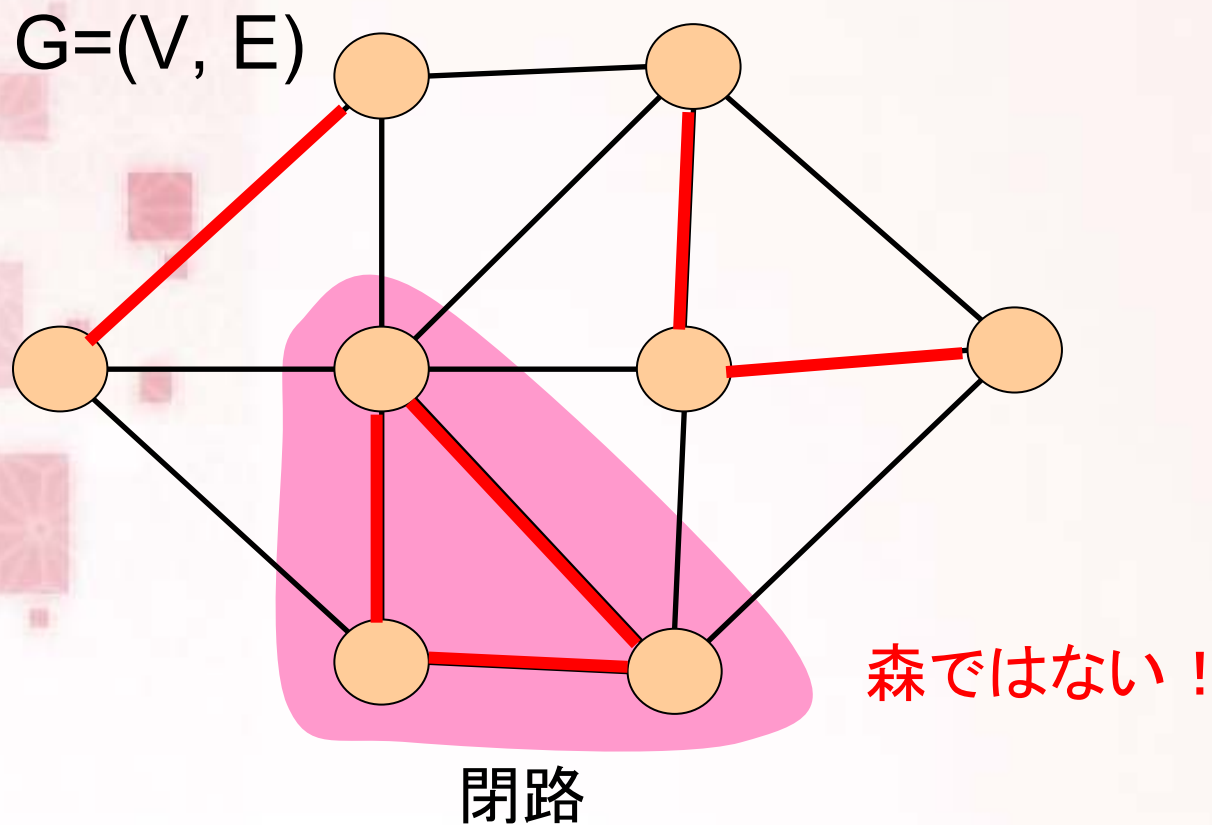
整数基集合の例（その１）

- 無向グラフの森, 全域木



整数基集合の例（その１）

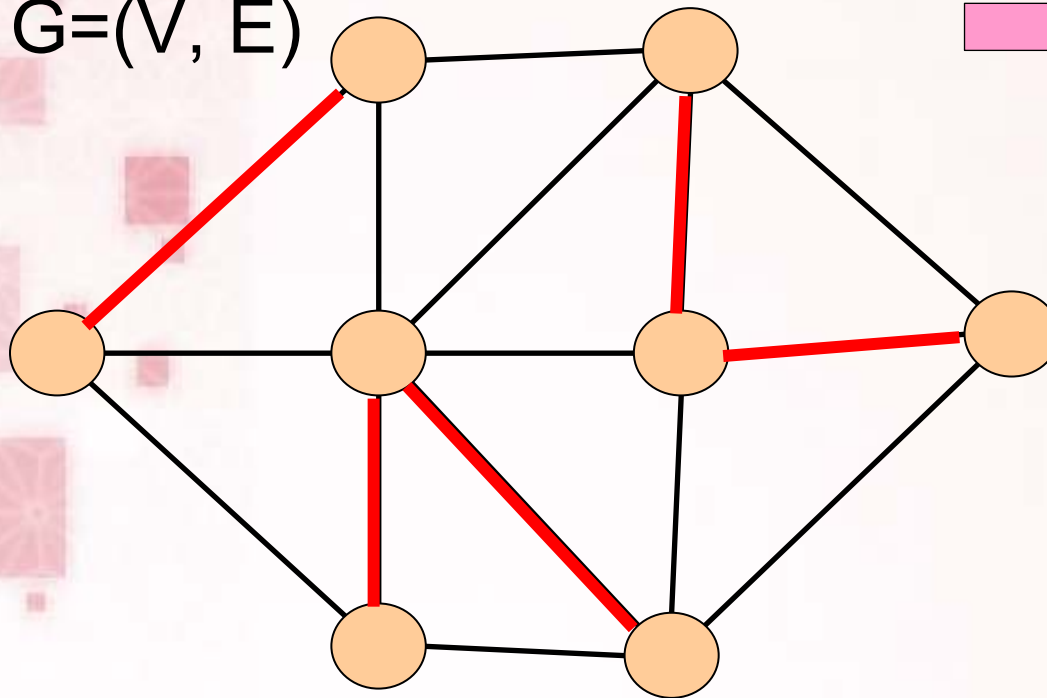
- 無向グラフの森, 全域木



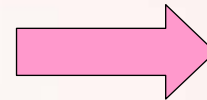
整数基集合の例（その１）

- 無向グラフの森, 全域木

$G=(V, E)$



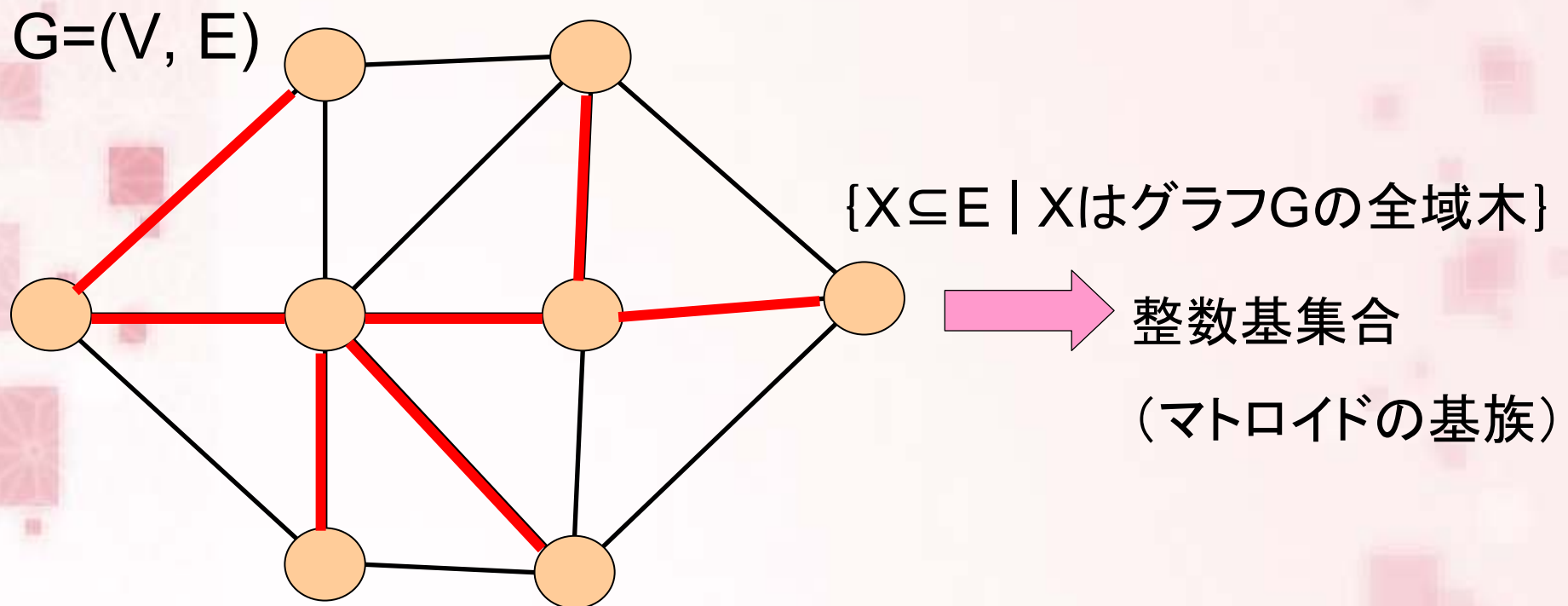
$\{X \subseteq E \mid X \text{ はグラフ } G \text{ の森}\}$



整ポリマトロイド
(マトロイドの
独立集合族)

整数基集合の例（その1）

- 無向グラフの森, 全域木



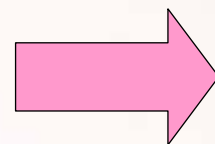
グラフの全域木 (極大な森)

整数基集合の例（その2）

- 行列の一次独立な列集合

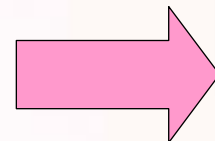
$$\begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 2 & -1 \end{bmatrix}$$

$\emptyset, \{1\}, \{2\}, \{3\}, \{4\},$
 $\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}$



整ポリマトロイド

■ 極大な一次独立列集合の
集まり



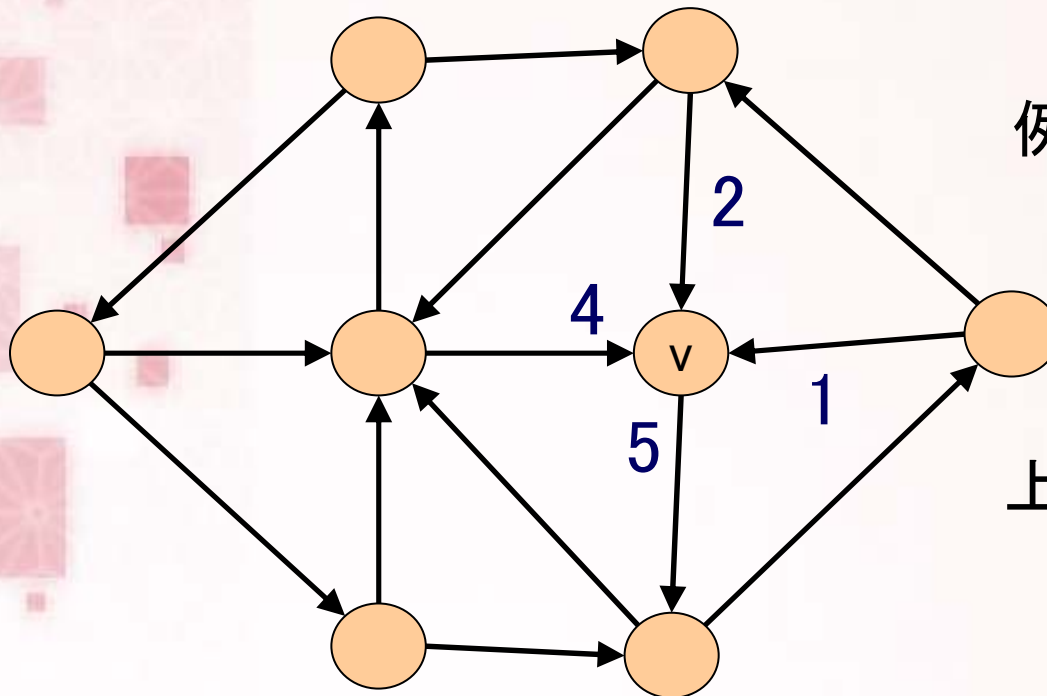
整数基集合

整数基集合の例（その3）

● ネットワークフロー

フロー $x \in \mathbb{Z}^A$ の境界

$G=(V, A)$



$\partial x(v) \equiv$ 頂点 v での
流出量－流入量

$$\text{例: } \partial x(v) = 5 - (4 + 2 + 1) = 2$$

上下限制約を満たすフローの
境界の集合

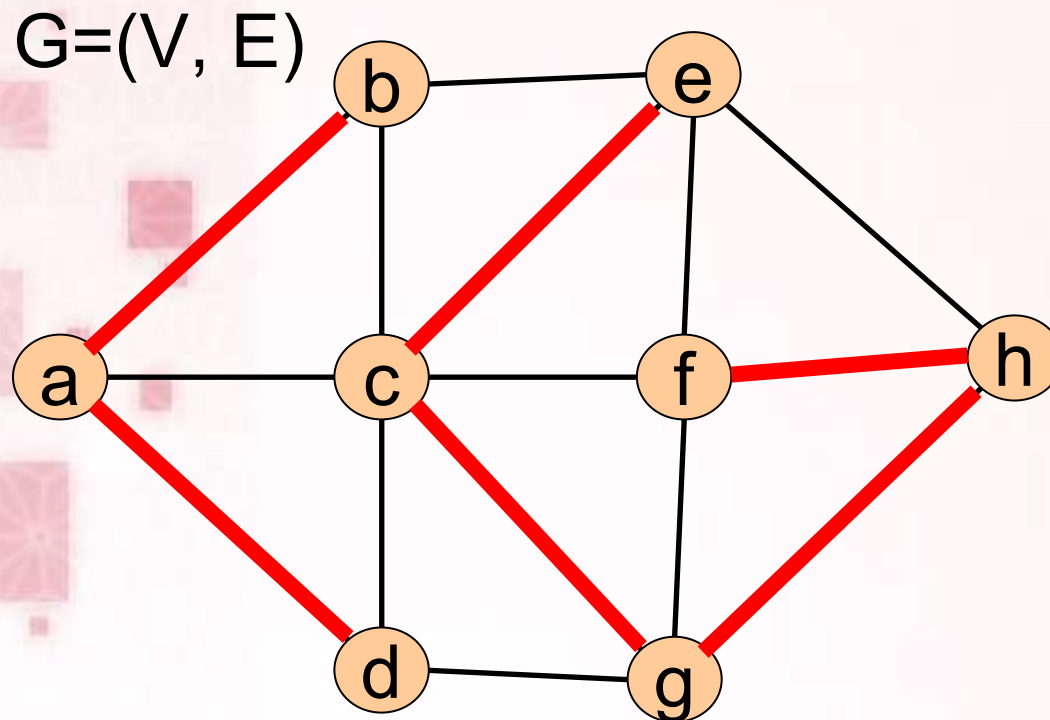


整数基集合

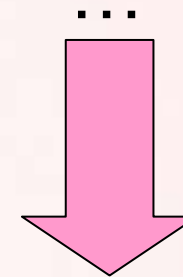
$$l(a) \leq x(a) \leq u(a) \quad (a \in A)$$

ジャンプシステムの例（その１）

- マッチングによりカバーされる頂点集合



$\{a, b, c, e, g, h\},$
 $\{a, c, d, f, g, h\},$



ジャンプシステム

ジャンプシステムの例（その2）

- 歪対称行列の正則小行列

$$\begin{bmatrix} \boxed{\begin{matrix} 0 & 2 \\ -2 & 0 \end{matrix}} & \begin{matrix} -1 & 3 & 5 \end{matrix} \\ \begin{matrix} 1 \\ -3 \\ -5 \end{matrix} & \boxed{\begin{matrix} -1 & 0 & 2 & -3 \\ 4 & -2 & 0 & 0 \\ -1 & 3 & 0 & 0 \end{matrix}} \end{bmatrix}$$

$\{1,2\}$

$\{2,3,4,5\}$

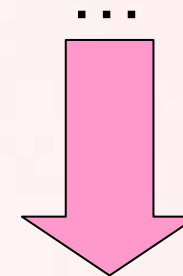
ジャンプシステムの例（その2）

- 歪対称行列の正則小行列

$$\begin{bmatrix} \boxed{\begin{matrix} 0 & 2 & -1 \\ -2 & 0 & 1 \\ 1 & -1 & 0 \end{matrix}} & \begin{matrix} 3 \\ -4 \\ 2 \end{matrix} \\ \begin{matrix} -3 \\ -5 \end{matrix} & \boxed{\begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix}} \end{bmatrix}$$

{1,2}

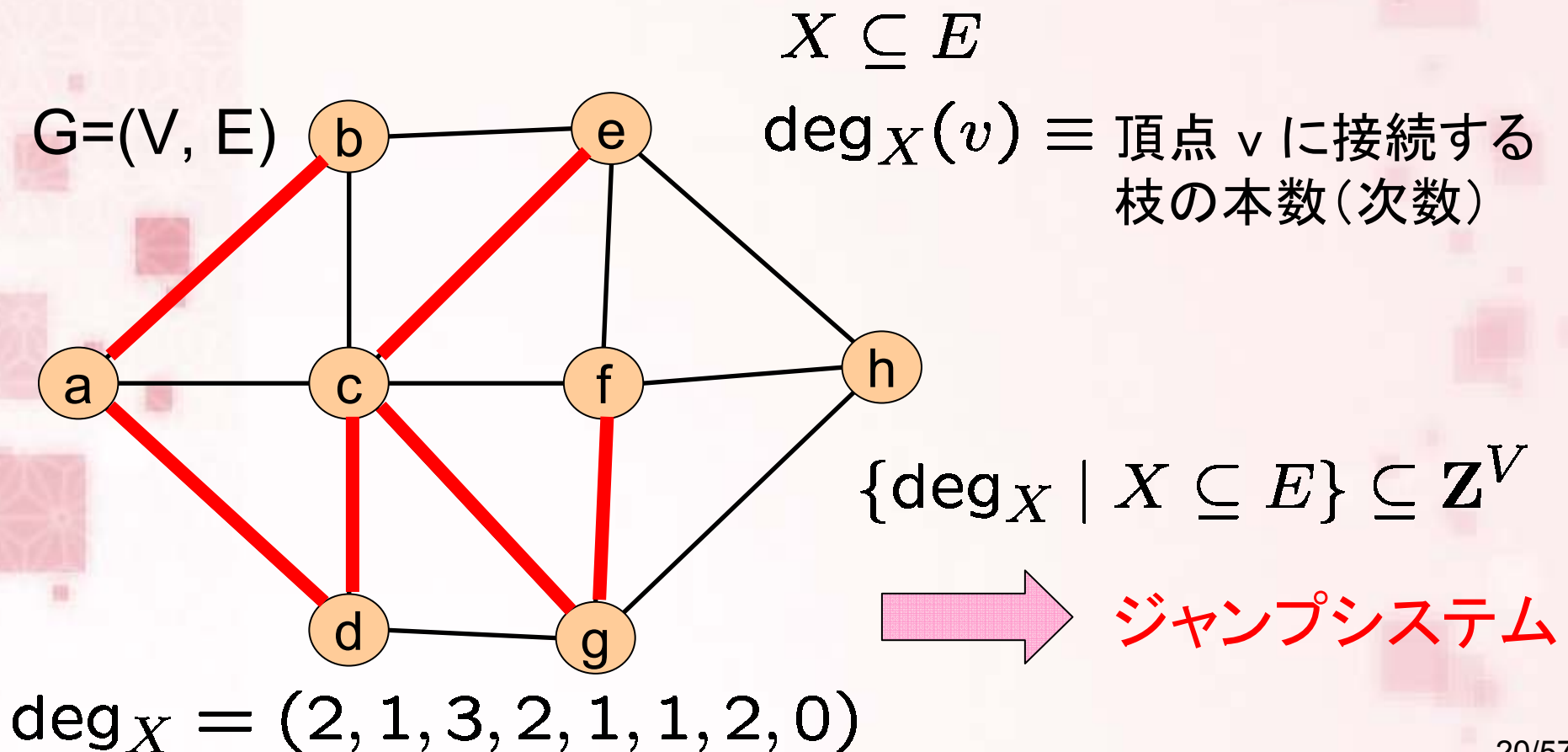
{2,3,4,5}



ジャンプシステム

ジャンプシステムの例（その3）

無向グラフの次数列



ジャンプシステムに対する 貪欲アルゴリズム

ジャンプシステム上での線形関数最小化は
貪欲アルゴリズムにより解ける

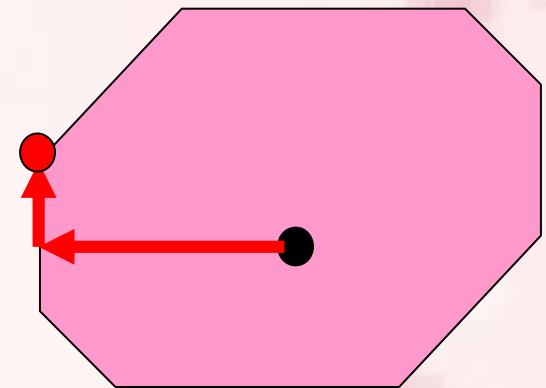
$$\text{Min. } w^T x \text{ s.t. } x \in S \subseteq \mathbb{Z}^V$$

1. $|w_1| \geq |w_2| \geq \dots \geq |w_k| > 0 = |w_{k+1}| = \dots = |w_n|$
となるように添え字を入れ替え

$$S_0 := S$$

2. 各 $i = 1, 2, \dots, k$ に対し, 以下を実行

- $w_i > 0 \rightarrow x_i^* := \max(x_i \mid x \in S_{i-1})$
- $w_i < 0 \rightarrow x_i^* := \min(x_i \mid x \in S_{i-1})$
- $S_i := \{x \in S_{i-1} \mid x_i = x_i^*\}$



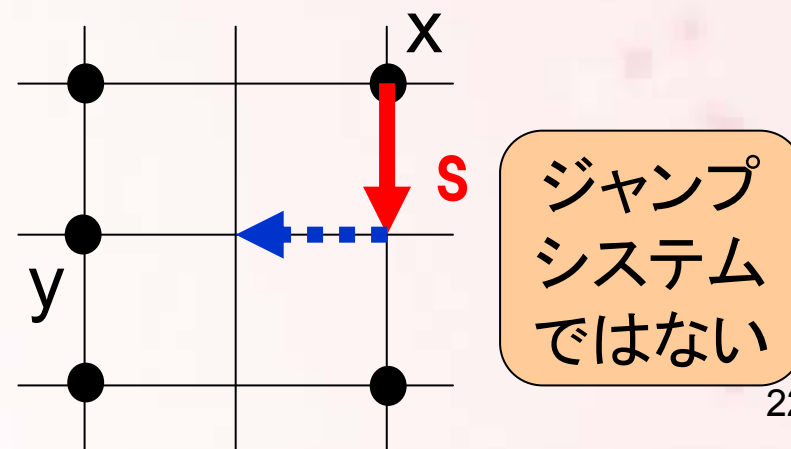
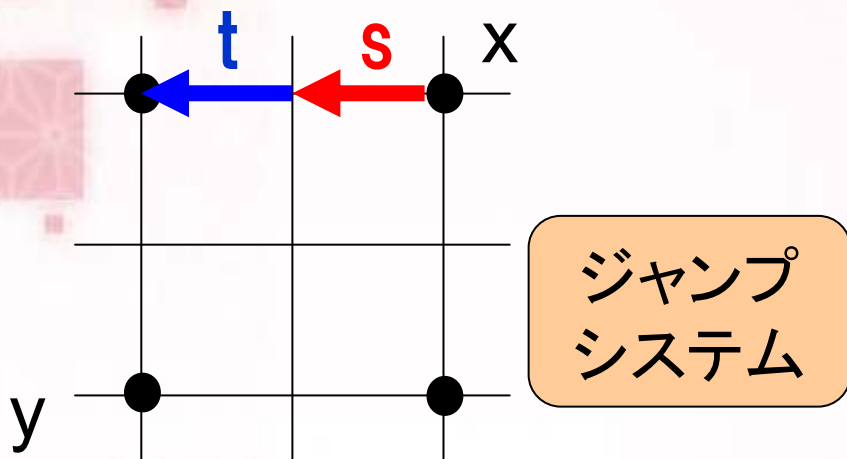
ジャンプシステムの定義

$S \subseteq \mathbb{Z}^V$ はジャンプシステム $\longleftrightarrow$ 2ステップ公理

$$\forall x, y \in S, \forall s \in \text{St}(x, y), \\ x + s \in S \quad \text{or} \quad x + s + t \in S \quad (\exists t \in \text{St}(x + s, y))$$

$\text{St}(x, y)$: (x, y) ステップの集合

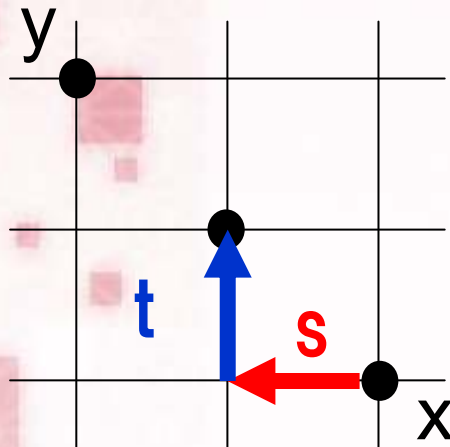
$$s \in \text{St}(x, y) \iff \begin{aligned} &s \in \mathbb{Z}^V, \|s\|_1 = 1, \\ &\|(x + s) - y\|_1 = \|x - y\|_1 - 1 \end{aligned}$$



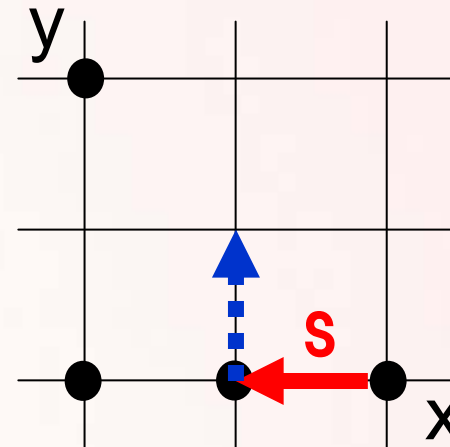
整数基集合の定義

$S \subseteq \mathbb{Z}^V$ は整数基集合

$\forall x, y \in S, \forall s \in \text{St}(x, y), s \leq 0,$
 $\exists t \in \text{St}(x + s, y) \text{ s.t. } t \geq 0, \text{ and } x + s + t \in S$



整数基集合

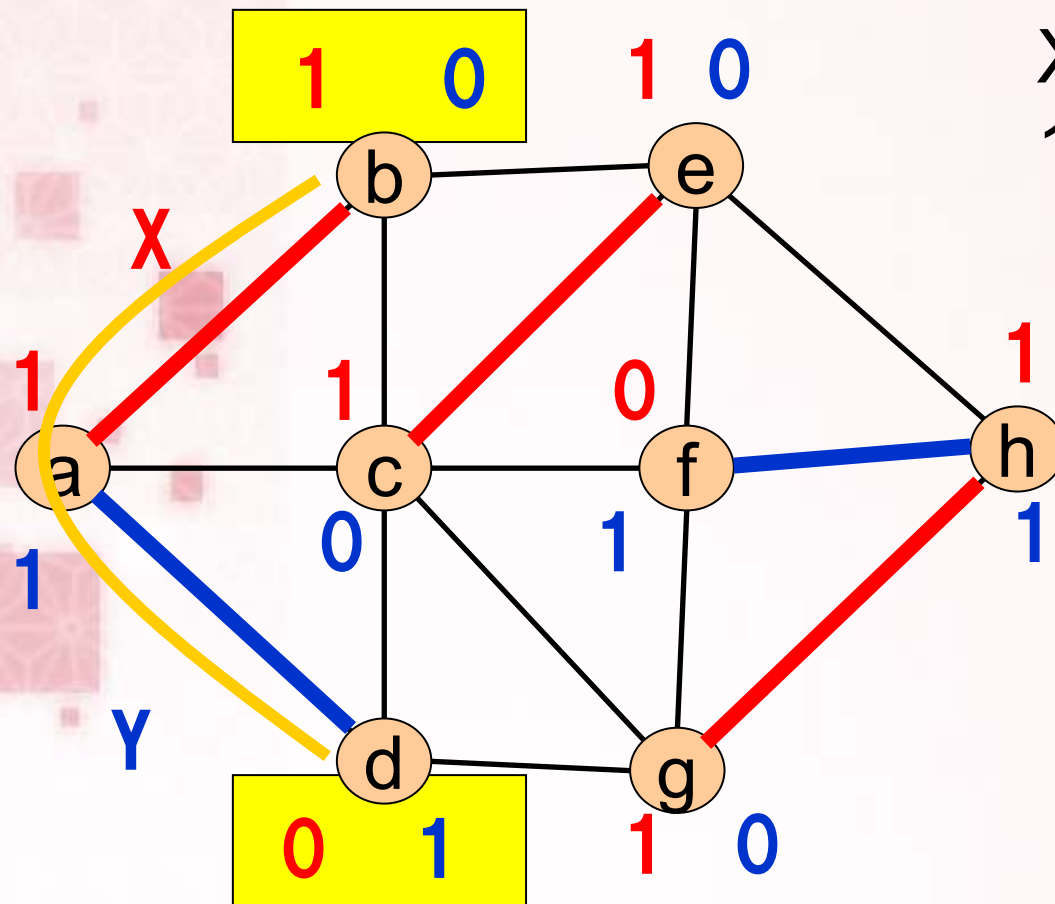


整数基集合
ではない

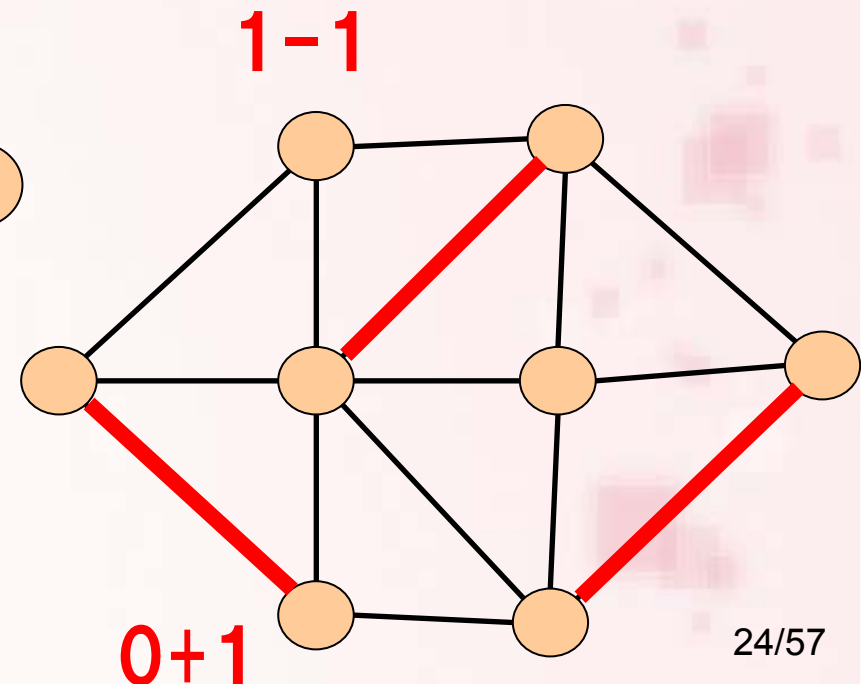
S : 整数基集合 $\rightarrow$ ベクトルの成分和 $\sum_{v \in V} x(v)$ ($x \in S$)
は一定

ジャンプシステムであることの証明

- マッチングによりカバーされる頂点集合

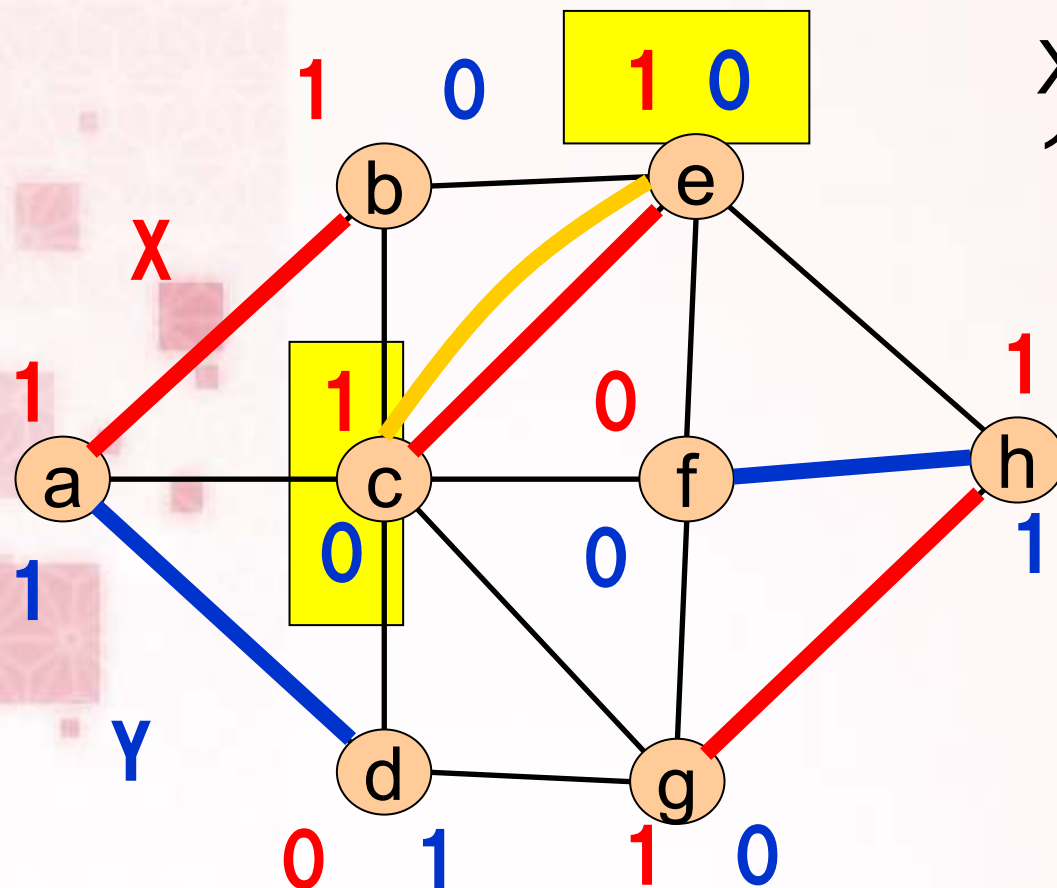


Xにおけるbの次数を
1減らしたい

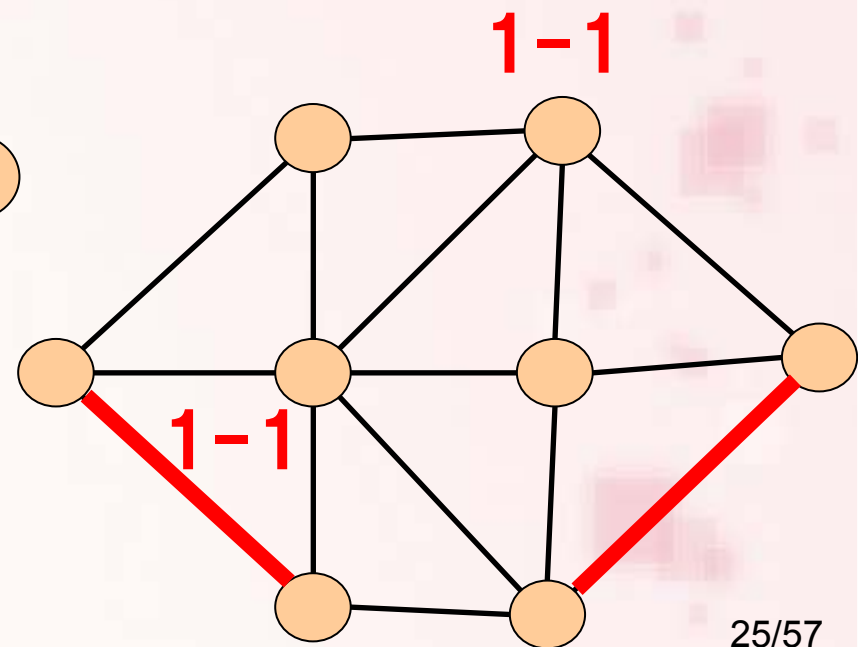


ジャンプシステムであることの証明

- マッチングによりカバーされる頂点集合



Xにおけるcの次数を
1減らしたい



ジャンプシステムに対する操作

- 整数ベクトルによる平行移動 $(x_1 + a_1, x_2 + a_2, \dots, x_n + a_n)$
- ある軸方向の反転 $(x_1, -x_2, \dots, x_n)$
- ある成分に関する制限 $(x_1, x_2, \dots, x_k, \boxed{x_{k+1}, \dots, x_n})$
- 整数区間との共通部分 $S \cap [a, b]$
- ジャンプシステム同士の和 $S_1 + S_2$
- ジャンプシステム同士の直積 $S_1 \times S_2$

いずれの操作も
ジャンプシステムの性質を保つ

発表の流れ

- ジャンプシステムとは？
- 離散凸関数の最小化手法
 - 鍵となる性質
 - 様々なアルゴリズム
- ジャンプシステムに対する多項式時間アルゴリズム

離散凸関数の最適化手法

Minimize $f(x)$ subject to $x \in S$

$S \subseteq \mathbf{Z}^V$, 離散凸集合, $f: S \rightarrow \mathbf{R}$, 離散凸関数,

- 貪欲アルゴリズム
 - 降下法 (指数時間)
 - 最急降下法 (擬多項式時間)
- 領域縮小法 (多項式時間)
- スケーリング法 (多項式時間)

アルゴリズムの計算時間について

Minimize $f(x)$ subject to $x \in S$

$S \subseteq \mathbb{Z}^V$, 離散凸集合, $f: S \rightarrow \mathbb{R}$, 離散凸関数,

仮定：

- 与えられたベクトル x に対し $x \in S$ か否かを単位時間で判定
- 与えられたベクトル x に対し関数値 $f(x)$ を単位時間で計算
- $x_0 \in S$ は前もって与えられている

問題のサイズ

ベクトルの次元 $n = |V|$

解集合 S の大きさ $L: \max_{x \in S} x(i) - \min_{x \in S} x(i) \leq L$

• n と L の多項式 $\rightarrow$ 擬多項式

• n と $\log L$ の多項式 $\rightarrow$ (弱)多項式

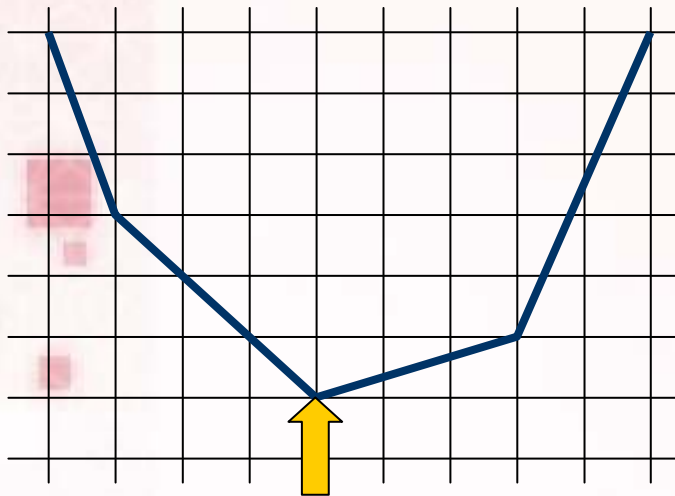
鍵となる性質

- 局所最適性 → 大域的最適性
 - 降下法 (指数時間)
- 最適解の分離可能性 (最小解カット)
 - 最急降下法 (擬多項式時間)
 - 領域縮小法 (多項式時間)
- 近接定理
 - スケーリング法 (多項式時間)

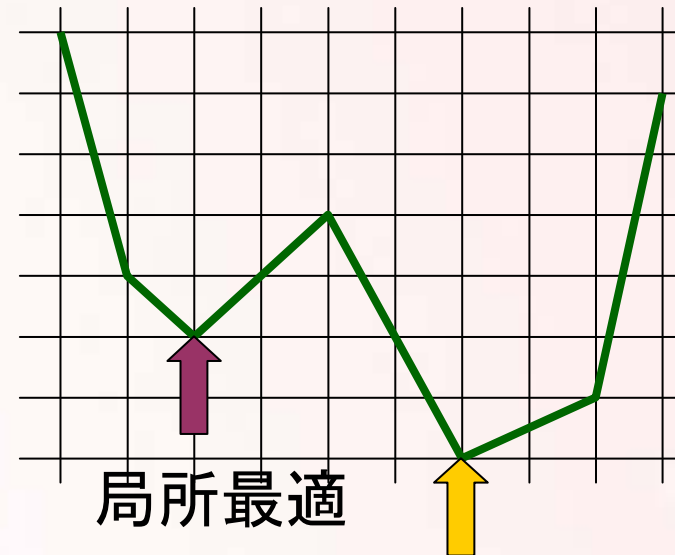
鍵となる性質（その1）

- 局所最適性 $\rightarrow$ 大域的最適性

$x \in S$ の近傍 $N(x)$ で $f(x)$ は最小 $\rightarrow x$ は最適解



局所最適 = 大域的最適



局所最適

大域的最適

貪欲アルゴリズム（降下法）

- 局所最適性 → 大域的最適性

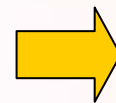
$x \in S$ の近傍 $N(x)$ で $f(x)$ は最小 → x は最適解

貪欲アルゴリズム（降下法）

以下を繰り返し実行

- $f(x) = \min\{f(y) \mid y \in N(x)\}$ を判定 (yes → 最適解)
- $f(y^*) = \min\{f(y) \mid y \in N(x)\}$ を満たす $y^* \in N(x)$ を求める
- $x := y^*$ とおく

- 最悪 S のすべての点を通る
- 同じ点は1度しか通らない



反復回数は指数オーダー

貪欲アルゴリズム（降下法）

- 局所最適性 → 大域的最適性

$x \in S$ の近傍 $N(x)$ で $f(x)$ は最小 → x は最適解

例1: $N(x) = \{y \in \mathbf{Z}^V \mid \|y - x\|_\infty \leq 1\}$ 指数サイズ

(S, f) が Miller(1971) の離散凸関数

2ステップ近傍

例2: $N(x) = \{y \in \mathbf{Z}^V \mid \|y - x\|_1 \leq 2\}$ サイズ $O(n^2)$

(S, f) が整数基集合上の M 凸関数 (室田1995)

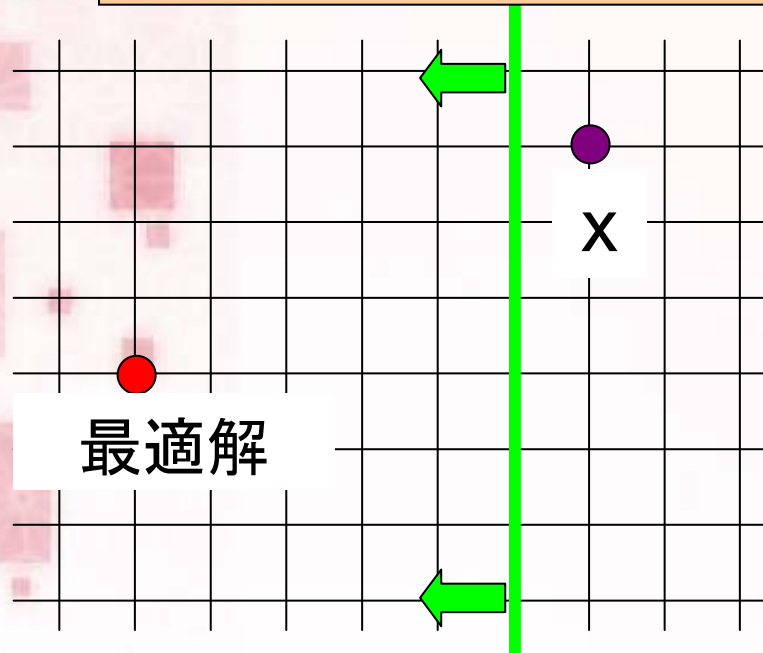
S がジャンプシステム, f が分離凸関数 (安藤-藤重-内藤(1995))

(S, f) がジャンプシステム上の M 凸関数 (室田2006)

鍵となる性質（その2）

■ 最適解の分離可能性（最小解カット）

与えられた $x \in S$ と最適解を分離する超平面が効率的に得られる

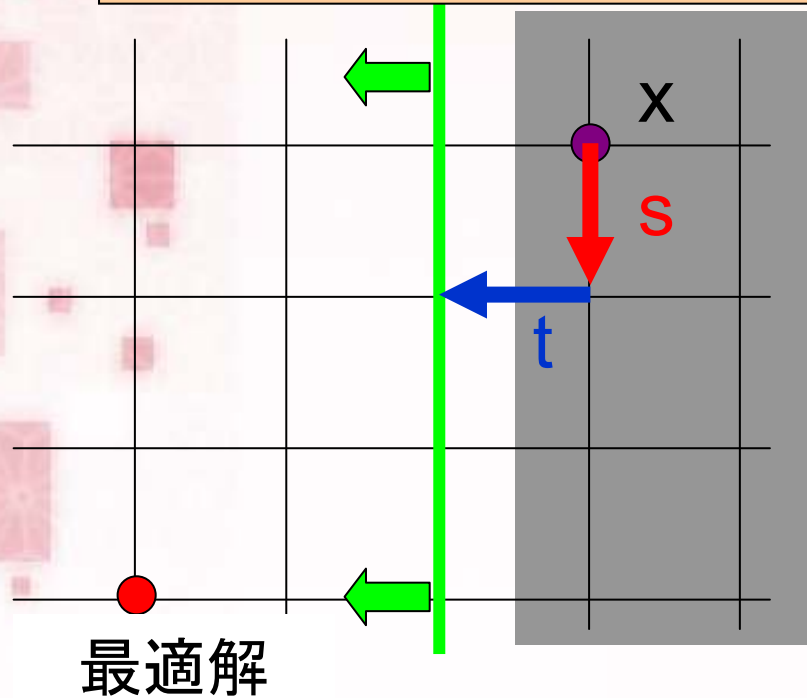


S が整数基集合, f が分離凸関数
(Girlich et al. 1996)
 (S, f) が整数基集合上のM凸関数
(塩浦1998)
 (S, f) がジャンプシステム上の
M凸関数(室田-田中2006)

鍵となる性質（その2）

■ 最適解の分離可能性（最小解カット）

与えられた $x \in S$ と最適解を分離する超平面が効率的に得られる



2ステップ近傍の中での
最急降下方向を利用

$$N(x) = \{x + s + t\}$$

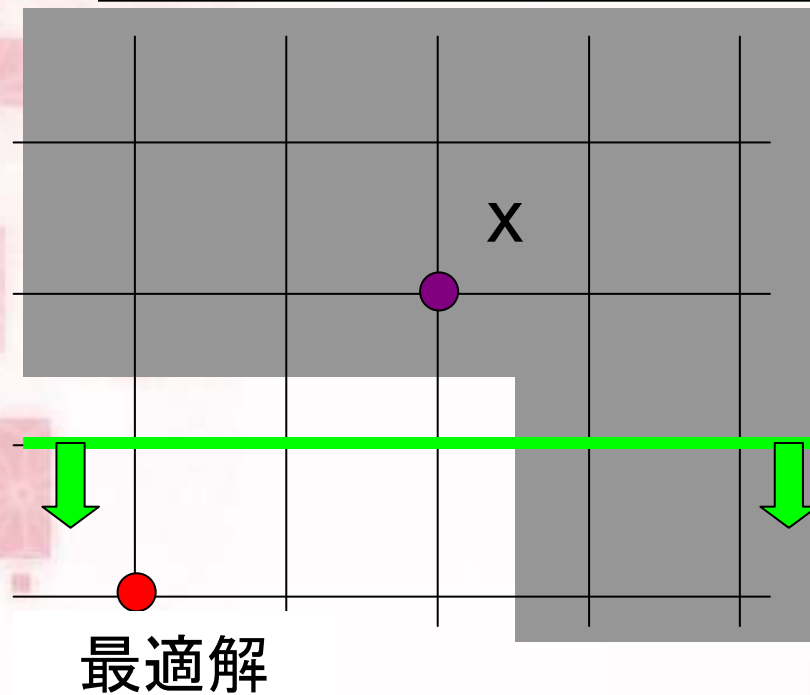
s を固定, 最適な t を求める

$O(n)$ 時間

鍵となる性質（その2）

- 最適解の分離可能性（最小解カット）

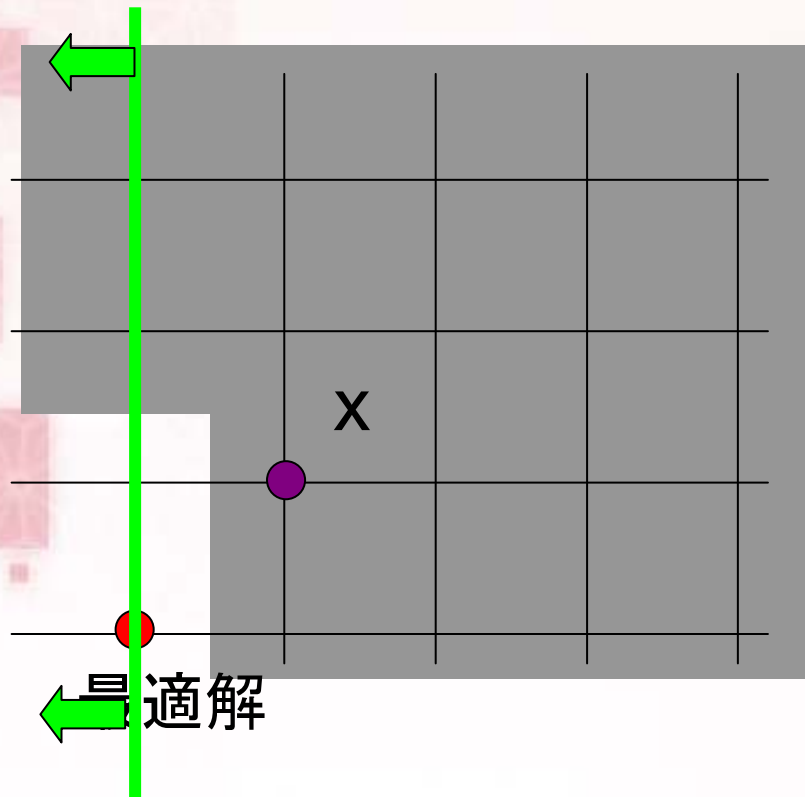
与えられた $x \in S$ と最適解を分離する超平面が効率的に得られる



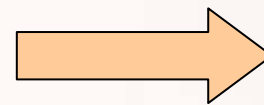
貪欲アルゴリズム（最急降下法）

最適解の分離可能性（最小解カット）

与えられた $x \in S$ と最適解を分離する超平面が効率的に得られる



各反復において、
ある軸方向の幅が1以上小さくなる



反復回数: nL

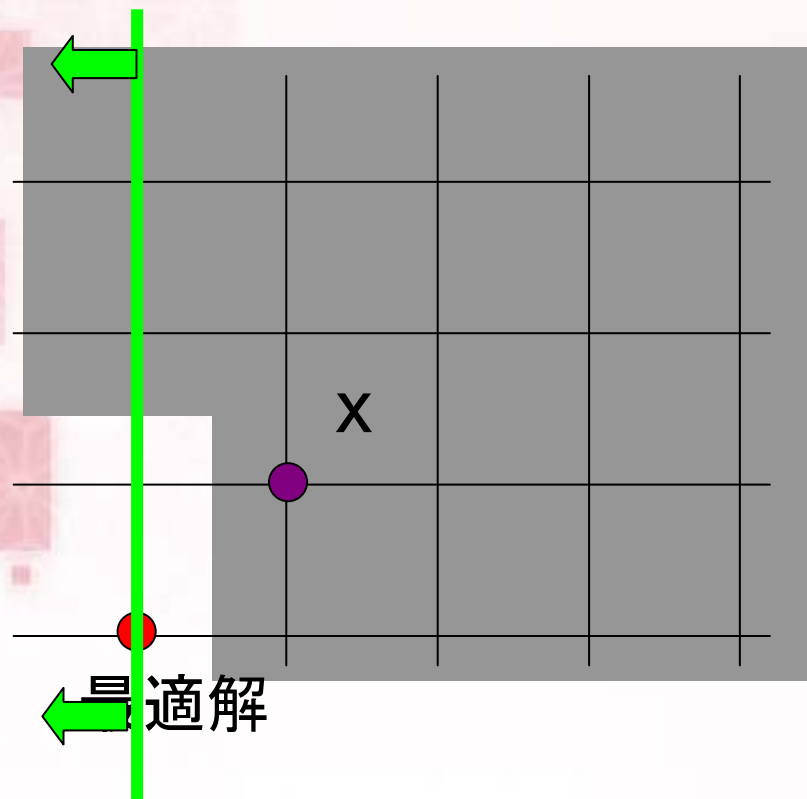
貪欲アルゴリズム
（最急降下法）

整数基集合, ジャンプシステムの
線形関数最小化, 分離凸関
数最小化に適用可

領域縮小法

- 最適解の分離可能性(最小解カット)

与えられた $x \in S$ と最適解を分離する超平面が効率的に得られる



高速化(領域縮小法)

各反復で x の選び方を工夫

反復回数: 弱多項式

(詳細は後ほど)

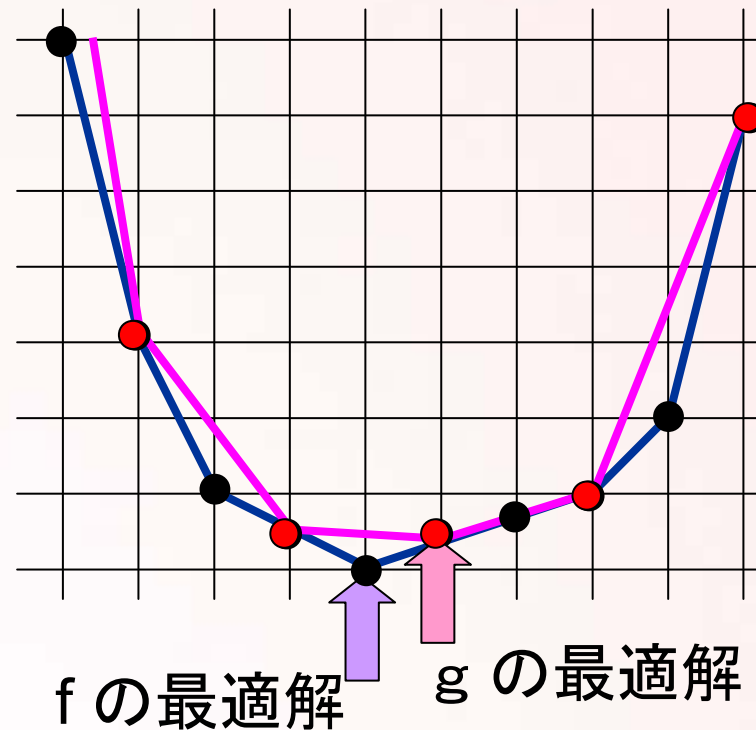
鍵となる性質（その3）

近接定理

オリジナルの
関数 f

スケール
リング

近似関数 g



スケールリングされた関数の最適解
≡元の関数の最適解

(S, f) が整数基集合上のM凸関数
(森口-室田-塩浦2002)

スケーリングアルゴリズム

近接定理

スケーリングされた関数の最適解
≡元の関数の最適解

アルゴリズム

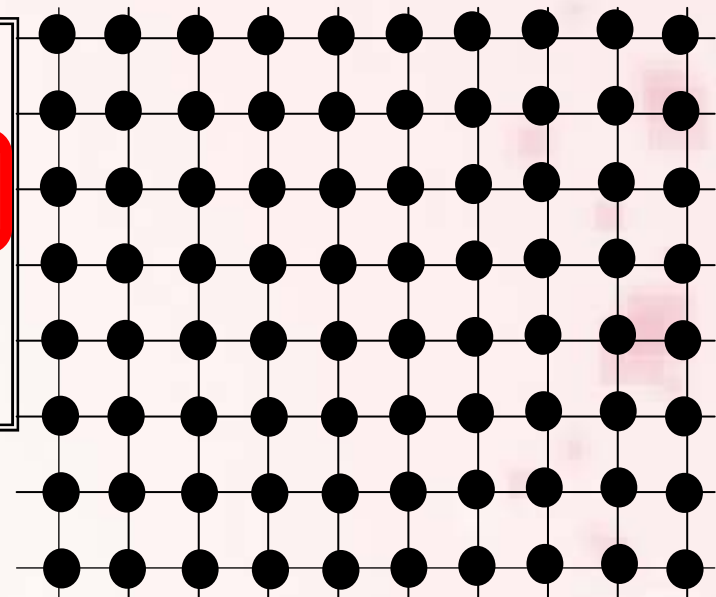
$\alpha := \lceil \log_2 L \rceil$

$2^\alpha \cdot \mathbf{Z}^V$ 上で f の最適解を求める

$\alpha = 0$ ならば終了

$\alpha := \alpha - 1$

反復回数 $O(\log L)$



$4 \mathbf{Z}^V$ $2 \mathbf{Z}^V$ $\mathbf{Z}^V$

スケーリングアルゴリズム

近接定理

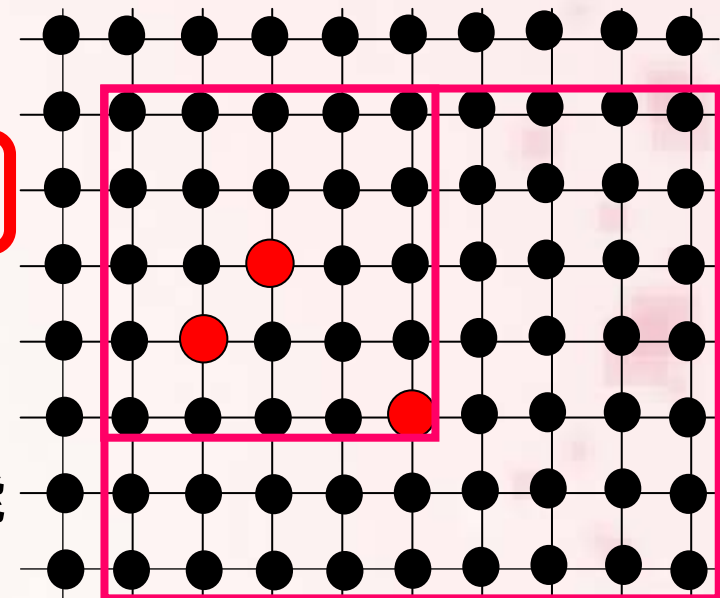
スケーリングされた関数の最適解
≡元の関数の最適解

アルゴリズム

$2^\alpha \cdot \mathbb{Z}^V$ 上で f の最適解を求める

- 近接定理により, 探索範囲を限定できる
- さらなる工夫により, 多項式時間で可能

(S, f) が整数基集合上の M 凸関数
(田村2005)(塩浦2004)



$4 \mathbb{Z}^V$ $2 \mathbb{Z}^V$ $\mathbb{Z}^V$

発表の流れ

- ジャンプシステムとは？
- 離散凸関数の最小化手法
- ジャンプシステムに対する多項式時間アルゴリズム
 - 領域縮小法の流れ
 - 各反復での x の選び方
 - 証明

ジャンプシステム上の分離凸関数 最小化に対する多項式時間アルゴリズム

$$\text{Minimize } \sum_{i \in V} f_i(x_i) \text{ subject to } x \in S$$

$S \subseteq \mathbf{Z}^V$: ジャンプシステム, $f_i: \mathbf{Z} \rightarrow \mathbf{R}$, 凸関数 ($i \in V$)

例: 無向グラフの次数列に関する最適化

Given: 無向グラフ $G=(V, E)$, ベクトル $b \in \mathbf{Z}^V$

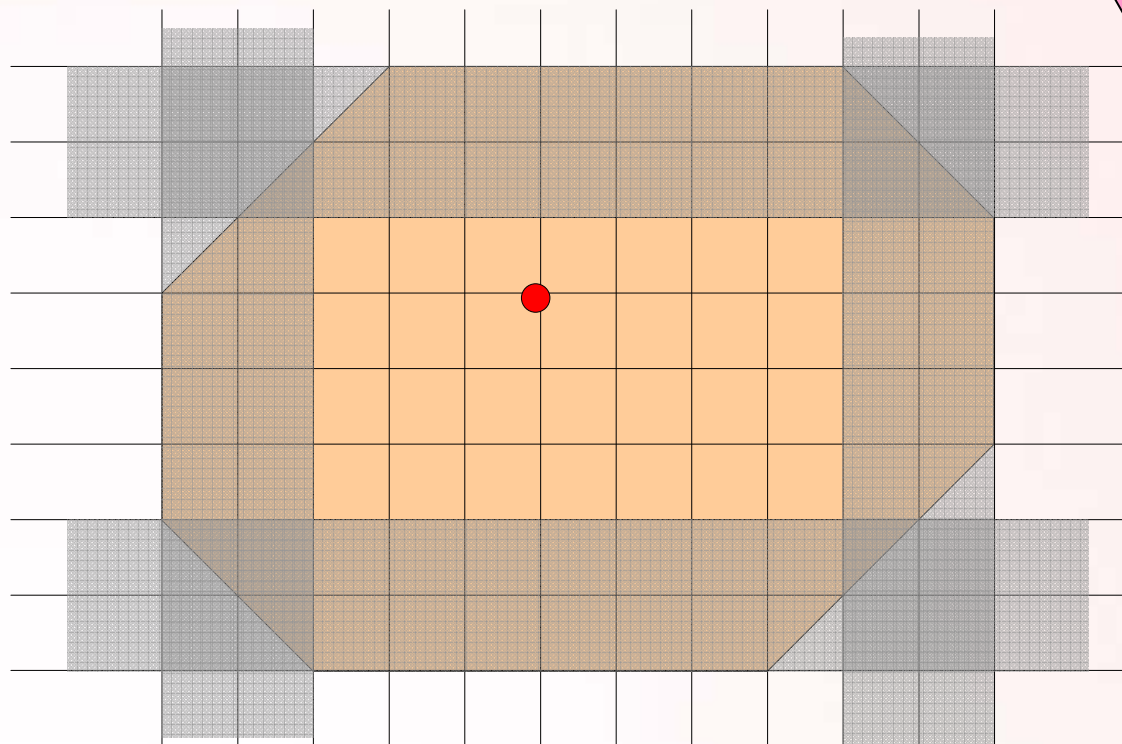
Find: 枝集合 $X \subseteq E$ minimizing $\sum_{v \in V} \{\deg_X(v) - b(v)\}^2$

- 領域縮小法を適用, 計算時間 $O(n^4 (\log L)^2)$

領域縮小法の流れ

- 与えられたベクトル x を最適解から分離
+ x の選び方の工夫 → 多項式時間アルゴリズム

x の選び方: 解候補の集合の境界から離れた点を選択

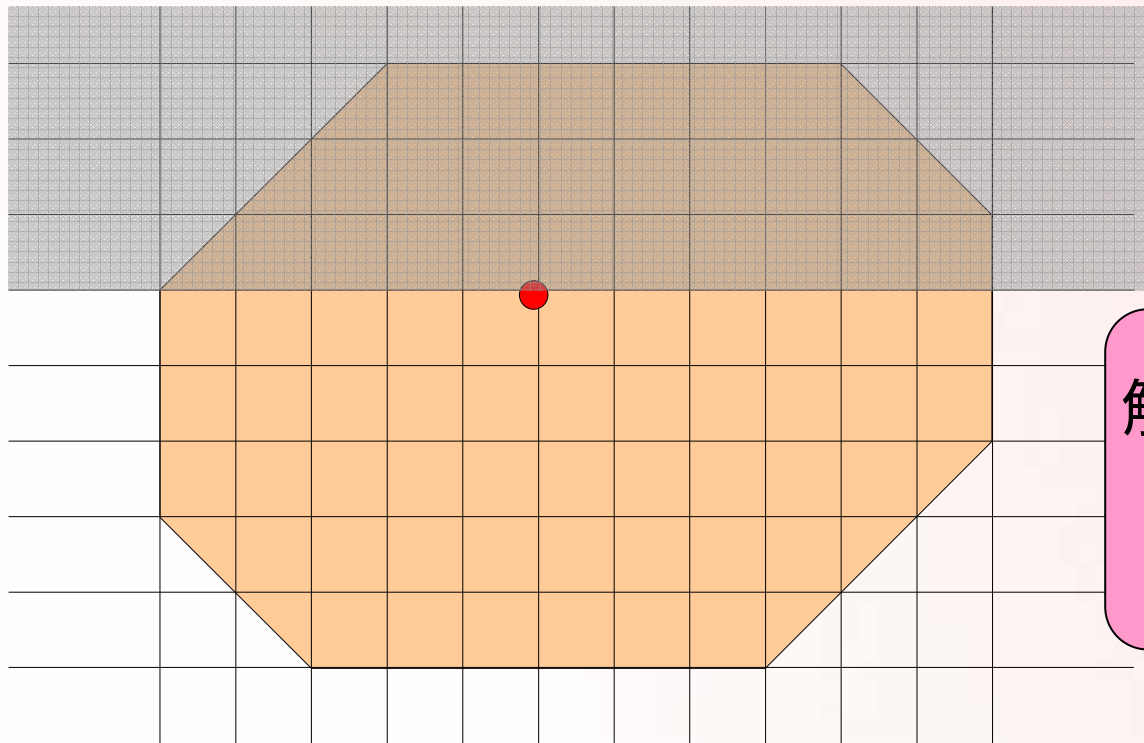


ジャンプ
システムの
中心部

領域縮小法の流れ

- 与えられたベクトル x を最適解から分離
+ x の選び方の工夫 → 多項式時間アルゴリズム

x の選び方: 解候補の集合の境界から離れた点を選択

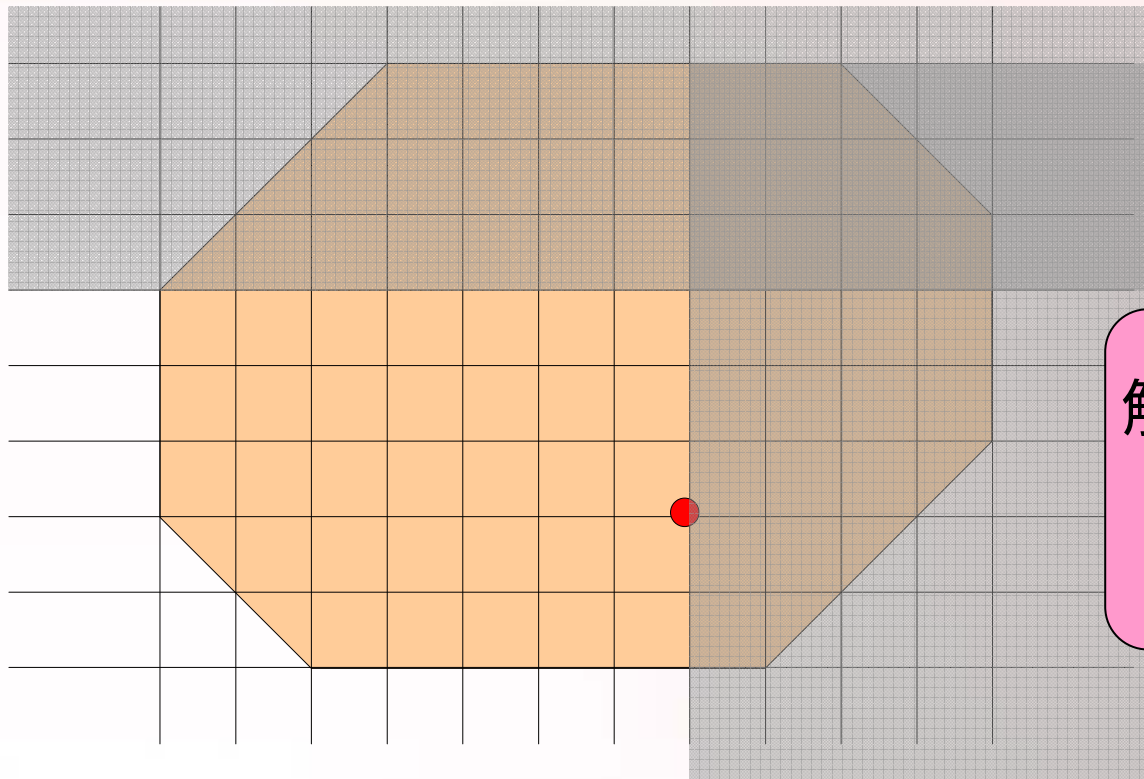


解候補の集合を
繰り返し
小さくしていく

領域縮小法の流れ

- 与えられたベクトル x を最適解から分離
+ x の選び方の工夫 → 多項式時間アルゴリズム

x の選び方: 解候補の集合の境界から離れた点を選択

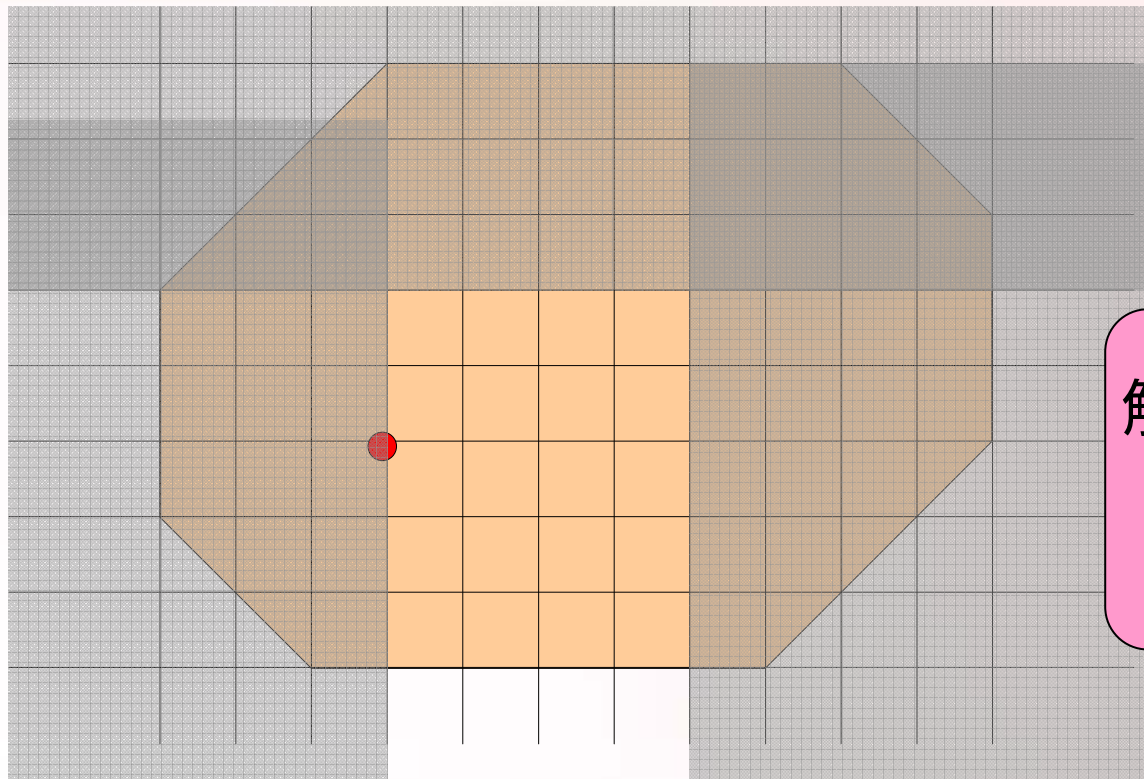


解候補の集合を
繰り返し
小さくしていく

領域縮小法の流れ

- 与えられたベクトル x を最適解から分離
+ x の選び方の工夫 → 多項式時間アルゴリズム

x の選び方: 解候補の集合の境界から離れた点を選択

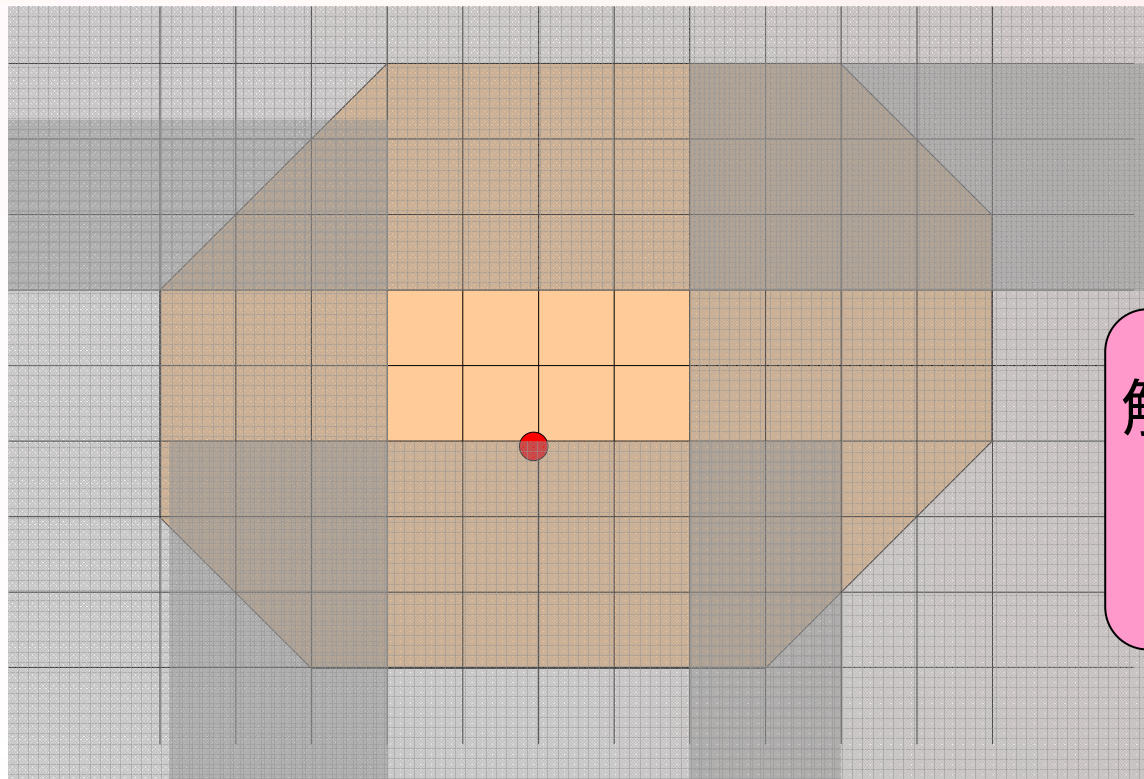


解候補の集合を
繰り返し
小さくしていく

領域縮小法の流れ

- 与えられたベクトル x を最適解から分離
+ x の選び方の工夫 → 多項式時間アルゴリズム

x の選び方: 解候補の集合の境界から離れた点を選択



解候補の集合を
繰り返し
小さくしていく

ジャンプシステムの中心部の決め方

検討すべき事項

- 「中心部」をどのように定義するか？
- 「中心部」は常に非空か？
- 「中心部」に含まれるベクトルを効率的に求めることは可能か？

ジャンプシステムの中心部の定義

S : ジャンプシステム (= 解候補集合)

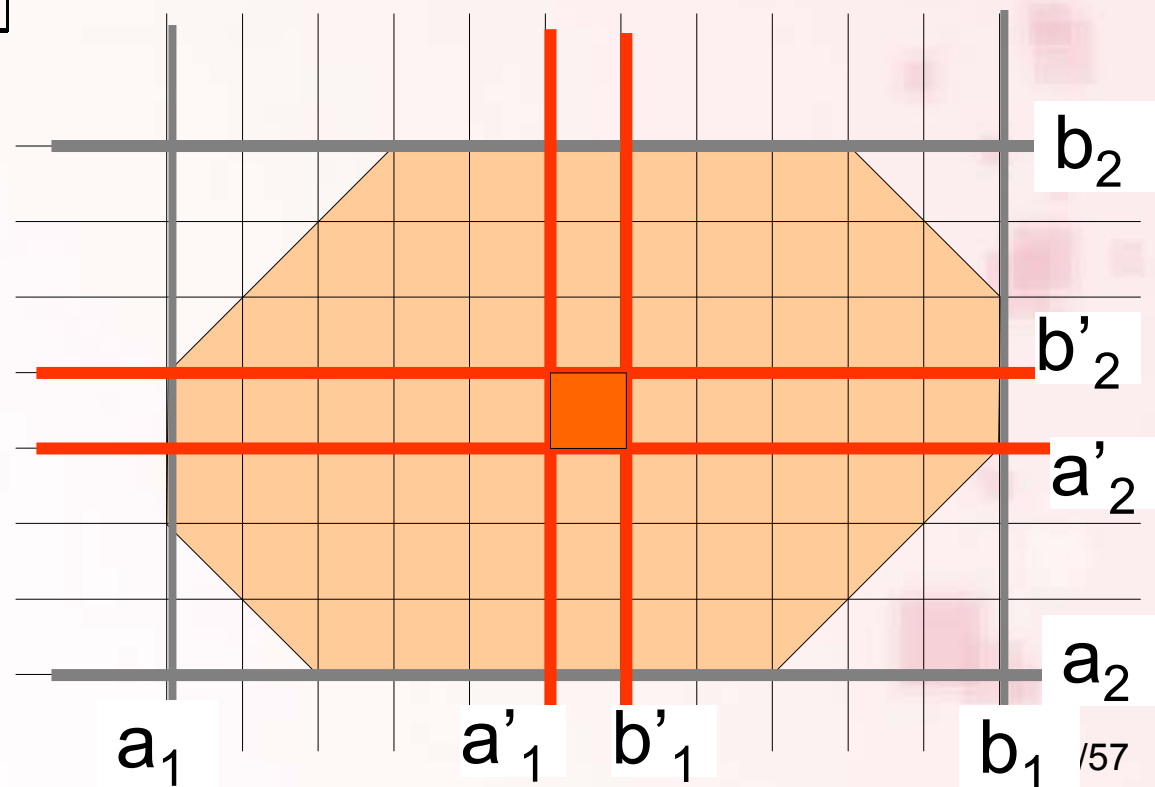
$$a_i = \min_{x \in S} x_i, \quad b_i = \max_{x \in S} x_i$$

$$a'_i = \lfloor a_i + \frac{1}{n}(b_i - a_i) \rfloor$$

$$b'_i = \lceil b_i - \frac{1}{n}(b_i - a_i) \rceil$$

➡ S の中心部
= $S \cap [a', b']$

整数基集合
の「中心部」と
同じ定義
(塩浦1998)



中心部に関する結果

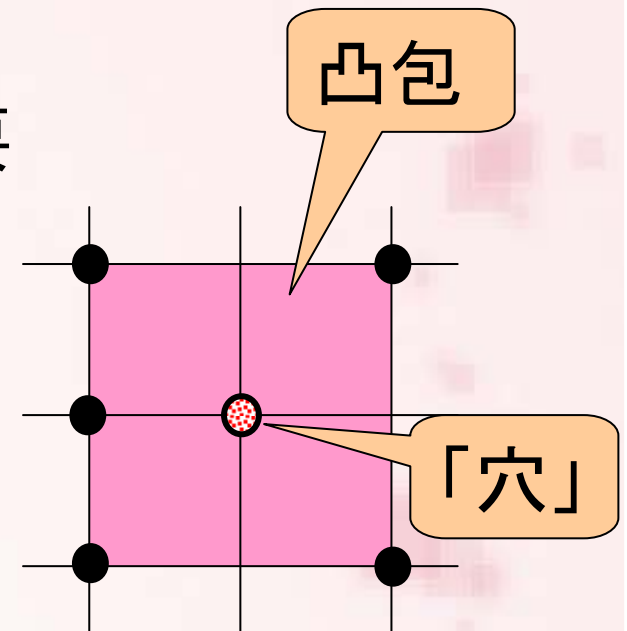
- 領域縮小法の反復回数は $O(n^2 \log L)$
 - 任意のジャンプシステムに対し「中心部」は常に**非空**
 - 「中心部」に含まれるベクトルを $O(n^2 \log L)$ 時間で求められる
- ➡ 領域縮小法の計算時間は $O(n^4 (\log L)^2)$

証明について

- 整数基集合のときの証明 + α が必要
- 証明の難しさ---「穴」の存在

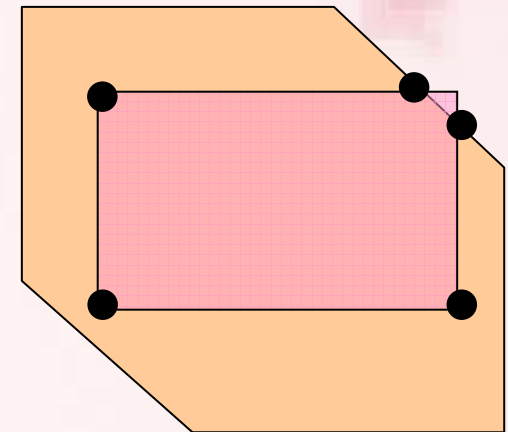
穴---Sの凸包に含まれ、
Sに含まれない格子点

※整数基集合は「穴」をもたない



「中心部」の非空性の証明

- 整数基集合の場合: 凸包の多面体構造を利用
 - 凸包は(整)基多面体
 - 整数値劣モジュラ関数により表現される
- S の凸包と区間 $[a', b']$ の共通部分は非空【劣モジュラ性を使って証明】
- 整基多面体と整数区間の共通部分は整数格子点を含む【既知】
- 整数基集合は「穴」をもたない【既知】



「中心部」の非空性の証明

- ジャンプシステムの場合：
 - 凸包は(整)双劣モジュラ多面体
 - 整数値双劣モジュラ関数により表現される
- S の凸包と区間 $[a', b']$ の共通部分は非空
【双劣モジュラ性を使って証明】
- 整双劣モジュラ多面体と整数区間の共通部分は
整数格子点を含む【既知】
- ~~● ジャンプシステムは「穴」をもたない~~

同様の証明ではうまくいかない...

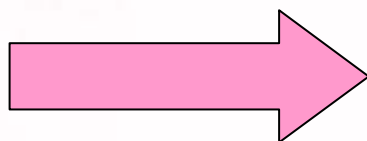
「中心部」の非空性の証明

Lovász(1997)の定理の帰結

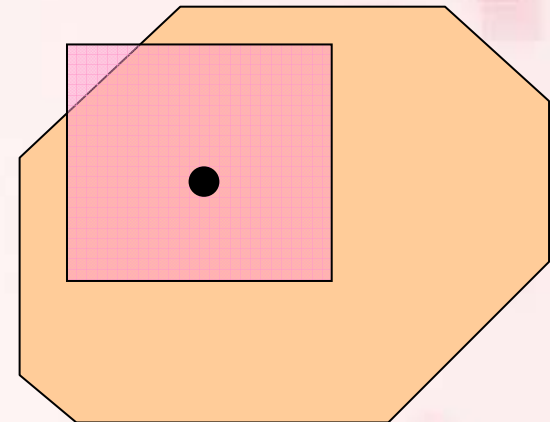
ジャンプシステム S の凸包と整数区間 $[x, y]$ が交わりをもつ
 $x(u) < y(u) \ (u \in V)$

→ S と $[x, y]$ は交わりをもつ

- S の凸包と区間 $[a', b']$ の共通部分は
非空【双劣モジュラ性を使って証明】
- $a'(u) < b'(u) \ (u \in V)$ と仮定して良い



ジャンプシステムの
「中央部」は非空



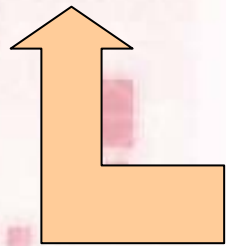
「中心部」の点の求め方

整数基集合の場合の手法を使う:

次の形の問題を繰り返し解く→「中心部」の点を得られる

Max. (Min.) $x(u)$
sub. to. $x \in S'$

(S' は S の部分集合,
ジャンプシステム)



$$\begin{pmatrix} 0 \\ +1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ +1 \\ 0 \\ \pm 1 \\ 0 \end{pmatrix}$$

$u \in V$ に対応

方向の移動可能距離を
繰り返し計算することで解ける

問題点

ジャンプシステムには「穴」がある

→ 移動可能距離の計算を効率的に行えるか？

「中心部」の点の求め方

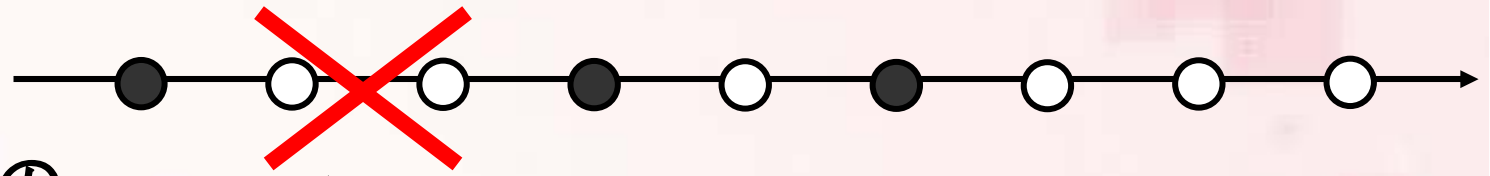
問題点

ジャンプシステムには「穴」がある

→ 移動可能距離の計算を効率的に行えるか？

$$\begin{pmatrix} 0 \\ +1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

方向の
場合

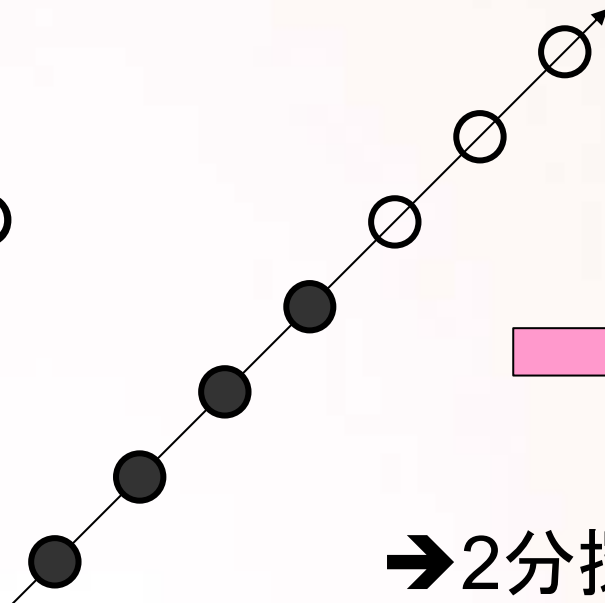


「穴」が2つ続くことはない

→ 2分探索で移動可能距離の計算が可能

$$\begin{pmatrix} 0 \\ +1 \\ 0 \\ \pm 1 \\ 0 \end{pmatrix}$$

方向の
場合



$$\begin{pmatrix} 0 \\ +1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

方向に最大限移動後

$$\begin{pmatrix} 0 \\ +1 \\ 0 \\ \pm 1 \\ 0 \end{pmatrix}$$

方向に「穴」は
存在しない

→ 2分探索で計算が可能

まとめ

- ジャンプシステム上の分離凸関数に対する多項式時間アルゴリズム(領域縮小法)を紹介
- ジャンプシステム上のM凸関数に対しても同様のアルゴリズムが適用可
- 近接定理は成り立つか否か未解決
- 応用は???