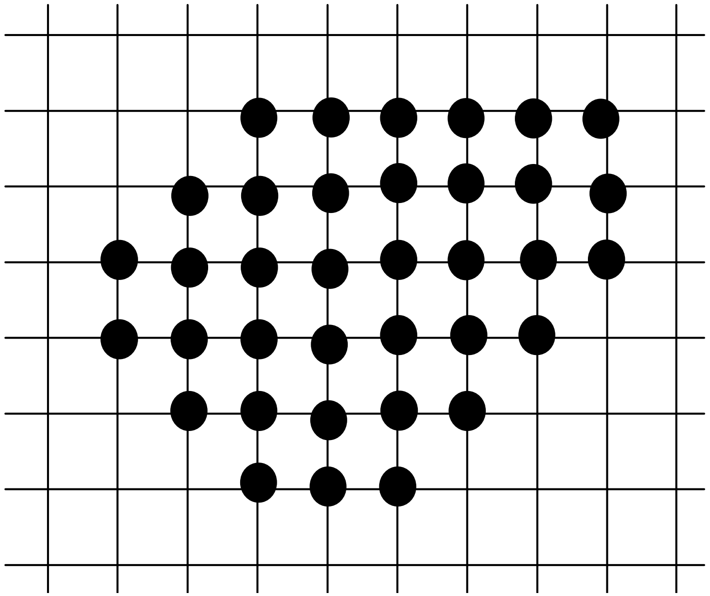
Polynomial-Time Algorithms for Convex Optimization on Jump Systems

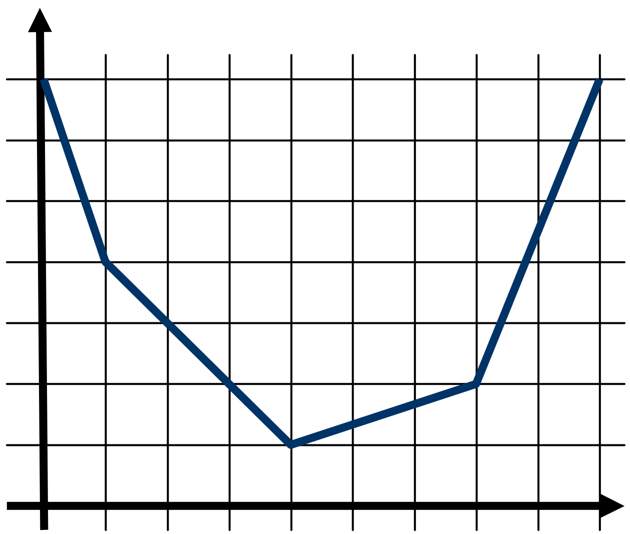
Akiyoshi Shioura
Tohoku University, Japan
(joint work with Ken’ichiro Tanaka)

Discrete Convex Optimization

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

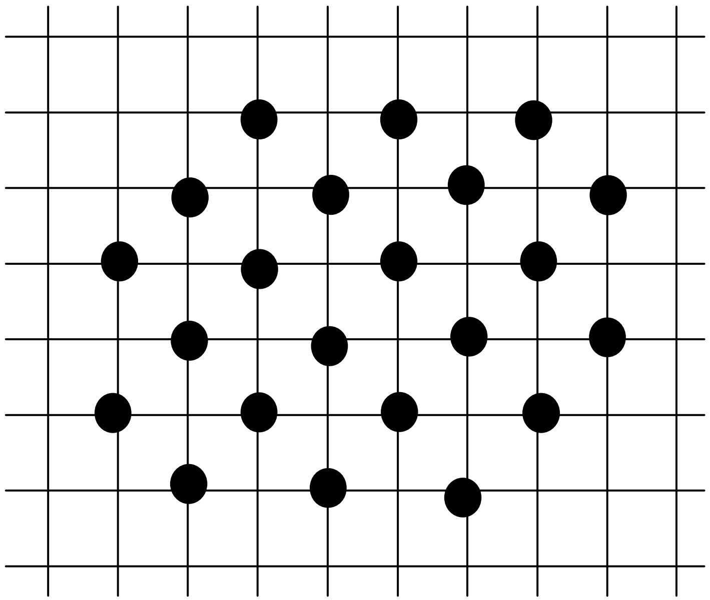
- $S \subseteq \mathbf{Z}^n$, discrete convex set
- $f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$, discrete convex function





Optimization on Jump Systems

- Our problem: Minimization of discrete conv. fn. f(x) on jump system S



(1) f: separable convex function

$$f(x) = \sum_{i=1}^{n} f_i(x_i)$$

(2) f: M-convex function (Murota2006)

Our Results:
first polynomial-time algorithms

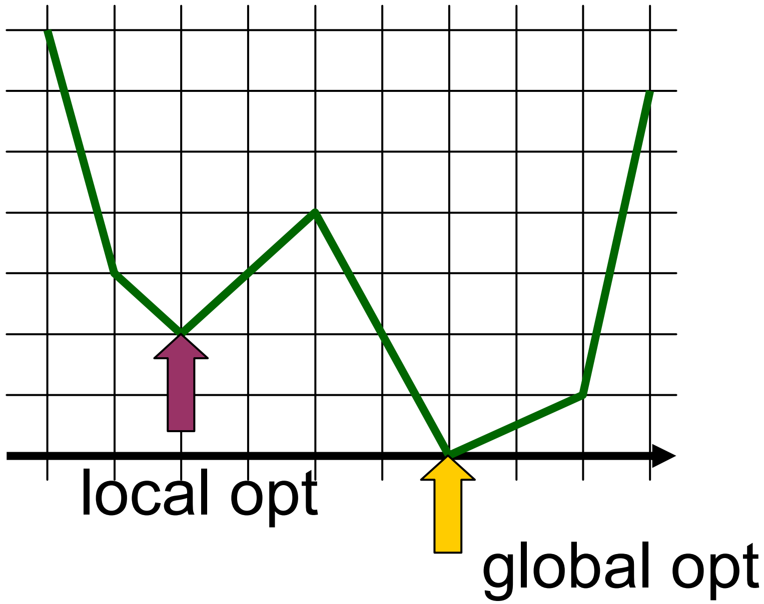
Previous Algorithms

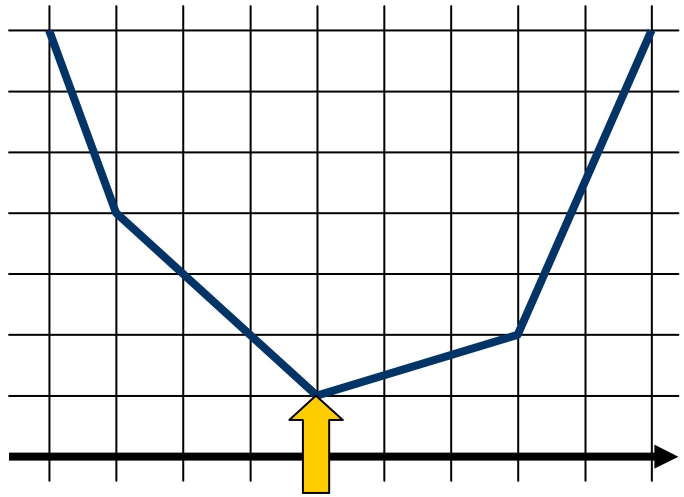
n: dimension, L: "size" of feasible region
$(L = \max\{||x - y||_1 \mid x, y \in \text{feas. region}\})$

- pseudo-polynomial time algorithm (polynomial in n & L)
  - Ando-Fujishige-Naitoh (1995)
    for separable-convex functions on jump systems
  - Murota-Tanaka (2006)
    for M-convex functions on jump systems
- no polynomial time algorithm (polynomial in n & log L)
  was known
- key properties
  - local optimality → global optimality
  - minimizer cut property

Key Properties

- local optimality ➔ global optimality



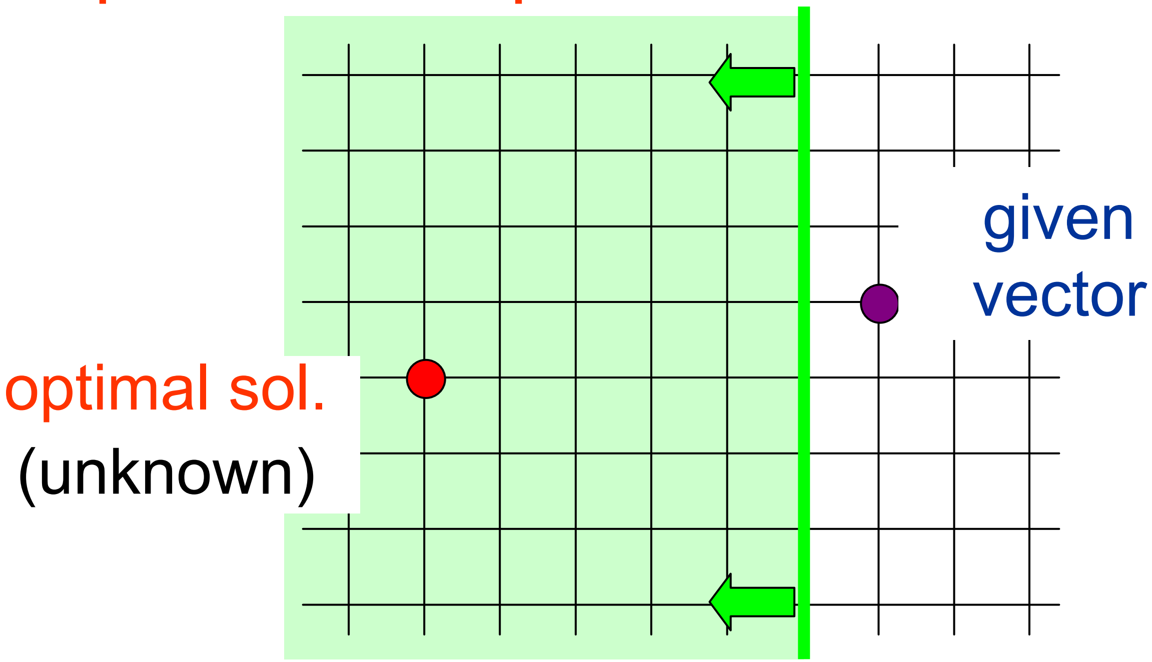


local opt = global opt

Key Properties

- minimizer cut property

--- separation of optimal solution from given vector



Outline of This Talk

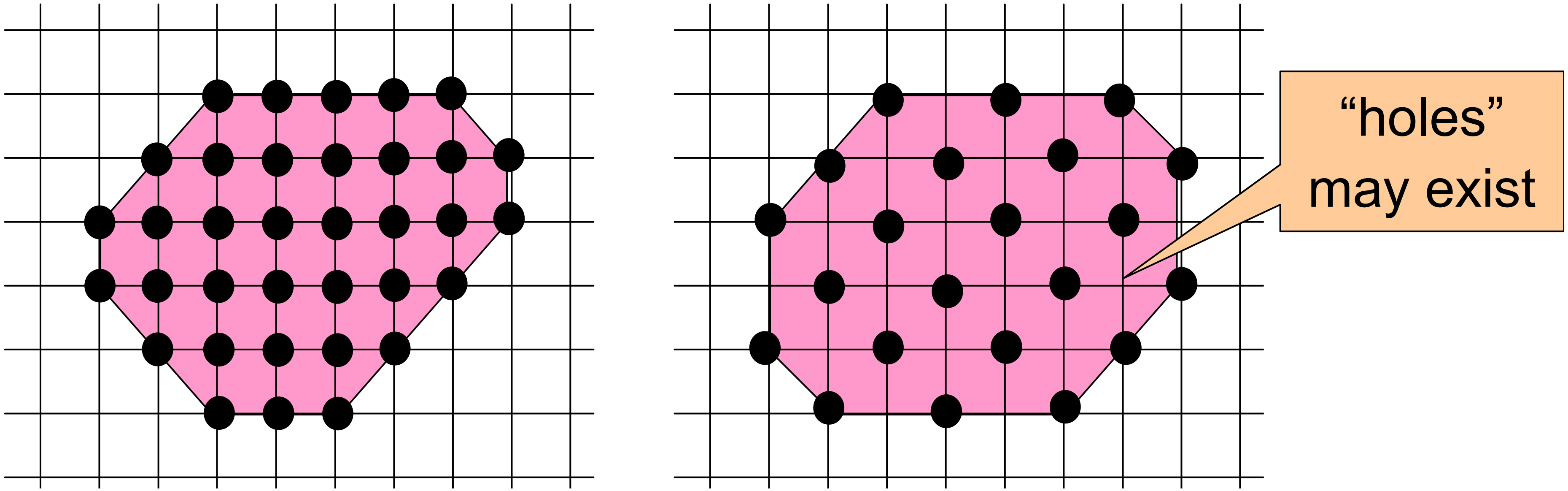
- Jump systems
- Key properties & greedy algorithm
- Polynomial time algorithm

Outline of This Talk

- Jump systems
  - an example: degree sequences of graphs
  - definition
- Key properties & greedy algorithm
- Polynomial time algorithm

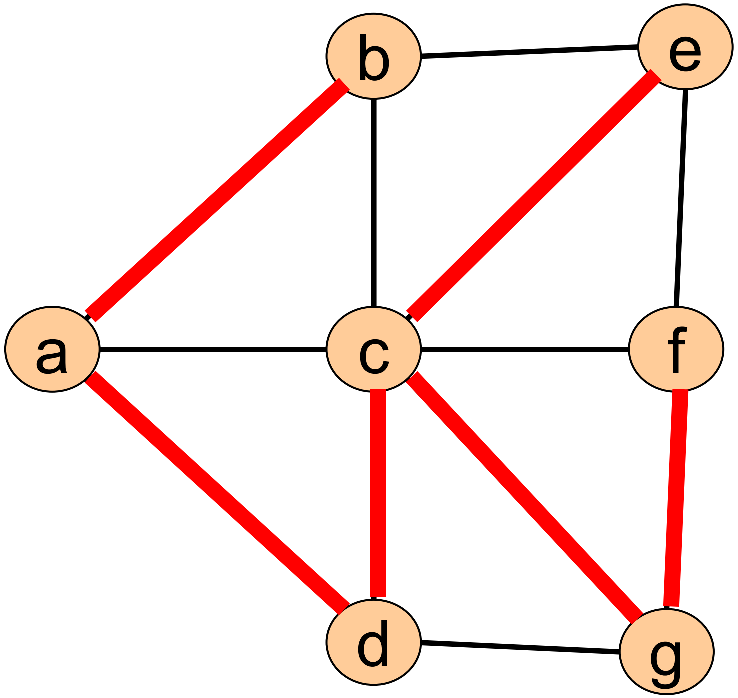
Jump System

- introduced by Bouchet-Cunningham (1995)
- set of integer vectors with nice combi. prop.
- common generalization of matroid, delta-matroid, and base polyhedron
- linear optimization can be solved by greedy algorithm



Example: Degree Sequences of Graphs

G=(V, E)



$\deg_X = (2,1,3,2,1,1,2)$

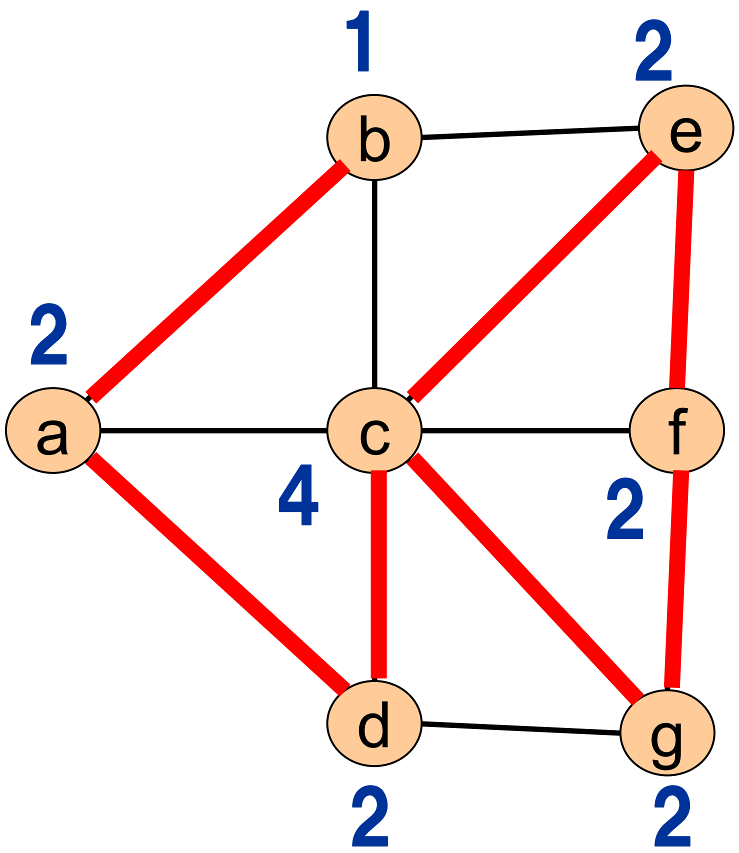
$X \subseteq E$

$\deg_X(v) \equiv$ # of edges in X incident to vertex v∈V

$\{\deg_X \mid X \subseteq E\} \subseteq \mathbf{Z}^V$

⇒ jump system

Example: Degree Sequences of Graphs



- **feasibility problem on degree sequences**

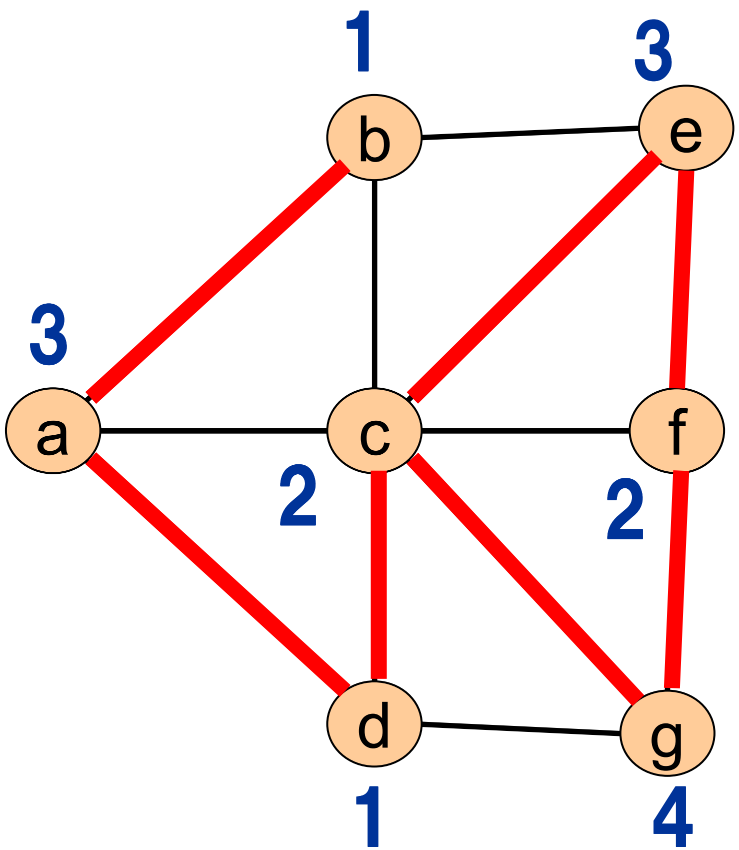
Given: graph G=(V, E), vector b∈$\mathbf{Z}^V$

Find: edge set X⊆E satisfying degree requirement $\deg_X = b$

What if there is no feasible solution?

➔ optimization

Example: Degree Sequences of Graphs



- **optimization problem on degree sequences**

Given: graph G=(V, E), vector $b \in \mathbf{Z}^V$

Find: edge set $X \subseteq E$ minimizing

$$\|\deg_X - b\|_2 = \sum_{v \in V} \{\deg_X(v) - b(v)\}^2$$

⇒ minimization of separable-convex fn on jump system

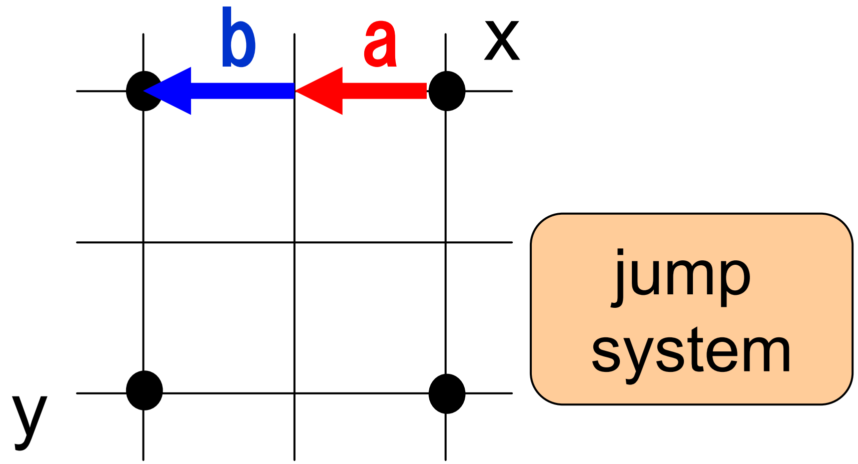
Definition of Jump System

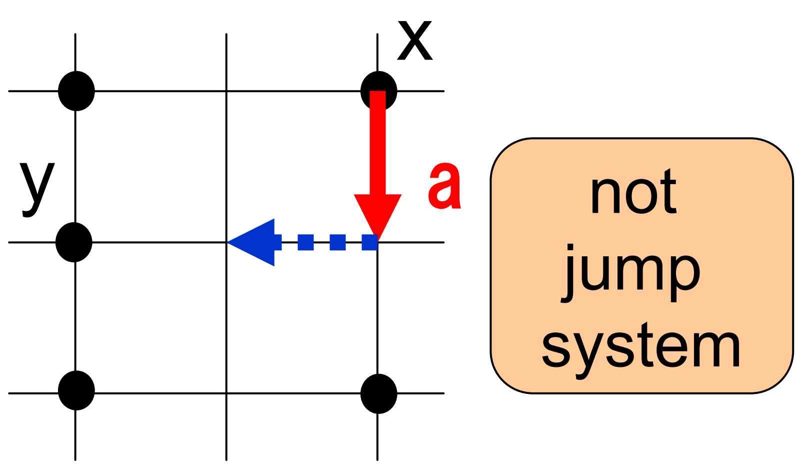
S⊆$\mathbf{Z}^V$: jump system ⟺ 2-step axiom

$\forall x, y \in S,\ \forall a \in \mathrm{St}(x,y),$
$x+a \in S$ or $x+a+b \in S\ (\exists b \in \mathrm{St}(x+a, y))$

St(x, y): set of (x, y)-steps

$$a \in \mathrm{St}(x,y) \iff \begin{cases} a: \text{ unit vector ("step")}, \\ ||(x+a)-y||_1 = ||x-y|| - 1 \end{cases}$$





Outline of This Talk

- Jump systems
- Key properties & greedy algorithm
  - local optimality → global optimality
  - minimizer cut property
  - greedy algorithm
- Polynomial time algorithm

Local Opt → Global Opt

Minimize $\sum_{i=1}^{n} f_i(x_i)$ subject to $x \in S$

jump system

$N(x) = \{y \in \mathbf{Z}^V \mid ||y - x||_1 \leq 2\}$

2-step neighborhood

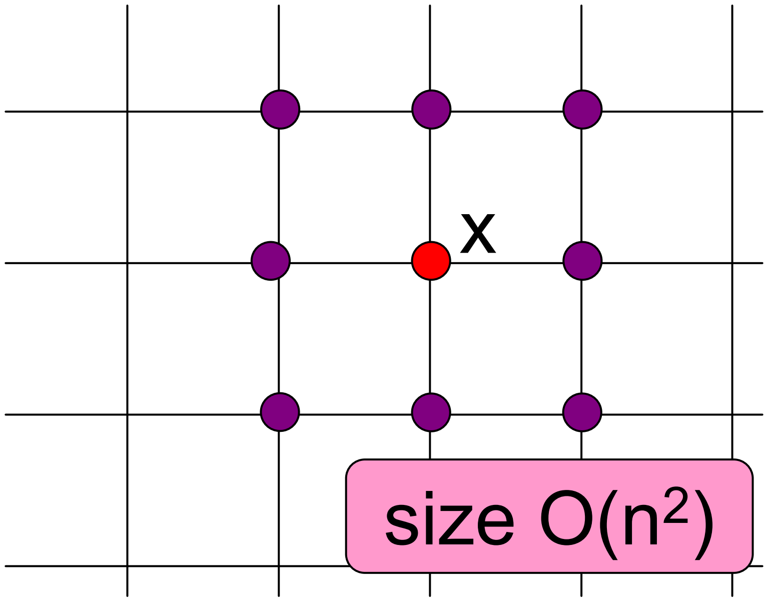
Theorem (Murota 2006)

$x \in S$, minimizer in $N(x)$

(local optimal)

$\Longrightarrow x$: optimal

(global optimal)



Greedy Algorithm

- x: local opt in N(x) ➔ global opt

S0: $x := x_0$

S1: Find a minimizer y in $N(x)$

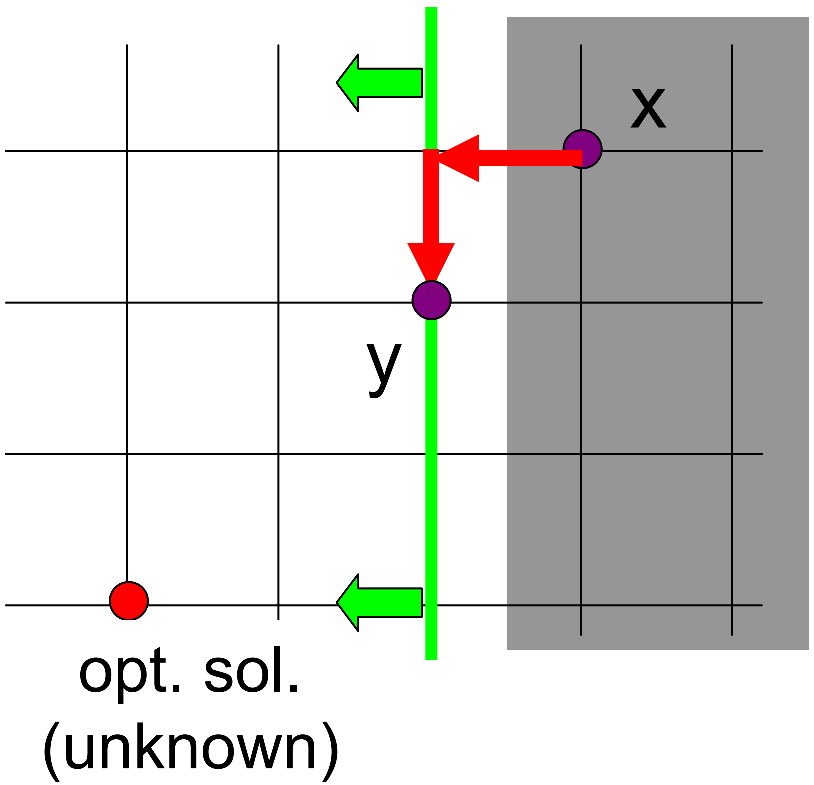
S2: if $y = x \Longrightarrow x$: optimal

S3: $y := x$, go to S1

- f(x) decreases strictly ➔ finite iterations
- exponential time

Minimizer Cut Property

separation of optimal solution from given vector



Theorem

$y \in N(x)$: minimizer in $N(x)$,
$y(i) < x(i)$

⇓

$\exists$ opt. sol. x_* s.t.
$x_*(i) < x(i)$

Improved Analysis of Greedy Algorithm

S0: $x := x_0$
S1: Find a minimizer y in $N(x)$
S2: if $y = x \implies x$: optimal
S3: $y := x$, go to S1

- distance $||x_* - x||$ decreases strictly

$$||x_* - (x + s + t)||_1 \leq ||x_* - x||_1 - 1$$

- O(nL) iterations $\quad (L = \max_{x,y \in S} ||x - y||_1)$

⇒ pseudo-polynomial time

Outline of This Talk

- Jump systems
- Key properties & greedy algorithm
- Polynomial time algorithm
  - Domain reduction algorithm

Minimizer Cut Property and Polynomial-time Algorithm

- Use of Minimizer Cut Property (MCP)

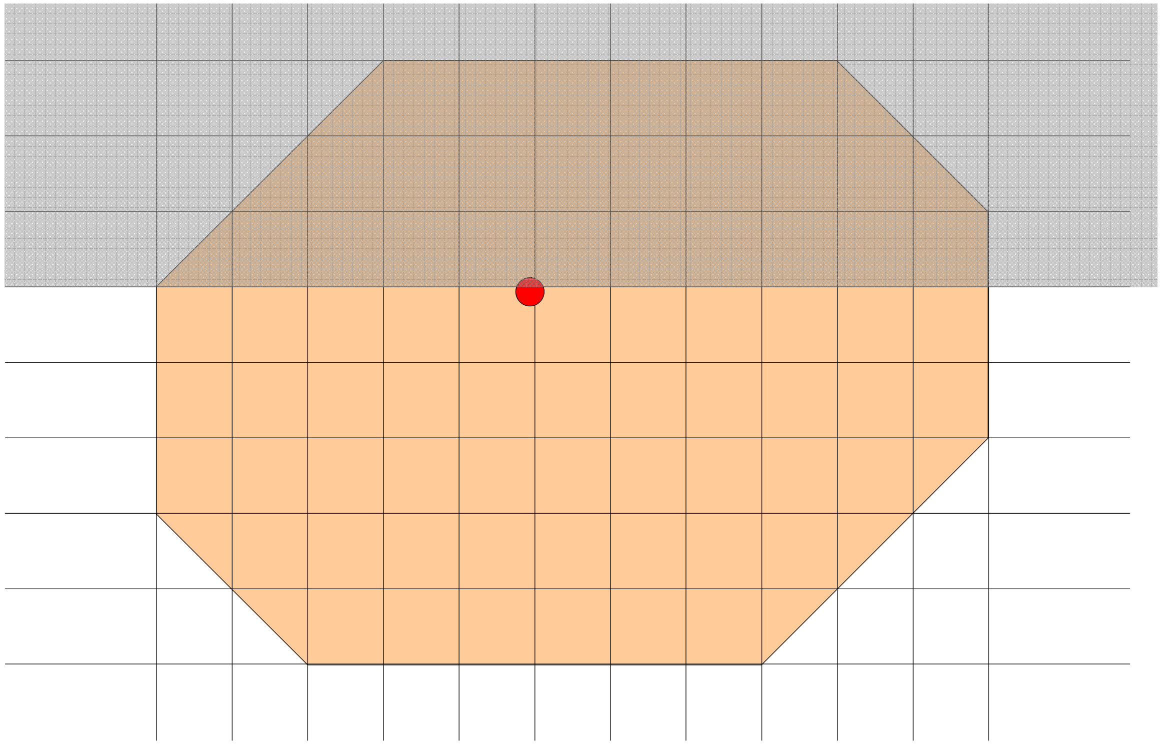
  --- detect the area containing an optimal solution
- apply MCP to appropriately chosen vectors

  ➔ polynomial-time algorithm
  - Domain reduction algorithm

    (Shioura(1998) for M-convex function on base polyhedron)

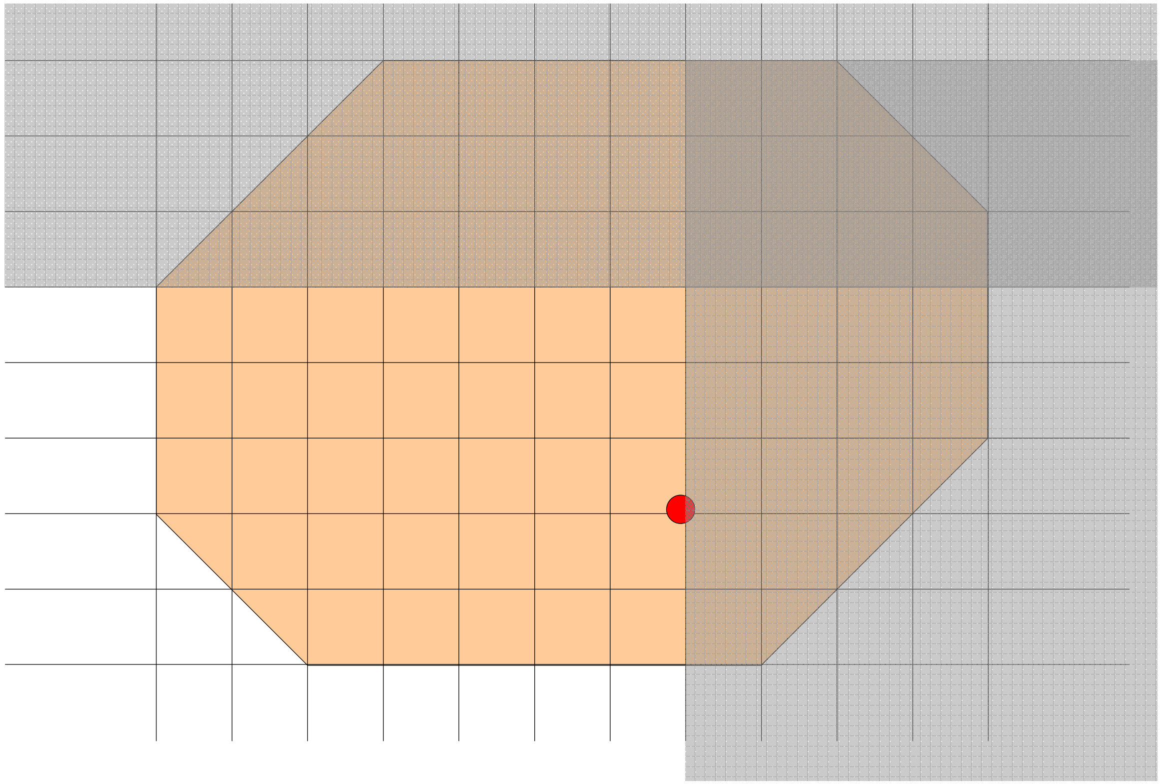
Domain Reduction Algorithm

- Idea: apply MCP to vector lying away from boundary of feasible region



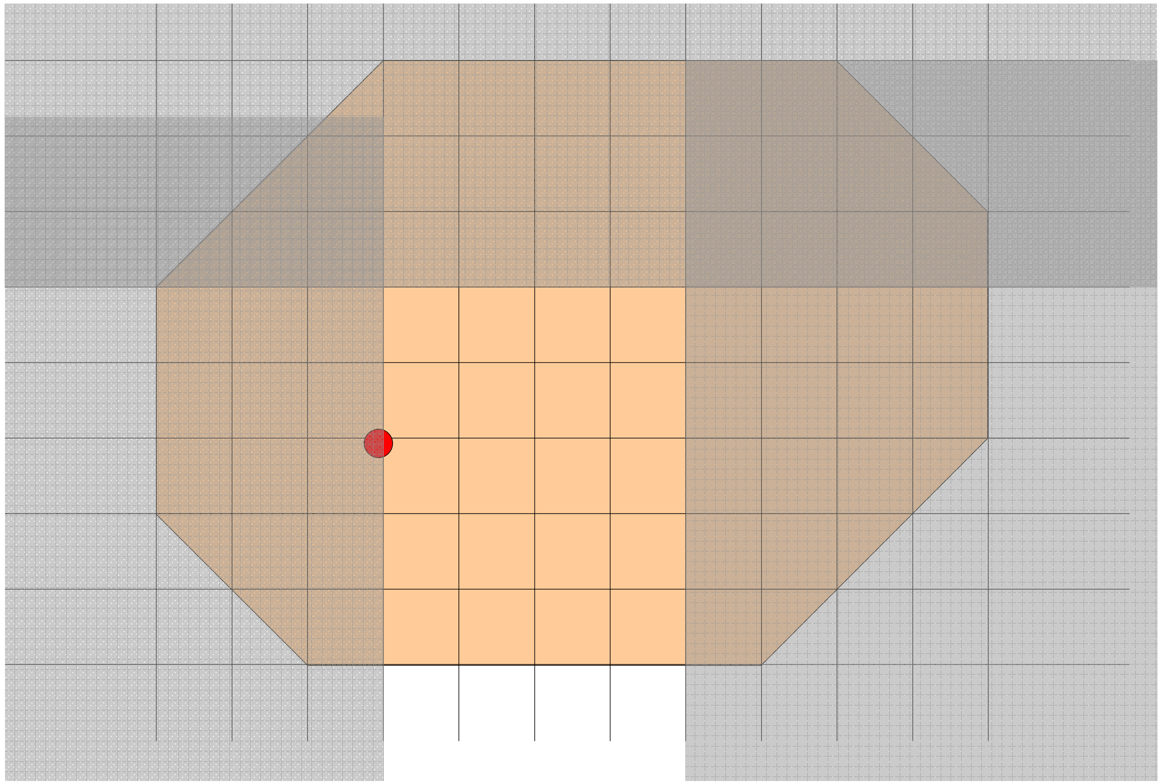
Domain Reduction Algorithm

- Idea: apply MCP to vector lying away from boundary of feasible region



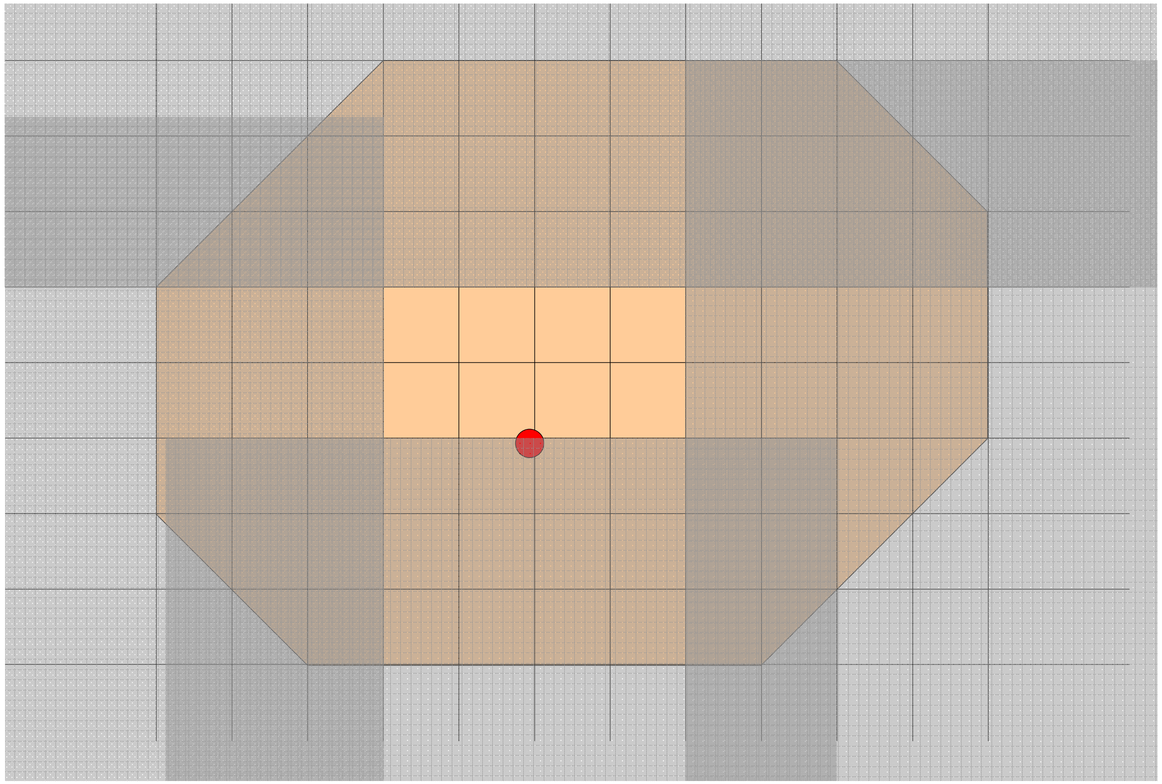
Domain Reduction Algorithm

- Idea: apply MCP to vector lying away from boundary of feasible region



Domain Reduction Algorithm

- Idea: apply MCP to vector lying away from boundary of feasible region



Which Vector to Choose?

Theorem:

S: jump system

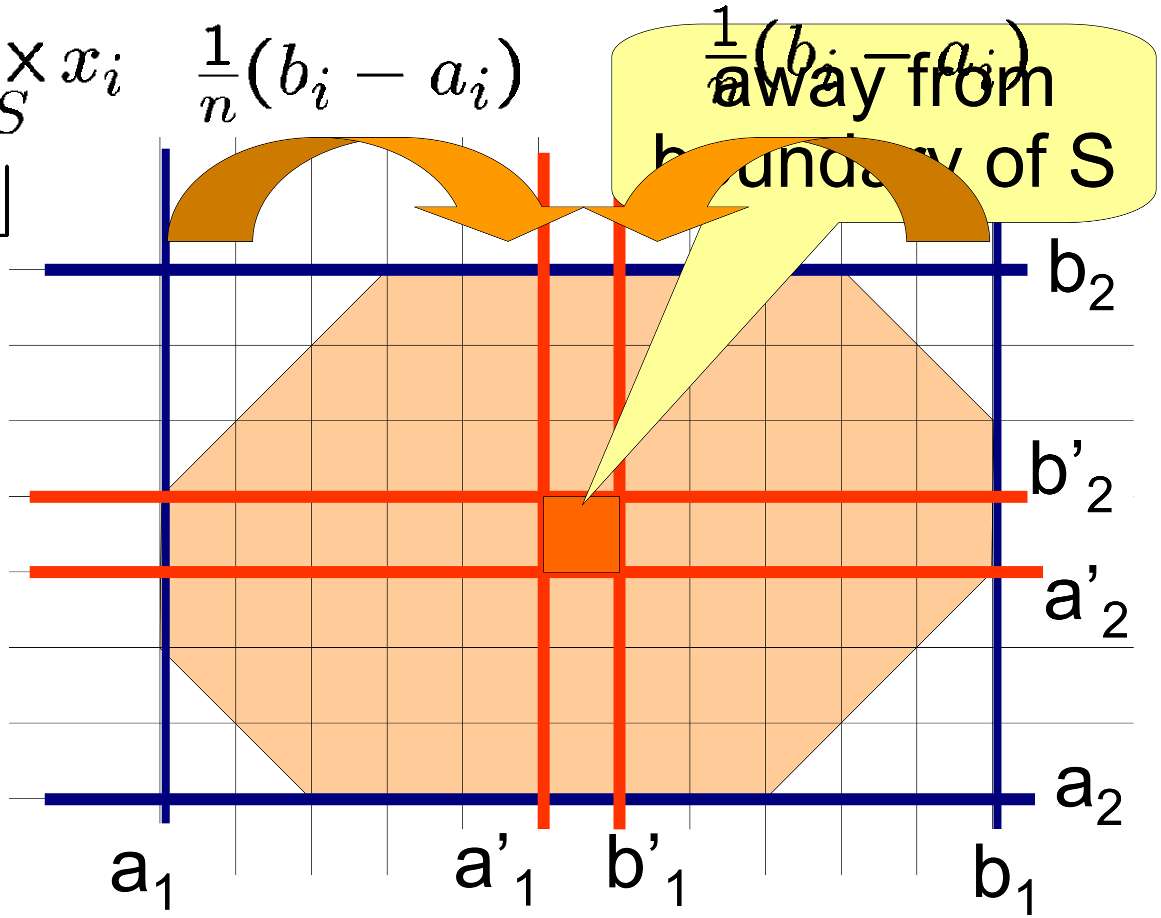
$a_i = \min_{x \in S} x_i,\ b_i = \max_{x \in S} x_i$

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

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

⇒ $S \cap [a', b'] \neq \emptyset$

Choose $x \in S \cap [a', b']$ in each iteration



Domain Reduction Algorithm

S1: Find $x \in S \cap [a', b']$

S2: Apply MCP to find a sep. ...
between x and opt. sol.

S3: Replace S with a smaller set. Go to S1

- nonempty in each iteration
- time complexity: O(n^2 log L)

jump system in each iteration

# of iterations: O(n^2 log L)

time complexity: $\mathrm{O}(n^4(\log L)^2)$

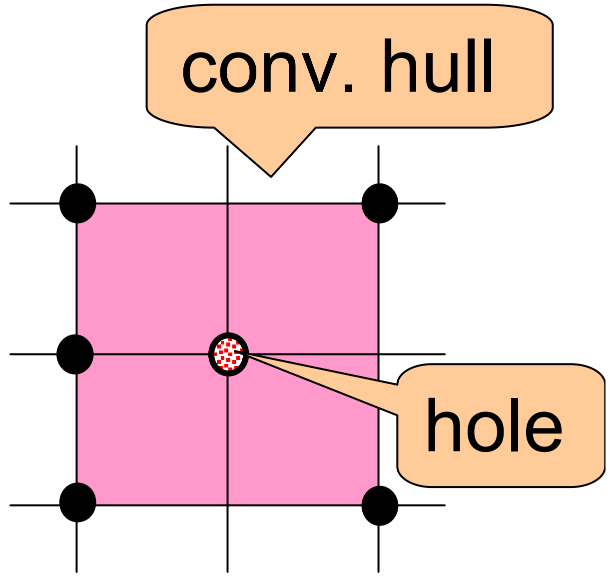
$$L = \max\{||x - y||_1 \mid x, y \in \mathrm{dom}\, f\}$$

Validity of Domain Reduction Algorithm

- originally proposed for M-convex fn. on base polyhedron

➔extended to sep.-conv. fn. on jump system

- difficulty: jump system may contain “holes”
- require new techniques for proofs



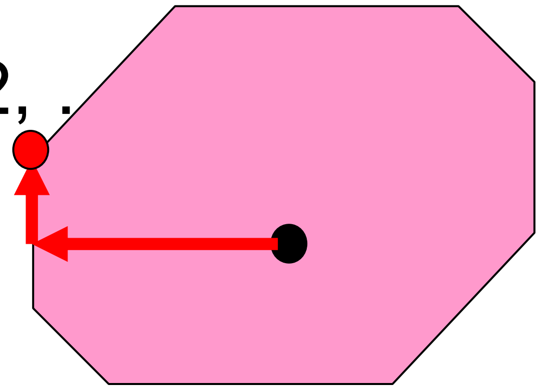
Thank you!

Greedy Algorithm for Linear Optimization

linear optimization can be solved by greedy algorithm

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

1. Assume $|w_1| \geqq |w_2| \geqq \cdots \geqq |w_k| > 0 = |w_{k+1}| = \cdots = |w_n|$
   Put $S_0 := S$
2. Perform the following for each i = 1, 2, ...
   - $w_i > 0 \rightarrow x_i^* := \max(x_i \mid x \in S_{i-1})$
   - $w_i < 0 \rightarrow x_i^* := \min(x_i \mid x \in S_{i-1})$
   - $S_i := \{x \in S_{i-1} \mid x_i = x_i^*\}$



Definition of M-convex Function on Jump System

- f: S→**R** is M-convex⇔

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

s.t. $x+s+t \in S,\ y-s-t \in S,$

and $f(x)+f(y) \geq f(x+s+t)+f(y-s-t)$