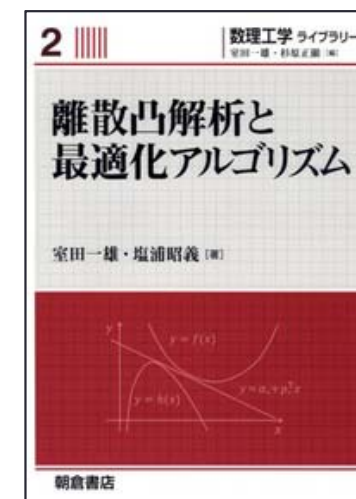


# 競争均衡の計算に対する 反復オークションの計算量解析 ---離散凸解析によるアプローチ---



塩浦昭義

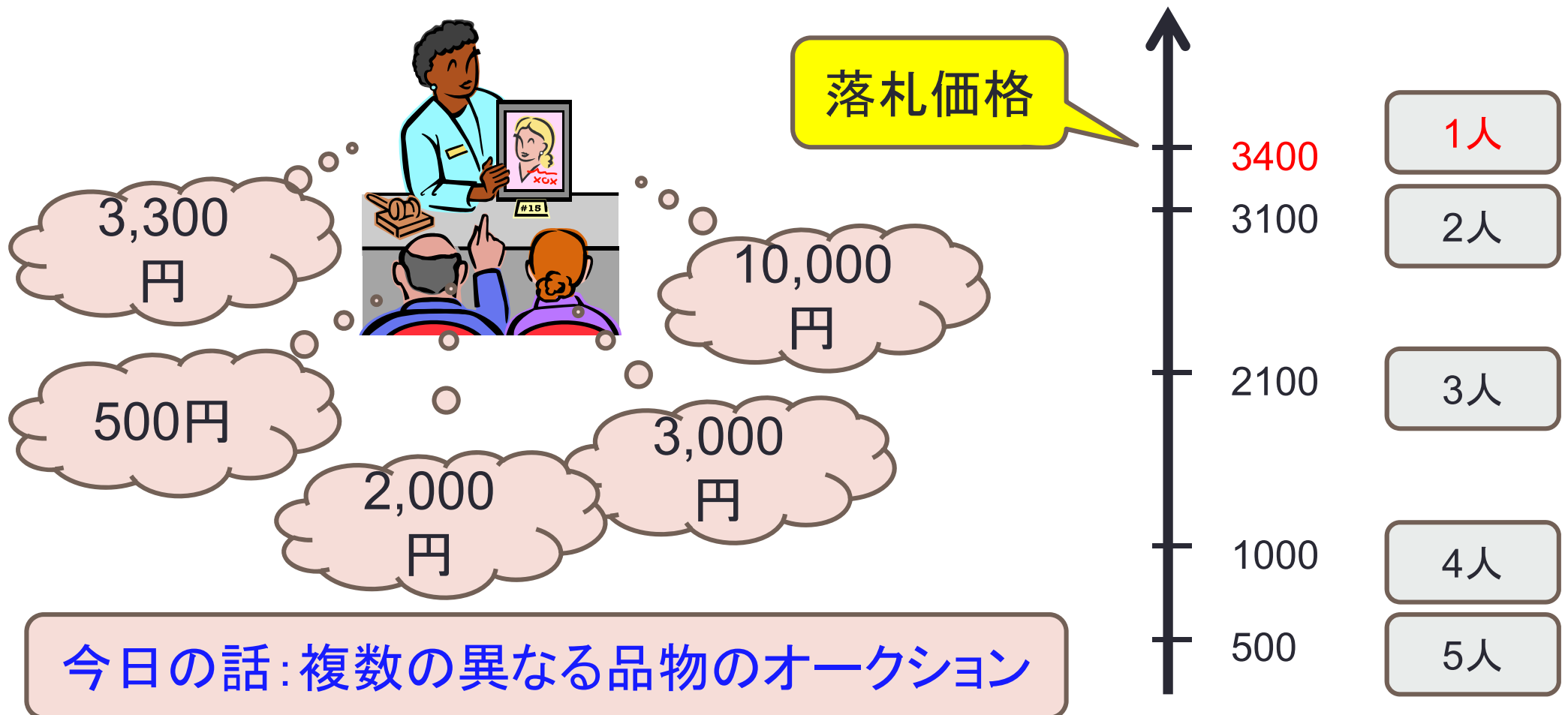
東北大学 大学院 情報科学研究科

2014/03/25 最適化研究集会（政策研究大学院大学）

室田一雄氏（東京大），  
Zaifu Yang 氏（Univ. York, UK）  
との共同研究

# 1つの財のオークション

- Ascending オークション (English オークション)
  - 価格を徐々に(1単位ずつ)上げる
  - 購入可能なオークション参加者が1人になったら終了



# 発表の概要

- Ausubel (2006)の複数財に対する Ascending Auction を  
離散凸解析の視点から理解
  - 効用関数 の 粗代替性 =  $M^{\sharp}$ 凹性
  - 間接効用関数 =  $L^{\sharp}$ 凸関数 (共役性)
  - Lyapunov 関数 の最小化 =  $L^{\sharp}$ 凸関数 の最小化
  - オークション =  $L^{\sharp}$ 凸関数最小化アルゴリズム

## 主結果:

- 種々の型のオークションの提案  
(English, Dutch, English-Dutch, Greedy)
- オークションの反復回数解析

# オークションのモデル

- 商品の集合  $N = \{1, \dots, n\}$ 
  - 各商品の個数は1と限定
- $m$  人の参加者(bidder)で商品を分け合う
- 参加者  $i = 1, \dots, m$  の効用関数  $f_i: 2^N \rightarrow \mathbb{Z}_+$



①



②



③



④



⑤

$f_i(X)$  = 商品セット  $X$   
の満足度  
(単位: お金)

# 効用関数の例



- ①を含む商品セットは**100**, それ以外は**0** (single-minded)
- 重み和(①:**50**, ②:**70**, ③:**40**, ④:**30**, ⑤:**100**) (additive)
- 商品セット(①:**50**, ②:**70**, ③:**40**, ④:**30**, ⑤:**100**) 中の一番良い商品にのみ依存 (unit-demand)
  - {①, ②, ③} → 満足度**70**, {③, ④, ⑤} → 満足度**100**
- 商品数依存 (symmetric & concave)
  - (1つ:**100**, 2つ:**180**, 3つ:**240**, 4つ:**280**, 5つ:**300**)

# 間接効用関数と需要集合



- 財の価格ベクトル  $p \in \mathbb{R}_+^n$
- 定義: 間接効用関数  $V_i: \mathbb{R}_+^n \rightarrow \mathbb{R}$ 
  - $V_i(p) = \max\{f_i(X) - \sum_{j \in X} p(j) \mid X \subseteq N\}$
- 定義: 需要集合  $D_i(p) \subseteq 2^N$ 
  - $D_i(p) = \arg \max\{f_i(X) - \sum_{j \in X} p(j) \mid X \subseteq N\}$

価格  $p$  の下で  
最も欲しい  
商品セット全体

例:  $p = (60, 60, 60, 60, 60)$  のとき

- Aさん: ①を含む商品セットは100, それ以外は0  
→  $V_A(p) = 40, D_A(p) = \{ \{1\} \}$
- Bさん: 重み和 (①: 50, ②: 70, ③: 40, ④: 30, ⑤: 100)  
→  $V_B(p) = 50, D_B(p) = \{ \{2, 5\} \}$
- Cさん: 商品数依存 (1つ: 100, 2つ: 180, 3つ: 240, 4つ: 280, 5つ: 300)  
→  $V_C(p) = 60, D_C(p) = \{ \text{商品数2または3} \}$

# 競争均衡

- 財の価格ベクトル  $p \in \mathbb{R}_+^n$
- 定義: 需要集合  $D_i(p) \subseteq 2^N$ 
  - $D_i(p) = \arg \max \{f_i(X) - \sum_{j \in X} p(j) \mid X \subseteq N\}$
- 定義: 価格ベクトル  $p^*$  と  $N$  の分割  $(X_1, \dots, X_m)$  の組は競争均衡
  - $\leftrightarrow X_i \in D_i(p^*) \ (i = 1, \dots, m)$

価格  $p$  の下で  
最も欲しい  
商品セット全体

$p^*$  は均衡価格

価格  $p^*$  の下で  
皆が最良の  
商品セット

例:  $p = (60, 60, 60, 60, 60)$  のとき

- Aさん:  $D_A(p) = \{ \{1\} \}$
- Bさん:  $D_B(p) = \{ \{2, 5\} \}$
- Cさん:  $D_C(p) = \{ \text{商品数2または3} \}$

$\rightarrow p$  と分割  $\{ \{1\}, \{2, 5\}, \{3, 4\} \}$  は競争均衡

# 効用関数の粗代替性

粗代替性  $\equiv$  iPad が値上がりしたら、他の商品の欲しさが高まる



- 定義: 効用関数  $f_i: 2^N \rightarrow \mathbb{Z}_+$  は  
粗代替性(gross-substitutes property)を満たす

$$\begin{aligned} \iff \forall p \in \mathbb{R}^N, q = p + \lambda e_j, \\ \forall X \in D_i(p), \exists Y \in D_i(q) \\ \text{s.t. } X \setminus \{j\} \subseteq Y \end{aligned}$$

$$D_i(p) \equiv \arg \max \{f_i(S) - \sum_{i \in S} p_i \mid S \subseteq N\}$$



# 粗代替性を満たす効用関数の例



- ①を含む商品セットは100, それ以外は0 (single-minded)
- 重み和(①:50, ②:70, ③:40, ④:30, ⑤:100) (additive)
- 商品セット(①:50, ②:70, ③:40, ④:30, ⑤:100) の中の一番良い商品にのみ依存 (unit-demand)  
 {①, ②, ③} → 満足度70, {③, ④, ⑤} → 満足度100
- 商品数依存 (symmetric & concave)  
 (1つ:100, 2つ:180, 3つ:240, 4つ:280, 5つ:300)

# 効用関数の粗代替性

粗代替性  $\equiv$  iPad が値上がりしたら、他の商品の欲しさが高まる

- 定義: 効用関数  $f_i: 2^N \rightarrow \mathbb{Z}_+$  は

粗代替性(gross-substitutes property)を満たす

$$\begin{aligned} \iff \forall p \in \mathbb{R}^N, q = p + \lambda e_j, \\ \forall X \in D_i(p), \exists Y \in D_i(q) \\ \text{s.t. } X \setminus \{j\} \subseteq Y \end{aligned}$$

$$D_i(p) \equiv \arg \max \{f_i(S) - \sum_{i \in S} p_i \mid S \subseteq N\}$$

定理[Kelso-Crawford(1982), et al.]:

$f_i$  は粗代替性を満たす  $\rightarrow$  競争均衡が存在

定理[藤重-Yang03]:

$f_i$  は粗代替性を満たす  $\iff f_i$  はM凹関数  $\rightarrow$  離散凸解析

# 反復オークション (Iterative Auction)

- 均衡価格 (および配分) を求めるためのプロトコル (アルゴリズム)
  - オークション主催者は繰り返し価格を更新
  - 参加者は要求された情報を提示.
  - ただし, 効用関数の情報は陽に公開しない

Step 0. initial prices  $p = (p_1, \dots, p_n)$

Step 1. bidder reports **all desired item sets in  $D_i(p)$**

Step 2. If  $\exists$  partition  $(S_1^*, S_2^*, \dots, S_m^*)$  s.t.  $S_i^* \in D_i(p)$   
 $\rightarrow$  stop ( $p$  is equilibrium prices)

Step 3. update  $p$ . Go to Step 1.

**ascending auction:**  $p$  が単調非減少

需要大  $\rightarrow$  価格を上げる  
 需要小  $\rightarrow$  価格を下げる

# Lyapunov 関数 (Ausubel 2006)

$$L(p) = p(N) + \sum_i V_i(p)$$

$$V_i(p) = \max\{f_i(X) - p(X) \mid X \subseteq N\}$$

- 性質: (i)  $L$  は劣モジュラ  $L(p) + L(q) \geq L(p \vee q) + L(p \wedge q)$   
 (ii)  $p$  は  $L$  の最小解  $\iff p$ : 均衡価格  
 (iii)  $L$  は整数最小解をもつ

$p$  は  $L$  の最小解  $\iff 0$  は  $L$  の劣微分に含まれる

$$L \text{ の劣微分} = \mathbf{1} + \sum_i (V_i(p) \text{ の劣勾配})$$

$$= \mathbf{1} - \sum_i D_i(p)$$

$$\iff \exists X_i \in D_i(p^*) \quad (i = 1, \dots, m)$$

供給と  
需要の差

# Lyapunov 関数 (Ausubel 2006)

$$L(p) = p(N) + \sum_i V_i(p)$$

$$V_i(p) = \max\{f_i(X) - p(X) \mid X \subseteq N\}$$

- 性質: (i)  $L$  は劣モジュラ  $L(p) + L(q) \geq L(p \vee q) + L(p \wedge q)$   
 (ii)  $p$  は  $L$  の最小解  $\iff p$ : 均衡価格  
 (iii)  $L$  は整数最小解をもつ

## Ascending Auction

Step 0:  $p :=$  sufficiently small integral vector (e.g.,  $\mathbf{0}$ )

Step 1: find  $X \subseteq N$  minimizing  $L(p + e_X)$  using  $D_i(p)$

Step 2:  $L(p + e_X) \geq L(p) \rightarrow$  stop ( $p$  is equilibrium)

Step 3:  $p := p + e_X$ , Go to Step 1

$$e_X = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

# 離散凸解析 (Murota 1996)

2種類の離散凸性:

$M_{\square}$ -convex /  $L_{\square}$ -convex

離散ルジヤンドル変換による共役性

# 発表の概要

- Ausubel (2006)の複数財に対する Ascending Auction を  
離散凸解析の視点から理解
  - 効用関数 の 粗代替性 =  $M^{\sharp}$ 凹性
  - 間接効用関数 =  $L^{\sharp}$ 凸関数 (共役性)
  - Lyapunov 関数 の最小化 =  $L^{\sharp}$ 凸関数 の最小化
  - オークション =  $L^{\sharp}$ 凸関数最小化アルゴリズム

## 主結果:

- 種々の型のオークションの提案  
(English, Dutch, English-Dutch, Greedy)
- オークションの反復回数解析

# $M^{\natural}$ -convex Function (Exchange Axiom)

$M^{\natural}$ -convex fn:

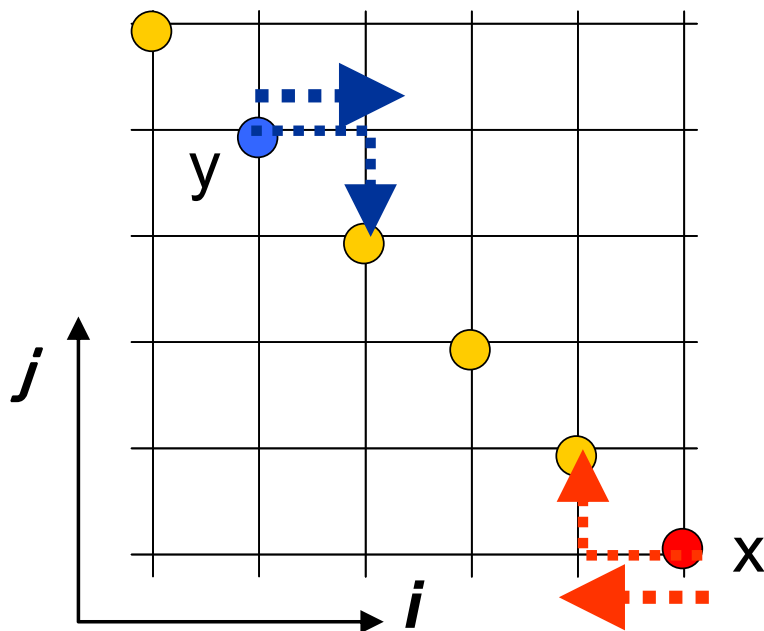
(Murota-Shioura 99)

**Def:**  $f: \mathbb{Z}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  is  $M^{\natural}$ -convex  $\iff$

$\forall x, y \in \mathbb{Z}^n, \forall i: x(i) > y(i):$

(i)  $f(x) + f(y) \geq f(x - \chi_i) + f(y + \chi_i)$ , or

(ii)  $\exists j: x(j) < y(j)$  s.t.  $f(x) + f(y) \geq f(x - \chi_i + \chi_j) + f(y + \chi_i - \chi_j)$



variant of M-convex (Murota 96)



# $L_q$ -convex Function

**Prop:** Continuous fn  $g: \mathbb{R}^n \rightarrow \mathbb{R}$  is **convex**

**$\leftrightarrow$  mid-point convex:**  $\forall p, q \in \mathbb{R}^n, g(p) + g(q) \geq 2g\left(\frac{p+q}{2}\right)$

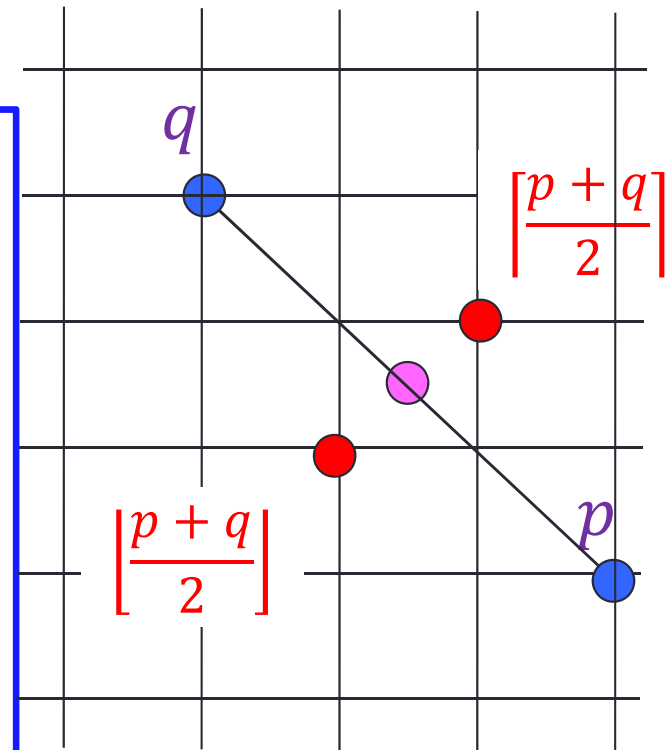
**Def:**  $g: \mathbb{Z}^n \rightarrow \mathbb{R}$  is  **$L_q$ -convex**

**$\leftrightarrow$  discrete mid-point convex:**

$\forall p, q \in \mathbb{Z}^n,$

$$g(p) + g(q)$$

$$\geq g\left(\left\lceil \frac{p+q}{2} \right\rceil\right) + g\left(\left\lfloor \frac{p+q}{2} \right\rfloor\right)$$



# 離散凸解析によるAscending Auctionの解析

定理[藤重-Yang03]:

$f_i$  は粗代替性を満たす  $\leftrightarrow$   $f_i$  はM凹関数



共役性定理

定理: Lyapunov 関数 --- L凹-convex

$$L(p) = p(N) + \sum_i V_i(p)$$

Ascending Auction --- L凹-convex 関数最小化の最急降下法

- 反復回数の既存のバウンド: Murota (2003), Kolmogorov-Shioura (2009)
- 解析の改善  $\rightarrow$  最急降下法の厳密なバウンド

定理:  $\min \left\{ \|p^* - p_0\|_\infty \mid p^*: \text{integer equil.}, p^* \geq p_0 \right\}$

$\rightarrow$  trajectory of  $p$  --- 均衡への「最短路」

# Lyapunov 関数 (Ausubel 2006)

$$L(p) = p(N) + \sum_i V_i(p)$$

$$V_i(p) = \max\{f_i(X) - p(X) \mid X \subseteq N\}$$

- 性質: (i)  $L$  は劣モジュラ  $L(p) + L(q) \geq L(p \vee q) + L(p \wedge q)$   
 (ii)  $p$  は  $L$  の最小解  $\iff p$ : 均衡価格  
 (iii)  $L$  は整数最小解をもつ

## Ascending Auction

Step 0:  $p :=$  sufficiently small integral vector (e.g.,  $\mathbf{0}$ )

Step 1: find  $X \subseteq N$  minimizing  $L(p + e_X)$  using  $D_i(p)$

Step 2:  $L(p + e_X) \geq L(p) \rightarrow$  stop ( $p$  is equilibrium)

Step 3:  $p := p + e_X$ , Go to Step 1

$$e_X = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

# その他のオークションアルゴリズム

- Ascending Auction (English)
- Descending Auction (Dutch)
- Greedy Auction
- Two-Phase Auction (English-Dutch)

# Greedy Auction

Step 0:  $p :=$  **any** integral vector

Step 1: find  $\sigma \in \{+1, -1\}$  and  $X \subseteq N$  minimizing  $L(p + \sigma e_X)$

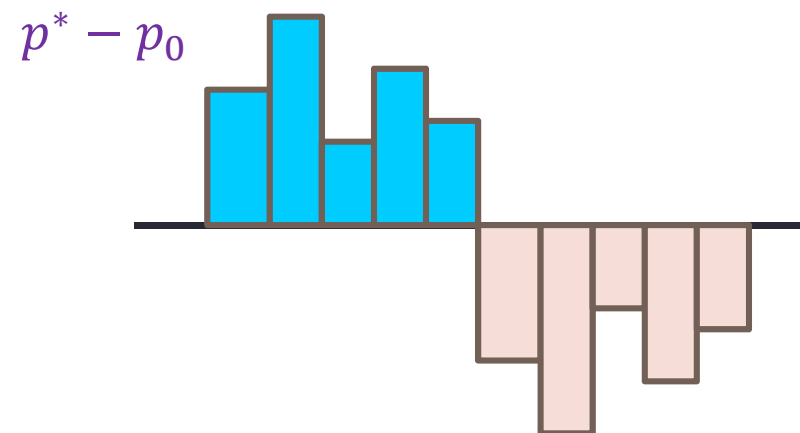
Step 2:  $L(p + \sigma e_X) \geq L(p) \rightarrow$  stop ( $p$  is equilibrium)

Step 3:  $p := p + \sigma e_X$ , Go to Step 1

Thm: tight bound  $\min \left\{ \|p^* - p_0\|_{\infty}^+ + \|p^* - p_0\|_{\infty}^- \mid p^*: \text{int. equil.} \right\}$

$$\|p^* - p_0\|_{\infty}^+ = \max\{0, \max(p^*(i) - p_0(i) \mid i \in N)\}$$

- merit
  - any initial vector can be used
  - less # of iter.
- demerit
  - **non-monotone** w.r.t. prices



# Two-Phase Auction (English-Dutch)

Ascending  
+ Descending

**Step 0:**  $p :=$  **any** integral vector

**Step A1:** find  $X \subseteq N$  minimizing  $L(p + e_X)$

**Step A2:**  $L(p + e_X) \geq L(p) \rightarrow$  go to **Step D1**

**Step A3:**  $p := p + e_X$ , Go to **Step A1**

**Step D1:** find  $Y \subseteq N$  minimizing  $L(p - e_Y)$

**Step D2:**  $L(p - e_Y) \geq L(p) \rightarrow$  stop ( $p$  is equilibrium)

**Step D3:**  $p := p - e_Y$ , Go to **Step D1**

- **merit**

- any initial vector can be used
- **almost monotone** w.r.t. prices

- **demerit**

- much more iterations than Greedy

**Thm:** # iters  $\leq 3 \times$  (# iters of Greedy)

# Andersson & Erlanson (2013) との関係

- Andersson, T., Erlanson, A. (2013):  
Multi-item Vickrey-English-Dutch auctions.  
Games Econ. Behav., 81, 116-129
- unit-demand auction: 一番良い item 一つだけを選択
- iterative auction の反復回数に注目
  - 既存のアルゴリズム: 反復回数大
    - English 型 (Demange et al. 1986, Mo et al. 1988)
    - Dutch 型 (Mishra & Parkes 2009)
- English-Dutch 型の iter. auction を提案
  - 反復回数の上界値(理論値), 数値実験による評価

gross subst.  
valuation  
の特殊ケース

# Andersson & Erlanson (2013) との関係： 我々の研究成果

- 既存のアルゴリズム：

  - English 型(Demange et al. 1986)

  - Dutch 型(Mishra & Parkes 2009)

- English-Dutch 型(Andersson & Erlanson 2013)

} 前出のiter. auction  
を特殊化したもの  
に一致

既存：over-demand, under-demand の用語で記述

→ Lyapunov fn を使って書き換え可能

- 正確な反復回数に関する我々の結果

→ English-Dutch型に対する結果(Andersson & Erlanson 2013)



# 発表の概要

- Ausubel (2006)の Ascending Auction を離散凸解析の視点から理解
  - 効用関数 の 粗代替性 =  $M$ 凹性
  - 間接効用関数 =  $L$ 凸関数 (共役性)
  - Lyapunov 関数 の最小化 =  $L$ 凸関数 の最小化
  - オークション =  $L$ 凸関数最小化アルゴリズム
- 種々の型のオークションの提案  
(English, Dutch, English-Dutch, Greedy)
- オークションの反復回数解析