数理経済学特講

複数財オークションのアルゴリズムと離散最適化

第9回　均衡を近似的に計算するアルゴリズム

塩浦昭義
東京工業大学 経営工学系 准教授
shioura.a.aa@m.titech.ac.jp

評価値が既知の場合のアルゴリズム

定理： 財の均衡配分 ←→ 評価値に関する最大重みマッチング

∴ 評価値が既知 →

均衡配分の計算は，最大重みマッチング問題に帰着可能

均衡配分が得られた →

均衡価格の計算は，最短路問題に帰着可能

評価値が分からない場合は？

評価値の扱いについて

入札者の評価値は個人情報
→ オークション主催者に 直接 知らせたくない

代案： 評価値の情報を **間接的に** 伝える
（例： 所与の価格に関して，利得最大の財）

1．主催者： 各財の暫定価格を決定
2．各入札者： 暫定価格の下で**利得最大の財**を報告
3．入札者全員に（重複無く）最も欲しい財を
配分可能 ➔ 終了．現在の価格は均衡価格
配分不可能 ➔ 主催者は暫定価格を適切に変更

反復オークション と呼ばれる

単一財の場合 → **イングリッシュ・オークション** など

反復オークションのアルゴリズム

以下の2つを紹介

- その1：均衡を近似的に計算
  - 単調に価格を増加，均衡配分（および（極小）均衡価格の近似値）を求める
  - 各反復で，入札者の利得最大の財ひとつの情報が必要
  - 価格増加のルールは簡単：　希望が重複→価格を増やす
- その2：均衡を厳密に計算
  - 単調に価格を増加，（極小）均衡価格（および均衡配分）を求める
  - 各反復で，入札者の利得最大の財すべての情報が必要
  - 価格増加のルールは複雑：
    - 得た情報を使い，価格を増やす財をうまく選ぶ

均衡を近似的に計算するアルゴリズム

アルゴリズムの流れ

- 各入札者は，現在の価格の下で利得最大の財を一つ選ぶ
- 同じ財を複数の入札者が選ぶ

  → ひとりに割り当て．　その財の価格を上昇させる．

異なる分野で独立に提案される

- Bertsekas (1979) --- 数理計画，オペレーションズ・リサーチ
  - 最大重みマッチング問題のアルゴリズムとして提案
  - この分野では「オークションアルゴリズム」とよばれる

- Crawford, Knoer (1981) --- 数理経済，オークション理論
  - ワルラス均衡（の近似解）を求めるアルゴリズムとして提案
  - Demange, Gale, Sotomayor (1986) に詳しい記述，解析あり

均衡を近似的に計算するアルゴリズム：詳細

δ: アルゴリズムのパラメータ, > 0

ステップ0：全ての財の価格 p(j) を 0 にする.
各入札者は財の割当なし, とする.

ステップ1：各入札者 i に対し,
財の割当あり or 最大利得≦0 ➔ 終了

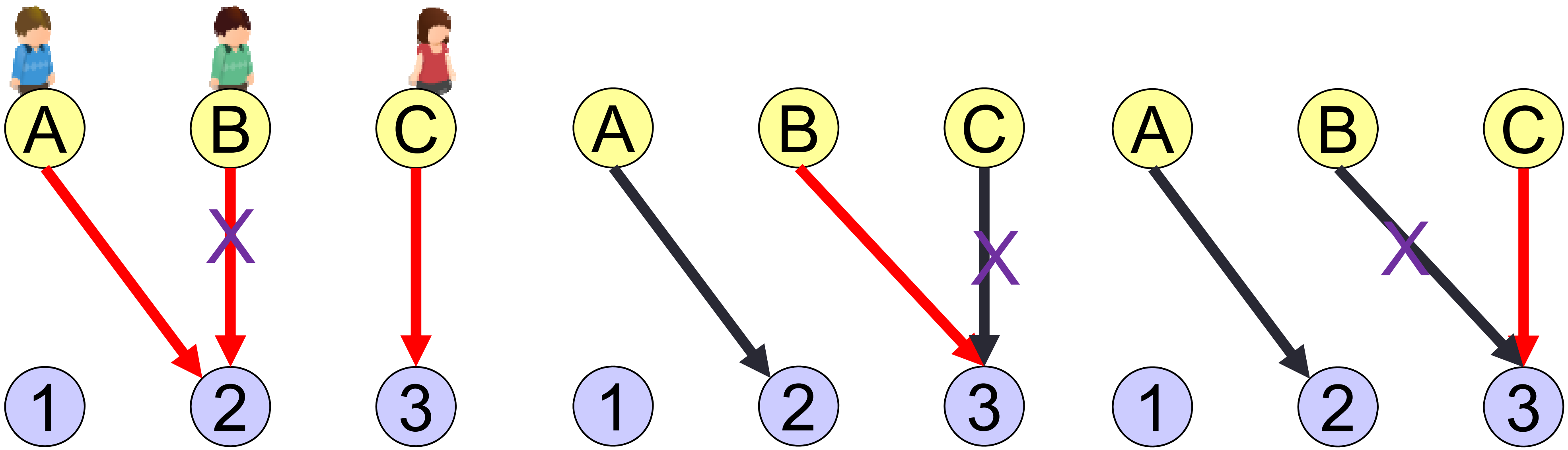
$\max_j\{v(i,j)-p(j)\}$

ステップ2：財の割当なし, かつ 最大利得 > 0 なる
入札者 i を選ぶ.

ステップ3： v(i,j)-p(j) 最大の j を選び, 入札者 i に財 j を割り当て.
入札者 k が既に財 j に割り当てられていた
→ k への j の割り当てを取消. p(j):=p(j) + δ
ステップ1へ.

アルゴリズムの実行例(1)

δ=2のとき



価格 0 0 0 | 0 2 0 | 0 2 2

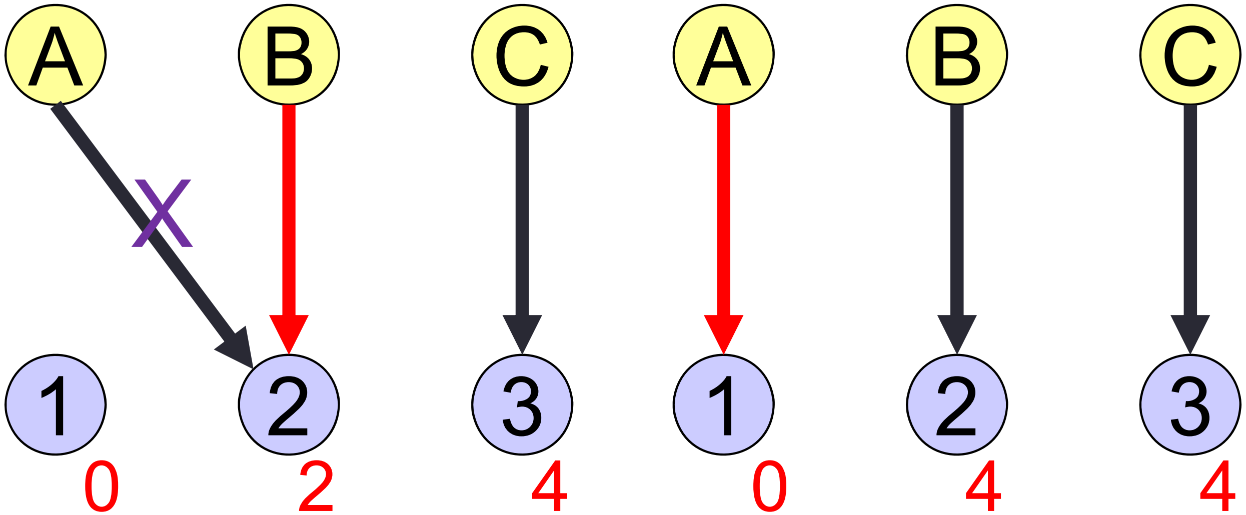
v(i,j)	A	B	C
①	3	1	1
②	6	5	3
③	2	4	4

利得	A	B	C
①	3	1	1
②	4	3	1
③	2	4	4

利得	A	B	C
①	3	1	1
②	4	3	1
③	0	2	2

アルゴリズムの実行例(1)

δ=2のとき



利得	A	B	C
①	3	1	1
②	4	3	1
③	-2	0	0

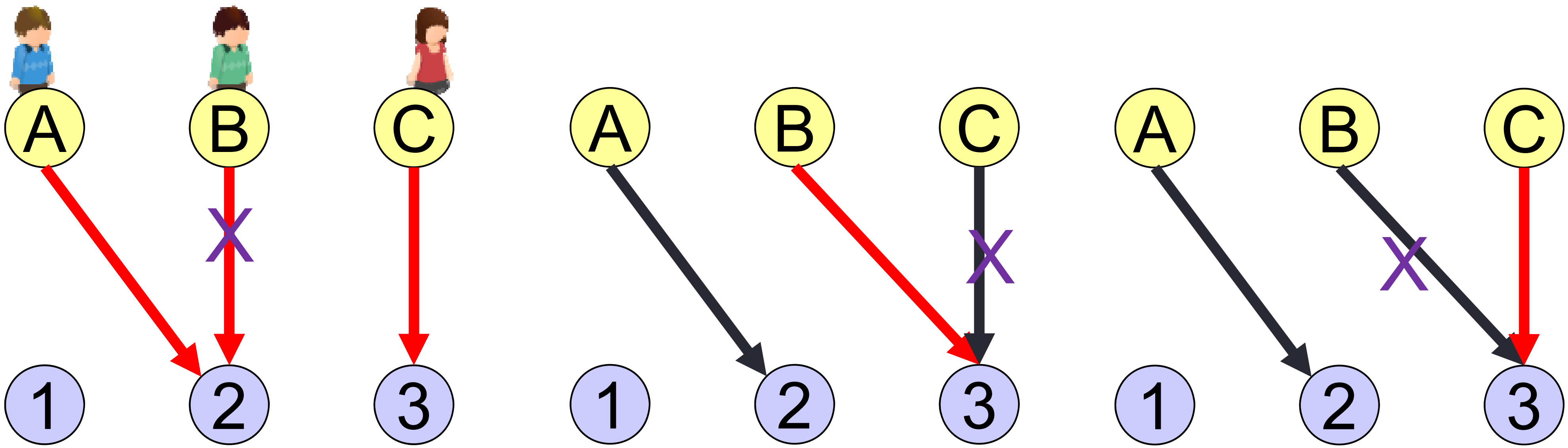
利得	A	B	C
①	3	1	1
②	2	1	-1
③	-2	0	0

終了
均衡配分○
均衡価格○

参考:
極小均衡価格
0　3　2

アルゴリズムの実行例(2)

δ=1のとき



価格　0　0　0　0　1　0　0　1　1

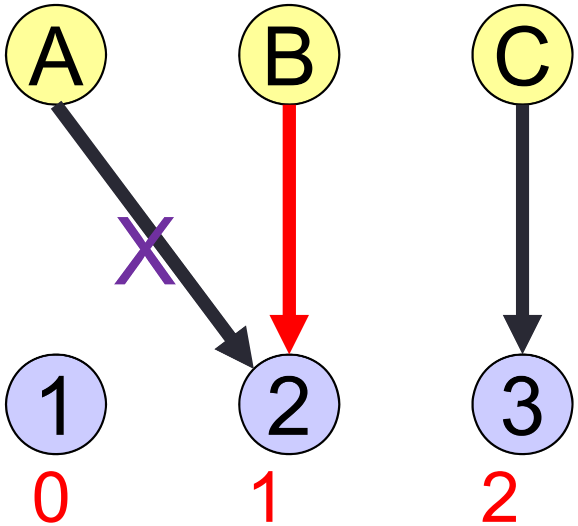
v(i,j)	A	B	C
①	3	1	1
②	6	5	3
③	2	4	4

利得	A	B	C
①	3	1	1
②	5	4	2
③	2	4	4

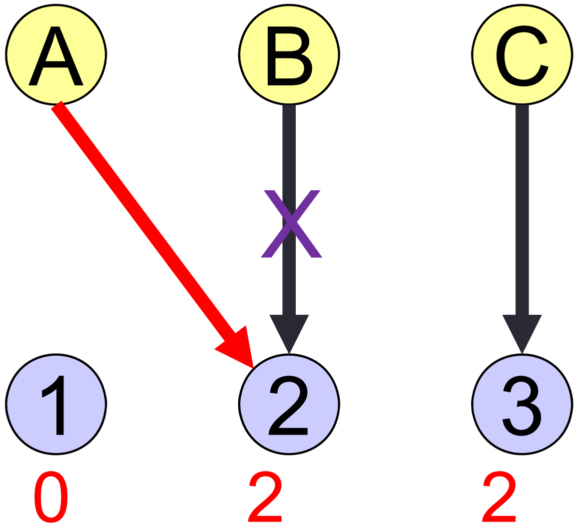
利得	A	B	C
①	3	1	1
②	5	4	2
③	1	3	3

アルゴリズムの実行例(2)

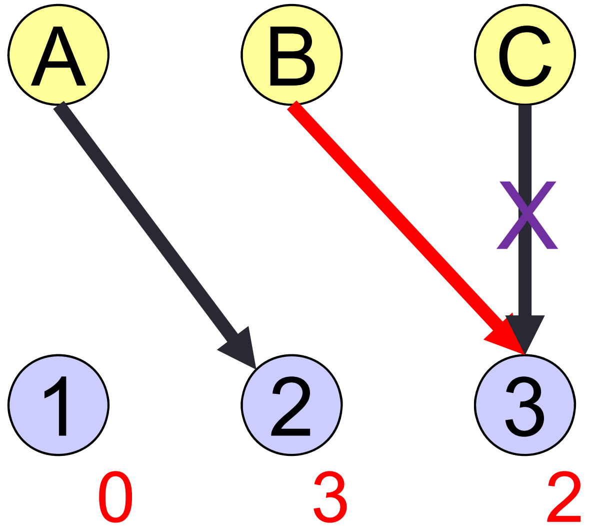
δ=1のとき



利得	A	B	C
①	3	1	1
②	5	4	2
③	0	2	2



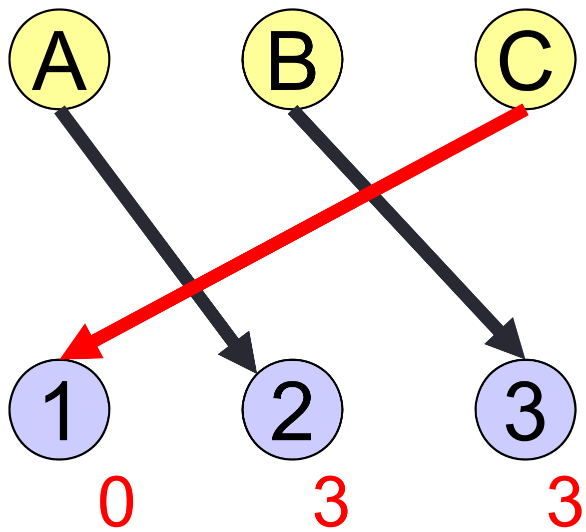
利得	A	B	C
①	3	1	1
②	4	3	1
③	0	2	2



利得	A	B	C
①	3	1	1
②	3	2	0
③	0	2	2

アルゴリズムの実行例(2)

δ=1のとき

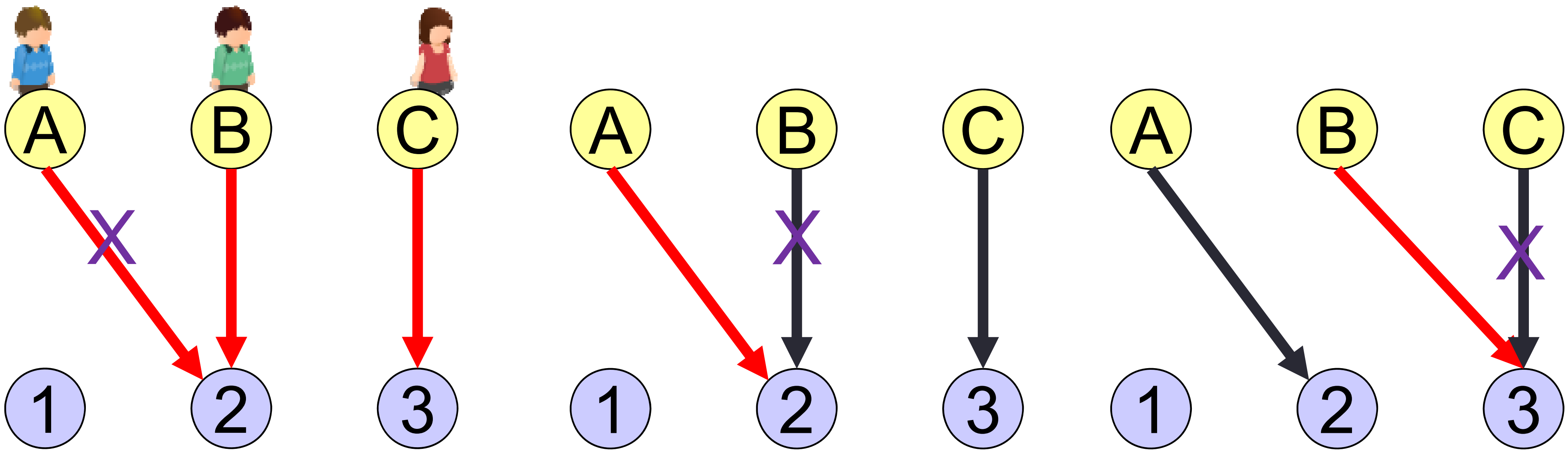


終了
均衡配分×
均衡価格○

利得	A	B	C
①	3	1	1
②	3	2	0
③	-1	1	1

アルゴリズムの実行例(3)

δ=0.5のとき



価格 0 0 0 | 0 0.5 0 | 0 1.0 0

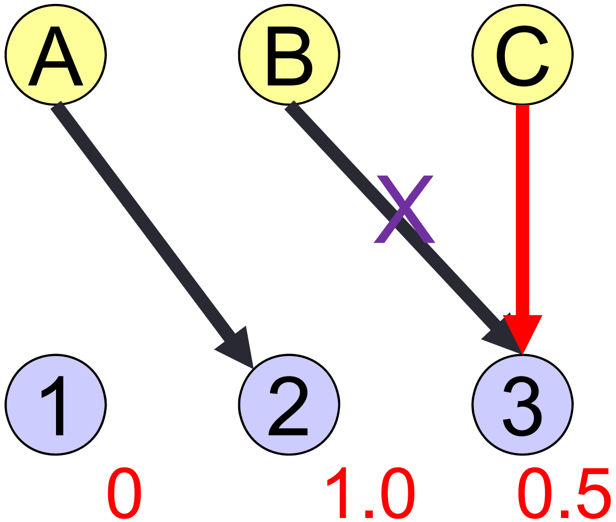
v(i,j)	A	B	C
①	3	1	1
②	6	5	3
③	2	4	4

v(i,j)	A	B	C
①	3	1	1
②	5.5	4.5	2.5
③	2	4	4

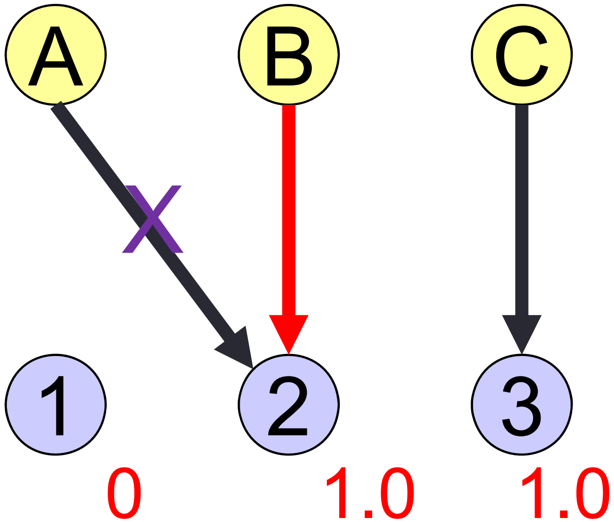
v(i,j)	A	B	C
①	3	1	1
②	5	4	2
③	2	4	4

アルゴリズムの実行例(3)

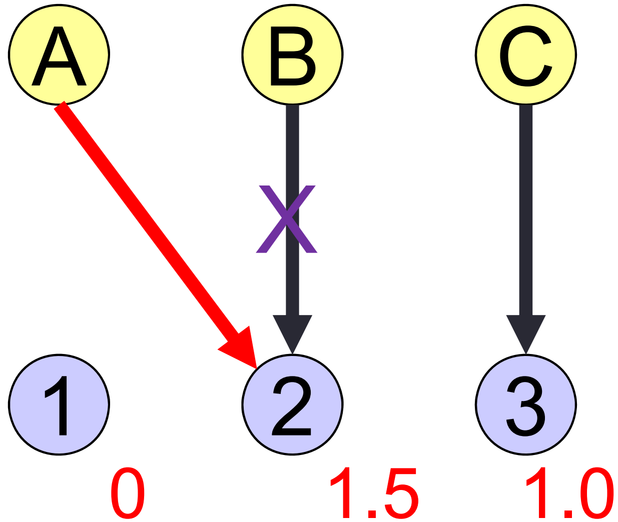
δ=0.5のとき



v(i,j)	A	B	C
①	3	1	1
②	5	4	2
③	1.5	3.5	3.5



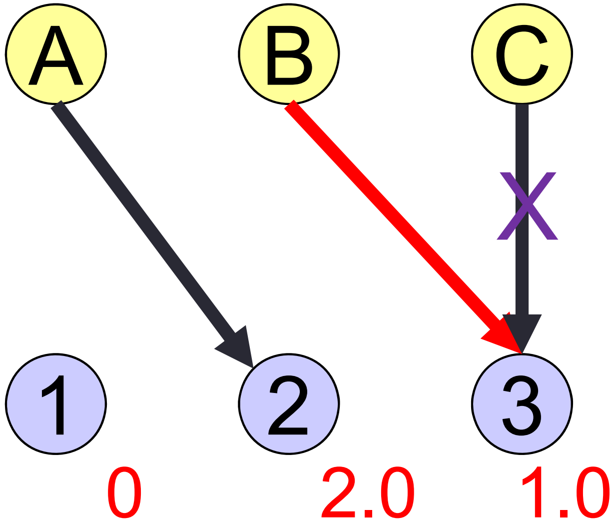
v(i,j)	A	B	C
①	3	1	1
②	5	4	2
③	1	3	3



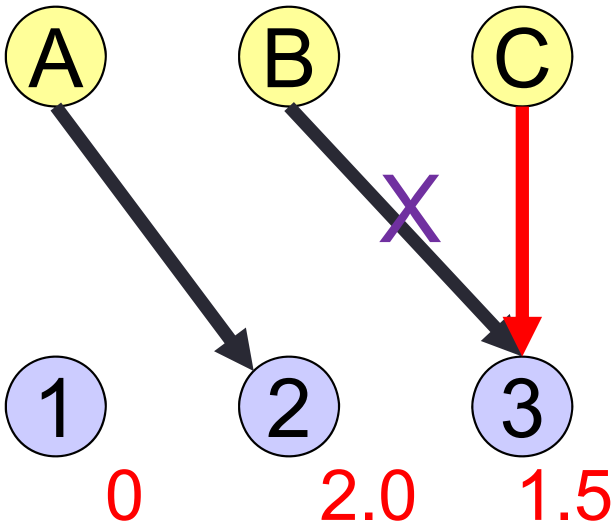
v(i,j)	A	B	C
①	3	1	1
②	4.5	3.5	1.5
③	1	3	3

アルゴリズムの実行例(3)

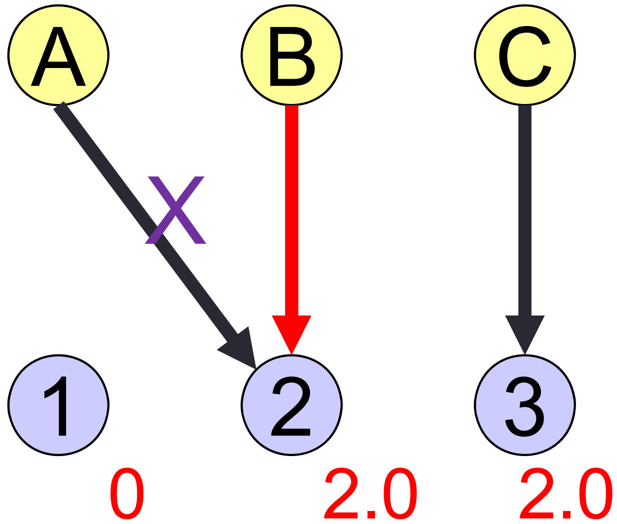
δ=0.5のとき



v(i,j)	A	B	C
①	3	1	1
②	4	3	1
③	1	3	3



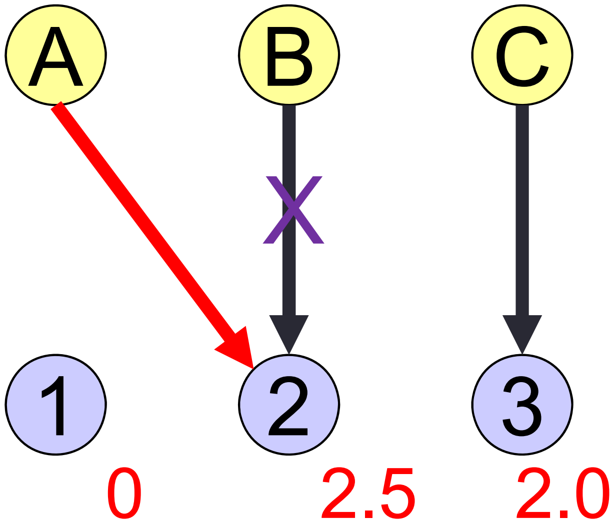
v(i,j)	A	B	C
①	3	1	1
②	4	3	1
③	0.5	2.5	2.5



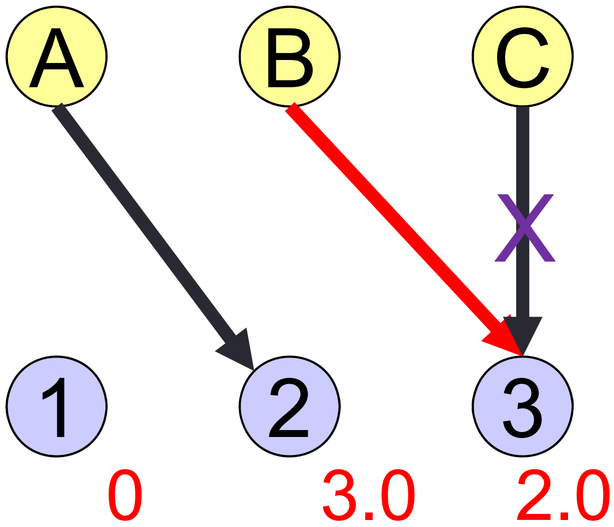
v(i,j)	A	B	C
①	3	1	1
②	4	3	1
③	0	2	2

アルゴリズムの実行例(3)

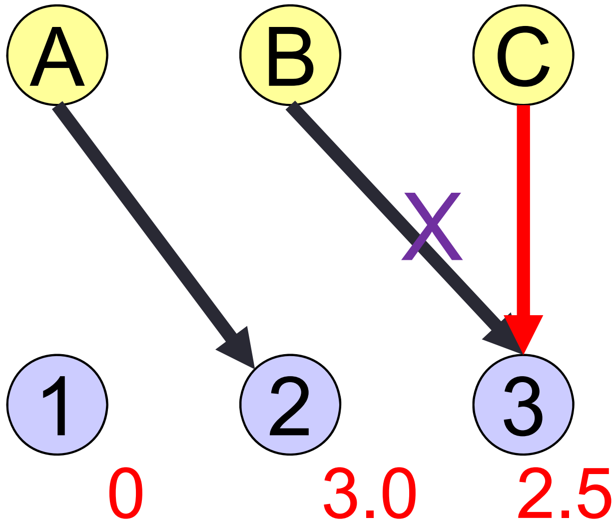
δ=0.5のとき



v(i,j)	A	B	C
①	3	1	1
②	3.5	2.5	0.5
③	0	2	2



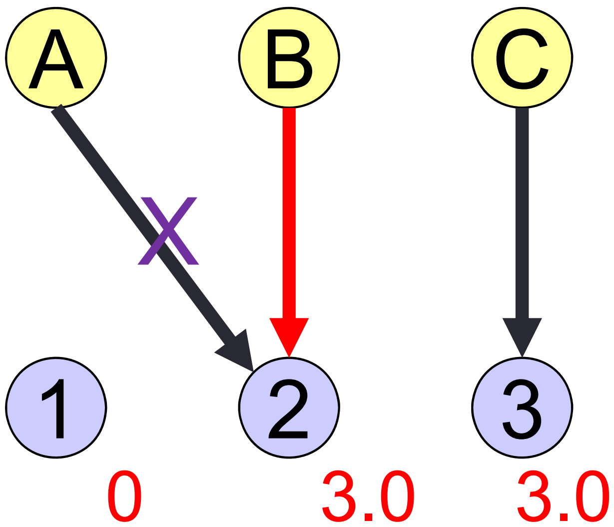
v(i,j)	A	B	C
①	3	1	1
②	3	2	0
③	0	2	2



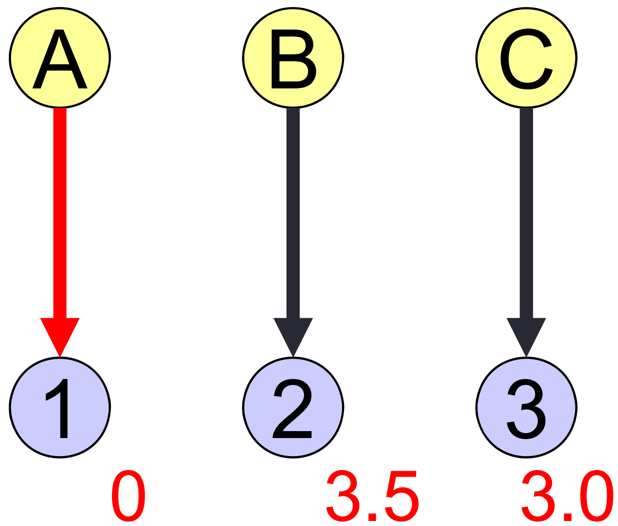
v(i,j)	A	B	C
①	3	1	1
②	3	2	0
③	-0.5	1.5	1.5

アルゴリズムの実行例(3)

δ=0.5のとき



v(i,j)	A	B	C
①	3	1	1
②	3	2	0
③	-1	1	1



v(i,j)	A	B	C
①	3	1	1
②	2.5	1.5	-0.5
③	-1	1	1

終了
均衡配分○
均衡価格○

アルゴリズムの性能評価

定理1 アルゴリズムは有限回の反復後に終了する

得られたマッチングの重み ≒ 最大重み

定理2 アルゴリズムにより得られたマッチング M に対し,
Mの重み ≧ 最大重みマッチングの重み － δ min{|B|, |N|}

得られた財の価格 ≒ 均衡価格

定理3 アルゴリズムにより得られた財の価格 p(j),
最小均衡価格 p*(j)
➔ |p(j) – p*(j) | ≦ δ min{|B|, |N|}

厳密な均衡を得る

評価値 v(i,j) がすべて整数
→ δ を調整して，均衡配分，均衡価格の厳密値を得ることが可能

定理2 アルゴリズムにより得られたマッチング M に対し，
Mの重み ≧ 最大重みマッチングの重み − δ min{|B|, |N|}

δ <1/min{|B|, |N|} とする
→ Mの重み > 最大重みマッチングの重み −1
→ マッチングの重みは整数なので， Mの重み = 最大重み

定理3 アルゴリズムにより得られた財の価格 p(j),
最小均衡価格 p*(j) ➔ |p(j) – p*(j) | ≦ δ min{|B|, |N|}

δ <1/2 min{|B|, |N|} とする
→ |p(j) – p*(j) | < 0.5
→ p*(j) は整数なので， p(j) を最も近い整数に丸め = p*(j)

定理1の証明

定理1 アルゴリズムは有限回の反復後に終了する

[証明]

- 各財の価格: 初期値＝0
- アルゴリズムの各反復:
  ある財 j の価格 p(j) が δ 増加 (→ v(i,j)-p(j) が δ 減少)
- すべての入札者 i に対し v(i,j)-p(j) < 0
  ➔ 財 j の価格は今後変化しない

∴財 j の価格 p(j) の増加回数≦ v(i,j) / δ

定理2の証明:準備その1

定理2 アルゴリズムにより得られたマッチング M に対し,
Mの重み ≧ 最大重みマッチングの重み − δ min{|B|, |N|}

命題1: アルゴリズム終了時に,
誰にも割り当てられなかった財 j の価格 p(j) = 0

[証明] アルゴリズムの途中で価格 >0 の財
→ 終了時まで常に誰かに割り当てられている
∴アルゴリズム終了時に割り当てなしの財 j: p(j)=0

定義: 入札者 i は δ-happy ←→条件 (a) or (b) を満たす
(a) [利得が「ほぼ」最大の財が割当]
i に財 j が割り当てられていて, $v(i,j) - p(j) \geq -\delta$
かつ $v(i,j) - p(j) \geq \max_{1 \leq h \leq n} \{v(i,h) - p(h)\} - \delta$
(b) [最大利得が非正]
i に財の割当なし, かつ $\max_{1 \leq h \leq n} \{v(i,h) - p(h)\} \leq 0$

定理2の証明：準備その2

命題2：

(i) 財の配分と価格が均衡 ➔ すべての入札者は 0-happy
（∵ 均衡の第一条件は，(a) において δ=0 とおいたものに一致）

(ii) ステップ3で入札者 i に財 j が割り当て
➔ i は δ-happy（財 j が他の入札者に奪われるまでずっと）
（∵ 財 j を選んだ時点では，j は最大利得 → 直後に価格が δ 減少.
他の入札者に奪われるまでは価格 p(j) 不変）

(iii) アルゴリズム終了時には，全員が δ-happy
（∵ 入札者に財の割り当てあり → (ii) より(a) 成立
なし → (b)成立）

定理2の証明（その1）

B_α =アルゴリズム終了時に財の割当のある入札者の集合
$\alpha(i)$=アルゴリズム終了時の入札者 $i \in B_\alpha$ への割り当て
B_{α^*} =最大重みマッチング M* において，財の割当のある入札者の集合
$\alpha^*(i)$=最大重みマッチングでの入札者 $i \in B_{\alpha^*}$ への割り当て
➔ 示したい不等式は $\sum_{i \in B_\alpha} v(i, \alpha(i)) \geq \sum_{i \in B_{\alpha^*}} v\left(i, \alpha^*(i)\right) - \delta \min\{|B|, |N|\}$

入札者 $i \in B_\alpha$ は，アルゴリズム終了時，δ-happy（∵命題2 (iii)）
➔ 財の割当があるので，δ-happy の条件 (a) が成立
→ $i \in B_{\alpha^*}$ならば　$v(i, \alpha(i)) - p\left(\alpha(i)\right) \geq \max_{1 \leq h \leq n} \{v(i, h) - p(h)\} - \delta$

$$\geq v\left(i, \alpha^*(i)\right) - p\left(\alpha^*(i)\right) - \delta \quad ①$$

　$i \notin B_{\alpha^*}$ならば　$v(i, \alpha(i)) - p\left(\alpha(i)\right) \geq -\delta$　②

①，② を辺々足すと

$$\sum_{i \in B_\alpha} v(i, \alpha(i)) - \sum_{i \in B_\alpha} p\left(\alpha(i)\right)$$

$$\geq \sum_{i \in B_\alpha^* \cap B_\alpha} v\left(i, \alpha^*(i)\right) - \sum_{i \in B_\alpha^* \cap B_\alpha} p\left(\alpha^*(i)\right) - \delta\, |B_\alpha| \quad ③$$

定理2の証明(その2)

入札者 $i \in B_\alpha^* \setminus B_\alpha$ は，アルゴリズム終了時，δ-happy（∵命題2 (iii)）
財の割当がない ➔ δ-happy の条件(b) が成立

→ $0 \geq v\left(i, \alpha^*(i)\right) - p\left(\alpha^*(i)\right)$ ④

③，④ を辺々足すと

$$\sum_{i \in B_\alpha} v(i, \alpha(i)) - \sum_{i \in B_\alpha} p\left(\alpha(i)\right)$$
$$\geq \sum_{i \in B_{\alpha^*}} v\left(i, \alpha^*(i)\right) - \sum_{i \in B_{\alpha^*}} p\left(\alpha^*(i)\right) - \delta\,|B_\alpha| \quad ⑤$$

$j \in N \setminus \{\alpha(i) | i \in B_\alpha\}$はアルゴリズム終了時に割り当てられていない財

➔ 命題1より p(j)=0　　$\therefore \sum_{i \in B_\alpha} p\left(\alpha(i)\right) = \sum_{j \in N} p(j)$ ⑥

価格の非負性より，　$\sum_{i \in B_{\alpha^*}} p\left(\alpha^*(i)\right) \leq \sum_{j \in N} p(j)$ ⑦

財の割当のある入札者の数$|B_\alpha|$ ≦入札者の数|B|, 財の数 |N|　⑧

定理2の証明（その3）

⑤, ⑥, ⑦, ⑧より

$$\sum_{i \in B_\alpha} v(i, \alpha(i)) - \sum_{j \in N} p(j)$$

$$= \sum_{i \in B_\alpha} v(i, \alpha(i)) - \sum_{i \in B_\alpha} p\big(\alpha(i)\big)$$

$$\geq \sum_{i \in B_{\alpha^*}} v\big(i, \alpha^*(i)\big) - \sum_{i \in B_{\alpha^*}} p\big(\alpha^*(i)\big) - \delta \, |B_\alpha|$$

$$\geq \sum_{i \in B_{\alpha^*}} v\big(i, \alpha^*(i)\big) - \sum_{j \in N} p(j) - \delta \min\{|B|, |N|\}$$

$$\therefore \sum_{i \in B_\alpha} v(i, \alpha(i)) \geq \sum_{i \in B_{\alpha^*}} v\big(i, \alpha^*(i)\big) - \delta \min\{|B|, |N|\}$$

証明終わり

演習問題

下記のように評価値が与えられたとき，δ＝1として，
均衡を近似的に計算するアルゴリズムを適用して
均衡の近似解を計算せよ．計算の過程も書くこと．

(1)

v(i,j)	A	B	C
①	2	3	6
②	6	7	7

(2)

v(i,j)	A	B	C
①	3	1	0
②	7	6	7
③	1	7	8

(3)

v(i,j)	A	B	C
①	3	1	0
②	7	6	7
③	1	7	8
④	0	0	4

それぞれの極小均衡価格は以下の通り
問1 : p(1)=2, p(2)=6
問2 : p(1)=0, p(2)=4, p(3)=5
問3 : p(1)=0, p(2) =3, p(3)=4, p(4)=0