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　均衡を厳密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

均衡価格の判定方法

- 所与の価格ベクトル p に対し，均衡価格か否かを判定したい

①入札者 i に財 j が割り当てられる（$\alpha(i) = j$）

➔ $0 \leq v(i,j) - p(j) = \max_{1 \leq h \leq n} \{v(i,h) - p(h)\}$

②入札者 i に財の割り当てがない（$\alpha(i) = 0$）

➔ $\max_{1 \leq h \leq n} \{v(i,h) - p(h)\} \leq 0$

③財 j が誰にも割り当てられない ➔ $p(j) = 0$

各入札者に対し，

最大利得 > 0 → 利得最大の財を割り当てる必要あり

各財に対し，

価格 > 0 → その財を欲しがる入札者に割り当てる必要あり

➔ 制約付きのマッチング問題

均衡価格の判定：マッチングへの帰着

- 頂点集合 B（入札者全体）と N（財全体）の二部グラフ
- B'=最大利得 > 0 の入札者全体
- N'=価格 > 0 の財全体

マッチングでカバーしたい

定義: マッチング M は頂点集合 X をカバーする

⟷ X の各頂点にマッチング M のいずれかの枝が接続する

- マッチングで使って良い枝 (i,j)

⟷ 入札者 i にとって財 j は最大利得，かつその利得≧0

E' = 上記の条件を満たす (i,j) 全体

均衡価格の条件を
グラフの言葉で書き換え

命題: 価格 p(j) (j∈N) は均衡価格
⟷ 枝集合 E' に含まれるマッチング M で,
B' および N' をカバーするものが存在

マッチングへの帰着：例

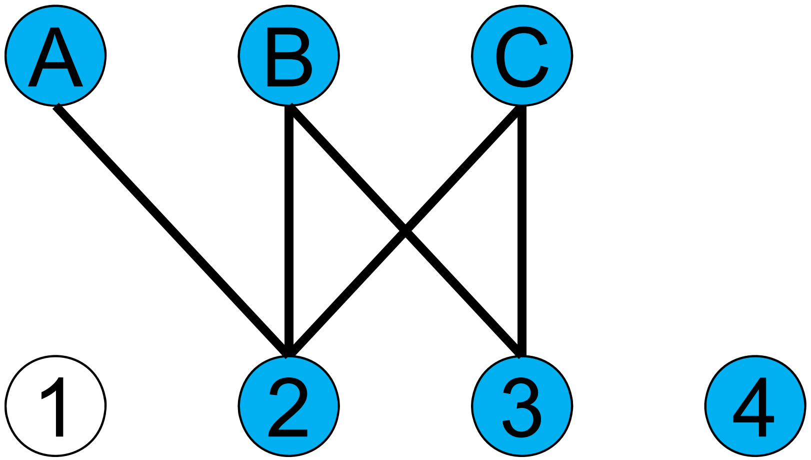
v(i,j)	A	B	C
①	3	1	0
②	7	6	7
③	1	7	8
④	0	0	3

利得	A	B	C
①	3	1	0
②	6	5	6
③	-1	5	6
④	-1	-1	2

価格(0,1,2,1) のとき

B’={A,B,C}, N’={2,3,4}

マッチングは存在せず

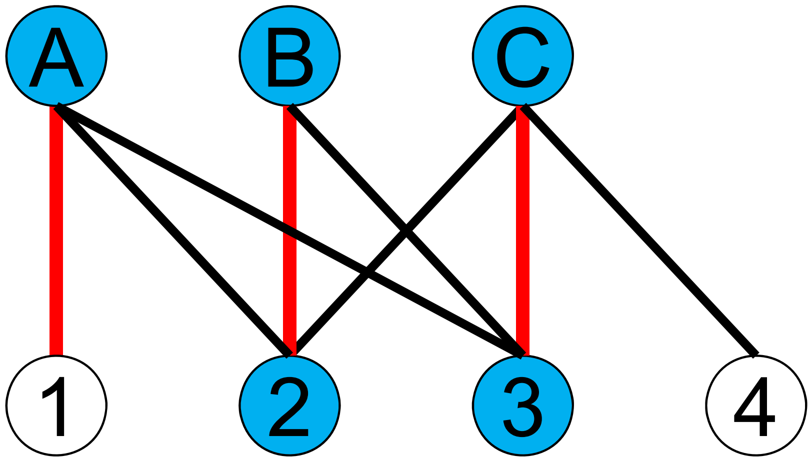


利得	A	B	C
①	3	1	0
②	3	2	3
③	3	2	3
④	0	0	3

価格(0,4,5,0) のとき

B’={A,B,C}, N’={2,3}

マッチングは存在



指定頂点をカバーするマッチング

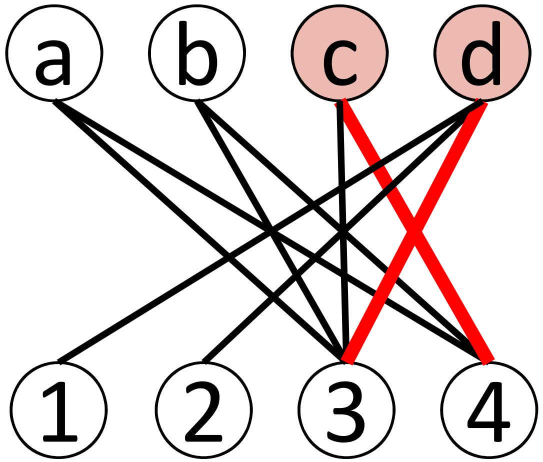
カバーする頂点の制約（片側）

カバーする頂点の制約付きマッチングに関する性質
← 均衡条件の書き換えに利用

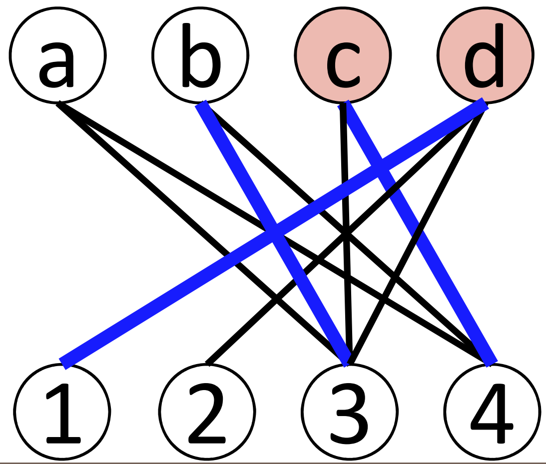
命題1

(i) B の部分集合 B′ に対し，B′ をカバーするマッチングが存在
➔B′をカバーする最大マッチングが存在

(ii) N の部分集合 N′ に対し，N′ をカバーするマッチングが存在
➔N′をカバーする最大マッチングが存在



赤いマッチングは
B’={c,d} をカバーする



青いマッチングは
最大マッチング

カバーする頂点の制約(片側)

命題1

(i) B の部分集合 B’ に対し，B’ をカバーするマッチングが存在

➔B’をカバーする最大マッチングが存在

(ii) N の部分集合 N’ に対し，N’ をカバーするマッチングが存在

➔N’をカバーする最大マッチングが存在

[証明] M: B’ をカバーするマッチング

M に増加路アルゴリズムを適用し，最大マッチング M* を求める

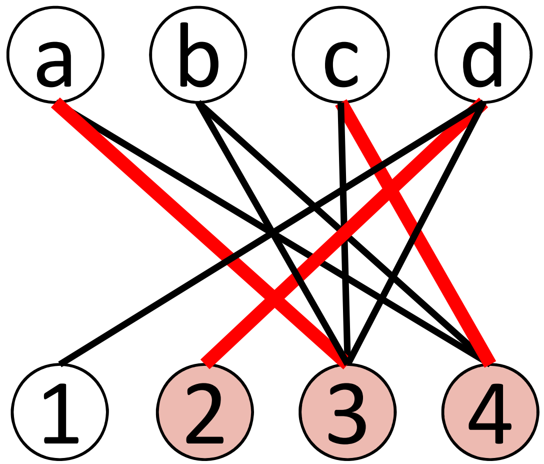
増加路アルゴリズムの各反復において，

マッチングのカバーする頂点は単調増加(減らない)

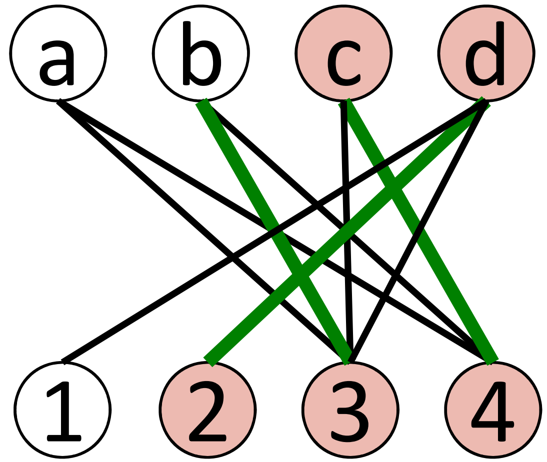
∴ M*のカバーする頂点 ⊇ M のカバーする頂点 ⊇ B’ ■

カバーする頂点の制約(両側)

命題2　B の部分集合 B’ および N の部分集合 N’ に対し，
B’ をカバーするマッチング，および N’ をカバーするマッチングがそれぞれ存在
➔B’ と N’を同時にカバーする(最大)マッチングが存在



赤いマッチングは
N’={2,3,4} をカバーする
最大マッチング



このマッチングは
B’={c,d}, N’={2,3,4}を同時に
カバーする最大マッチング

均衡価格の条件の書き換え

- 頂点集合 B（入札者全体）と N（財全体）の二部グラフ
- B’=最大利得 > 0 の入札者全体
- N’=価格 > 0 の財全体
- 枝集合 E’ ={(i,j) | 入札者 i にとって財 j は最大利得,
  かつその利得≧0 }

命題2を使って，さらに書き換え可能

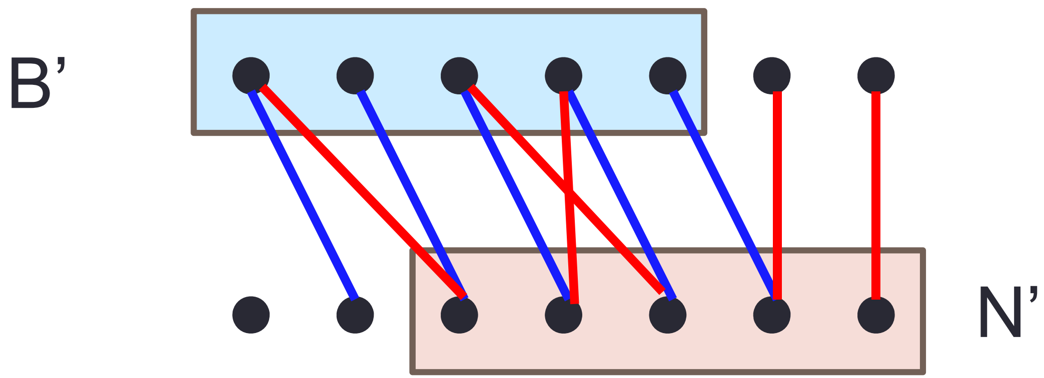
命題: 価格 p(j) (j∈N) は均衡価格
⇔ マッチング M⊆ E’ で，B’ および N’ をカバーするものが存在
⇔ マッチング M_1⊆ E’ で B’ をカバーするものが存在，
かつ マッチング M_2⊆ E’ で N’ をカバーするものが存在

カバーする頂点の制約(両側)

命題2　B の部分集合 B’ および N の部分集合 N’ に対し,
B’ をカバーするマッチング, および N’ をカバーするマッチングが
それぞれ存在
➔B’ と N’を同時にカバーする(最大)マッチングが存在

[証明]

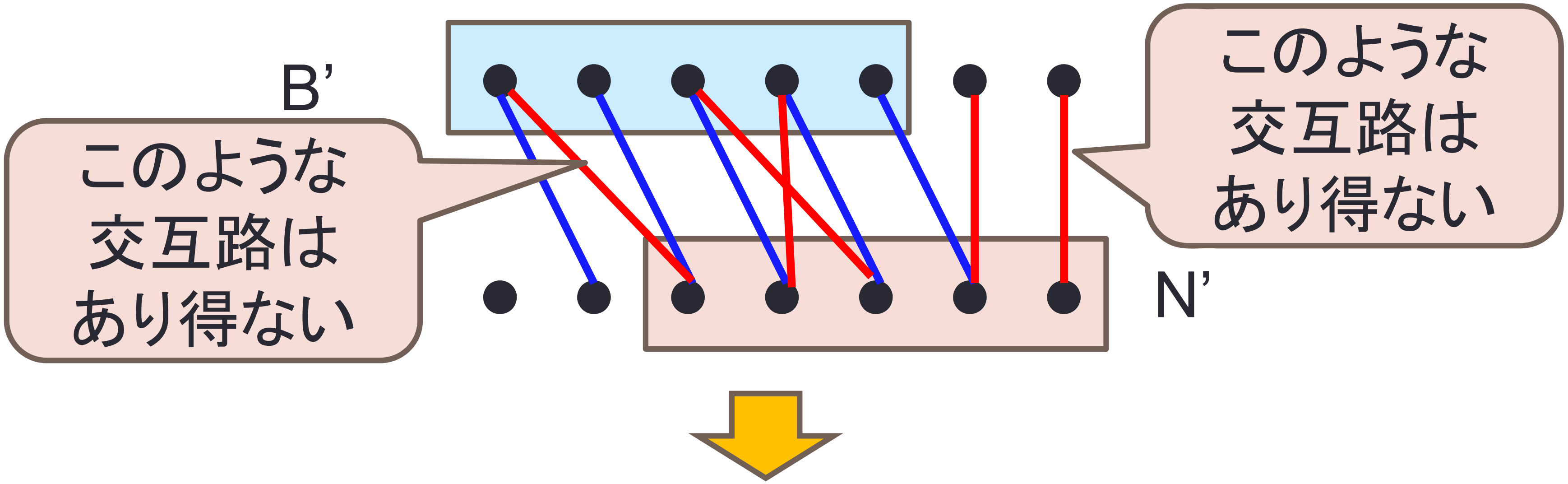
- M: B’ をカバーする最大マッチング(命題1(i)より存在)
- そのようなマッチングの中で, カバーする N’ の頂点数が最大とする
- M が N’ すべてをカバーしていないと仮定 → 矛盾を導く
- M*: N’ すべてをカバーする最大マッチング

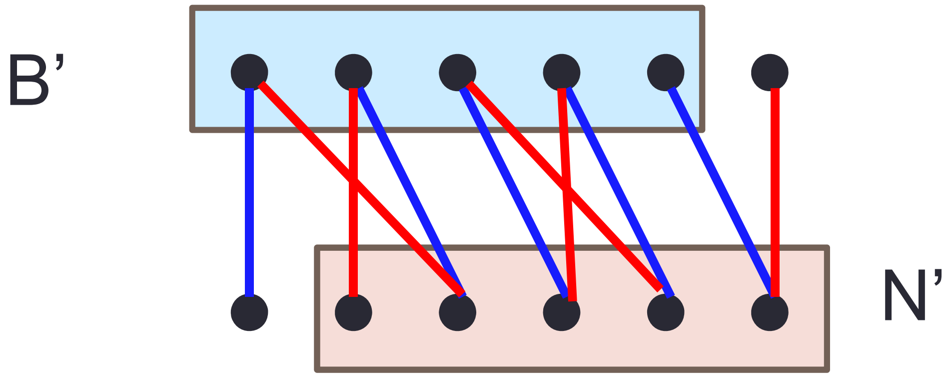


カバーする頂点の制約（両側）

[証明のつづき]

- 枝集合 M∪M* からなるグラフ → 交互路と交互閉路に分解できる
  - M に関する増加路なし，M* に関する増加路なし
    （M, M* ともに最大マッチングだから）



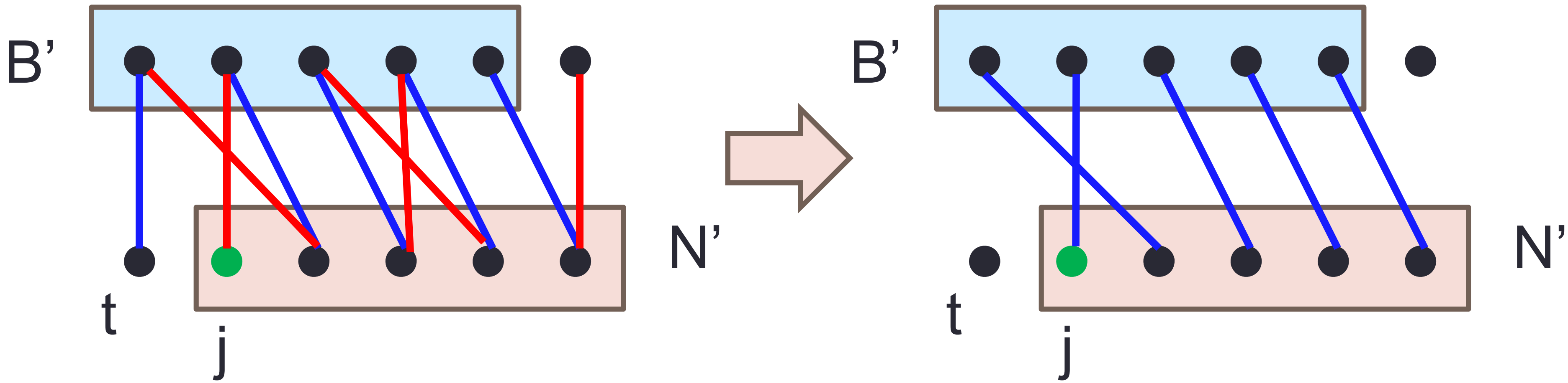


∴交互路の枝数は
すべて偶数

カバーする頂点の制約(両側)

[証明のつづき]

- j∈N': マッチング M でカバーされていない頂点



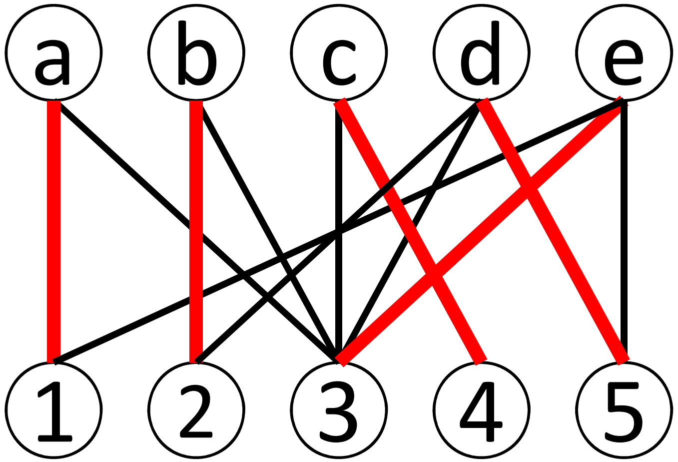
- j はマッチング M* でカバーされている
  → j から始まる交互路 P が存在, M* の枝から始まる
- 交互路 P の枝数は偶数
  → 最後の枝は M, 最後の頂点 t は N' に含まれない
  (∵ t∈N' → t に接続する M* の枝が存在 → 交互路は終わらない)

∴ 交互路 P を使って M の枝を入れ替え → j がカバーされる
→ カバーできる N' の頂点数が増える(矛盾)

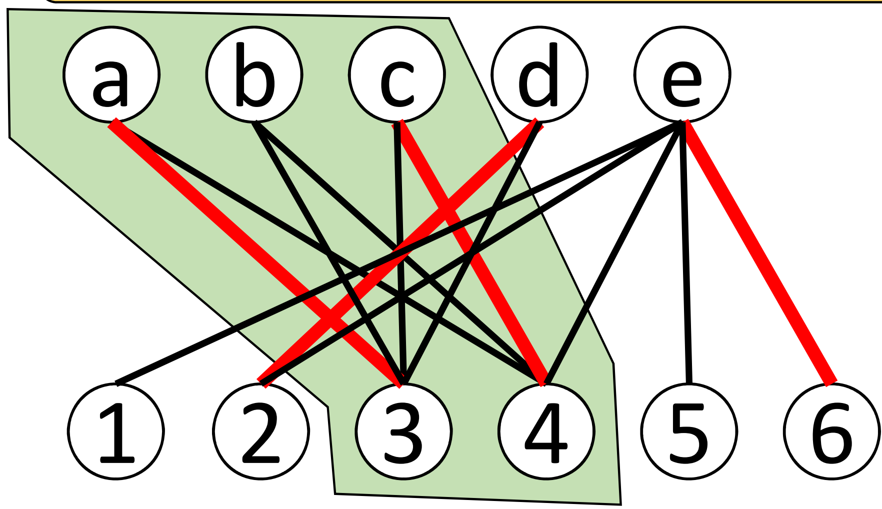
指定頂点集合をカバーするマッチング：存在するための必要条件

所与の B’⊆B をカバーするマッチングは存在するか？

B’={a,b,c,d,e}をカバー可能



B’={a,b,c,d,e}をカバー可能？



命題
ある X⊆B’に対し,
X の隣接頂点の数 < |X|
➔ B’ すべてをカバーする
マッチングは存在しない

a,b,c の隣接頂点 = {3,4}
∴ a,b,cのうち，高々2つしか
カバーできない
→ B’ はカバーできない

指定頂点集合をカバーするマッチング：存在するための必要十分条件

命題 あるX⊆B'に対し，X の隣接頂点の数 < |X|
➔ B' すべてをカバーするマッチングは存在しない
（対偶：B' すべてをカバーするマッチングが存在
➔ 任意のX⊆B'に対し，X の隣接頂点の数 ≧ |X|

逆も成り立つ

定理（Hall（ホール）の定理）
B' すべてをカバーするマッチングが存在
⇔ 任意のX⊆B'に対し，X の隣接頂点の数 ≧ |X|

ホールの定理：証明

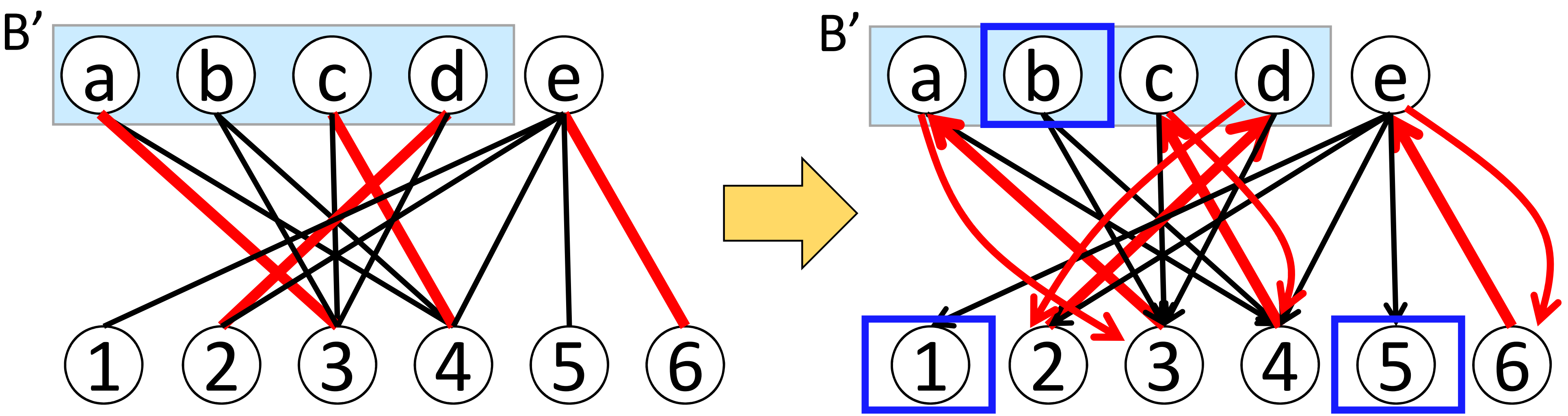
[🡐] 対偶「B′ すべてをカバーするマッチングは存在しない

→ ある $X \subseteq B'$ に対し,　|X|> Xに隣接する頂点の数」を証明

- M: マッチング, カバーする B′ の頂点数が最大
- M に関する有向グラフを作成

（マッチングの枝：両方向きの枝，それ以外：下向きの枝）

- S⊆B′: マッチング枝が接続していない B′ の頂点集合
- T⊆N: マッチング枝が接続していない N の頂点集合



ホールの定理：証明のつづき

Mのカバーする B’ の頂点数は最大

→ S の頂点から始まる，Mに関する増加路は存在しない

（∵存在する → カバーできる B’ の頂点数が増える）

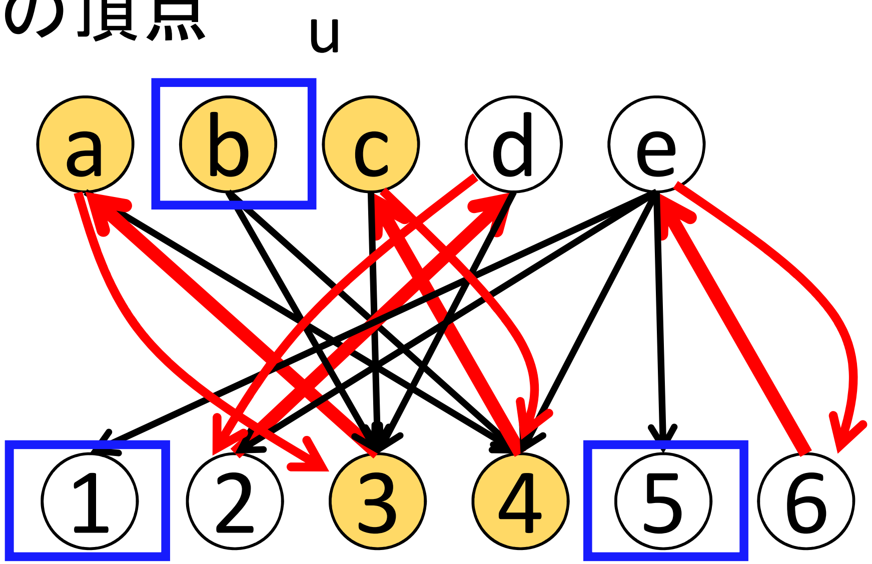
→ 有向グラフで S から T への有向路が存在しない

仮定より，S は非空 → u∈S とする.

X⊆B’: 有向グラフで u から到達可能な B’ の頂点

Y⊆N: 有向グラフで u から到達可能な N の頂点

→ 二部グラフで X に隣接する頂点＝ Y



ホールの定理：証明のつづき

また Y∩T=∅ （∵ u から始まる増加路が存在しない）

→ Y の各頂点にはマッチング枝が接続

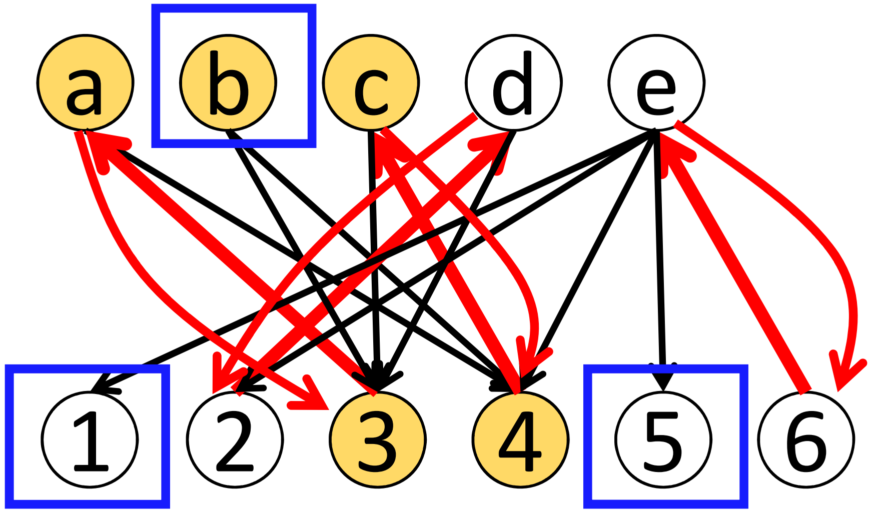
Z⊆B': マッチング M での Y の「相手」

→ Z⊆X, |Z|=|Y|

さらに u∉Z, u∈X

∴ |X|≧|Z|+1>|Y|

= Xに隣接する頂点数



均衡価格の必要十分条件

- 頂点集合 B（入札者全体）と N（財全体）の二部グラフ
- B’=最大利得 > 0 の入札者全体
- N’=価格 > 0 の財全体
- 枝集合 E’ ={(i,j) | 入札者 i にとって財 j は最大利得, かつその利得≧0 }

ホールの定理を使って書き換え

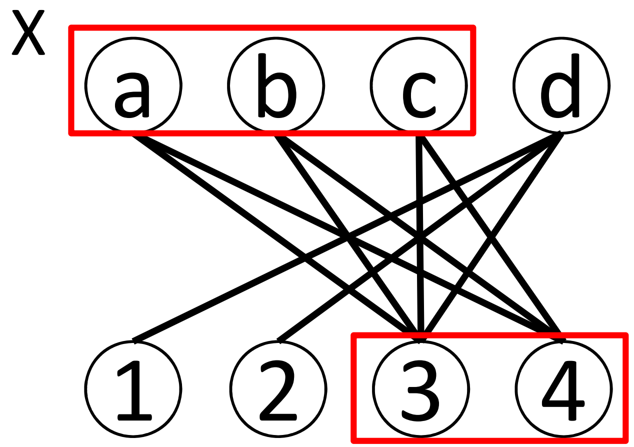
価格 p(j) (j∈N) は均衡価格
⇔ マッチング $M_1 \subseteq E'$ で B’ をカバーするものが存在,
かつ　マッチング $M_2 \subseteq E'$ で N’ をカバーするものが存在
⇔ 二部グラフにおいて,
任意のX⊆B’に対し,　X の隣接頂点の数 ≧ |X|
任意のY⊆N’に対し,　Y の隣接頂点の数 ≧ |Y|

均衡価格の必要十分条件：書き換え

二部グラフにおいて,
任意のX⊆B’に対し, Xの隣接頂点の数 ≧ |X|
任意のY⊆N’に対し, Yの隣接頂点の数 ≧ |Y|

⇕ オークションの言葉で書き換え

任意のX⊆B’に対し,
Xに含まれる入札者が欲しい財の数 ≧ |X|
任意のY⊆N’に対し,
Yに含まれる財を欲しがる入札者の数 ≧ |Y|



Xに含まれる入札者が欲しい財

均衡価格の必要十分条件：書き換え

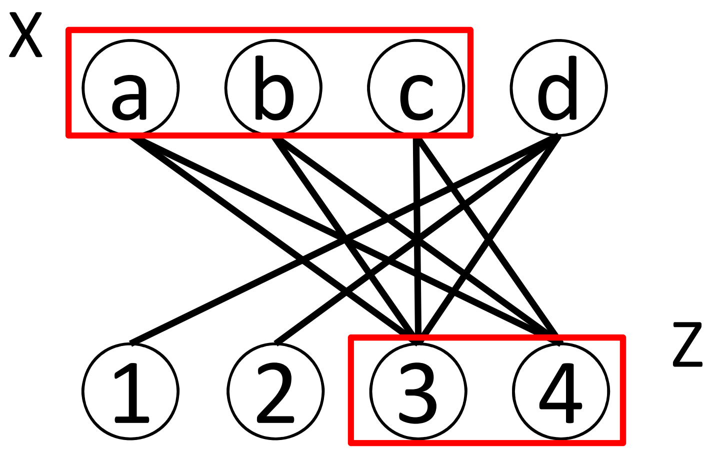
さらに，1つめの条件を書き換え

任意のX⊆B’に対し，

X に含まれる入札者が欲しい財の数 ≧ |X|

←→ 任意のZ⊆Nに対し，

|Z| ≧ 欲しい財が皆 Z に含まれ，かつ最大利得>0の入札者の数



Z = X に含まれる入札者が欲しい財

X = 欲しい財が皆 Z に含まれ，かつ最大利得>0 の入札者全員

これらの条件を踏まえ，反復オークションを設計

均衡を厳密に計算するアルゴリズム

均衡を厳密に計算する

- 入札者の評価値の情報を使わず，均衡（価格）を厳密に計算したい
- 使える情報：

  価格 (p(1), p(2),…,p(n)) を入札者に提示

  → N∪{0} の中で最も欲しい財（v(i,j)-p(j)最大の財）を全て答える
- 計算の方針
  - 現在の価格ベクトルが均衡か否かを判定
    - 最大マッチング問題を利用
  - 均衡価格でない → 価格を更新
    - 最大マッチング問題の結果を利用，価格を更新する財を選ぶ

アルゴリズムの方針

二部グラフ (B, N; E’),　B’ = 最大利得 > 0 の入札者全員

N’=価格 > 0 の財全体

満たすべき条件 :

① 任意のX⊆B’に対し，　X の隣接頂点 Z の数 ≧ |X|

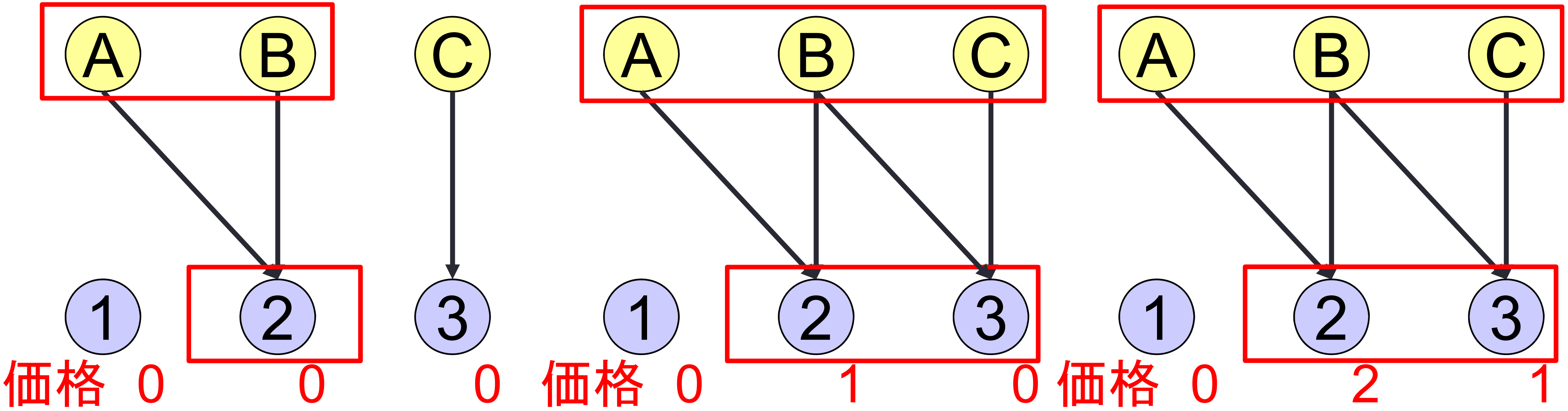
② 任意のY⊆N’に対し，　Y の隣接頂点の数 ≧ |Y|

- 価格p=(0,0,…,0)から開始

  ➔N’=∅ なので，②はただちに成立　①は一般に成り立たない
- ②を満たしつつ，①の条件を満たさない X を減らしていく
  - ①の条件を満たさない X で，隣接頂点集合Zが極小なものを選ぶ

    → Z に含まれる財の価格を1ずつ増やす

アルゴリズムの実行例(1)

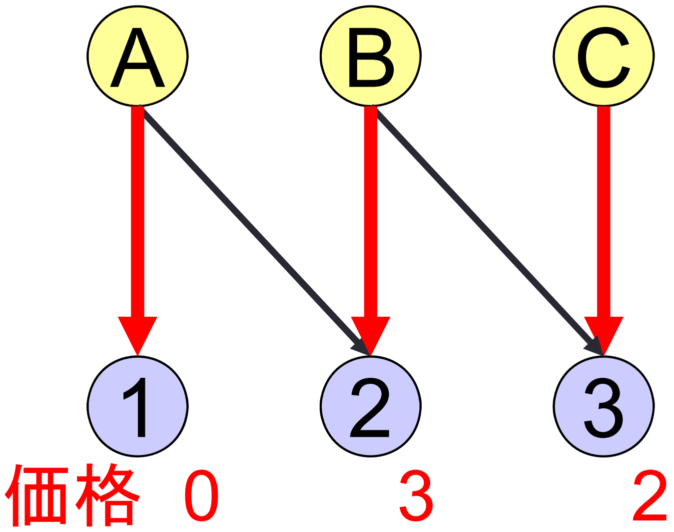


v(i,j)	A	B	C
①	3	1	1
②	6	5	3
③	2	4	4

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	4	3	1
③	1	3	3

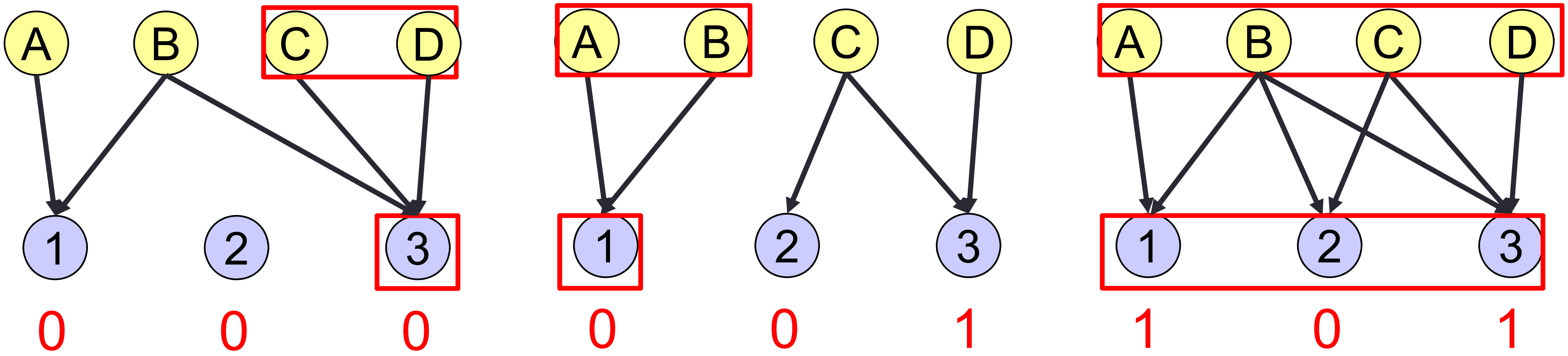
アルゴリズムの実行例(1)



利得	A	B	C
①	3	1	1
②	3	2	0
③	0	2	2

①の条件を満たさない X なし
→ アルゴリズム終了
　所望のマッチング存在

アルゴリズムの実行例(2)

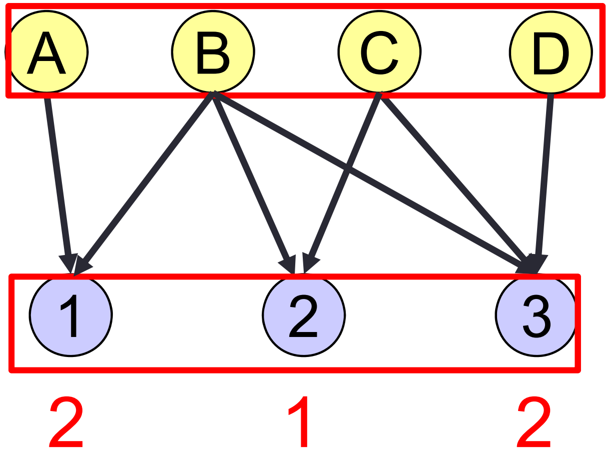


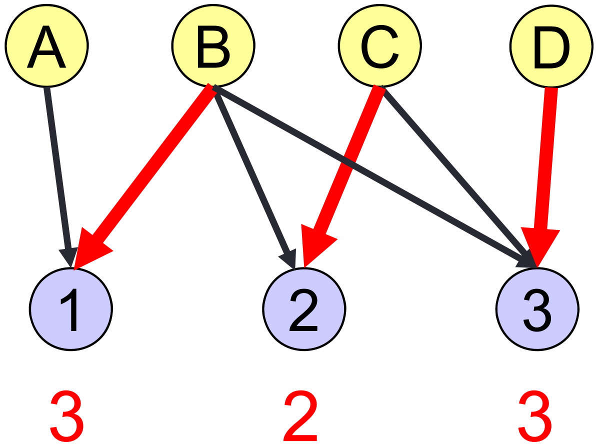
v(i,j)	A	B	C	D
①	3	7	1	0
②	1	6	7	0
③	0	7	8	4

利得	A	B	C	D
①	3	7	1	0
②	1	6	7	0
③	-1	6	7	3

利得	A	B	C	D
①	2	6	0	-1
②	1	6	7	0
③	-1	6	7	3

アルゴリズムの実行例(2)





利得	A	B	C	D
①	1	5	-1	-2
②	0	5	6	-1
③	-2	5	6	2

利得	A	B	C	D
①	0	4	-2	-3
②	-1	4	5	-2
③	-3	4	5	1

①の条件を
満たさない X なし
→ アルゴリズム終了
所望の
マッチング存在

Aの最大利得=0

アルゴリズムの正当性の証明

二部グラフ (B, N; E'),　 B' = 最大利得 > 0 の入札者全員

N'=価格 > 0 の財全体

満たすべき条件:

① 任意のX⊆B'に対し,　X の隣接頂点 Z の数 ≧ |X|

② 任意のY⊆N'に対し,　Y の隣接頂点の数 ≧ |Y|

反復が続けば, いつかは条件①成立

(∵反復回数が大きくなる → 各財の価格が十分に大きくなる

→ どの財も利得≦0になる

→ 最大利得 > 0 の入札者不在　→ 条件① 成立　)

よって, 条件②が各反復で成り立つことを証明すれば十分.

条件②が成り立つことの証明

② 任意のY⊆N'に対し，Y の隣接頂点の数 ≧ |Y|

条件②が各反復で成り立つことを証明（概略）

- 入札者集合 X に隣接する財集合 Zの価格を増やしたと仮定，

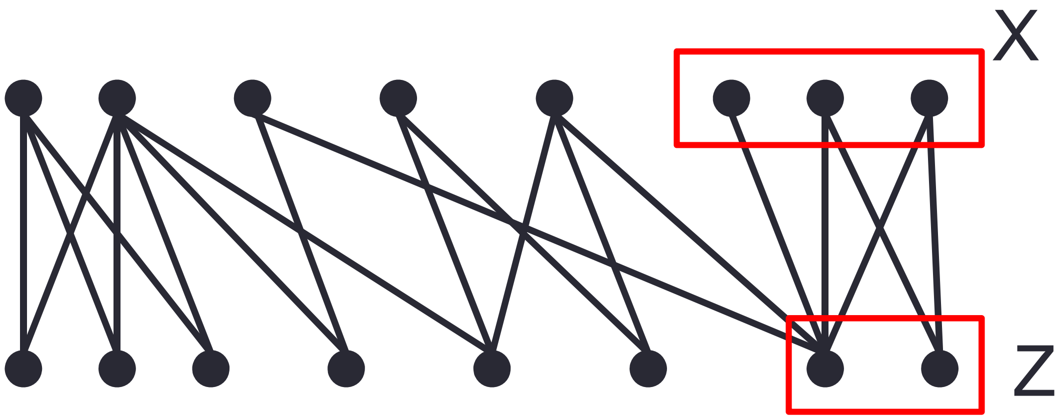
  → X と N – Z を結ぶ枝は存在しない

  Z は N' に追加される

- 価格増加前の入札者集合 X の最大利得> 0

  → 価格増加後も，Xに接続する枝は消えない

∴ 消える可能性のある枝は，B – X と Z を結ぶ枝のみ

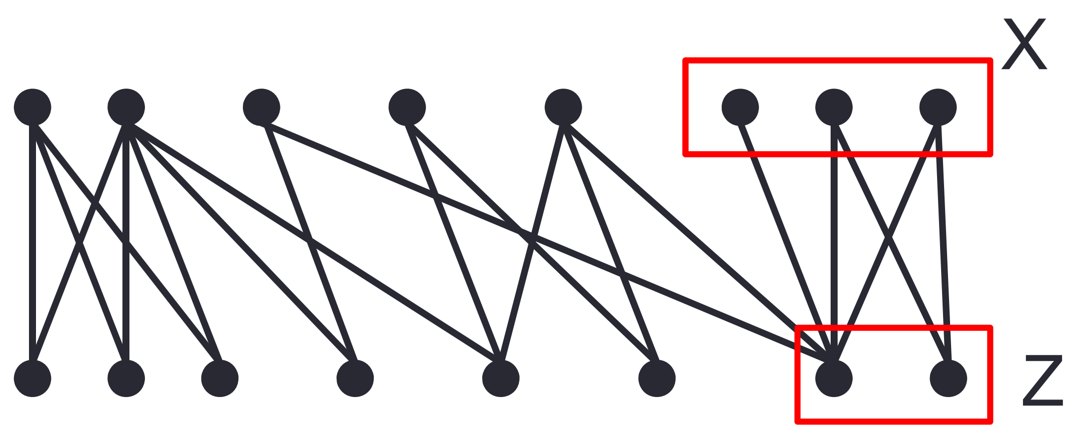


条件②が成り立つことの証明

(i) Y⊆Z の場合: Z の選び方より, XとZの間の枝のみを使って,
条件②が示せる.

(ii) Y⊆N – Z の場合: 価格増加前は条件②が成立.
Y から出る枝は価格増加後も同じ ∴条件②成立

(iii)それ以外の場合: Y は Z および N– Z と共通部分をもつ
(i),(ii)の結果より, Y∩Z は条件②を満たす
Y∩(N– Z)は条件②を満たす
∴ Y = (Y∩Z)∪(Y∩(N– Z))も条件②を満たす



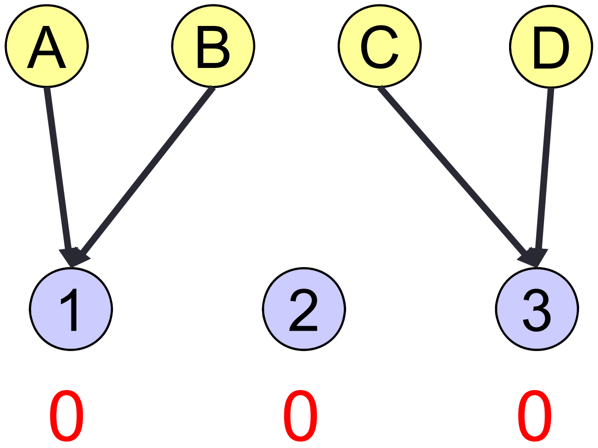
価格の増やし方の改良案

価格の増やし方

- 現状：以下の条件を満たす，極小な財集合 Z を使う

  X⊆B’に対し， X の隣接頂点 Z の数 ＜ |X|
- 効率が悪いことも．．．
  - 下記の例：財1と財3の価格を交互に増加する必要性

    同時に増やす方が反復回数が減る
- 改良案：二部グラフのDM分解を利用，財の価格をまとめて増やす



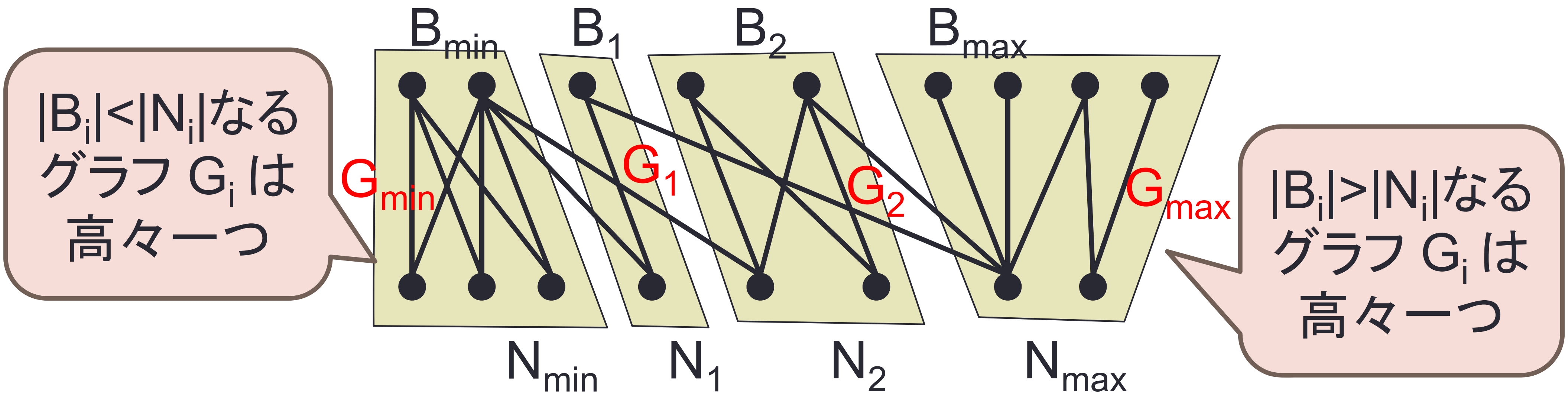
v(i,j)	A	B	C	D
①	3	7	1	0
②	0	4	7	0
③	0	4	8	4

二部グラフのDulmage-Mendelsohn分解

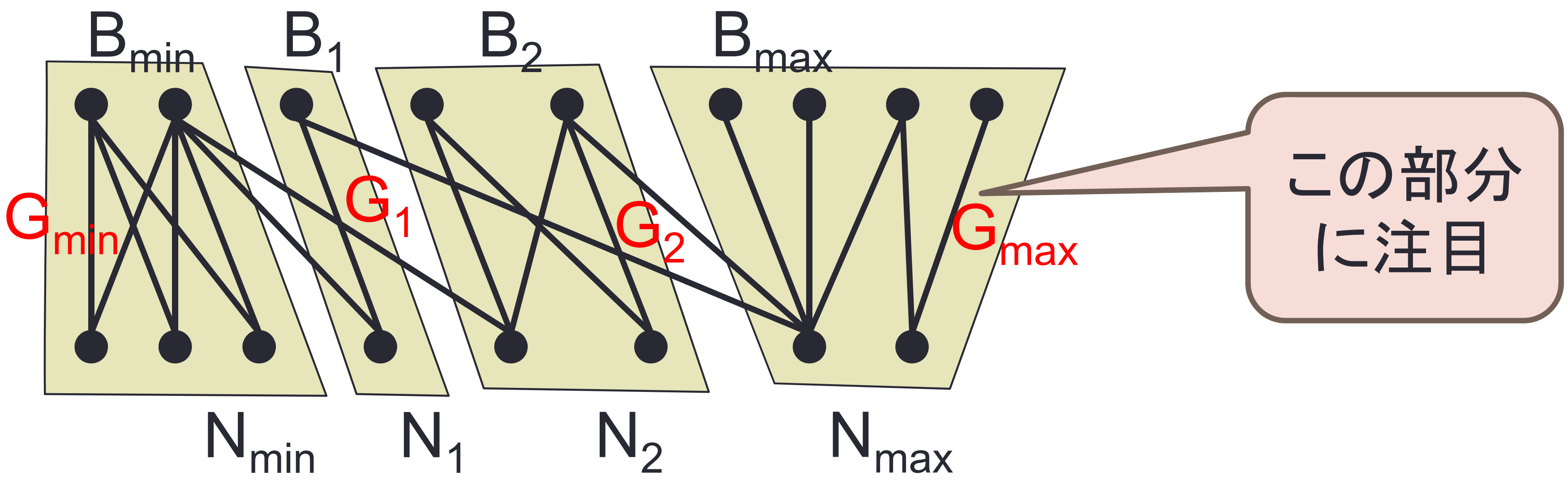
DM分解←→最大マッチング全体の構造に基づく分解
以下の条件を満たす

厳密な定義は省略

- Gの最大マッチングは，各 G_i の最大マッチングを合わせたもの
  - 異なる Gi にまたがる枝は最大マッチングに含まれない
- $i < j$ に対し，N_i と B_j を結ぶ枝は存在しない
- 最大マッチングは，N_{min}, B_{max} 以外の全頂点を必ずカバー
  - N_{min}, B_{max}の各頂点をカバーしない最大マッチングが存在



価格の増やし方の改良案

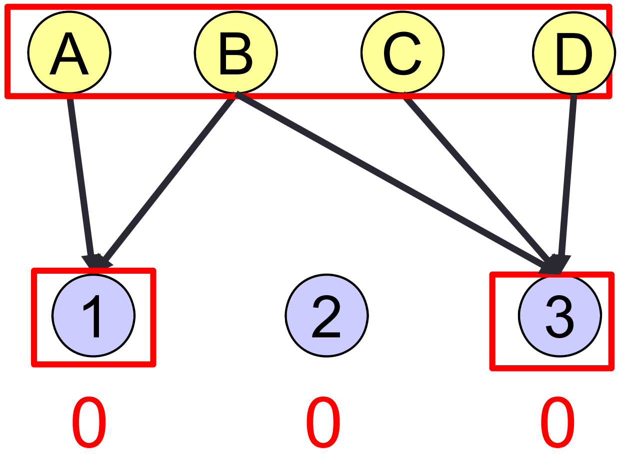


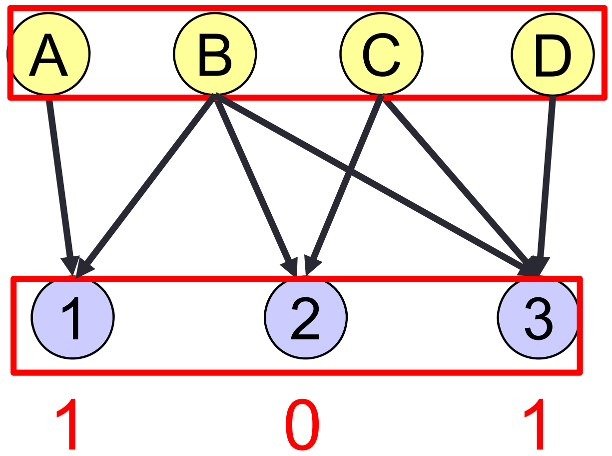
- B_{max} の入札者=最大マッチングでカバーされるとは限らない入札者
  → 隣接頂点集合 N_{max} の財の価格を上げる必要あり
  - 元々のアルゴリズムで価格増加に使ったZは，
    必ず N_{max} に含まれる

∴ 修正案：各反復で，$N_{max} \neq \emptyset$ → N_{max} の各財の価格を1増やす

アルゴリズムの実行例(2')

修正案を適用 → 1，2回目の価格更新がひとまとめになる





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

利得	A	B	C	D
①	2	6	0	-1
②	1	6	7	0
③	-1	6	7	3

改良による反復回数の減少

- 価格増加する財の選び方の変更により，

  アルゴリズムの反復回数が減少（増えない）
- 極小均衡価格=(p*(1),…,p*(n))のとき，

  アルゴリズムの反復回数 ≧ max p*(j)

定理[Andersson-Erlanson(2013), Murota-Shioura-Yang(2013)]：

改良後のアルゴリズムの反復回数＝ max p*(j)

つまり，改良によって，

反復回数が最小のアルゴリズムになった

演習問題

下記のように評価値が与えられたとき，
均衡を厳密に計算するアルゴリズムを適用して
均衡価格を計算せよ.

問1

v(i,j)	A	B	C
①	2	3	6
②	6	7	7

問2

v(i,j)	A	B	C
①	3	1	0
②	7	6	7
③	1	7	8

問3

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