数理経済学特講

複数財に対するオークションの数理とアルゴリズム

第4回　二部グラフのマッチング問題

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

マッチングと交互路

マッチングに関する交互路

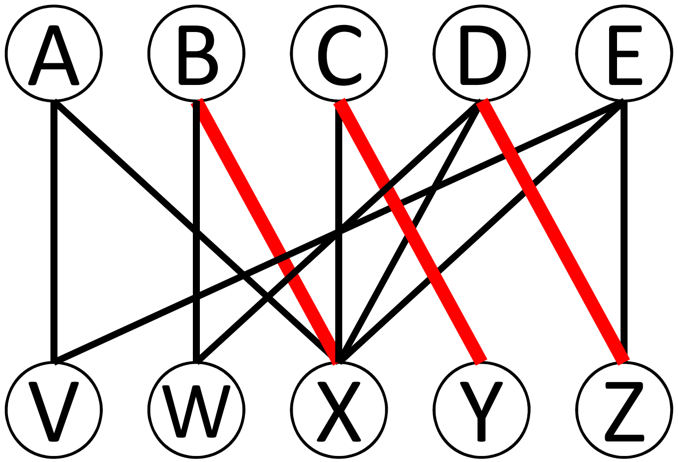
交互路 – 新しいマッチングを作るための「道具」

定義：マッチング Mに関する**交互路**

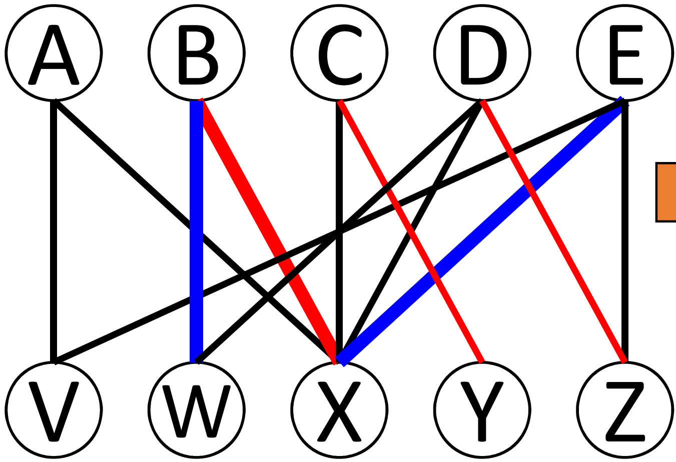
(i) Mの枝とM以外の枝が交互に現れる路

(ii) 路の最初の枝がM以外➔路の始点は，Mの枝の端点ではない

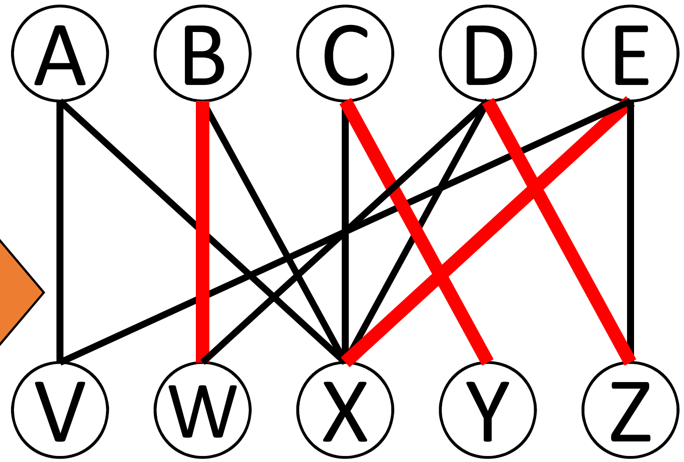
(iii) 路の最後の枝がM以外➔路の終点は，Mの枝の端点ではない



マッチング M



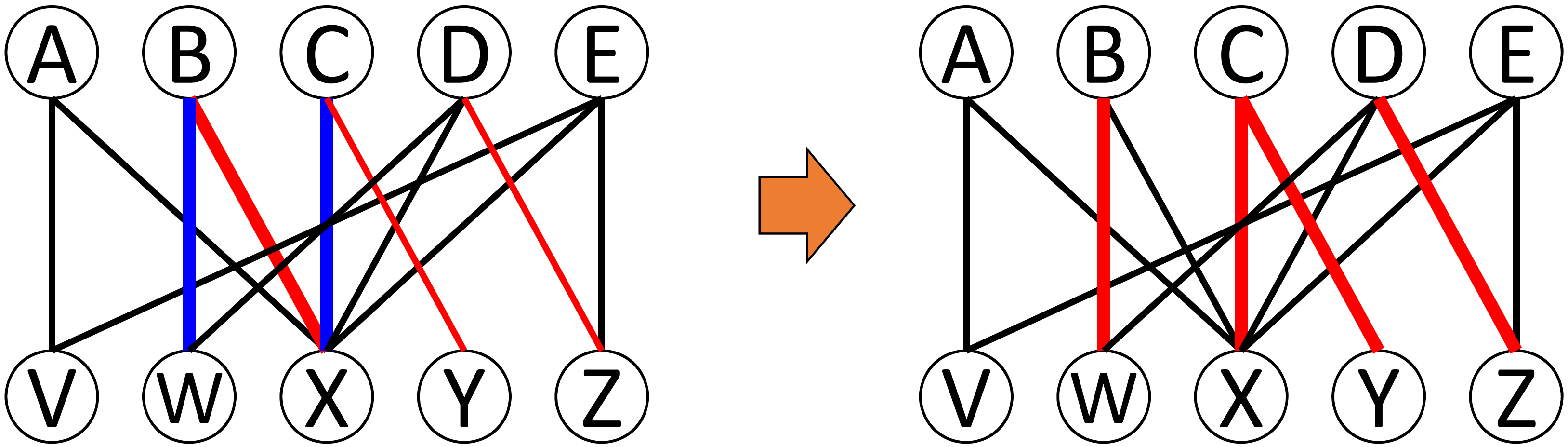
交互路



交互路を使って，
マッチングの枝を
入れ替えした結果

マッチングに関する交互路

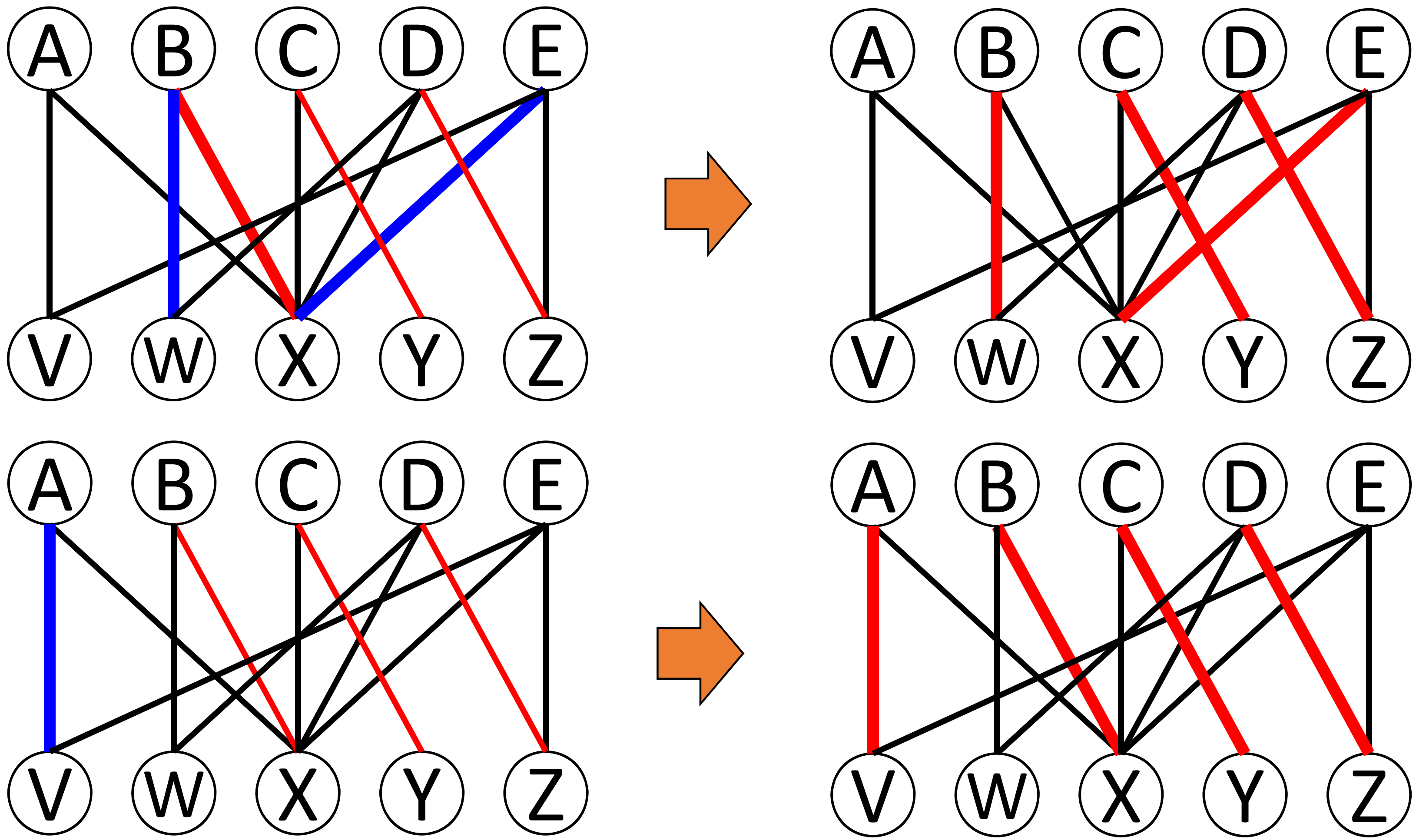
交互路でない例



条件(ii)を満たさない
➔枝を入れ替えても
マッチングにならない

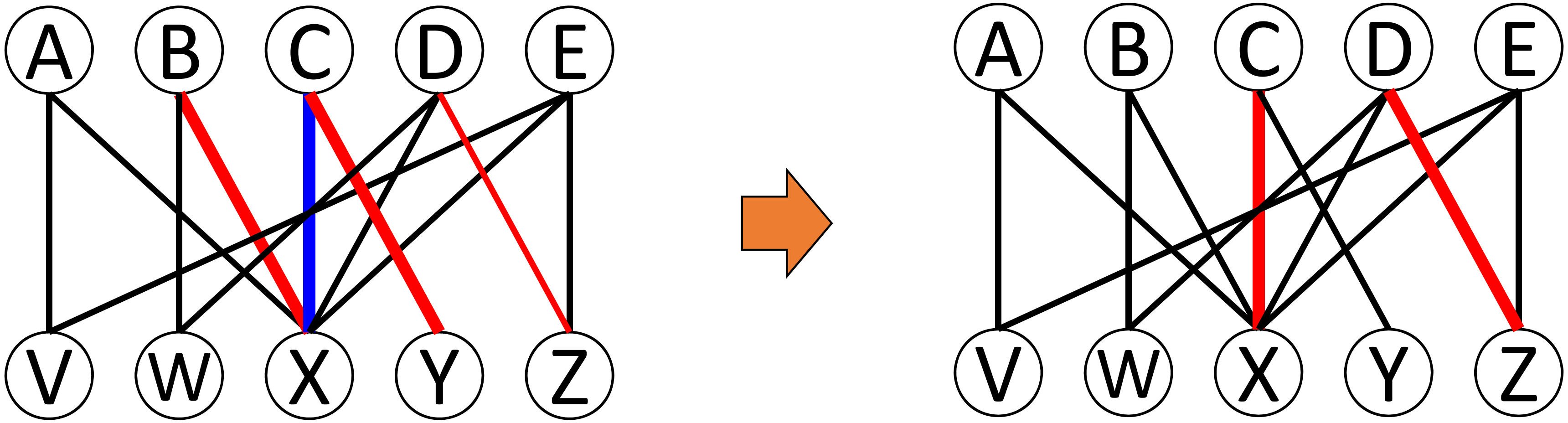
マッチングに関する交互路

交互路タイプ1：M以外の枝数＝Mの枝数＋1
（枝を入れ替えるとマッチングの枝数が1増える）
➔ 増加路と呼ばれる

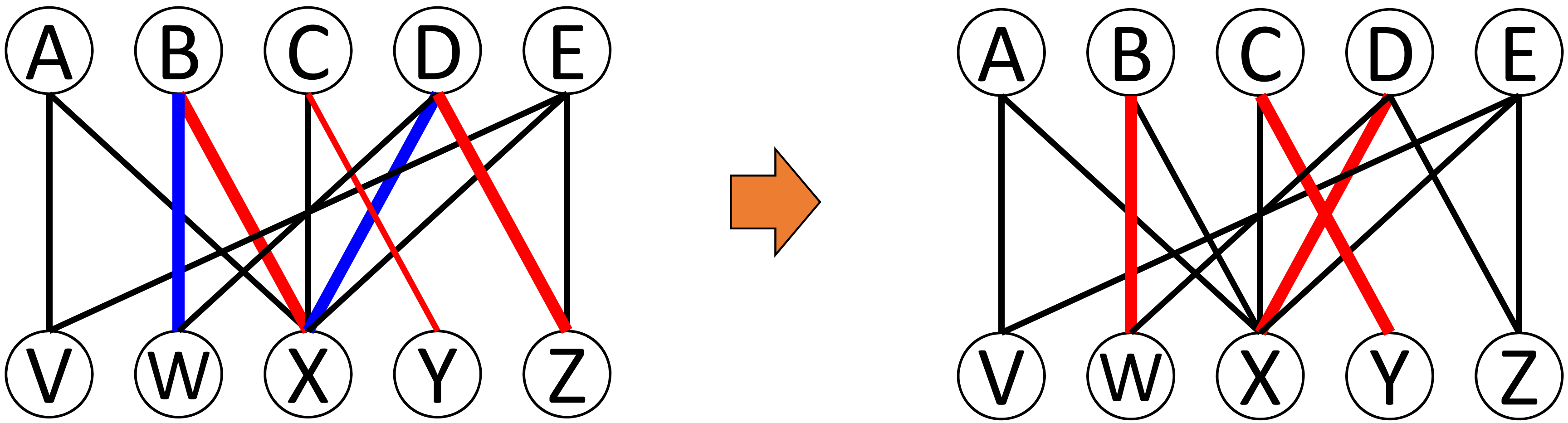


マッチングに関する交互路

交互路タイプ2：M以外の枝数＝Mの枝数−1
（枝を入れ替えるとマッチングの枝数が1減る「減少路」）



交互路タイプ3：M以外の枝数＝Mの枝数
（枝を入れ替えても，マッチングの枝数は不変）

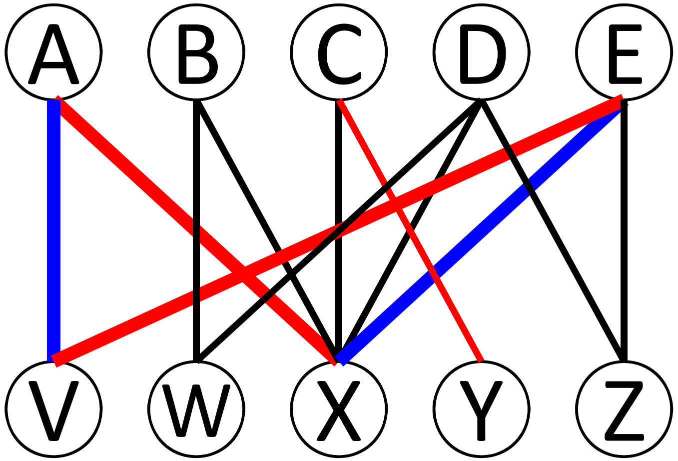


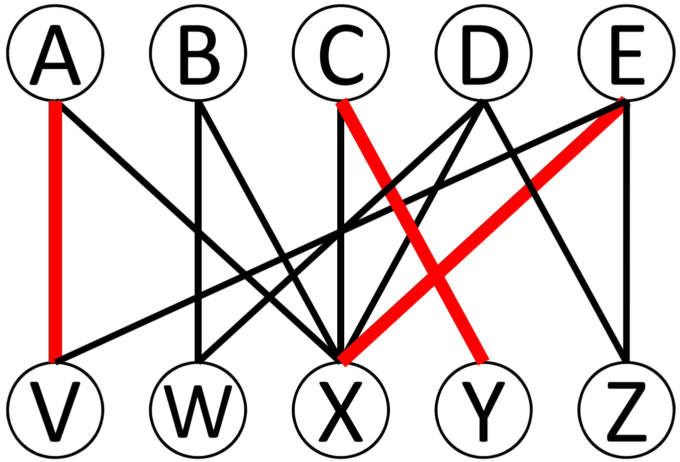
マッチングに関する交互閉路

定義：マッチング Mに関する交互閉路
　　　＝Mの枝とM以外の枝が交互に現れる閉路

交互閉路では，「M以外の枝数＝Mの枝数」
➔枝を入れ替えても，マッチングの枝数は不変

交互閉路の例





交互路によるマッチングの更新

命題 マッチングMに対し,
路Pは(Mに関する)交互路
⇔Pを使ってMの枝を入れ替えて得られる
枝集合 $M' = (M \setminus P) \cup (P \setminus M)$ はマッチング

命題 マッチングMに対し,
Mに関する**増加路**が存在➔Mは最大マッチングではない
(対偶:Mは最大マッチング➔ Mに関する**増加路**が存在しない)

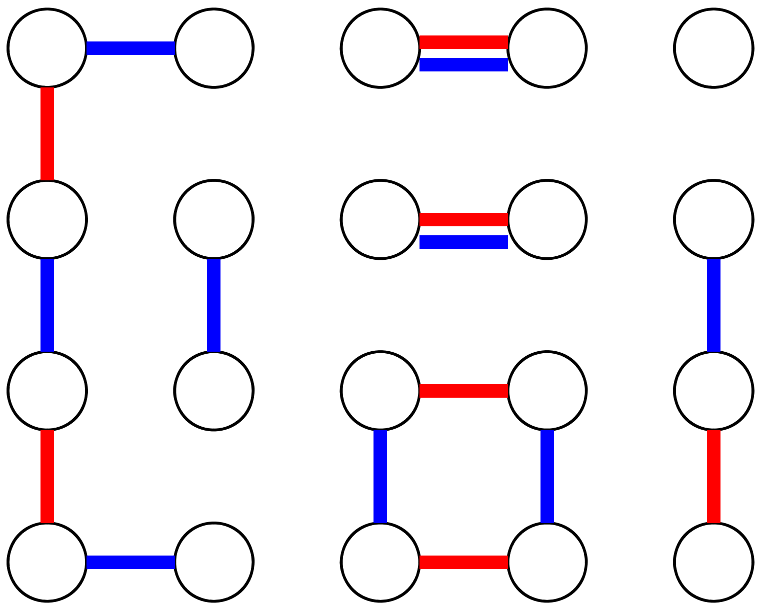
逆も成立

定理 マッチングMに対し,
Mに関する**増加**路が存在⇔Mは最大マッチングではない
(Mは最大マッチング⇔ Mに関する**増加路**が存在しない)

増加路なし➔最大マッチング　の証明

命題 マッチングMに対し,
Mに関する**増加路**が存在🡐Mは最大マッチングではない
(対偶：Mは最大マッチング🡐 Mに関する**増加路**が存在しない)

(証明)最大マッチングをM*とおく➔|M*| > |M|
枝集合$(M^* \setminus M) \cup (M \setminus M^*)$に注目
➔ 幾つかの(Mに関する)
　交互路・交互閉路からなる
(∵各頂点にはMの枝が高々1つ,
　M*の枝が高々一つ，接続しているから)
その中に，増加路が必ず存在
(∵タイプ2，3の交互路，交互閉路は
　いずれも「Mの枝数≦M*の枝数」
　➔合計すると，|M*|≦|M|となり矛盾) ■
※ 減少路は存在しない(∵M* にとっての増加路に対応するから)



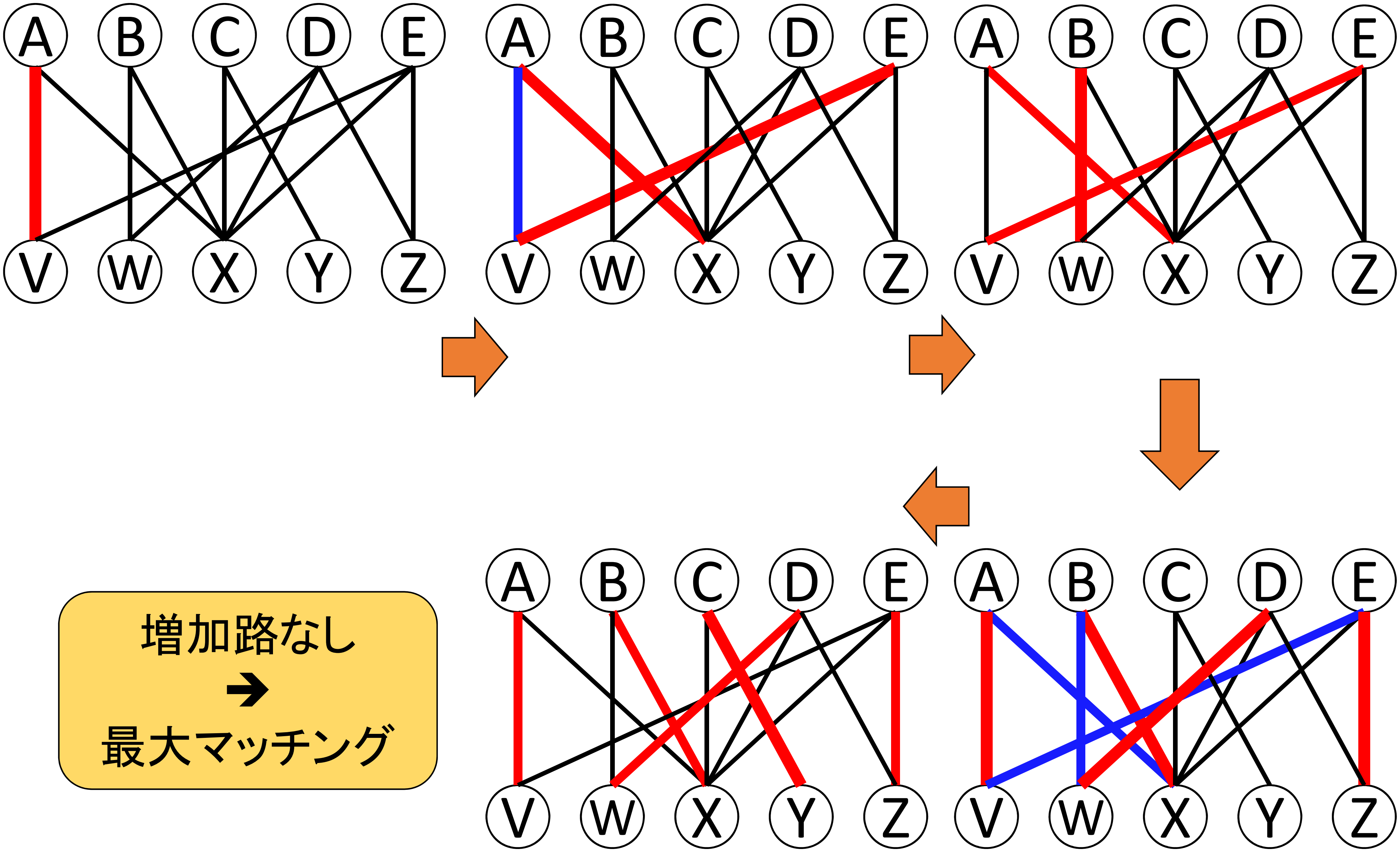
最大マッチングの計算

最大マッチング問題のアルゴリズム

「Mは最大マッチング🡄🡆Mに関する増加路が存在しない」
これに基づき，次のアルゴリズムが得られる

ステップ0：初期マッチングを$M := \emptyset$とする.
ステップ1： Mに関する増加路の存在をチェック.
存在しない🡆現在のMは最大マッチング（終了）
存在する🡆ステップ2へ.
ステップ2：Mに関する増加路 P をひとつ見つけ,
P を用いて M を更新（ $M := (M \setminus P) \cup (P \setminus M)$）
ステップ1へ.

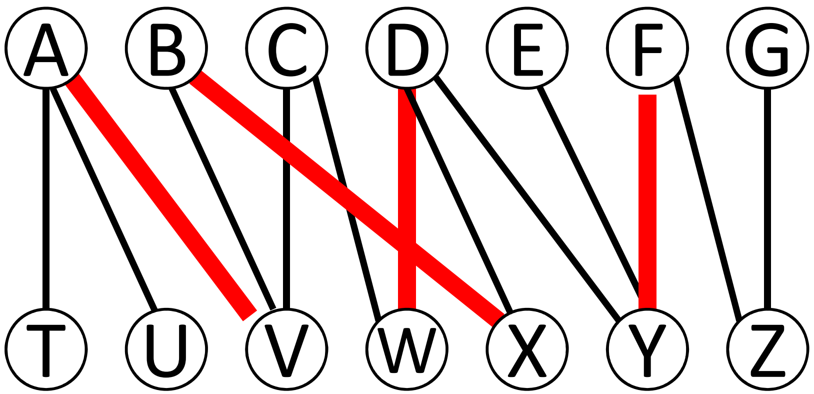
アルゴリズムの実行例

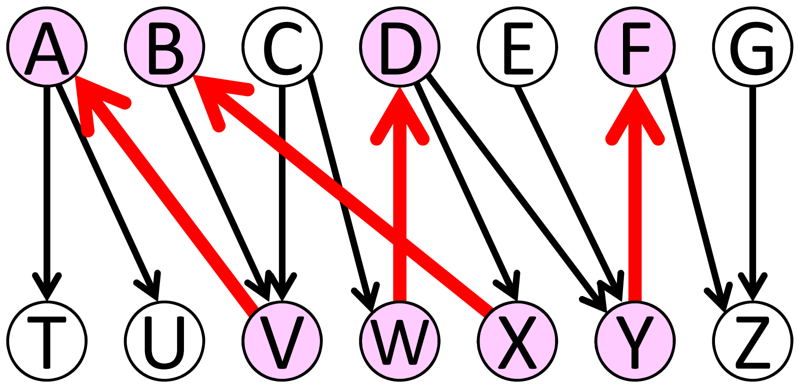


増加路の見つけ方

- 増加路を見つける問題は，有向路を見つける問題に帰着できる
- 二部グラフの頂点集合が B, N に分かれているとき，
  - マッチングMの枝は，N の頂点から B の頂点に向き付け
  - それ以外の枝は，B の頂点から N の頂点に向き付け
- B の頂点のうち，
  - マッチングの枝が接続している頂点の集合→U_1
  - 残りの頂点の集合→T_1
- U_2, T_2 も N に関して同様に定義

ピンクの頂点 : U_1とU_2
白い頂点 : T_1とT_2

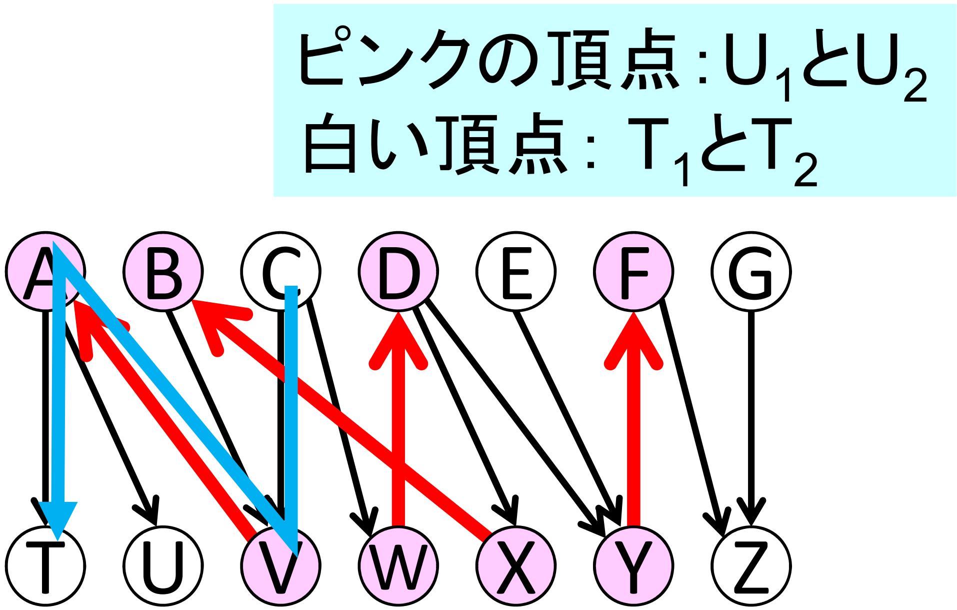




増加路の見つけ方

- 新しい有向グラフにおける T_1 の頂点から T_2 の頂点への有向路
  ←→元の二部グラフの（Mに関する）増加路

∴有向グラフにおいて，上記のような有向路を求めればよい
（有向路を見つける効率的なアルゴリズムは幾つか存在）



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

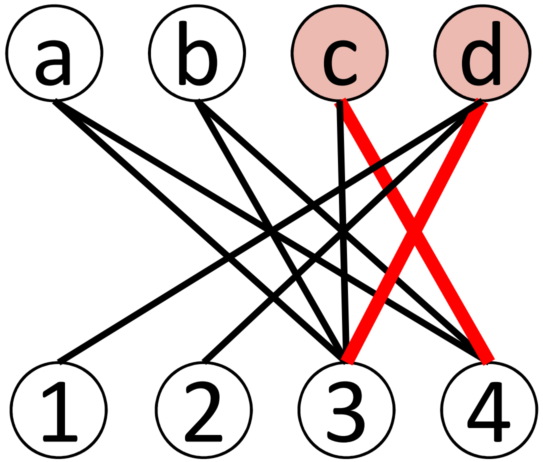
命題1

(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

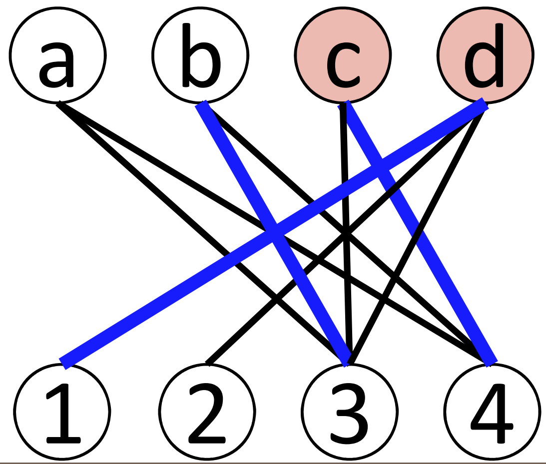
➔B’をカバーする最大マッチングが存在

(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

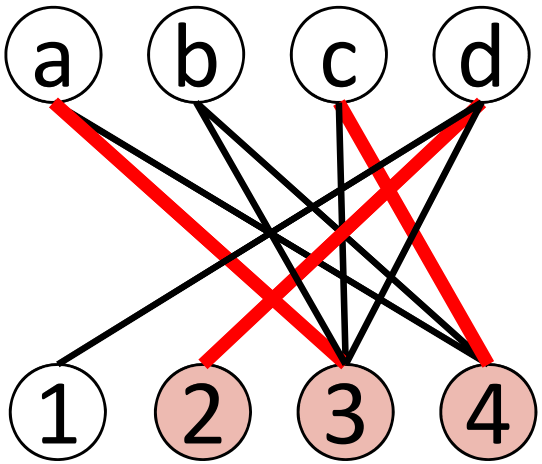
カバーする頂点の制約(片側)

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

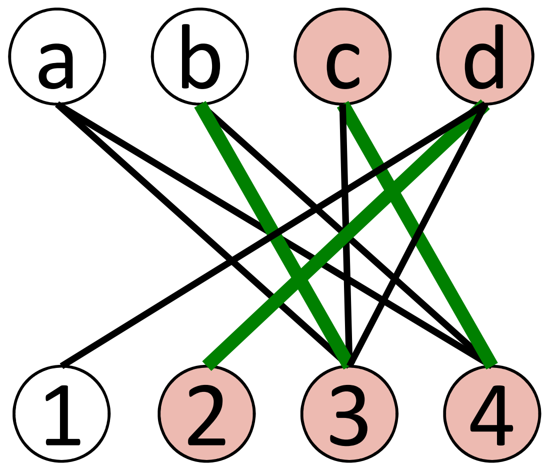
[証明] M: B' をカバーするマッチング
M に増加路アルゴリズムを適用し，最大マッチング M* を求める
増加路アルゴリズムの各反復において，
マッチングのカバーする頂点は単調増加(減らない)
∴ M*のカバーする頂点 ⊇ M のカバーする頂点 ⊇ B' ■

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



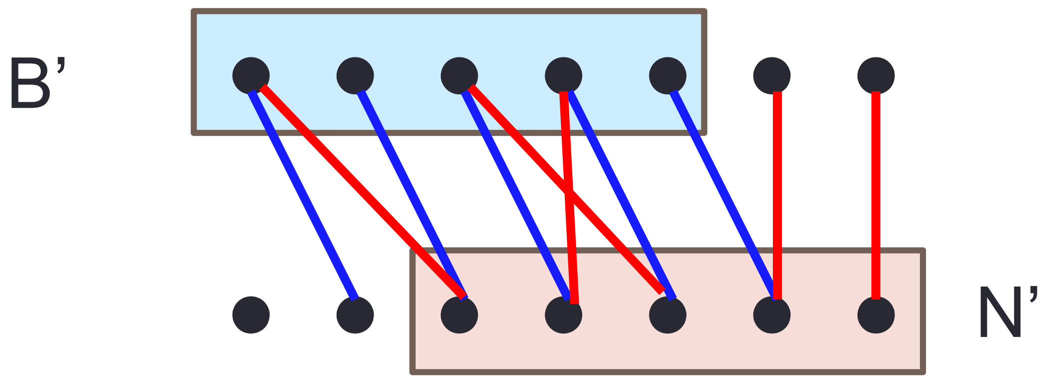
このマッチングは
B’={c,d}, N’={2,3,4}を同時に
カバーする最大マッチング

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]

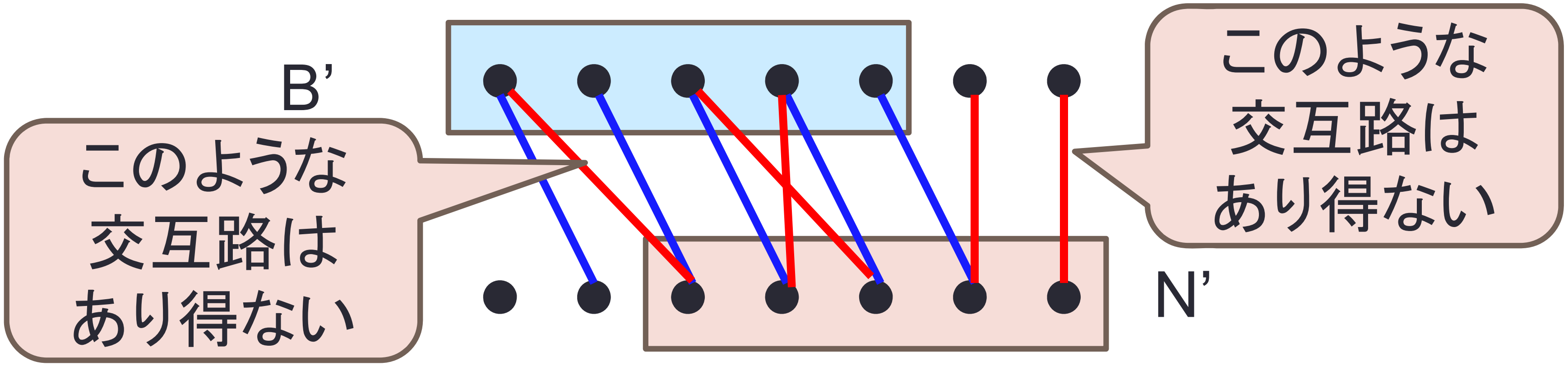
- M: B’ をカバーする最大マッチング(命題1(i)より存在)
- そのようなマッチングの中で，カバーする N’ の頂点数が最大とする
- M が N’ すべてをカバーしていないと仮定 → 矛盾を導く
- M*: N’ すべてをカバーする最大マッチング

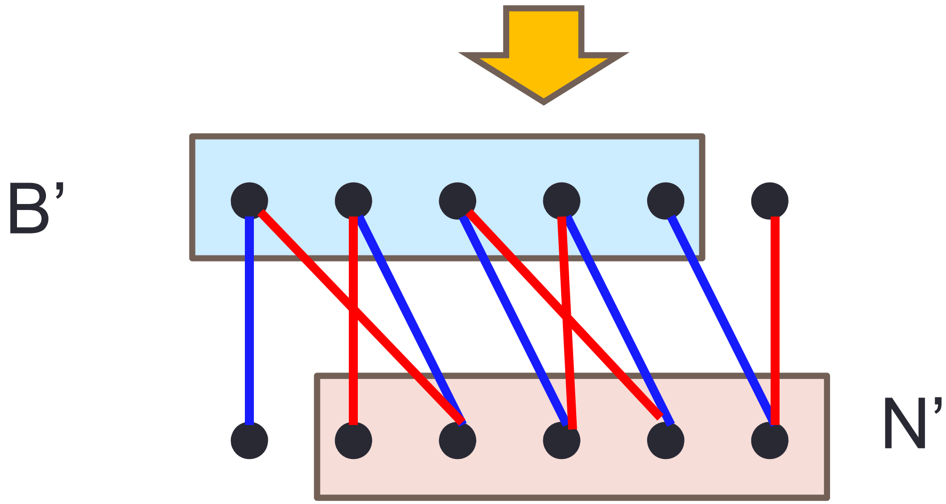


カバーする頂点の制約（両側）

[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）



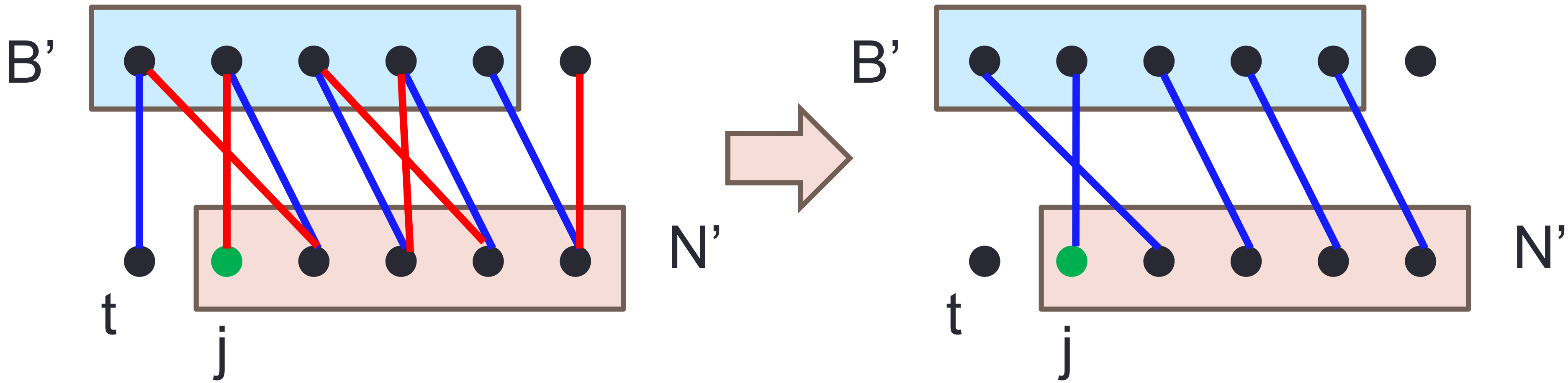


∴交互路の枝数は
すべて偶数

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



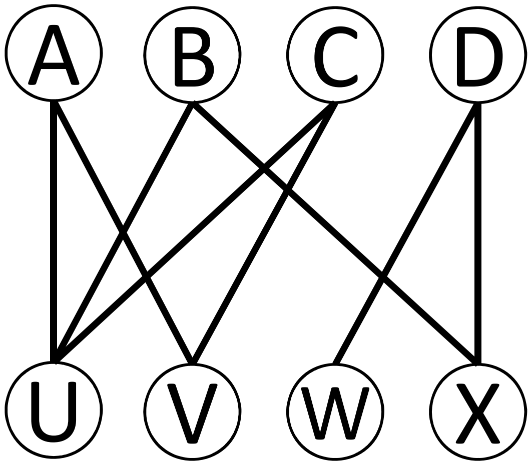
- j はマッチング M* でカバーされている
  → j から始まる交互路 P が存在, M* の枝から始まる
- 交互路 P の枝数は偶数
  → 最後の枝は M, 最後の頂点 t は N' に含まれない
  (∵ t∈N' → t に接続する M* の枝が存在 → 交互路は終わらない)

∴ 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)

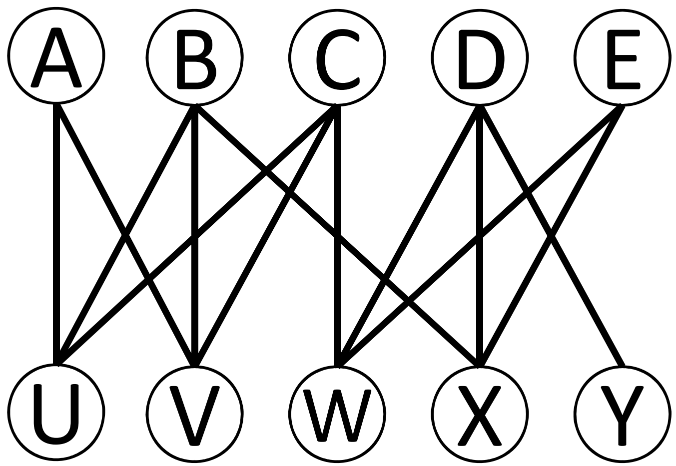
演習問題

問1: 以下のグラフの最大マッチングを，前のスライドのアルゴリズムにより求めよ．各反復でのマッチングと使った増加路を（図を使って）明記すること．
（増加路の求め方は説明不要）

（1）



（2）



演習問題

問2: 下記のように評価値と財の価格（赤い数字）が与えられたとき，
均衡価格か否か判定したい．
（1）(i), (ii), (iii) それぞれに対し，判定のための二部グラフを書け．
　　また，頂点集合B’, N’ を明記せよ．
（2）頂点集合 B’ および N’ をカバーするマッチングが存在するか否か，調べよ．
　　存在する場合は具体的に求めること．存在しない場合はその理由を述べよ．

(i)

	v(i,j)	A	B	C
3	①	2	3	6
6	②	6	7	7

(ii)

	v(i,j)	A	B	C
1	①	3	1	0
5	②	7	6	7
6	③	1	7	8

(iii)

	v(i,j)	A	B	C
0	①	3	1	0
5	②	7	6	7
6	③	1	7	8
0	④	0	0	4