

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

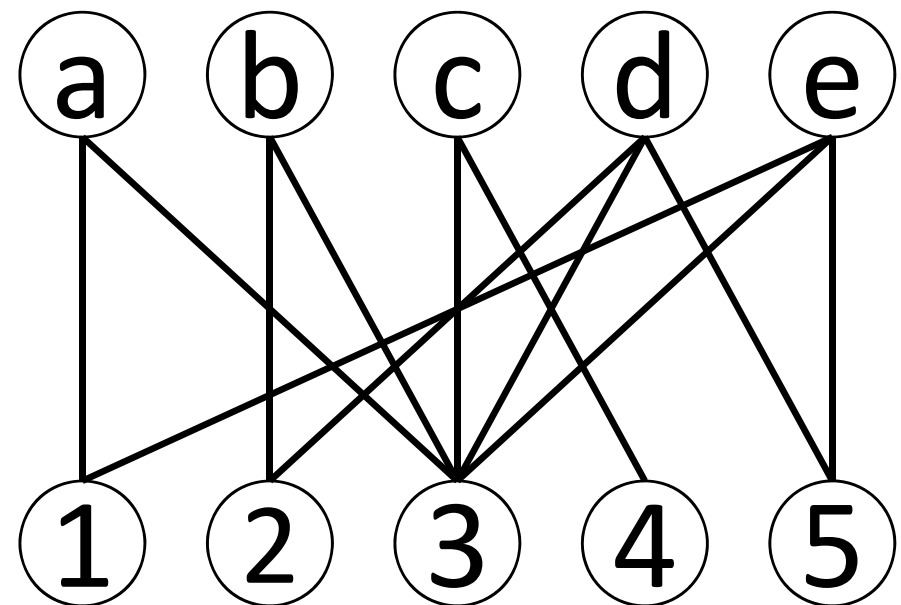
shioura.a.aa@m.titech.ac.jp

二部グラフのマッチング問題

- (グラフ理論における) マッチング問題:
「ひと」や「もの」の間の最適なペアの集合を求める問題
- 例: 入札者への財の割り当て
 - 5人の入札者 $B=\{a, b, c, d, e\}$ 5つの財 $N=\{1, 2, 3, 4, 5\}$
 - 各々の入札者は欲しい財と欲しくない財がある
→ グラフで表現可能

頂点の2つのグループ
B と N の間にのみ, 枝が存在
← **二部グラフ**と呼ばれる

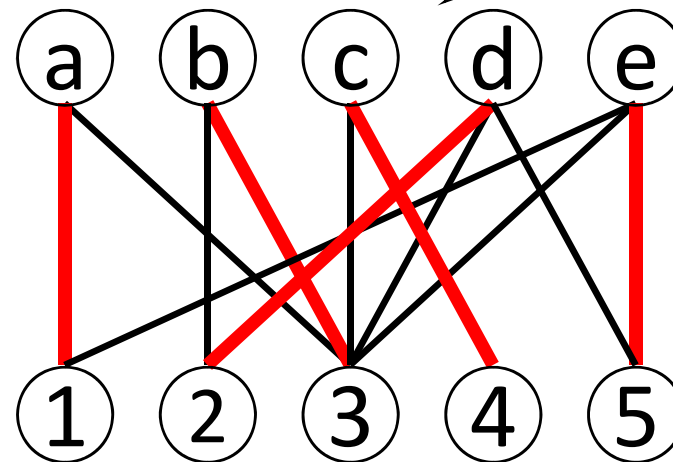
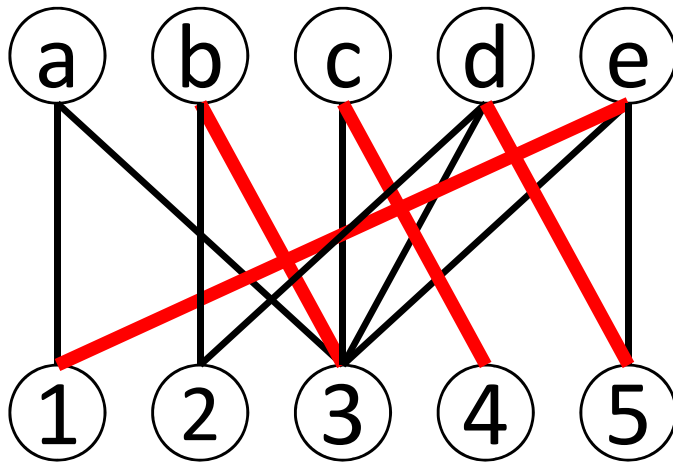
※2部グラフにおいて, B と N の
頂点数は同じでなくても良い



二部グラフの最大マッチング問題

- できるだけ多くの入札者に財を割り当てたい
 - 各入札者には, 高々1つの財が割り当て可能
 - 各財は, 高々一人の入札者に割り当て可能
- ➔ 財の割当を枝集合で表現可能

最大
マッチング



- (グラフの) マッチング:
グラフの枝集合であって, 各頂点に接続する枝の数(次数)が1以下
- 最大(サイズ)マッチング問題:
与えられたグラフのマッチングの中で, 枝数最大のものを求める

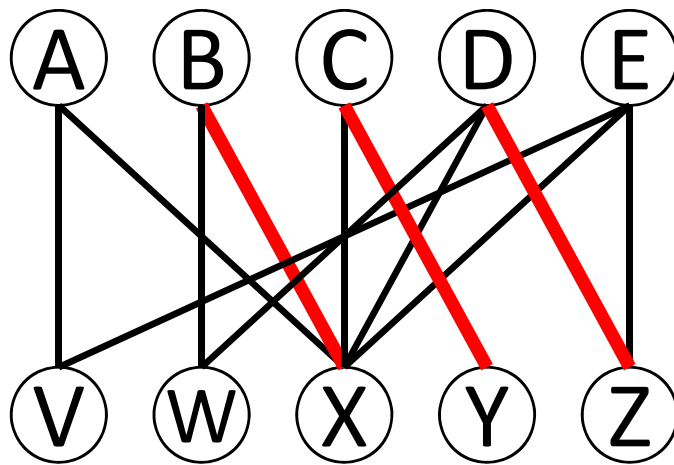
マッチングと交互路

マッチングに関する交互路

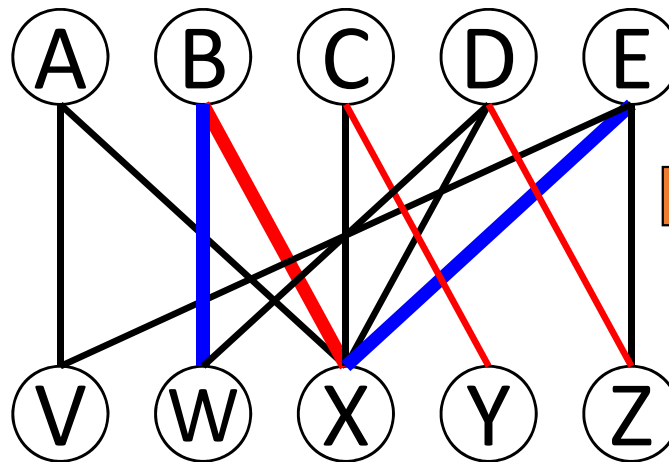
交互路 – 新しいマッチングを作るための「道具」

定義: マッチング M に関する**交互路**

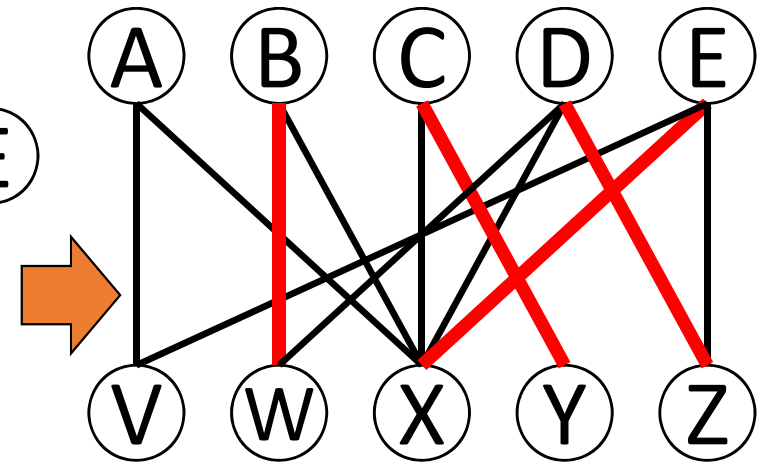
- (i) M の枝と M 以外の枝が交互に現れる路
- (ii) 路の最初の枝が M 以外 $\rightarrow$ 路の始点は, M の枝の端点ではない
- (iii) 路の最後の枝が M 以外 $\rightarrow$ 路の終点は, M の枝の端点ではない



マッチング M



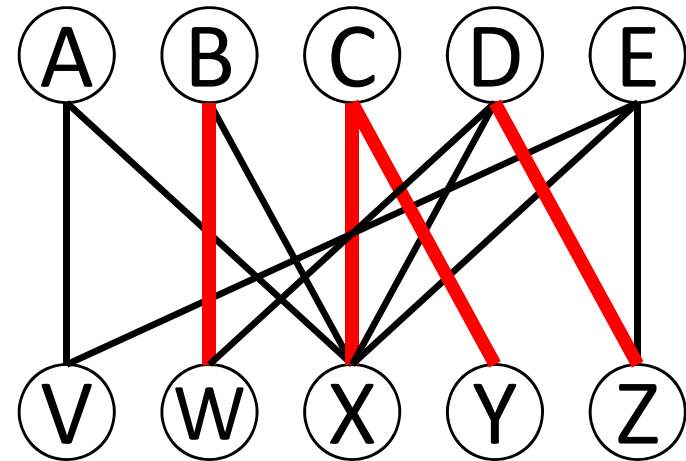
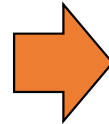
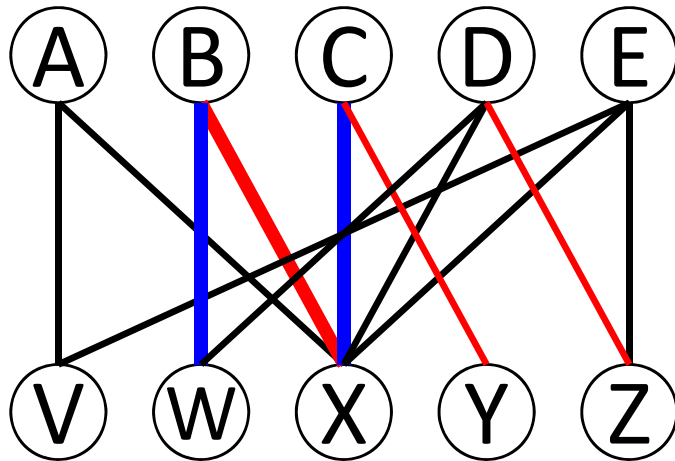
交互路



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

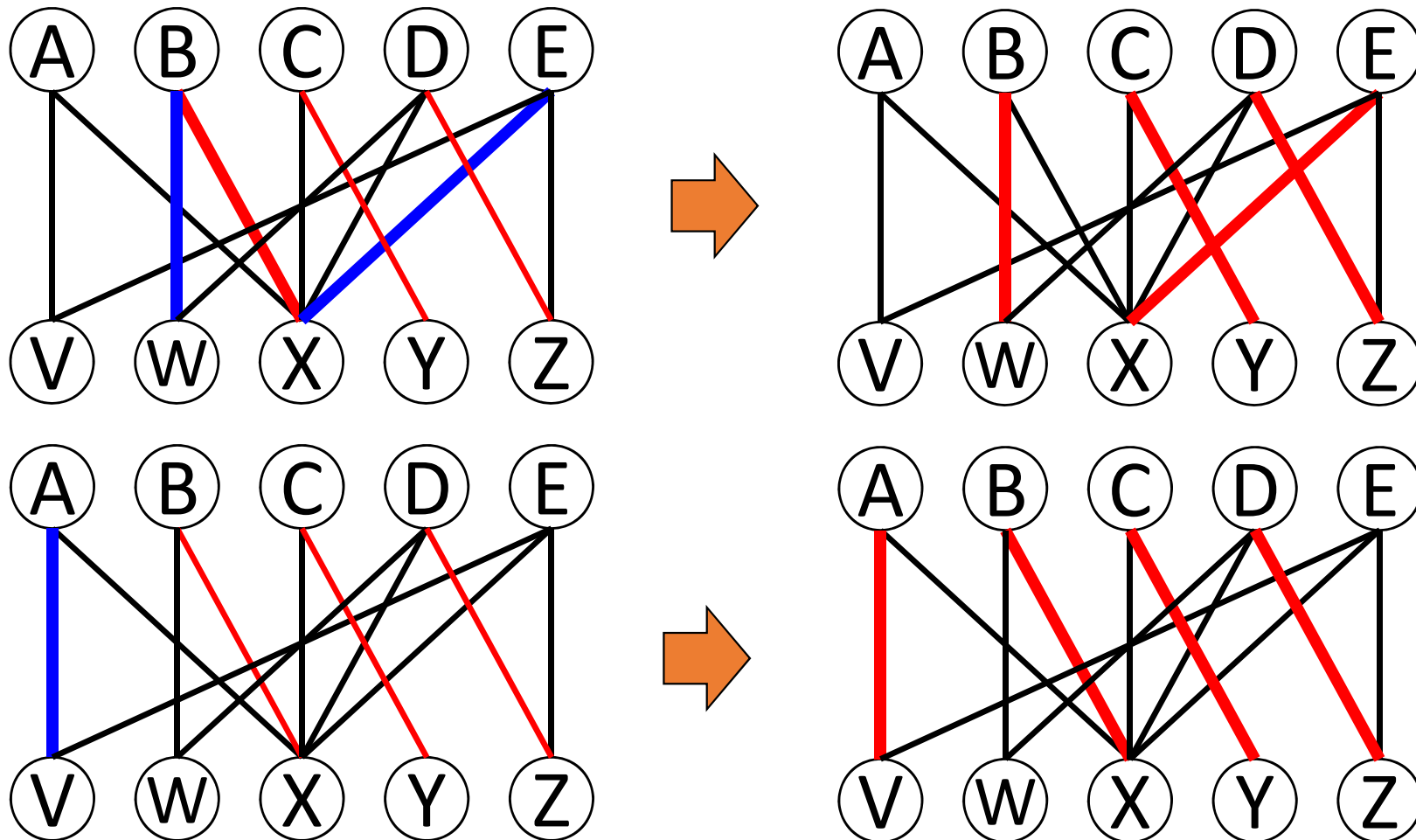


条件(ii)を満たさない
 ➡ 枝を入れ替えても
 マッチングにならない

マッチングに関する交互路

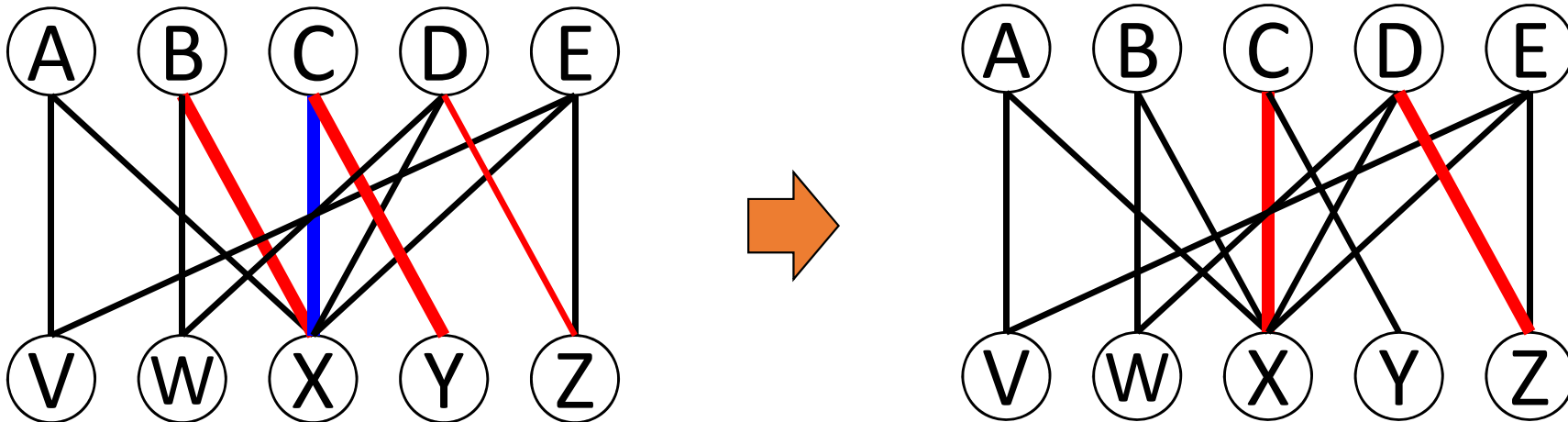
交互路タイプ1: M 以外の枝数 = M の枝数 + 1
 (枝を入れ替えるとマッチングの枝数が1増える)

→ 増加路と呼ばれる

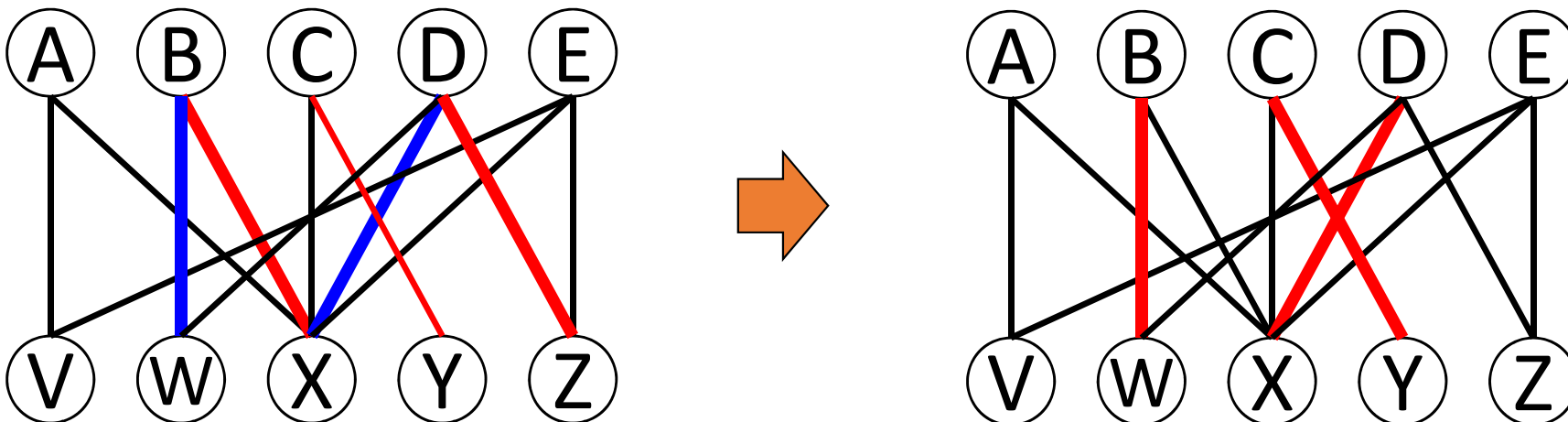


マッチングに関する交互路

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、マッチングの枝数是不変)



マッチングに関する交互閉路

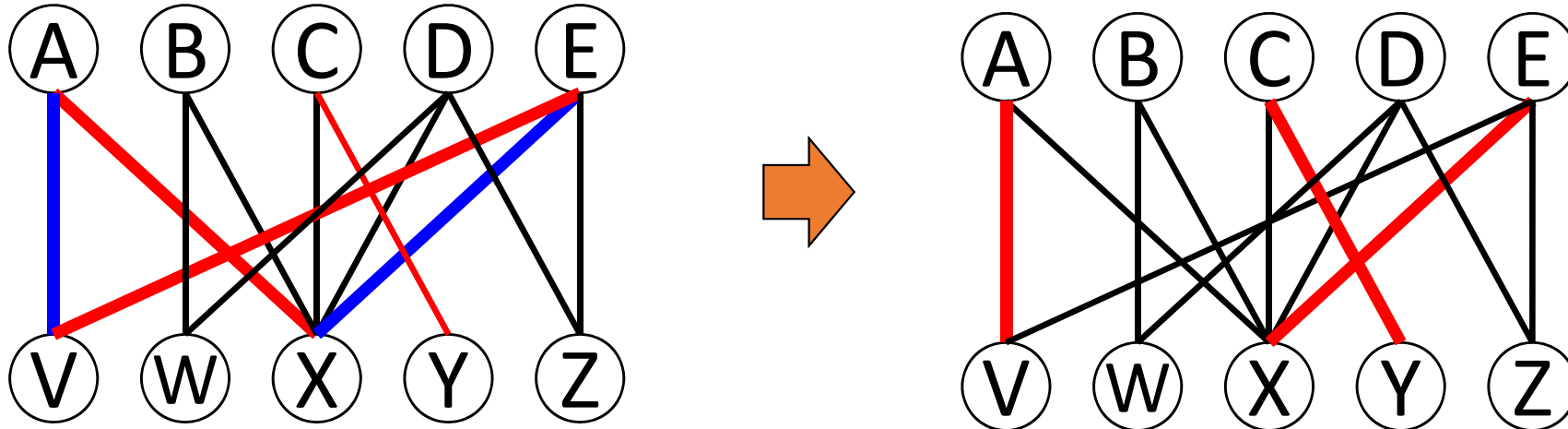
定義: マッチング M に関する**交互閉路**

= 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、マッチングの枝数是不変

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
 $\longleftrightarrow P$ 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 $\rightarrow M$ は最大マッチングではない
 (対偶: M は最大マッチング $\rightarrow M$ 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 $\longleftrightarrow M$ は最大マッチングではない
 (M は最大マッチング $\longleftrightarrow M$ に関する **増加路** が存在しない)

増加路なし $\rightarrow$ 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 $\leftarrow M$ は最大マッチングではない
 (対偶: M は最大マッチング $\leftarrow M$ 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 $\rightarrow |M^*| > |M|$

枝集合 $(M^* \setminus M) \cup (M \setminus M^*)$ に注目

$\rightarrow$ 幾つかの (M に関する)

交互路・交互閉路からなる

($\because$ 各頂点には M の枝が高々1つ,

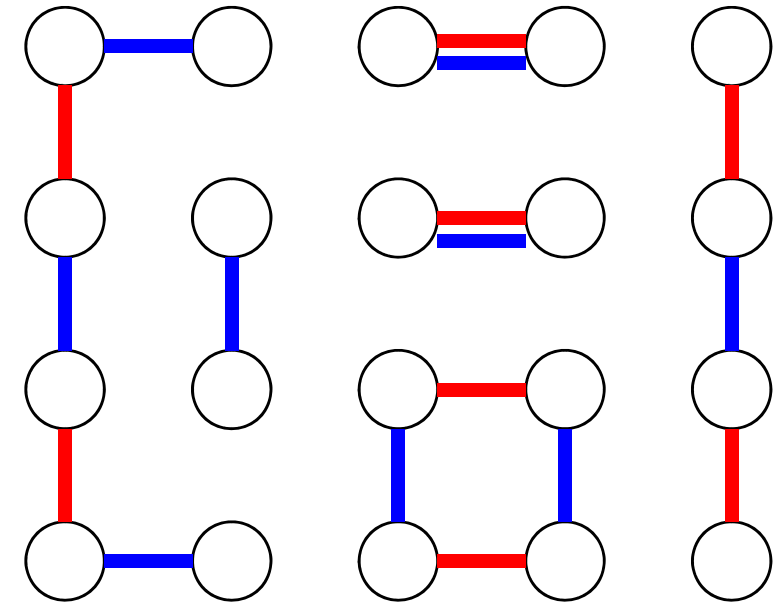
M^* の枝が高々一つ, 接続しているから)

その中に, 増加路が必ず存在

($\because$ タイプ2, 3の交互路, 交互閉路は

いずれも「 M の枝数 $\leq M^*$ の枝数」

$\rightarrow$ 合計すると, $|M^*| \leq |M|$ となり矛盾)



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 $\iff M$ 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 $\rightarrow$ 現在の M は最大マッチング(終了)

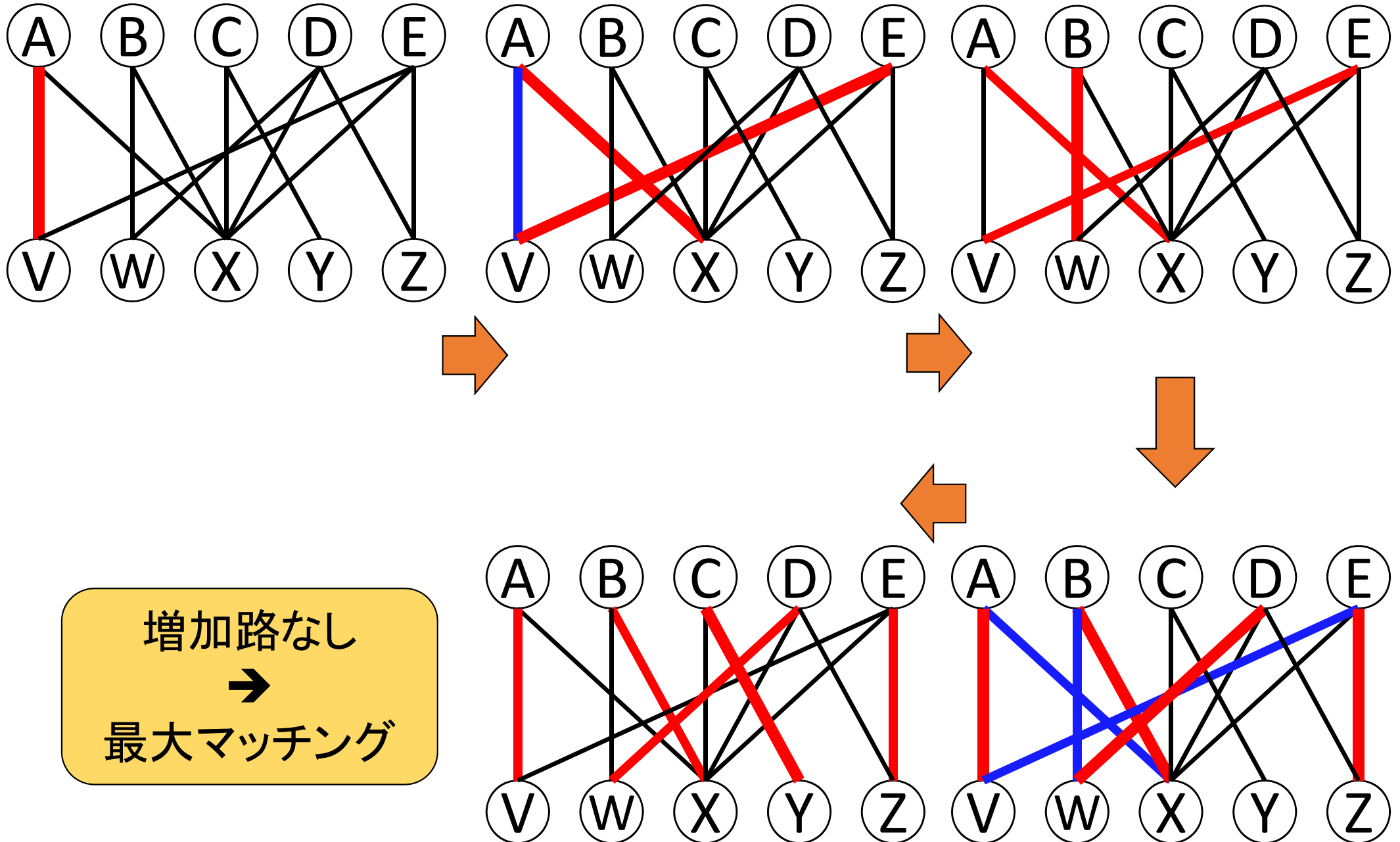
存在する $\rightarrow$ ステップ2へ.

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.

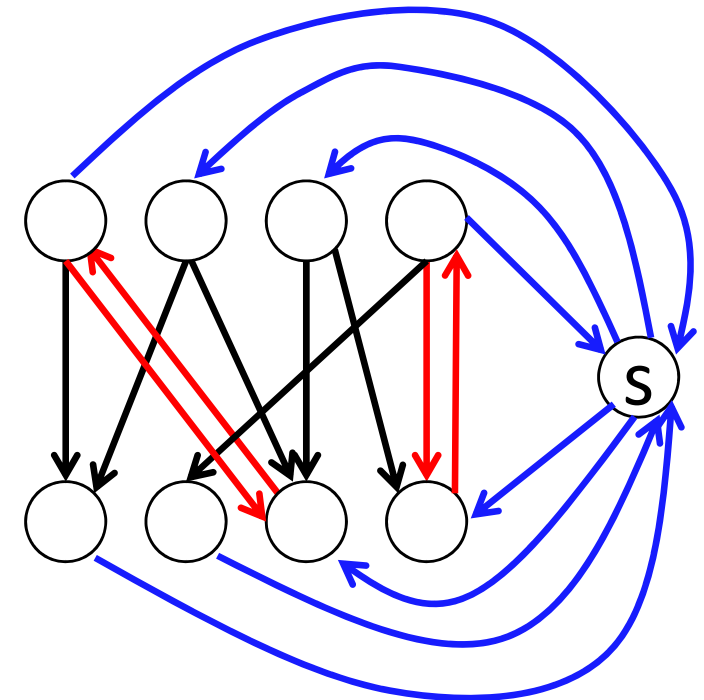
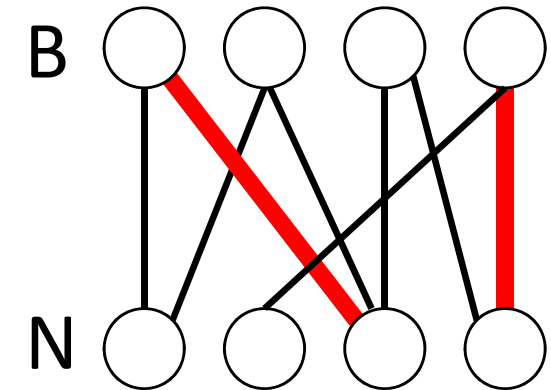
アルゴリズムの実行例



増加路の見つけ方：準備

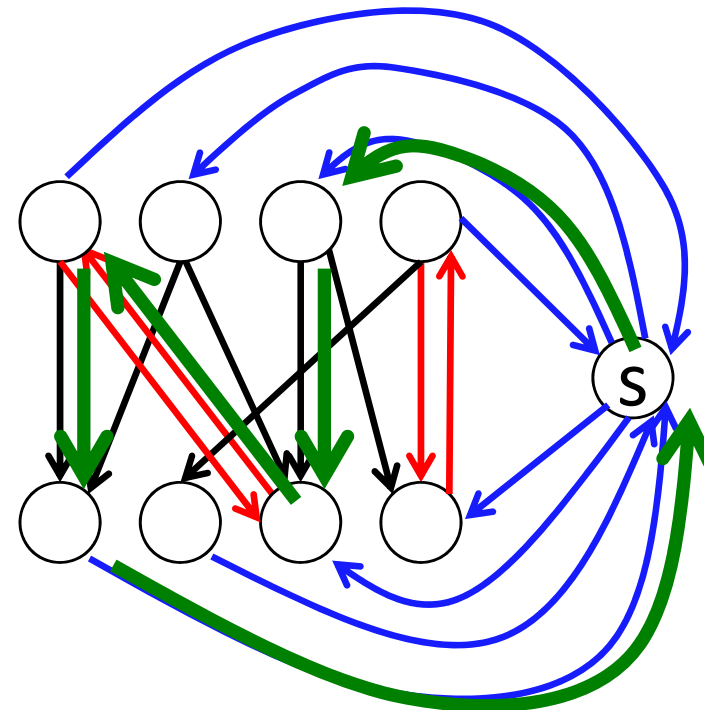
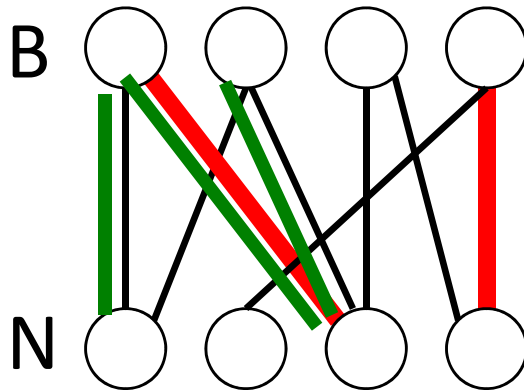
増加路の検出→ある有向グラフでの
有向(閉)路の検出に帰着可能

- 全ての枝を, B 側から N 側に向き付け
- マッチング M の各枝に対し, さらに
逆向きの枝 ($N \rightarrow B$) を追加
- 新たな頂点 s をおく
- B 側の各頂点 u に対し,
 - マッチング枝が接続 $\rightarrow$ 枝 (u, s) をおく
 - 接続なし $\rightarrow (s, u)$ をおく
- N 側の各頂点 v に対し,
 - マッチング枝が接続 $\rightarrow$ 枝 (s, v) をおく
 - 接続なし $\rightarrow (v, s)$ をおく



増加路の見つけ方

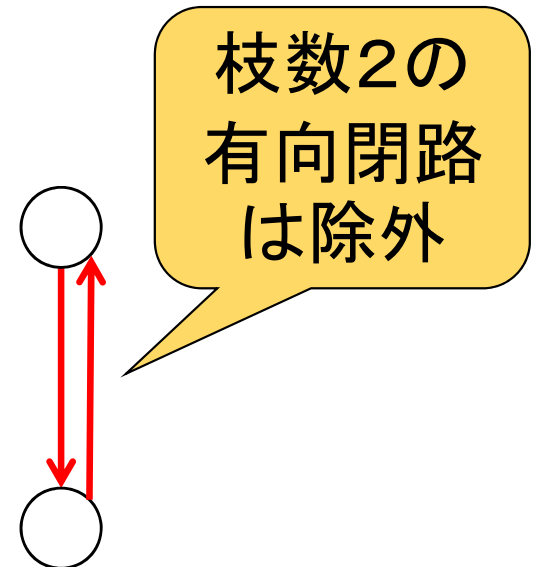
- 新しい有向グラフにおいて,
 頂点 $s \rightarrow B$ 側の頂点 $\rightarrow \dots \rightarrow N$ 側の頂点 $\rightarrow$ 頂点 s の形の有向閉路
 $\longleftrightarrow$ 元の二部グラフの (M に関する) 増加路
 $\therefore$ 有向グラフにおいて, 上記のような有向閉路を求めればよい



有向閉路と交互路の関係

- 新しい有向グラフの有向閉路(のほとんど)は,
元の二部グラフの交互路と対応

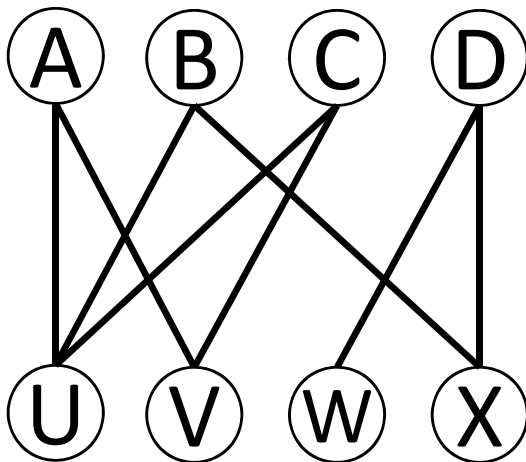
- (1) 頂点 $s \rightarrow B$ 側の頂点 $\rightarrow \dots \rightarrow N$ 側の頂点 $\rightarrow$ 頂点 s の形の有向閉路
 $\longleftrightarrow$ 元の二部グラフの増加路(タイプ1)
- (2) 頂点 $s \rightarrow N$ 側の頂点 $\rightarrow \dots \rightarrow B$ 側の頂点 $\rightarrow$ 頂点 s の形の有向閉路
 $\longleftrightarrow$ 元の二部グラフの「減少路」(タイプ2)
- (3) 上記以外で, 頂点 s を通過する有向閉路
 $\longleftrightarrow$ 元の二部グラフのタイプ3の交互路
- (4) 頂点 s を通過しない(枝数4以上の)有向閉路
 $\longleftrightarrow$ 元の二部グラフの交互閉路



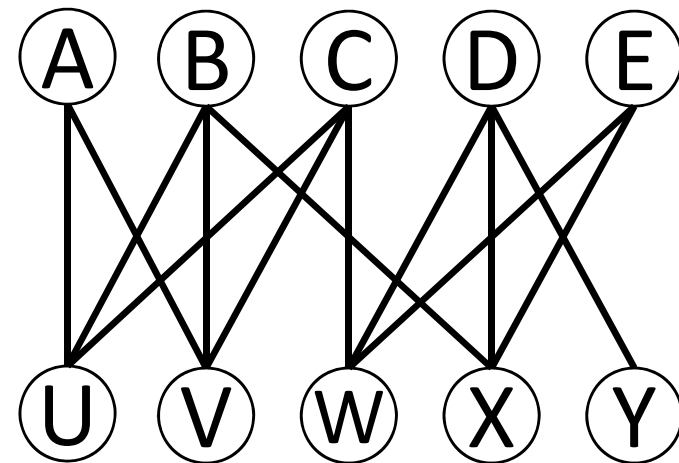
演習問題

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, 説明する必要なし)

(1)



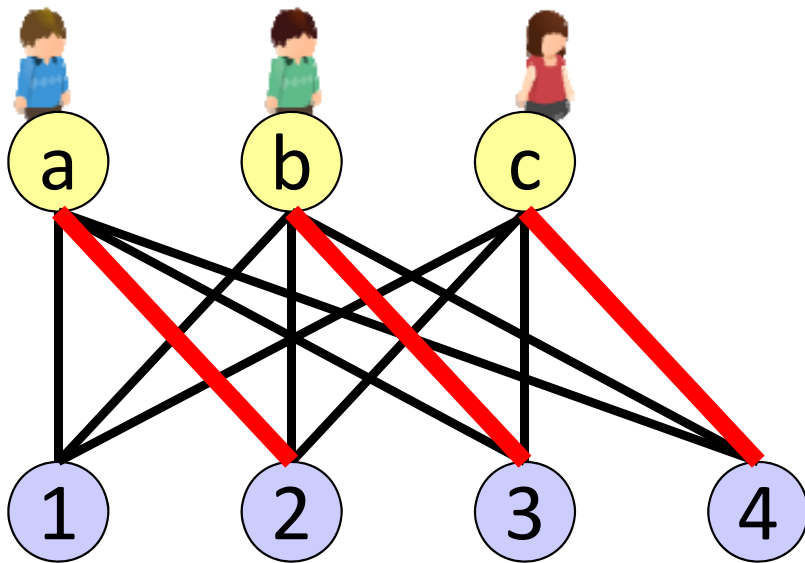
(2)



最大重みマッチング問題

最大重みマッチング問題

- 各入札者の各財への評価値 $v(i,j)$ が与えられている状況で
入札者に割り当てられた財の評価値の合計を最大化したい



$v(i,j)$	a	b	c
①	3	1	0
②	7	6	7
③	1	7	8
④	0	0	4

- 最大重みマッチング問題:**
枝に重みが与えられたグラフにおいて,
枝重みの和が最大のマッチングを求める

最大重みkマッチング問題の定義

最大重みkマッチング問題

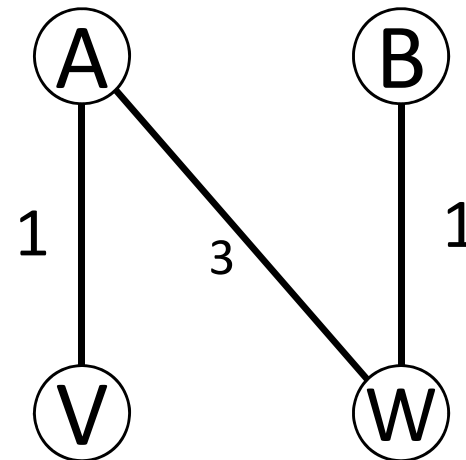
- **入力:** 二部グラフ $G=(V,E)$ (頂点集合 V , 枝集合 E),
各枝 (u,v) の重み $w(u,v)$
正整数 k
- **出力:** 枝数= k のマッチングの中で枝重みの和が最大のもの
(最大重みkマッチング)

例: 最大重み1マッチングは

$\{(A,W)\}$, 重み3

最大重み2マッチングは

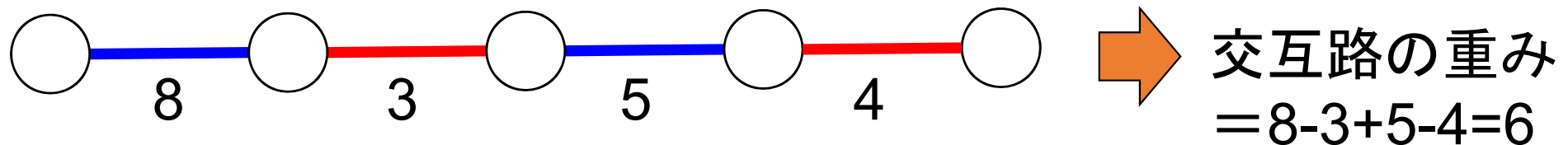
$\{(A,V), (B,W)\}$, 重み2



最大重みマッチングと交互路

交互路・交互閉路による マッチングの更新

定義: マッチング M に関する交互路 (交互閉路) P の **重み**
 $= P \setminus M$ の枝重みの和 $- P \cap M$ の枝重みの和



命題 M : マッチング, P : (M に関する) 交互路

$\leftrightarrow$ P を使って M の枝を入れ替えて得られる

マッチング $(M \setminus P) \cup (P \setminus M)$ の重み $= M$ の重み $+ P$ の重み



命題 正の重みの交互路 (交互閉路) が存在

$\rightarrow$ 現在のマッチング M は最大重みではない

(対偶: マッチング M は最大重みマッチング

$\rightarrow$ 正の重みの交互路 (交互閉路) は存在しない)

最大重みマッチングの最適性条件: 交互路を用いた条件

命題 正の重みの交互路(交互閉路)が存在
→現在のマッチングMは最大重みではない
(対偶: マッチングMは最大重みマッチング
→正重みの交互路(交互閉路)は存在しない)

逆も成り立つ

命題 正の重みの交互路(交互閉路)が存在
←現在のマッチングMは最大重みではない
(対偶: マッチングMは最大重みマッチング
←正重みの交互路(交互閉路)は存在しない)

マッチングが最適解であるための必要十分条件

定理: マッチング M は最大重みマッチング
↔正重みの交互路および交互閉路は存在しない

交互路による最適性条件：証明

最大重みマッチングではない $\rightarrow$ 正の重みの交互路(閉路)が必ず存在

命題 正の重みの交互路(交互閉路)が存在
 $\leftarrow$ 現在のマッチング M は最大重みではない

(証明) 最大重みマッチングを M^* とおく

枝集合 $(M^* \setminus M) \cup (M \setminus M^*)$ に注目

$\rightarrow$ 幾つかの (M に関する)

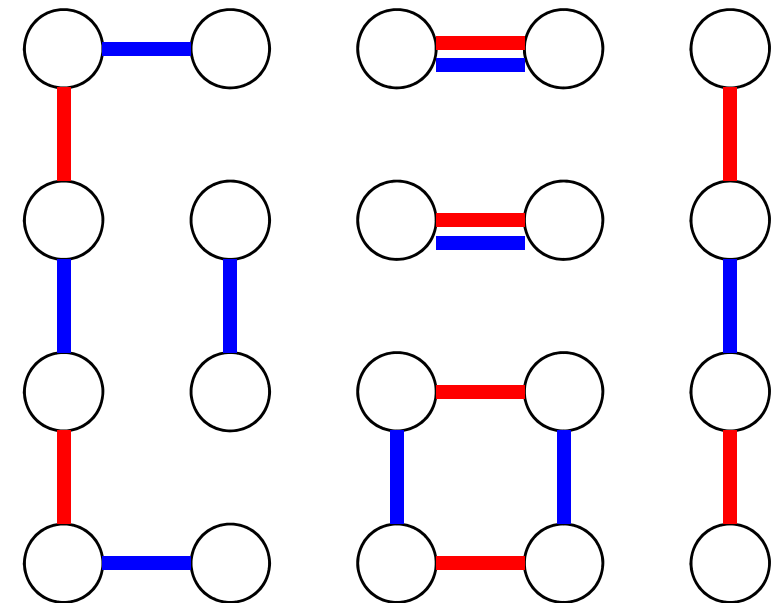
交互路・交互閉路からなる.

それらの重みの合計

$= M^* \setminus M$ の重み $- M \setminus M^*$ の重み

$= M^*$ の重み $- M$ の重み > 0

$\therefore$ 正の重みの交互路または交互閉路が
必ずひとは存在



最大重みマッチングのアルゴリズム

最大重み増加路を用いたアルゴリズム

ハンガリー法とよばれる

- 各反復で、重み最大の増加路を利用
- 高々 $|V|/2$ 回の反復で終了

アルゴリズム

ステップ0: $M_0 := \emptyset$, $k=0$ とする.

ステップ1: M_k に関する増加路が存在しない $\rightarrow$ 終了

ステップ2: M_k に関する最大重みの増加路 P をひとつ見つけ,
 P を用いて M_k を更新 ($M_{k+1} := (M_k \setminus P) \cup (P \setminus M_k)$)
 $k:=k+1$ においてステップ1へ.

命題 各 $k=0,1,\dots$ に対し, M_k は最大重み k マッチング

求めた M_k の中で重み最大のマッチングが最大重みマッチング

アルゴリズムの正当性

命題 各 $k=0,1,\dots$ に対し, M_k は最大重み k マッチング

Proof. 帰納法で証明. $k=0$ は自明. よって, $k>0$ の場合について考える.

帰納法の仮定により M_{k-1} は最大重み $k-1$ マッチングである. 第 k 反復で求めた (M_{k-1} に関する) 増加路を P とすると, $M_k = M_{k-1} \Delta P$ である. また, N を最大重み k マッチングとすると, 枝集合 $M_{k-1} \Delta N \equiv (M_{k-1} \setminus N) \cup (N \setminus M_{k-1})$ には N の枝が M_{k-1} の枝より多く含まれるので, M_{k-1} に関する増加路 P' が必ず含まれる. 増加路 P は重みが最大なので,

$$P \text{ の重み} \geq P' \text{ の重み}$$

が成り立つ. また, P' は N に対する交互路でもあり, $N' = N \Delta P'$ は枝数 k のマッチングである. ここで N' の重みは

$$w(N') = w(N) - P' \text{ の重み}$$

であるが, M_{k-1} が最大重み $k-1$ マッチングなので

$$w(N) - P' \text{ の重み} \leq w(M_{k-1})$$

が成り立つ. 以上のことから,

$$\begin{aligned} w(M_k) &= w(M_{k-1}) + P \text{ の重み} \\ &\geq (w(N) - P' \text{ の重み}) + P' \text{ の重み} = w(N) \end{aligned}$$

となり, M_k は最大重み k マッチングである. $\square$

重み最大増加路の見つけ方: 準備

- 増加路検出でつかった有向グラフを利用
- 有向枝に長さを与える → 最短路問題に帰着

長さ = - (枝重み)

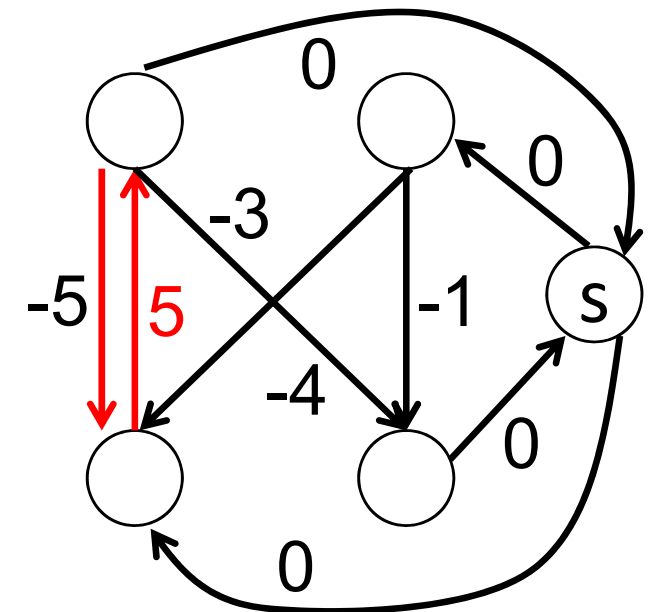
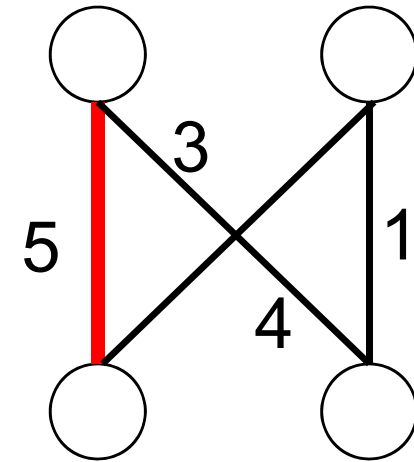
- 全ての枝を, B 側から N 側に向き付け
- マッチング M の各枝に対し, さらに

逆向きの枝 ($N \rightarrow B$) を追加

長さ = 枝重み

- B 側の各頂点 u に対し,
 - マッチング枝が接続 $\rightarrow$ 枝 (u, s) をおく
 - 接続なし $\rightarrow (s, u)$ をおく

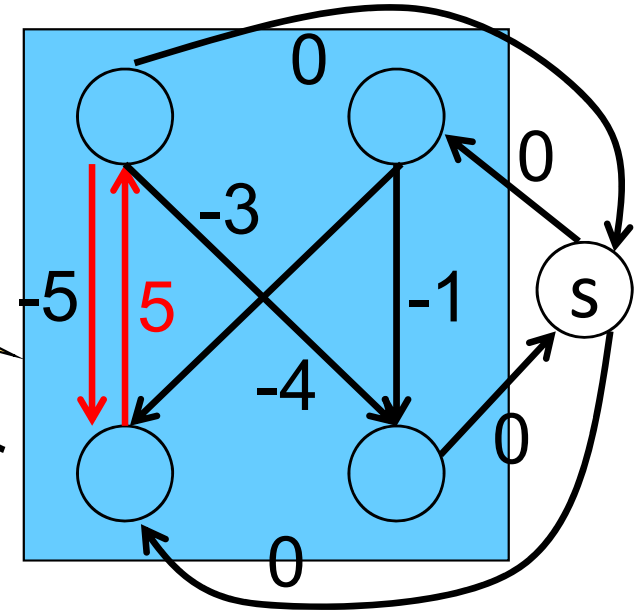
- N 側の各頂点 v に対し,
 - マッチング枝が接続 $\rightarrow$ 枝 (s, v) をおく
 - 接続なし $\rightarrow (v, s)$ をおく



最大重み増加路のを見つけ方

この部分のみ
注目

- 有向グラフにおいて,
B 側の頂点 $\rightarrow \dots \rightarrow$ N側の頂点 の形の有向路
 $\longleftrightarrow$ 元の二部グラフの(Mに関する)増加路
 有向路の長さ = 増加路の重み $\times (-1)$
 $\therefore$ 有向グラフにおいて, 上記のような有向路のうち,
 最短なものを求めればよい



注意: M が最大重み k マッチング

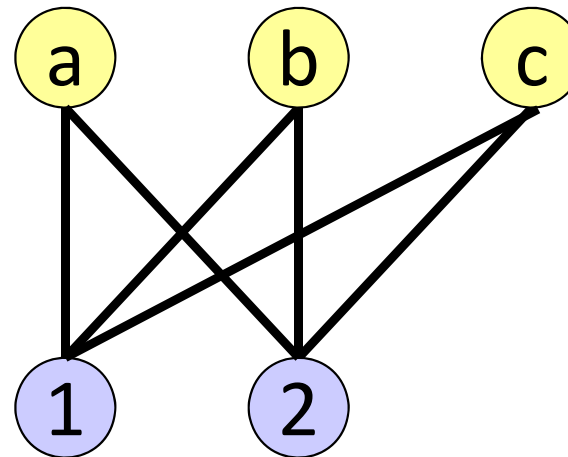
- ➔ 対応する有向グラフに負閉路は存在しない
 ($\because$ 頂点 s を通らない有向閉路は交互閉路に対応)
 $\therefore$ 最短路は存在

演習問題

問2: 下記のように枝重みが与えられたときの
最大重みマッチングを求めよ.

(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

