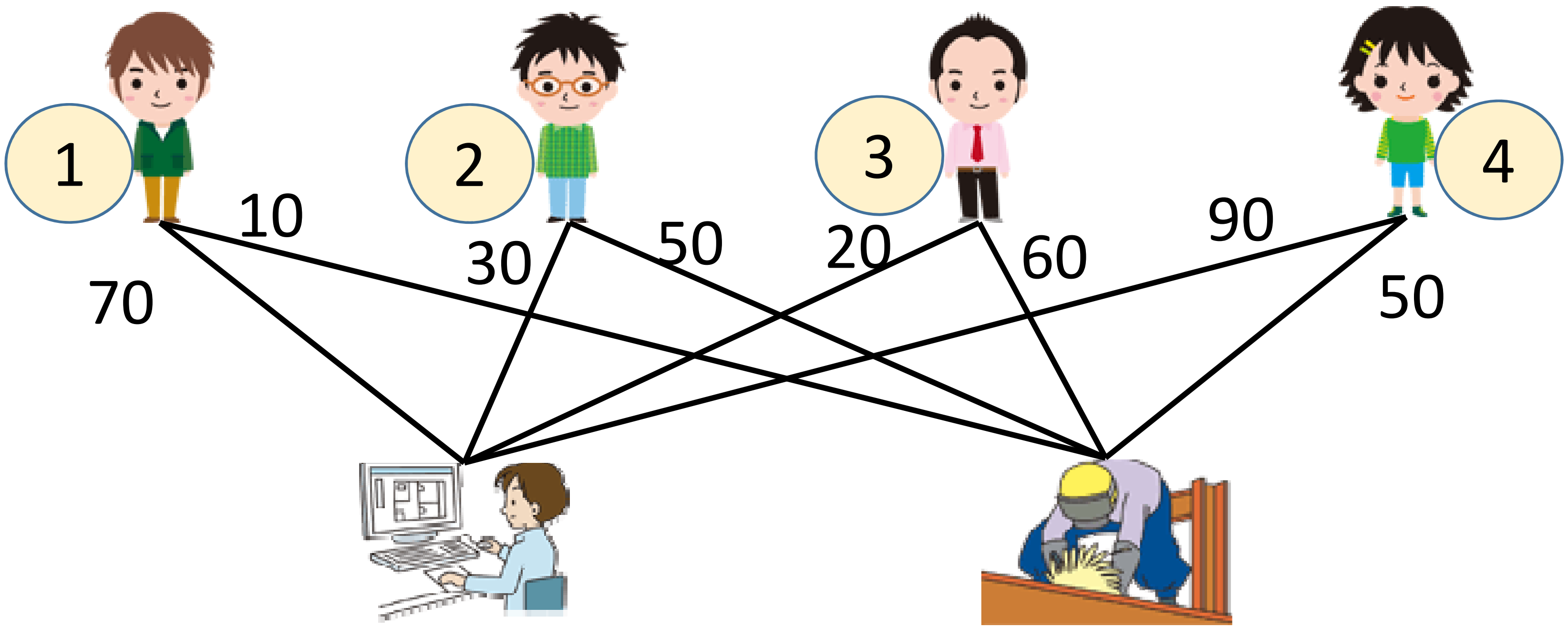
数理経済学
第4回
最大重みマッチング問題

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

最大重みマッチング問題の
定義と例

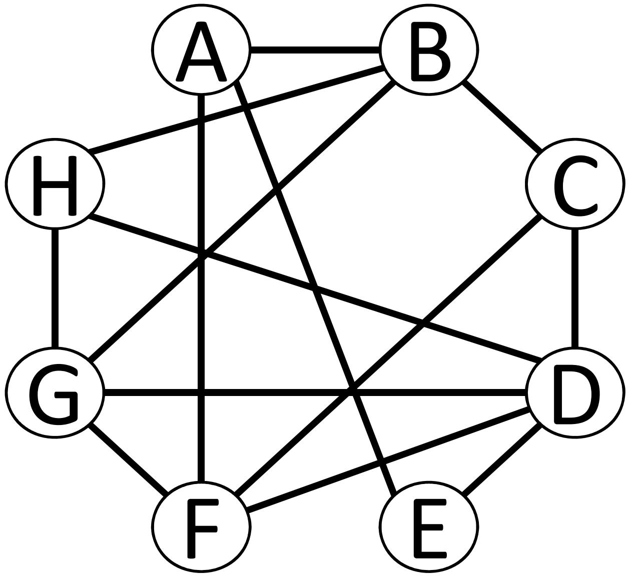
最大重みマッチング問題

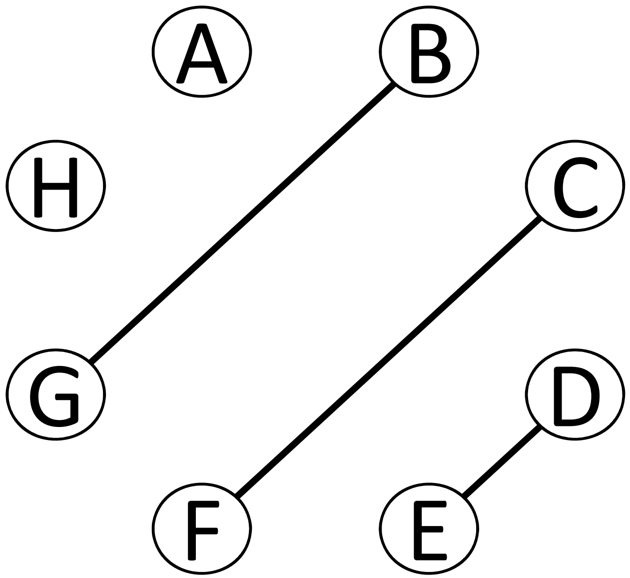
- 労働者に仕事を割り当てたい
  - 各労働者には，高々1つの仕事が割り当て可能
  - 各仕事は，高々一人の労働者に割り当て可能
  - 労働者を仕事に割り当てると，利益が得られる
- 総利益を最大にする割当を見つけたい➔最大重みマッチング



一般グラフの最大重みマッチング問題

- 一般のグラフでも，最大重みマッチング問題を考えることができる
    - 例：2人チーム（ペア）で作業を行う
    - 相性の良いチームを多く作りたい
    - 相性を数値化，相性の合計値を最大にするチームを求めたい

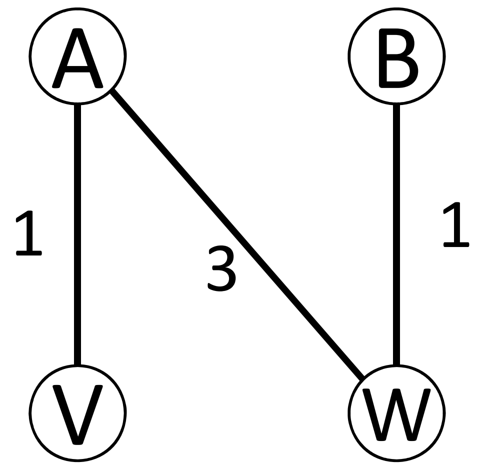




最大重みマッチング問題の定義

最大重みマッチング問題
- 入力：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)
- 出力：枝重みの和が最大のマッチング（最大重みマッチング）

※「最大重みマッチング➔最大サイズマッチング」は
一般に成り立たない



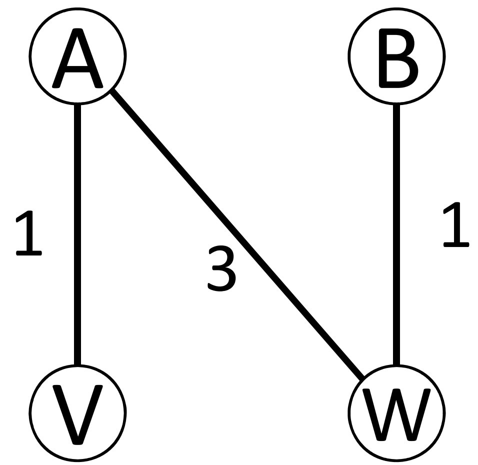
例： 最大重みマッチングは
{(A,W)}, 重み3
最大サイズマッチングは
{(A,V), (B, W)}, 重み2

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: 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問題の例1：研究室配属問題

各研究室に学生数人を割り当てる

学生A,B,C,Dの４人を研究室X,Yへ

各研究室に配属できる人数には上限がある

	X研究室	Y研究室
定員	2	2

学生の満足度の合計を最大にしたい

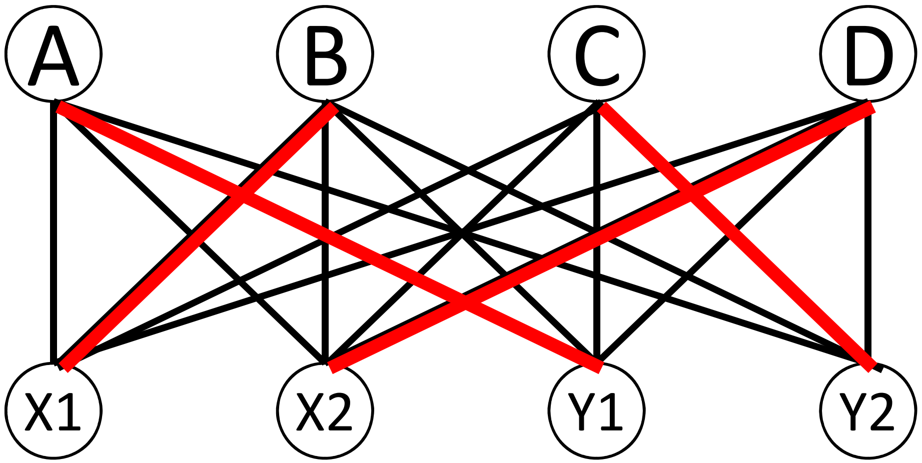
満足度	A	B	C	D
X	6	8	5	9
Y	9	1	5	3

研究室配属問題から
最大重みkマッチング問題へ

各学生に対応する頂点，各研究室の「枠」に対応する頂点

各学生と「研究室枠」の間に枝，重み＝満足度

k＝学生数



w(u,v)	A	B	C	D
X1	6	8	5	9
X2	6	8	5	9
Y1	9	1	5	3
Y2	9	1	5	3

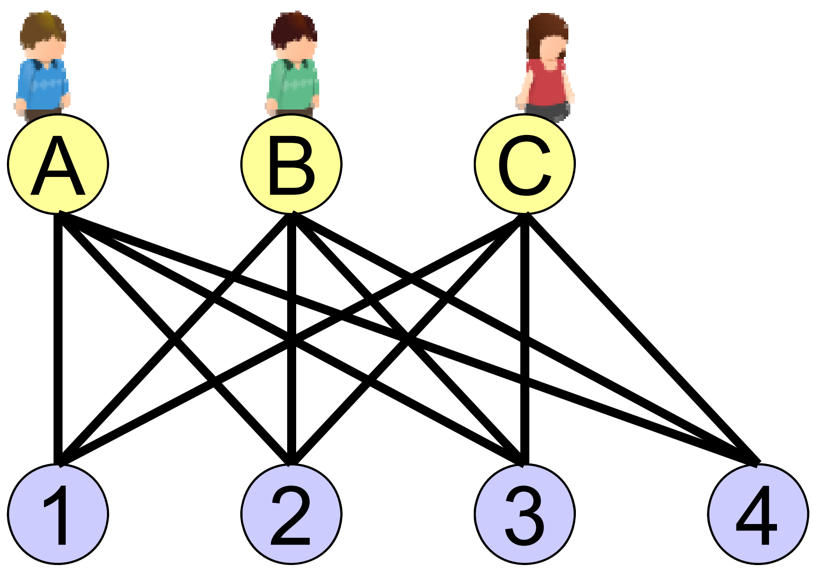
最大kマッチング➔研究室配属案

※注意：この配属決定法では，
学生が嘘をつくことで得する可能性がある．

最大重みマッチング問題の例2：複数財オークション

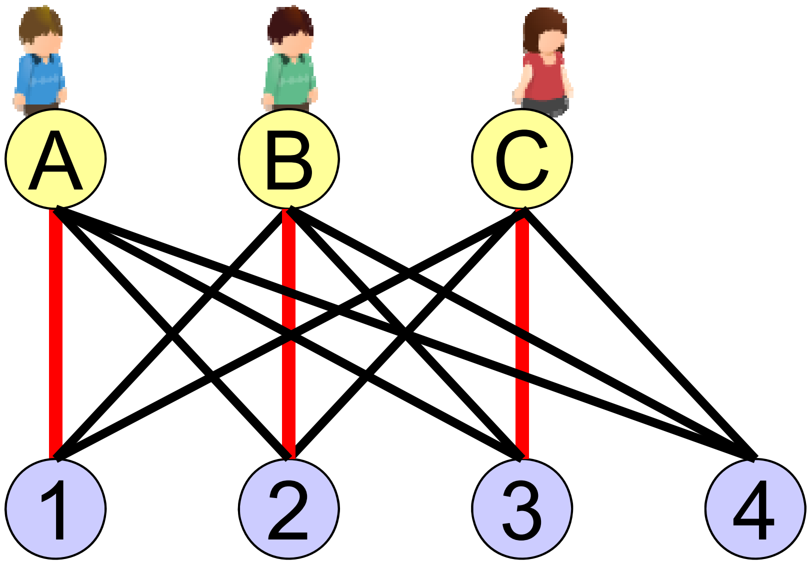
- 複数の財（商品），複数の入札者が存在するオークション
- 入札者に財が割り当てられたら，
  入札者は指定された金額のお金（価格）を払う
- 各入札者は各財に対する満足度（価値）の情報を提示

➔ オークション主催者は，入札者全員が「満足する」ように
入札者と財のマッチング，および財の価格を決める
「満足」＝実質的な満足度（満足度-価格）が最大の財を得る



満足度	A	B	C
①	3	1	0
②	7	6	7
③	1	7	8
④	0	0	4

入札者が不満を持つ例1



価格　2　2　1　1

満足度	A	B	C
①	3	1	0
②	7	6	7
③	1	7	8
④	0	0	4

上記のマッチングと価格を提示→入札者は満足？

入札者Aは不満: 入手した財①の実質的な満足度=3-2=1

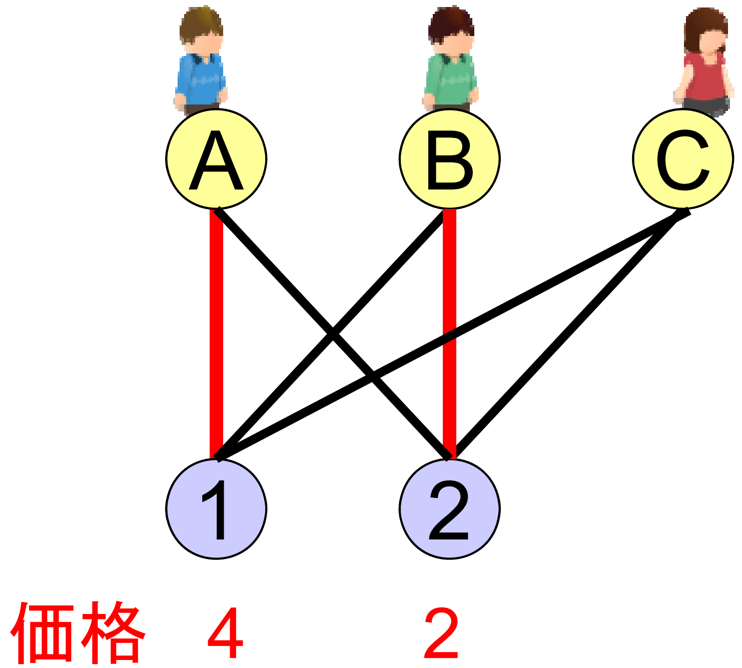
< ②の実質的な満足度=7-2=5

入札者Bは不満: 入手した財②の実質的な満足度=6-2=4

< ③の実質的な満足度=7-1=6

入札者Cは満足

入札者が不満を持つ例2



満足度	A	B	C
①	3	1	0
②	7	6	7

上記のマッチングと価格を提示→入札者は満足？

入札者Aは不満：入手した財①の実質的な満足度=3-4=-1 < 0

入札者Bは満足：入手した財②の実質的な満足度=6-2=4

≧ ①の実質的な満足度=1-4=-3

入札者Cは不満：入手できなかった財②の実質的な満足度=7-2>0

複数財オークションのワルラス均衡

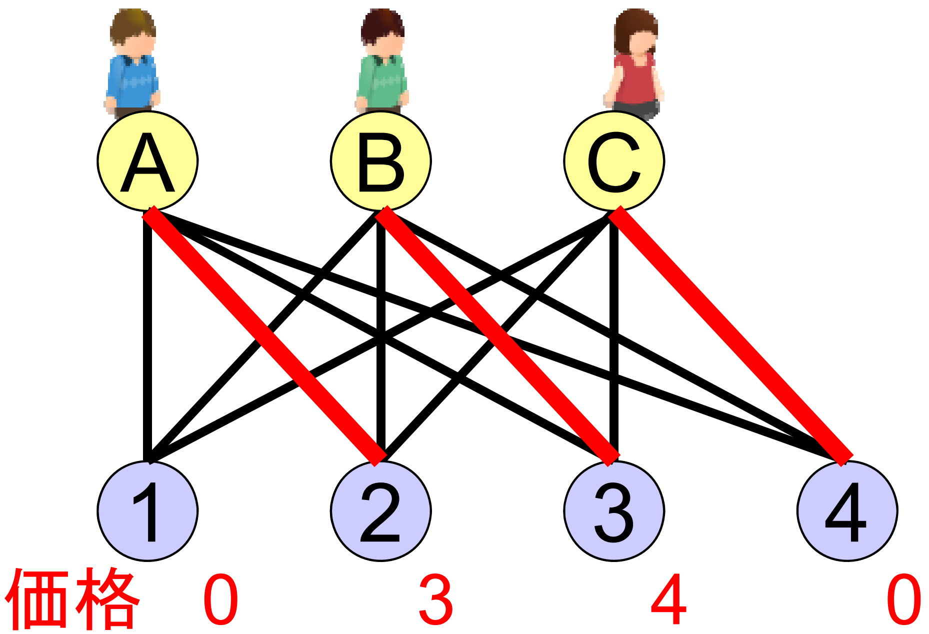
- ワルラス均衡：入札者全員が「満足する」財の割当＆財の価格

ワルラス均衡：以下の条件を満たす価格と財の割当

- 入札者 i に財 j が割り当てられる
  ➔ 他の財より利得 w(i,j)-p(j) が大きい（または等しい），
  かつ w(i,j)-p(j)≧0
- 入札者 i に財の割り当てがない ➔ $\max_{1\le j\le n}\{w(i,j)-p(j)\} \le 0$
- 財 j が誰にも割り当てられない ➔ p(j)=0

均衡割当，均衡価格 ＝ 均衡におけるマッチングと価格

均衡の例1

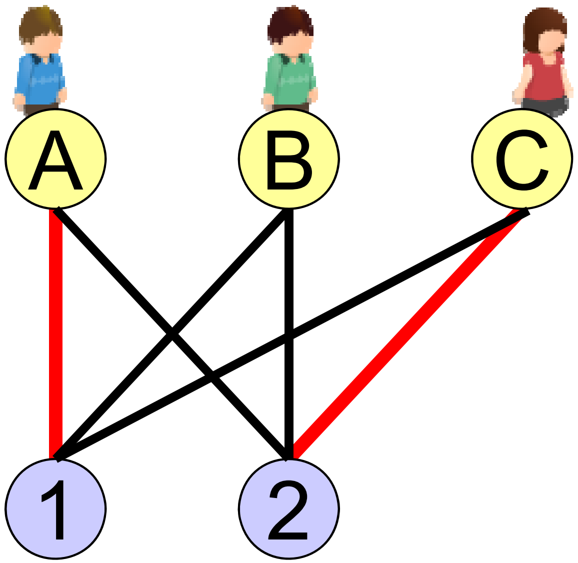


w(i,j)	A	B	C
①	3	1	0
②	7	6	7
③	1	7	8
④	0	0	4

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

- 各入札者が得た財の利得は最大
- 財1は売れ残り➔価格0

均衡の例2



価格　1　　6

満足度	A	B	C
①	3	1	0
②	7	6	7

利得	A	B	C
①	2	0	-1
②	1	0	1

- 入札者A, Cの得た財の利得は最大
- 入札者Bは利得≦0の財のみ
  ➔財の割当がなくてもよい

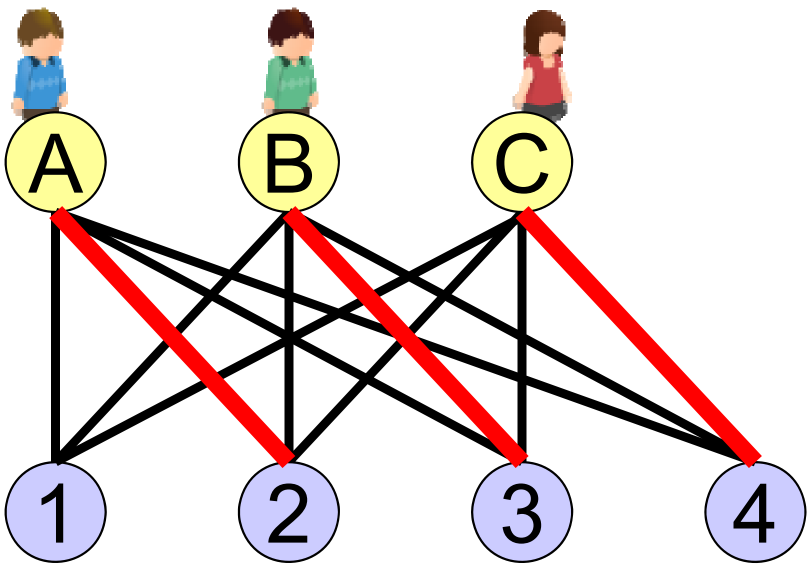
複数財オークションと最大重みマッチングの関係

最大重みマッチング問題を解くと，均衡割当が得られる

定理

参加者と財の間のマッチングMは均衡割当

⟷ Mは重み=w(i,j) としたときの最大重みマッチング



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

均衡価格と差分不等式系

均衡割当が分かっているとき，

均衡価格は差分不等式系で表現可能

- 入札者 i に財 j_i が割り当てられる

  ➔ 他の財より利得 $w(i,j_i)-p(j_i)$ が大きい（または等しい），

  かつ $w(i,j_i)-p(j_i)\geq 0$

  ➔ $w(i,j_i)-p(j_i)\geq w(i,j)-p(j) \quad (\forall j=1,2,\ldots,n)$

  $w(i,j_i)-p(j_i)\geq 0$
- 入札者 i に財の割り当てがない

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

  ➔ $w(i,j)-p(j)\leq 0 \quad (\forall j=1,2,\ldots,n)$
- 財 j が誰にも割り当てられない

  ➔ p(j)=0

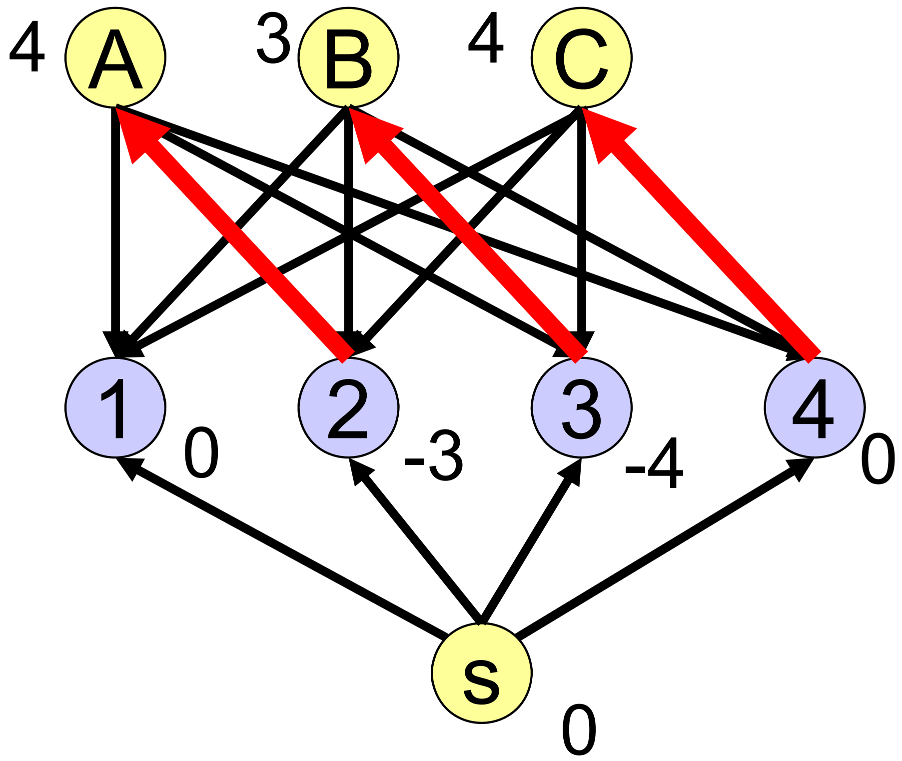
∴ 均衡価格の計算は
最短路問題に帰着可能

均衡価格の計算

※前のスライドの差分不等式系に
対応する最短路問題を
巧みに表現している

均衡価格は最短路問題を解いて求めることができる

- 最大重みマッチングの枝：財→参加者に向き付け．長さ＝重み
- その他の枝：参加者→財に向き付け．長さ＝－重み
- 新しい頂点 s，および sから各財への長さ0の枝をつける．
- 財 j の価格＝ －（頂点 j への最短距離） ➔ 均衡価格



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

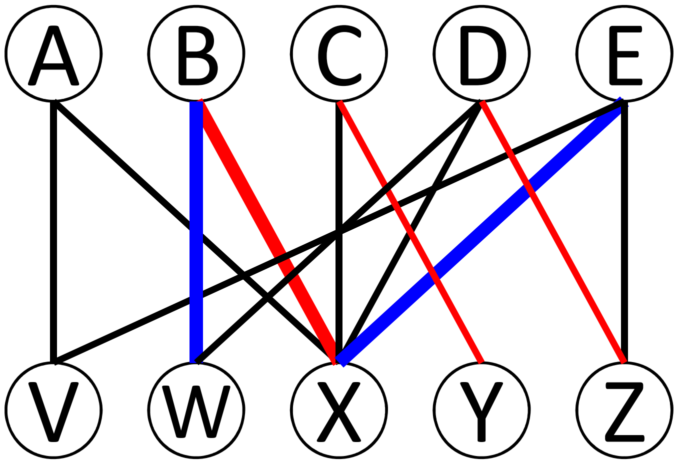
最大重みマッチングと交互路

（復習）交互路・交互閉路

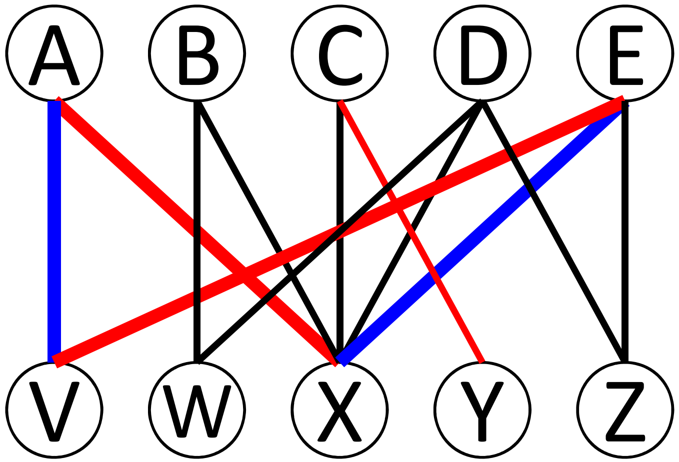
定義：路PはマッチングMに関する交互路 ←→ 次の条件を満たす
(a) 路PにMの枝とM以外の枝が交互に現れる
(b) 枝集合 $M' = (M \setminus P) \cup (P \setminus M)$ はマッチング

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

交互路の例

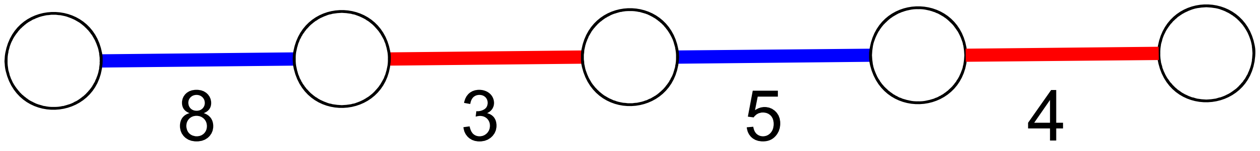


交互閉路の例



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 ー P∩Mの枝重みの和



交互路の重み
＝8-3+5-4=6

命題 マッチングMに対し，Pは（Mに関する）交互路または交互閉路
➔Pを使ってMの枝を入れ替えると，新しいマッチングM’ができる

$$M' = (M \setminus P) \cup (P \setminus M)$$

M’の重み ー Mの重み ＝ P の重み

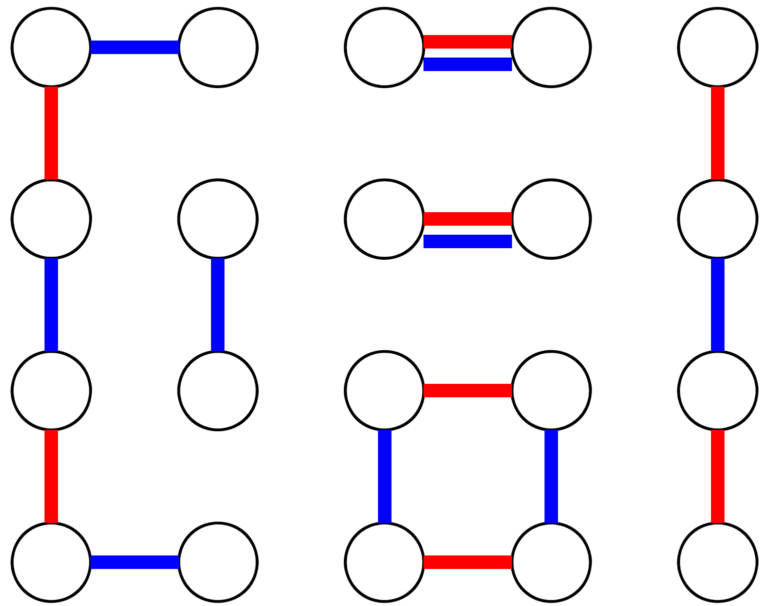
命題　正の重みの交互路（交互閉路）が存在
➔現在のマッチングMは最大重みではない
（対偶：マッチングMは最大重みマッチング
➔正の重みの交互路（交互閉路）は存在しない）

交互路によるマッチングの最適性条件

最大重みマッチングではない➔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
　　　　🡐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
　　　　🡐正の重みの交互路(交互閉路)は存在しない)

(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
➔ 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
∴ 正の重みの交互路または交互閉路が
　必ずひとつは存在



最大重みマッチングの計算

最大重みマッチング問題のアルゴリズム

「Mは最大重みマッチング⇔正の重みの交互路(閉路)がない」

これに基づき，次のアルゴリズムが得られる

アルゴリズムA

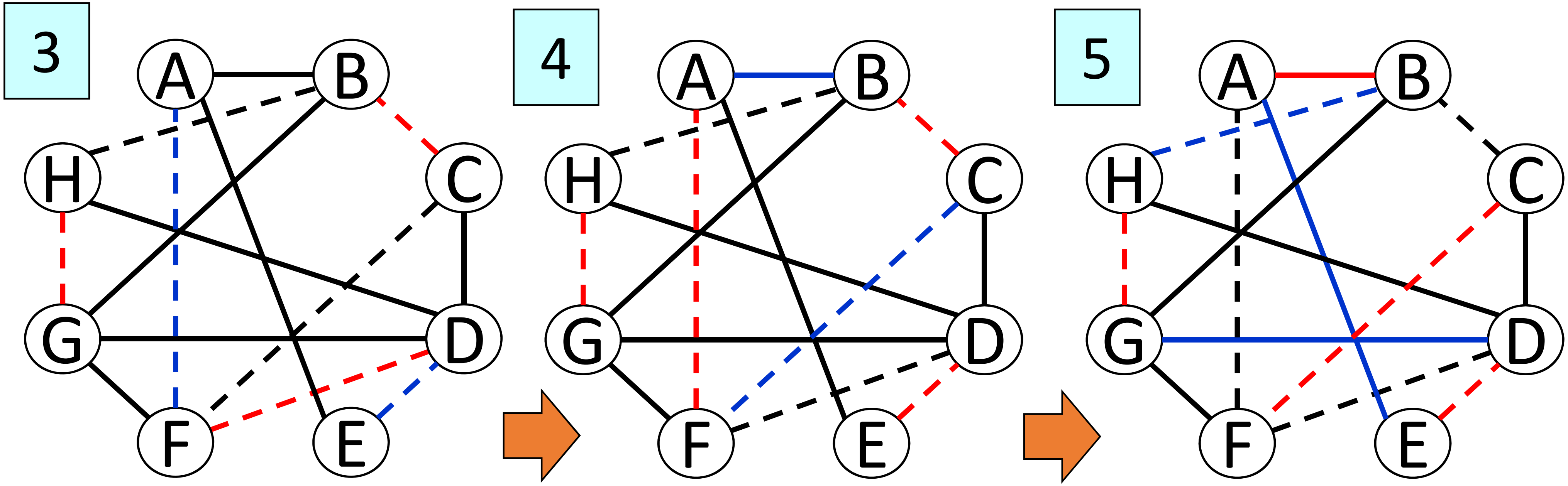
ステップ0：初期マッチングMを任意に選ぶ（例えば $M := \emptyset$）

ステップ1：Mに関する正の重みの交互路(閉路)の存在をチェック.
　存在しない➔現在のMは最大重みマッチング（終了）
　存在する➔ステップ2へ.

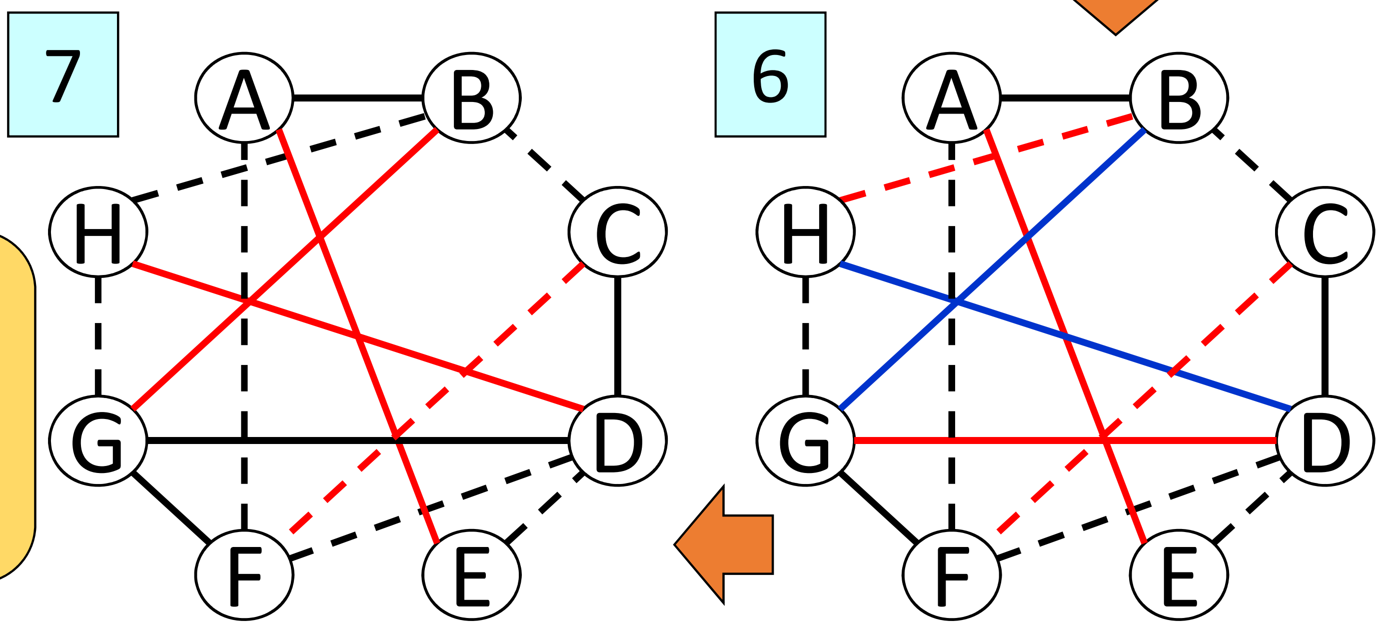
ステップ2：Mに関する正の重みの交互路(閉路) P をひとつ見つけ，
　P を用いて M を更新（ $M := (M \setminus P) \cup (P \setminus M)$ ）
　ステップ1へ.

反復回数が多くなる可能性あり

アルゴリズムの実行例



実線の枝：重み2
破線の枝：重み1



正重みの
交互路（閉路）なし
➔
最大重みマッチング

最大重み増加路を用いたアルゴリズム

- 各k=0,1,2,... に対する最大重みkマッチングを求める
- 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B

ステップ0 : M_0 := ∅ ，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➔ 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 とおいてステップ1へ.

命題 各k=0,1,... に対し，M_kは最大重みkマッチング

アルゴリズムの正当性

命題 各k=0,1,... に対し，M_kは最大重みkマッチング

（証明の概略）

ある k に対し，M_k は最大重み k マッチングとする.

このとき，ある最大重み k+1 マッチング M* で，

$(M^* \setminus M_k) \cup (M_k \setminus M^*)$ が（一つの）増加路となるようなものが存在することを示せば良い.

M* として，|M*∩ M_k |最大の最大重み k+1 マッチングを選ぶ

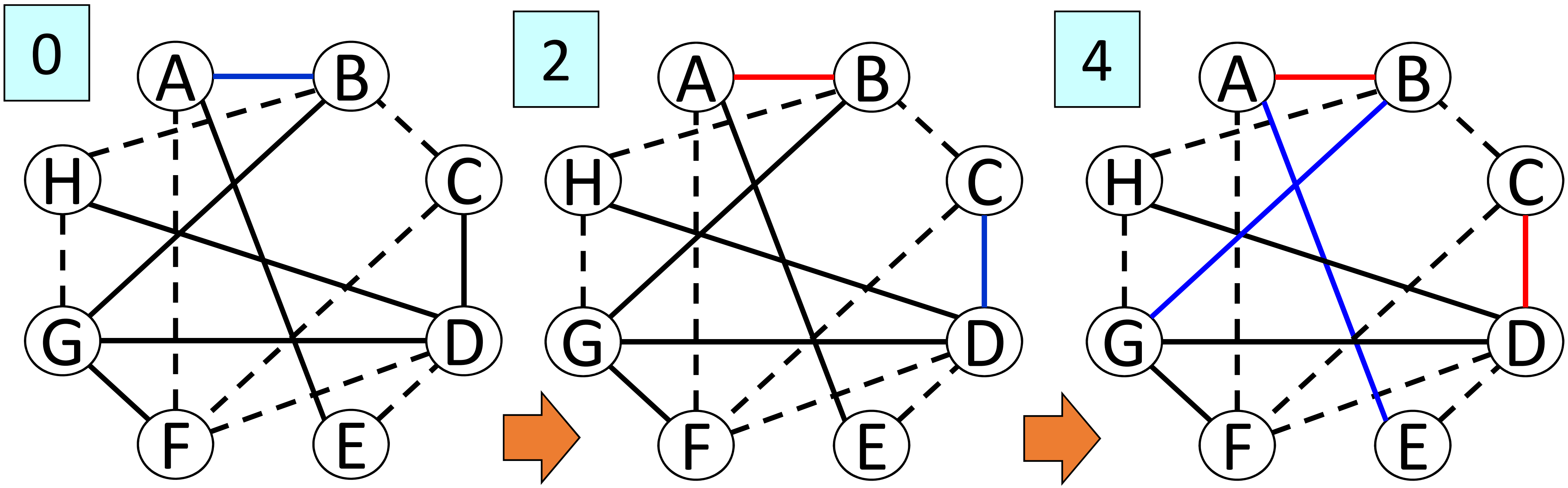
➔ $(M^* \setminus M_k) \cup (M_k \setminus M^*)$ は交互路･交互閉路の集まり

（注意：M* の方が枝数が多いので，増加路が必ず存在）

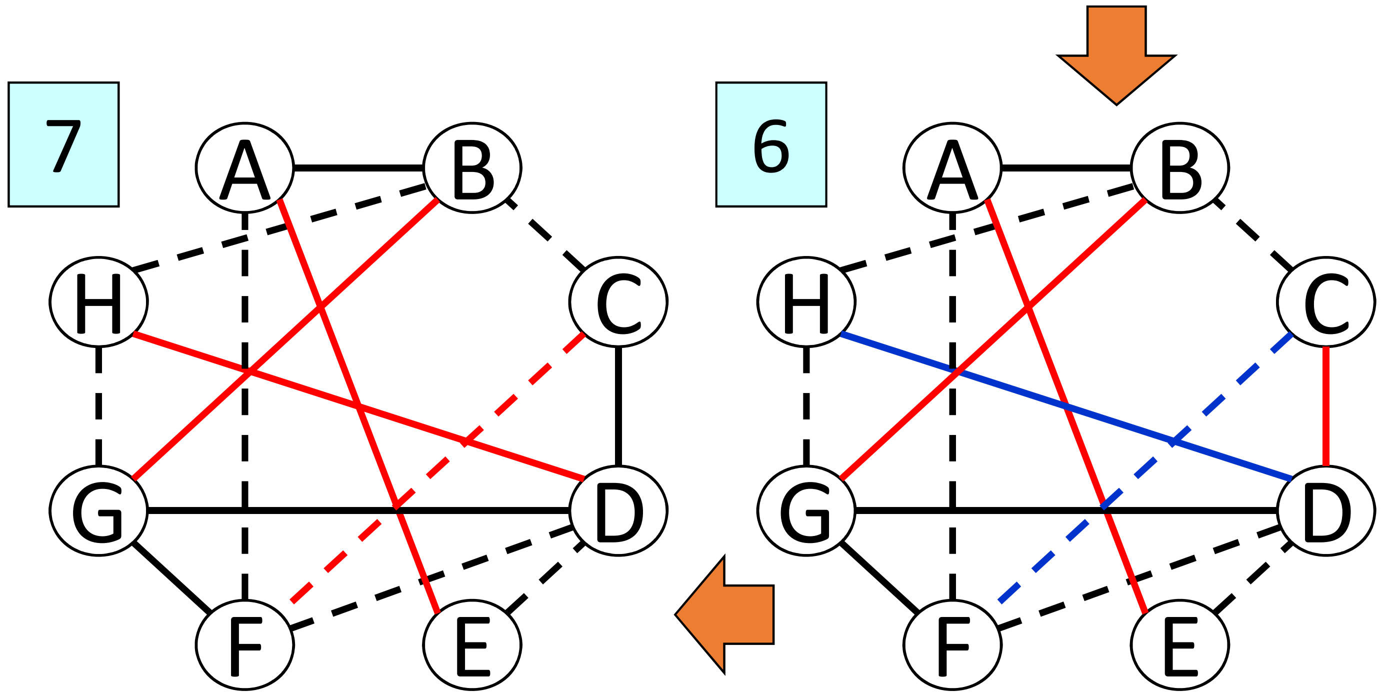
交互路･交互閉路が2つ以上存在する

➔ |M*∩ M_k |最大に矛盾することが示せる（なぜ？）

アルゴリズムの実行例



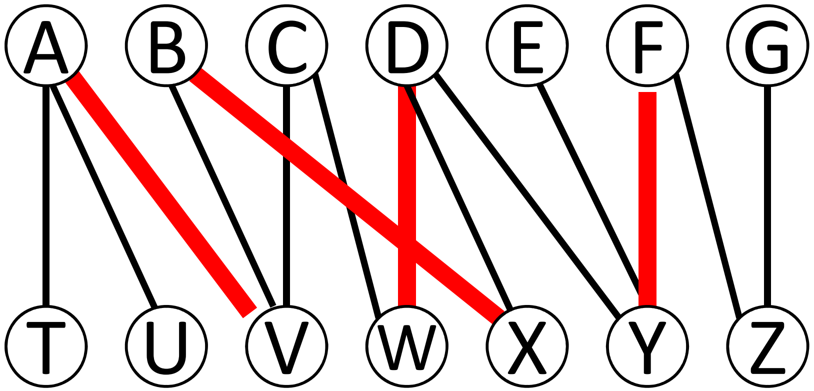
実線の枝：重み2
破線の枝：重み1

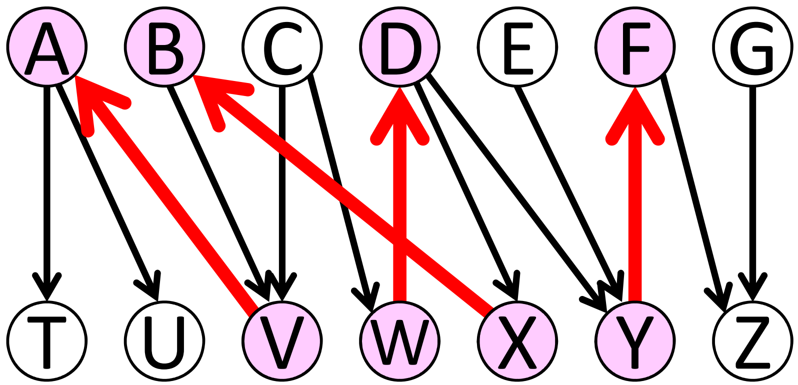


重み最大の増加路の見つけ方: 二部グラフの場合

- 増加路を求める場合と同様に計算可能.
- 二部グラフの頂点集合が V_1, V_2 に分かれているとき,
  - マッチングMの枝は, V_2 の頂点から V_1 の頂点に向き付け
  - それ以外の枝は, V_1 の頂点から V_2 の頂点に向き付け
- V_1 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

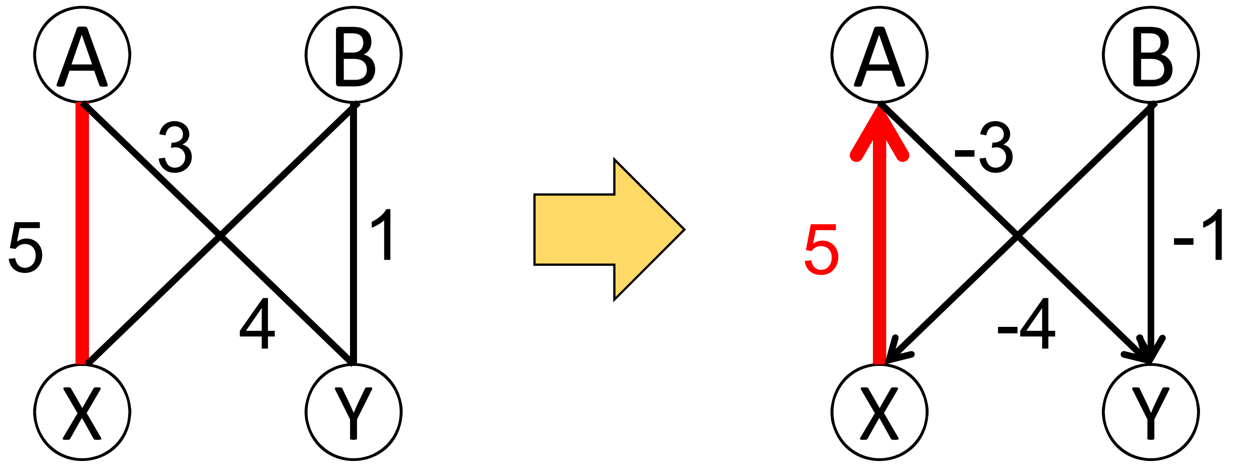




重み最大の増加路の見つけ方：二部グラフの場合

- 各有向枝の長さ：
  - マッチングの枝の逆向き枝の長さ＝元の長さ
  - それ以外の枝の長さ　＝元の長さ×（－1）
- 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，T_1 の頂点から T_2 の頂点への有向路のうち，最も短いものを求めればよい



最大重みマッチングを求める際の工夫

アルゴリズムB を使って,
　すべてのkに対し最大重みkマッチングを求める
➔ 最も重みの大きいマッチングが最大重みマッチング

しかし,最大重みマッチングが見つかった後も
（無駄な）反復を繰り返す必要あり

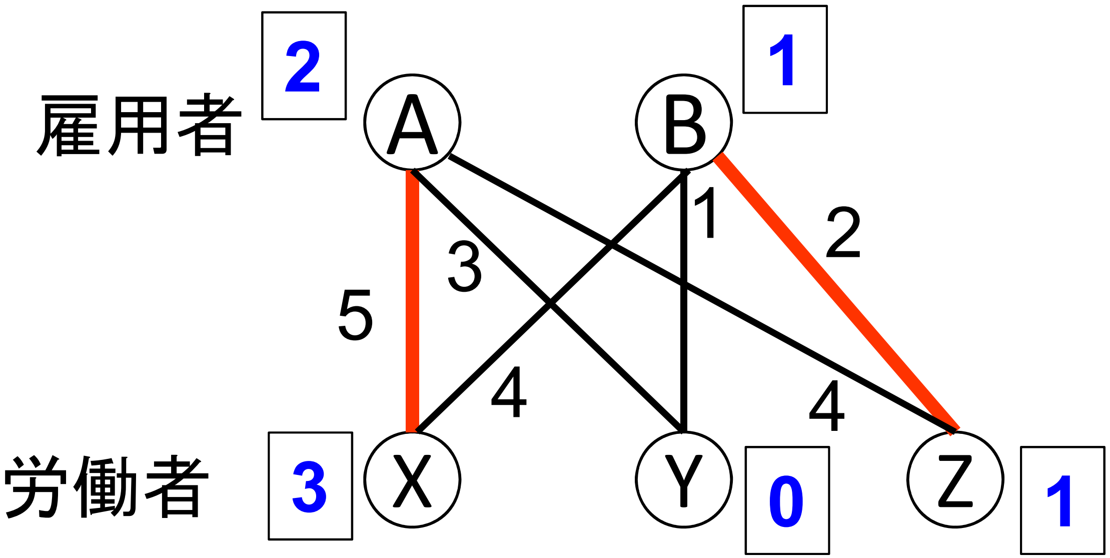
実は,無駄な反復は回避可能

命題　あるk>0 において$w(M_{k-1}) > w(M_k)$
➔ $w(M_{k-1}) > w(M_k) > w(M_{k+1}) > w(M_{k+2}) > \cdots$

$\therefore$ $w(M_k)$ の値が次の反復ではじめて減少したら,
M_k は最大重みマッチング

最大重みマッチング問題の例3：割当ゲーム

- m 人の雇用者（仕事）と n 人の労働者
  雇用者 i と労働者 j のペアに対する利益 w(i,j) > 0
- 雇用者 i が労働者 j を雇う（ペアになる）
  ➔ 得た利益 w(i,j) を2人で分配:
  i の利得 p(i) + j の利得 q(j) = 利益 w(i,j)
- 雇用者 i，労働者 j は誰ともペアを組まない ➔ 利得 p(i) = q(j) = 0
- 求めたいこと: 皆が「満足する」ペアの組み方（マッチング）
  と全員の利得



皆が満足する割当と利得

「満足」「不満」とは？

- ある雇用者 i，労働者 j は現在ペアになっておらず，

  かつ　$p(i) + q(j) < w(i,j)$　成立と仮定

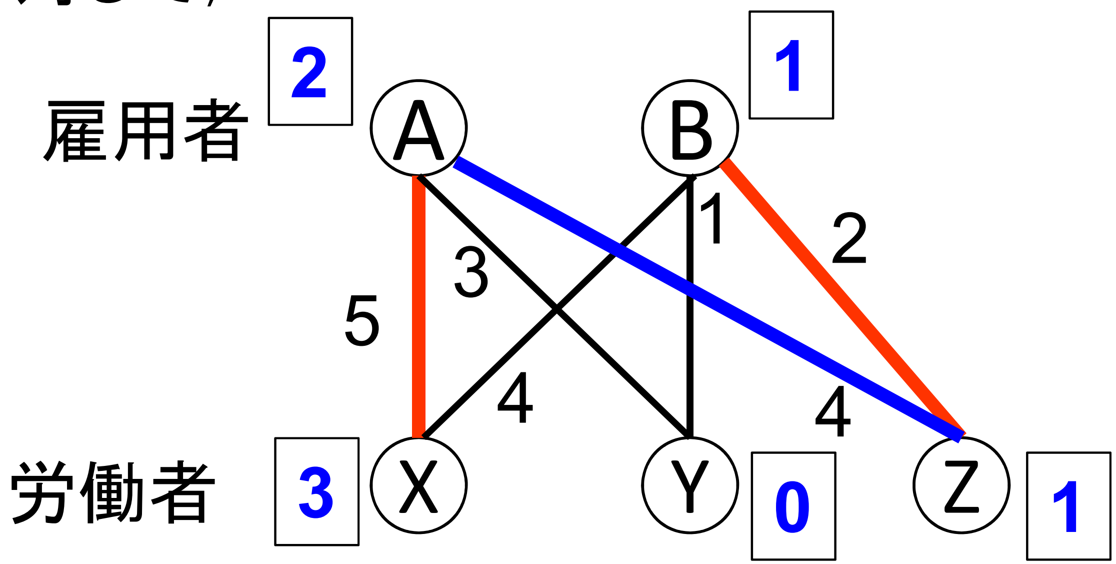
➔ 現在のペアを解消し，i と j でペアになると，i と j の利得が増える

➔ 現状は不満

∴ 皆が満足するための条件：

すべての雇用者 i，労働者 j に対して，

$p(i) + q(j) \geqq w(i,j)$

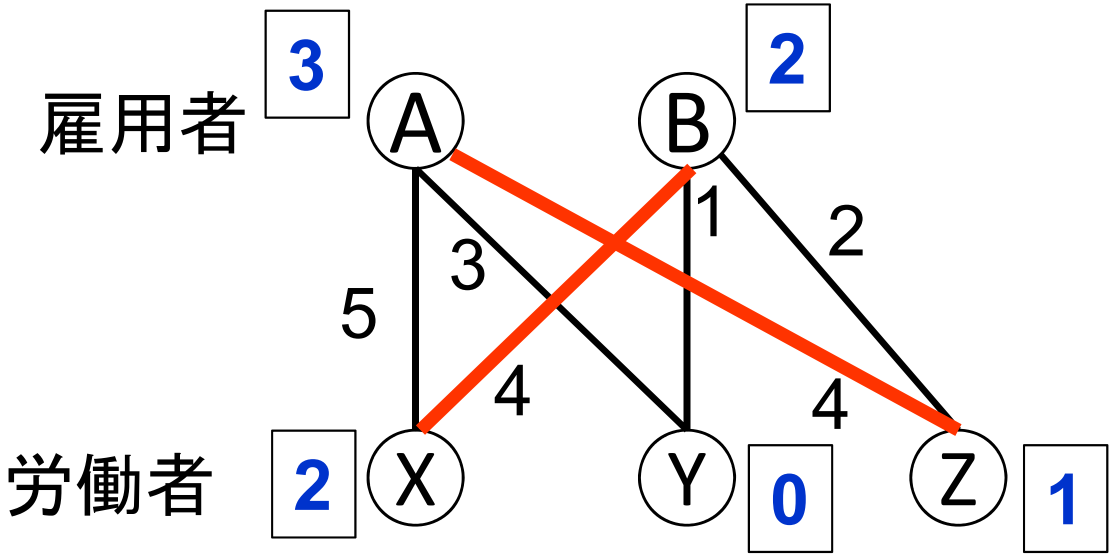


割当ゲームの定義

割当ゲームの目的:
以下の条件を満たすマッチング M と全員の利得 p(i), q(j) を求める

- (i,j) ∈M ➔ p(i) + q(j) = w(i,j)
- i, j に接続する M の枝がない ➔ p(i) = q(j) = 0
- p(i) + q(j) ≦ w(i,j) (各雇用者 i, 各雇用者 j に対して)
- p(i) ≧0, q(j) ≧0 (各雇用者 i, 各雇用者 j に対して)

ゲーム理論における
「協力ゲーム」
の一種



割当ゲームと最大重みマッチングの関係

定理:

割当ゲームの条件を満たすマッチング

　＝ w(i,j) に関する最大重みマッチング