

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6回 ホールの定理 最大重みマッチングの最適性条件

塩浦昭義

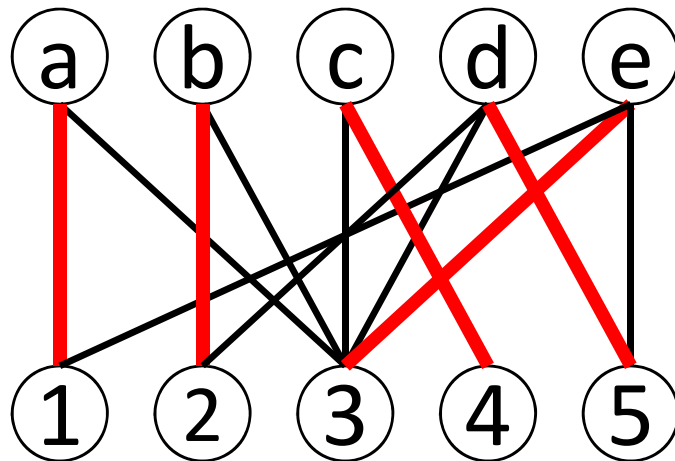
東京工業大学 経営工学系 准教授

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

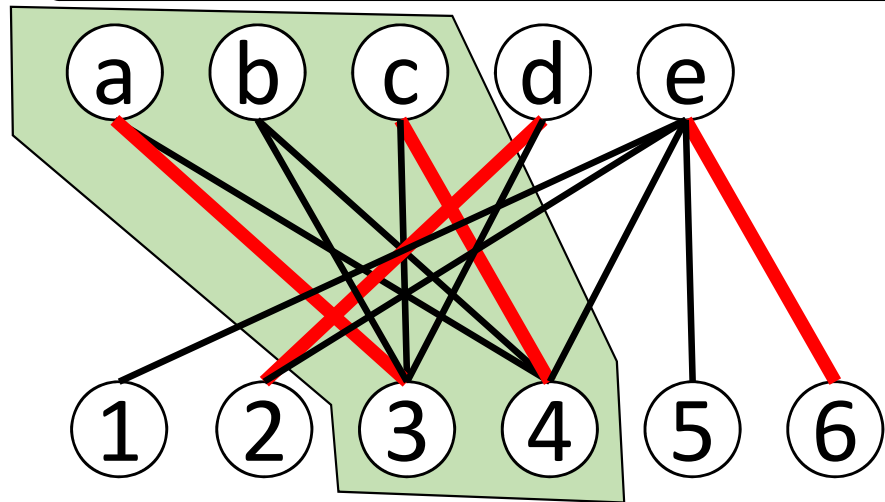
指定頂点集合をカバーするマッチング: 存在するための必要条件

所与の $B' \subseteq B$ 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 $X \subseteq B'$ 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 $X \subseteq B'$ 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 $X \subseteq B'$ に対し, X の隣接頂点の数 $\geq |X|$)

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 $X \subseteq B'$ に対し, X の隣接頂点の数 $\geq |X|$

ホールの定理：証明

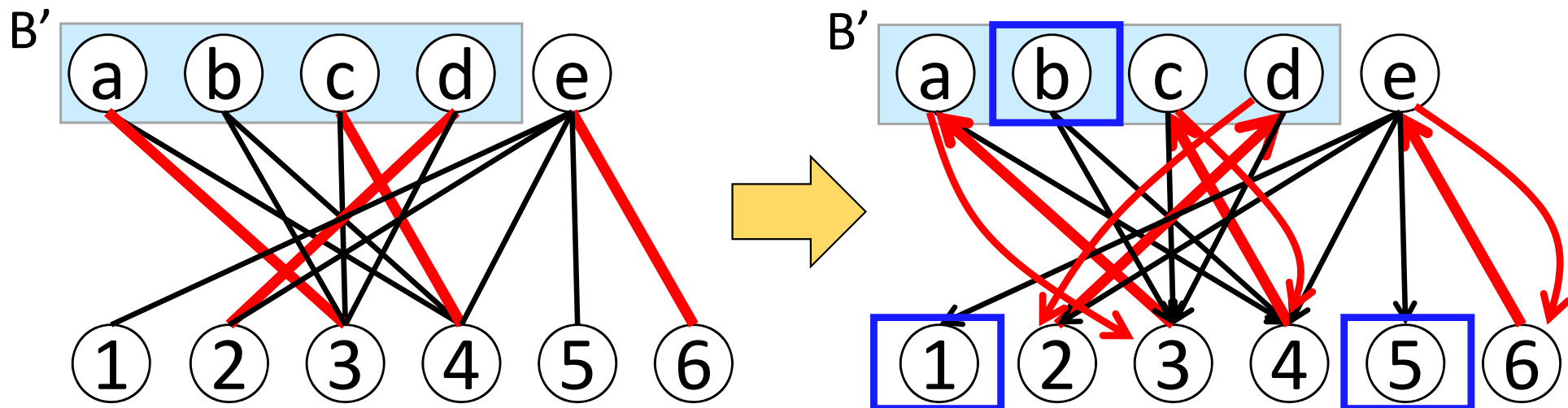
[←] 対偶「 B' すべてをカバーするマッチングは存在しない

→ 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)

- $S \subseteq B'$: マッチング枝が接続していない B' の頂点集合
- $T \subseteq N$: 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

($\because$ 存在する → カバーできる B' の頂点数が増える)

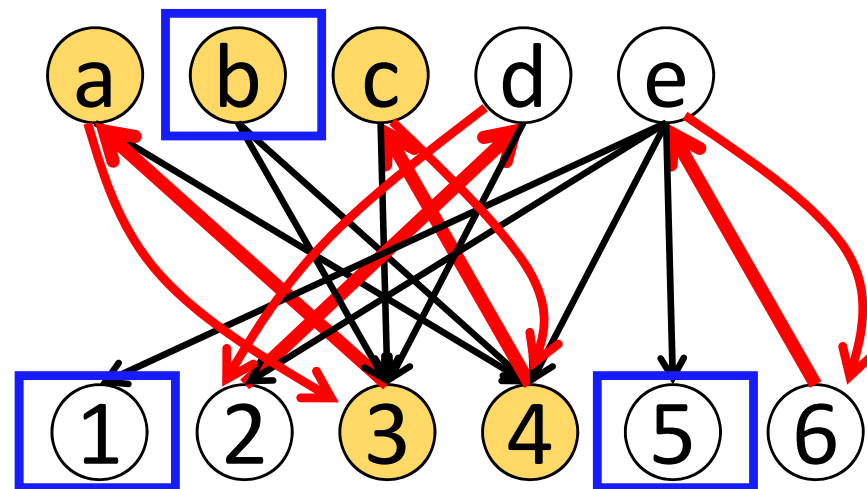
→ 有向グラフで S から T への有向路が存在しない

仮定より, S は非空 → $u \in S$ とする.

$X \subseteq B'$: 有向グラフで u から到達可能な B' の頂点

$Y \subseteq N$: 有向グラフで u から到達可能な N の頂点 u

→ 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 $Y \cap T = \emptyset$ ($\because u$ から始まる増加路が存在しない)

$\rightarrow Y$ の各頂点にはマッチング枝が接続

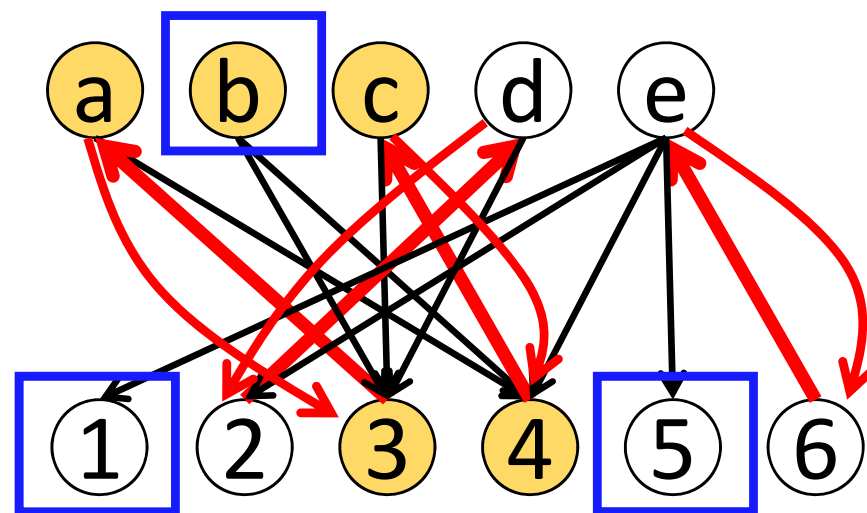
$Z \subseteq B'$: マッチング M での Y の「相手」

$\rightarrow Z \subseteq X, |Z| = |Y|$

さらに $u \notin Z, u \in X$

$\therefore |X| \geq |Z| + 1 > |Y|$

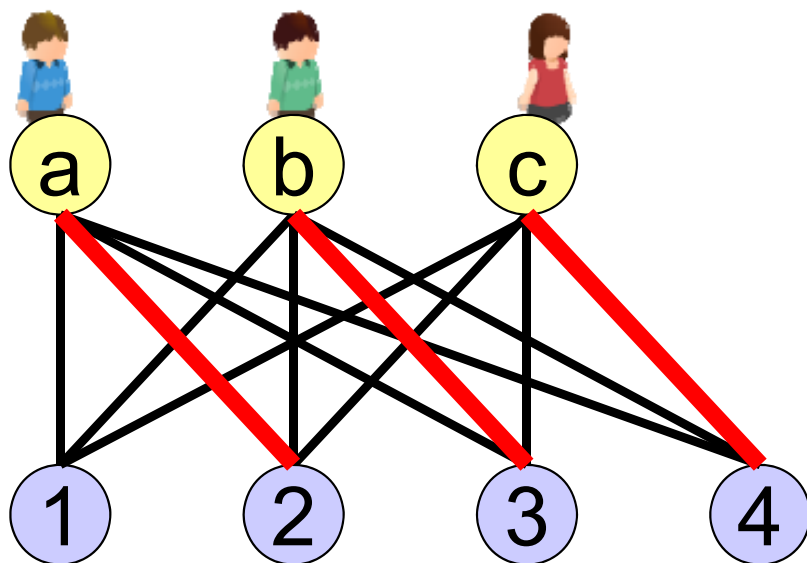
= X に隣接する頂点数



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:**
枝に重みを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

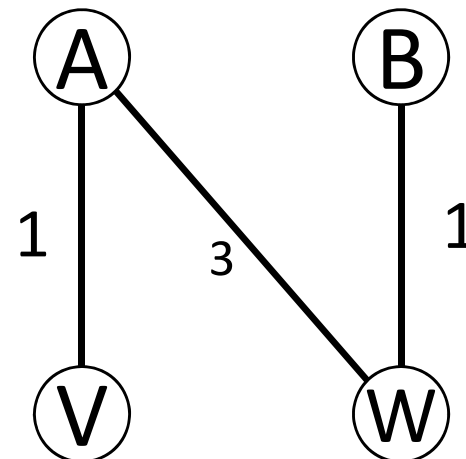
- **入力:** 二部グラフ $G=(V,E)$ (頂点集合 V , 枝集合 E),
各枝 (u,v) の重み $w(u,v)$
正整数 k
- **出力:** 枝数= k のマッチングの中で枝重みの和が最大のもの
(最大重みkマッチング)

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問題の読み替え

- 入札者 → ある仕事の雇用者
- 財 → 労働者
- 評価値 $v(i,j)$ → 仕事 i に労働者 j を割り当てたときの収益

※ 各仕事に従事できる労働者は高々ひとり
各労働者が従事できる仕事は高々ひとつ

- **最大重みマッチング問題**: 雇用者と労働者の仲介役の問題
総収益が最大となるような, 労働者の仕事への割当を求める

最大重みマッチングの双対問題

仲介役の別の仕事:

労働者の仕事への割当てで得た収益をうまく配分

- $q(i)$ = 仕事 i の雇用者の取り分 (≥ 0)
- $p(j)$ = 労働者 j の取り分 (≥ 0)
- 労働者 j が仕事 i に従事 $\rightarrow$ 収益 $v(i,j)$
 $\therefore q(i) + p(j) \geq v(i,j)$ を満たさないと, 不満が生じる

仲介役の解くべき問題:

全労働者および全雇用者が不満をもつことなく,
配分するお金の総額を最小に ← 双対問題

双対問題の定式化

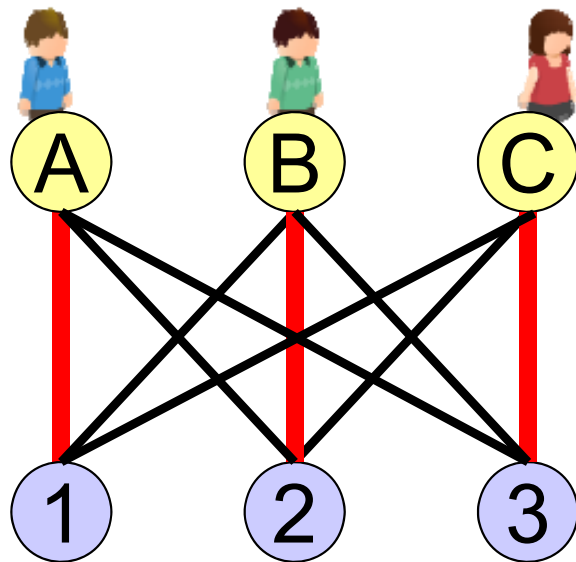
最小化 $\sum_{i \in B} q(i) + \sum_{j \in N} p(j)$

条件 $q(i) + p(j) \geq v(i, j) \quad (\forall i \in B, \forall j \in N)$

$$q(i) \geq 0 \quad (\forall i \in B)$$

$$p(j) \geq 0 \quad (\forall j \in N)$$

オークションと最大重みマッチング問題の関係



$v(i,j)$	A	B	C
①	3	1	1
②	6	5	3
③	2	4	4

二部グラフを次のように定義

- 頂点集合 $V = B \cup N$
- 枝集合 $E = B \times N \quad (= \{ (i,j) \mid i \in B, j \in N \})$
- 各枝 (i,j) の重み $= v(i,j)$

➔ 最大重みマッチング問題が得られる

財の配分と二部グラフのマッチングは1対1対応

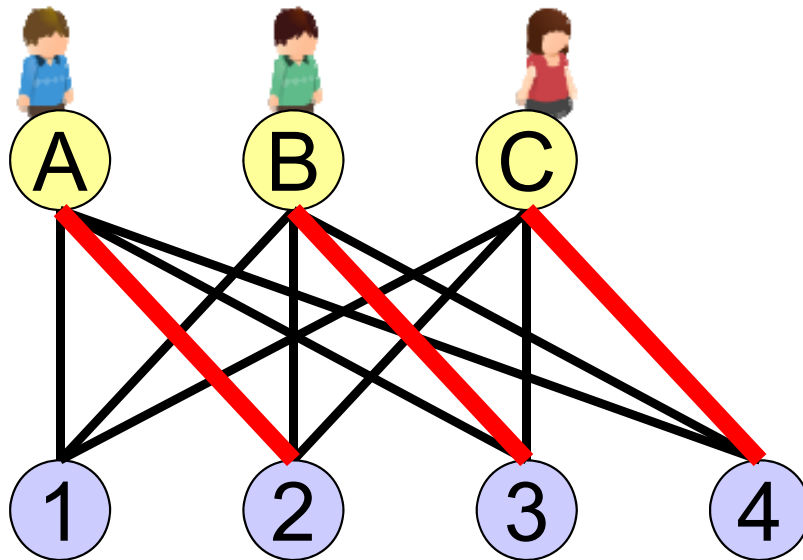
- 配分 $\alpha(i)$ に対応するマッチング $M = \{ (i, \alpha(i)) \mid i \in B, \alpha(i) \neq 0 \}$

均衡配分と最大重みマッチングの関係

オークションの均衡配分 = 最大重みマッチング

定理: $\alpha^*(i) \in N \cup \{0\}$ は均衡配分

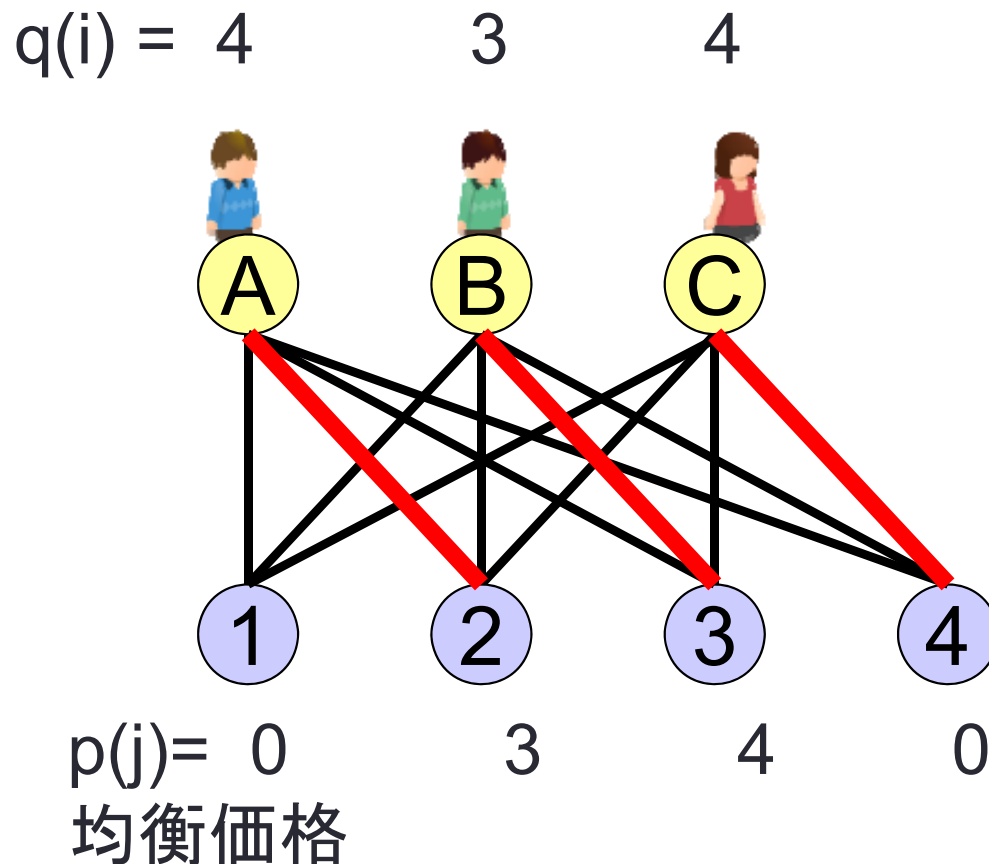
$\longleftrightarrow \alpha^*(i)$ に対応するマッチングは最大重み



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

均衡価格と最大重みマッチングの双対問題の関係

オークションの均衡価格 $\leftrightarrow$ 双対問題の最適解



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

均衡配分ならば最大重みマッチング(1)

命題: α^* は均衡配分

→ α^* に対応するマッチングは最大重み

[証明の概略]

任意のマッチング(に対応する配分) $\alpha(i) \in N \cup \{0\}$ に対し,

α の重み $\leq \alpha^*$ の重み ①

を示せばよい. 以下では, α^*, α 共にすべての入札者に
財の割当がある場合のみ考える

→ α^* の重み = $\sum_{i \in B} v(i, \alpha^*(i))$, α の重み = $\sum_{i \in B} v(i, \alpha(i))$ と書ける
 $S^* = \{\alpha^*(i) | i \in B\}$, $S = \{\alpha(i) | i \in B\}$ とおく.

α^* は均衡配分 → ある財の価格 $p(j) \in \mathbb{R}_+$ が存在して
配分 α^* と価格 p のペアはワルラス均衡

$$v(i, \alpha^*(i)) - p(\alpha^*(i)) \geq v(i, h) - p(h) \quad (\forall h \in N) \quad \textcircled{2}$$

$$p(j) = 0 \quad (\forall j \in N \setminus S^*) \quad \textcircled{3}$$

均衡配分ならば最大重みマッチング(2)

(証明のつづき)

② より, 各入札者 i に対し

$$v(i, \alpha^*(i)) - p(\alpha^*(i)) \geq v(i, \alpha(i)) - p(\alpha(i))$$

$$\therefore \sum_{i \in B} v(i, \alpha^*(i)) \geq \sum_{i \in B} v(i, \alpha(i)) + \sum_{i \in B} p(\alpha^*(i)) - \sum_{i \in B} p(\alpha(i))$$

$$\sum_{i \in B} p(\alpha^*(i)) - \sum_{i \in B} p(\alpha(i)) = \sum_{j \in S^* \setminus S} p(j) - \sum_{j \in S \setminus S^*} p(j) \geq 0$$

($\because$ 価格の非負性と③)

よって,
$$\sum_{i \in B} v(i, \alpha^*(i)) \geq \sum_{i \in B} v(i, \alpha(i))$$

最大重みマッチングならば均衡配分

命題: 均衡が存在するとき,
配分 α に対応するマッチングが最大重み $\rightarrow \alpha$ は均衡配分

[証明の概略]

α^*, p はワルラス均衡とする

$\rightarrow$ 直前の命題より, α^* もまた最大重み

$$\therefore \sum_{i \in B} v(i, \alpha^*(i)) = \sum_{i \in B} v(i, \alpha(i))$$

前のスライドの証明において, この等号が成り立つためには,

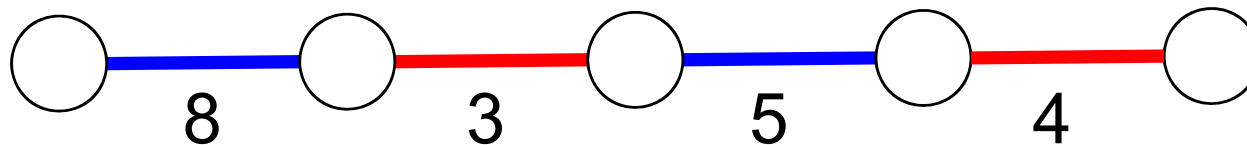
不等式② および $p(j) \geq 0$ ($\forall j \in S^* \setminus S$) が等号で成り立つ必要あり

$\rightarrow \alpha, p$ もワルラス均衡の条件を満たす

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



交互路の重み
 $= 8 - 3 + 5 - 4 = 6$

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)

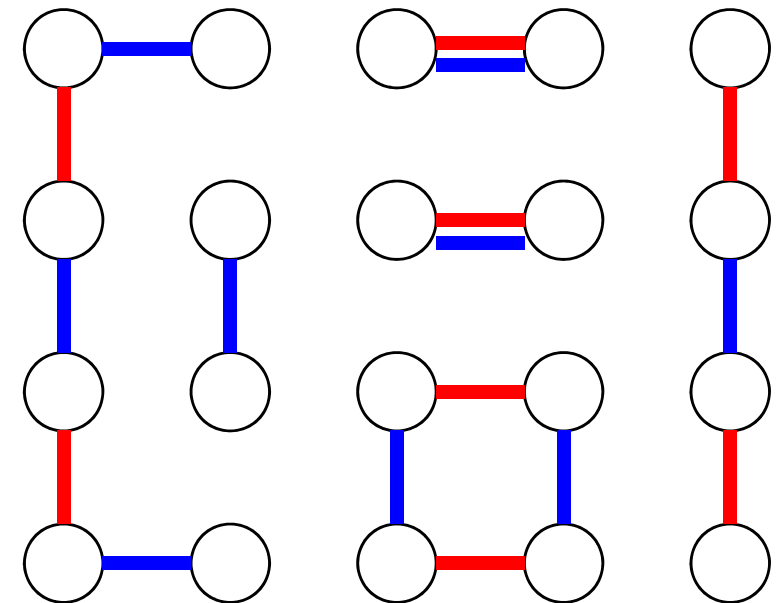
交互路・交互閉路からなる.

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)

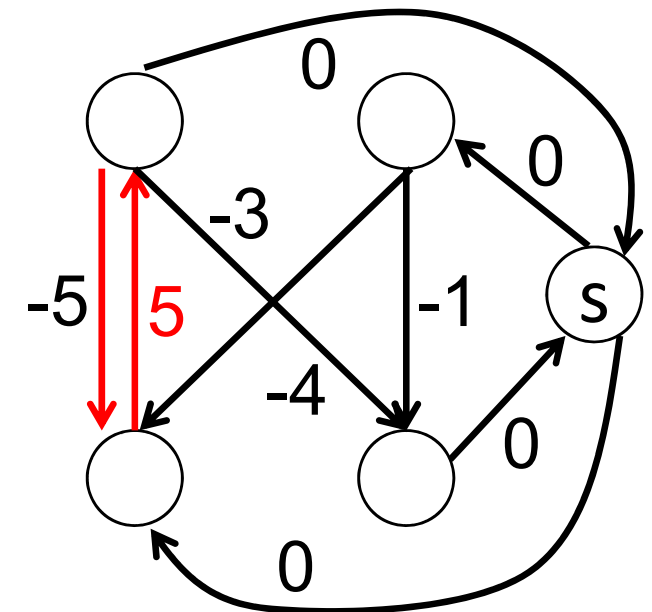
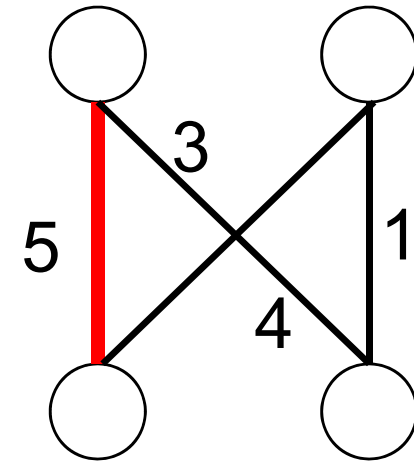
- 全ての枝を, B 側から N 側に向き付け
- マッチング M の各枝に対し, さらに

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 → 枝 (u, s) をおく
 - 接続なし → (s, u) をおく

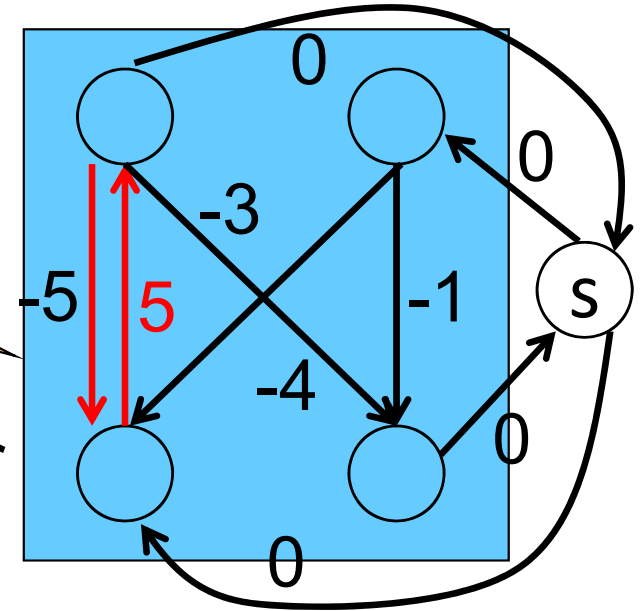
- N 側の各頂点 v に対し,
 - マッチング枝が接続 → 枝 (s, v) をおく
 - 接続なし → (v, s) をおく



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

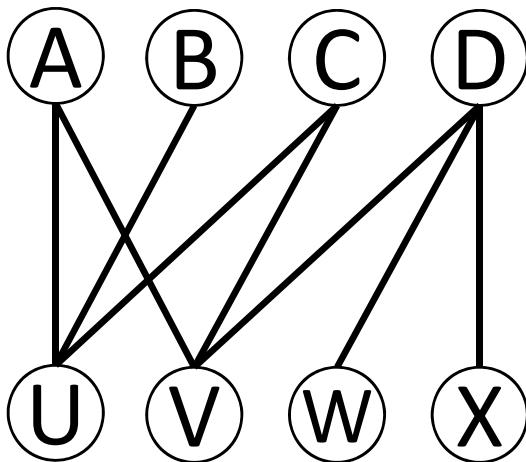
($\because$ 頂点 s を通らない有向閉路は交互閉路に対応)

$\therefore$ 最短路は存在

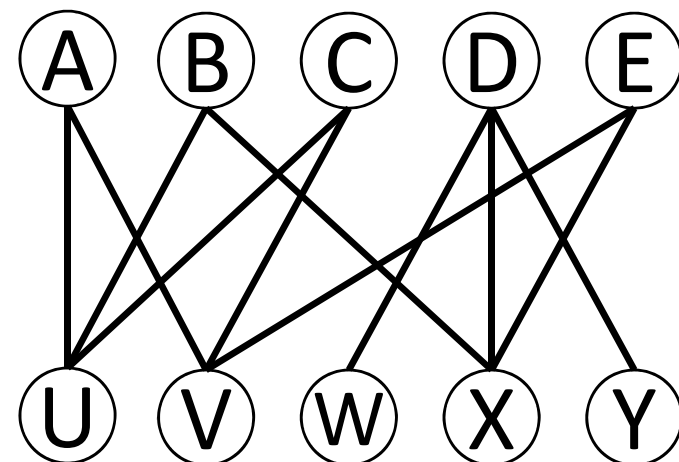
演習問題

問1: 以下の二部グラフでは, 頂点集合 R をカバーする
マッチングが存在しない.
ホールの定理に基づき, その証拠を示せ.

(1) $R = \{A, B, C, D\}$



(2) $R = \{A, B, C, D, E\}$

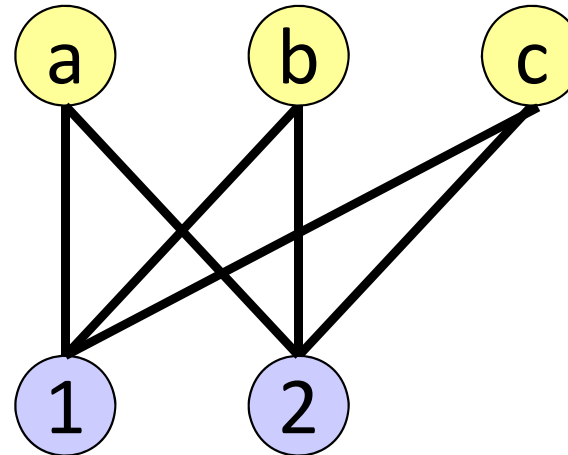


演習問題

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ハンガリー法により求めよ.
各反復で使った交互路および得られたマッチングを書くこと.

(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

