離散最適化基礎論　第 13 回
一般グラフの最小費用完全マッチング：アルゴリズム (詳細)

岡本 吉央
okamotoy@uec.ac.jp

電気通信大学

2021 年 1 月 26 日

最終更新：2021 年 1 月 29 日　21:16

スケジュール 前半

1. マッチングの用語 (10/6)
2. 二部グラフの最大マッチング (10/13)
3. 二部グラフの最大マッチング：アルゴリズム (10/20)
4. 一般グラフの最大マッチング (10/27)
- ⋆ 祝日 のため 休み (11/3)
5. 一般グラフの最大マッチング：アルゴリズム (11/10)
6. 線形計画法の復習 (11/17)
- ⋆ 調布祭片付け のため 休み (11/24)
7. 整数計画法の復習 (12/1)

スケジュール 後半 (予定)

8 二部グラフの最小費用完全マッチング：線形計画法 (12/8)

⋆ 国際会議 のため 休み (12/15)

9 二部グラフの最小費用完全マッチング：アルゴリズム (12/22)

10 一般グラフの最小費用完全マッチング：線形計画法 (1/5)

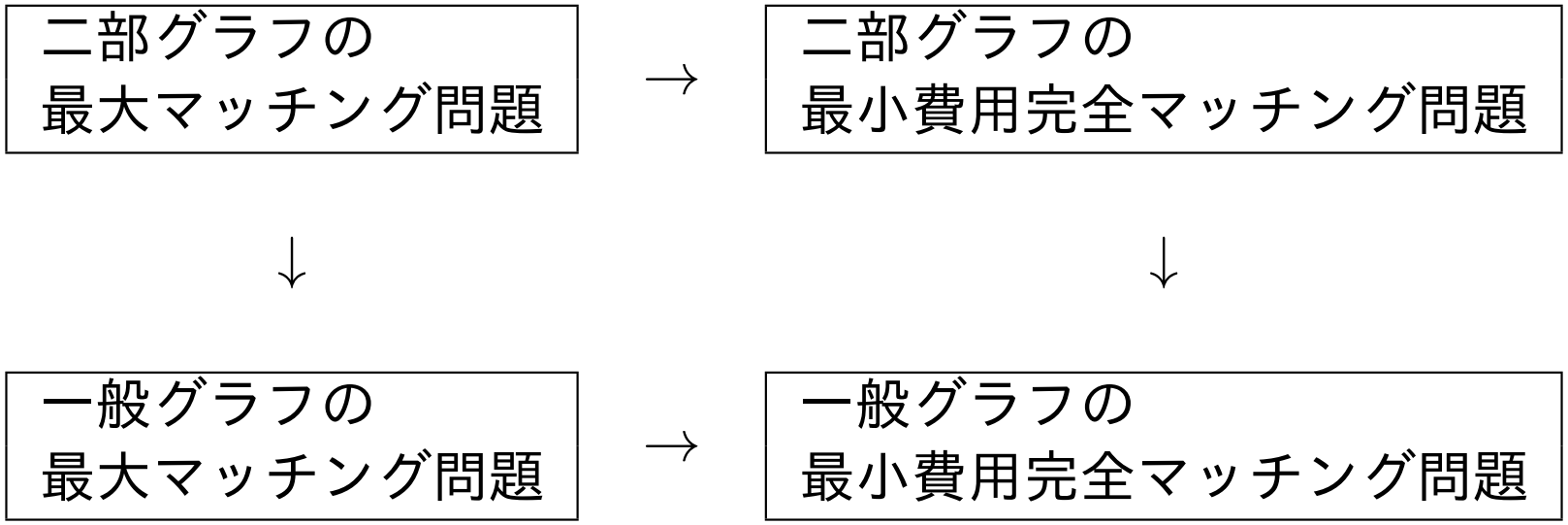
11 一般グラフの最小費用完全マッチング：完全整数双対性 (1/12)

12 一般グラフの最小費用完全マッチング：アルゴリズム (概要) (1/19)

13 一般グラフの最小費用完全マッチング：アルゴリズム (詳細) (1/26)

14 予備 (2/2)

この講義が対象とする 4 つの問題



この講義で行うこと

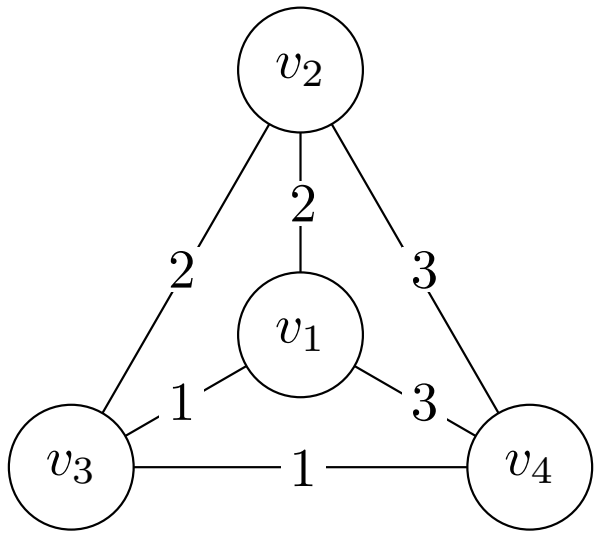
「これら 4 つの問題は多項式時間で解けること」の説明

重要な考え方

- 最適化における最大最小定理 (強双対性)
- 費用無し問題のアルゴリズム + 線形計画法 $\rightsquigarrow$
  費用有り問題のアルゴリズム (主双対法)

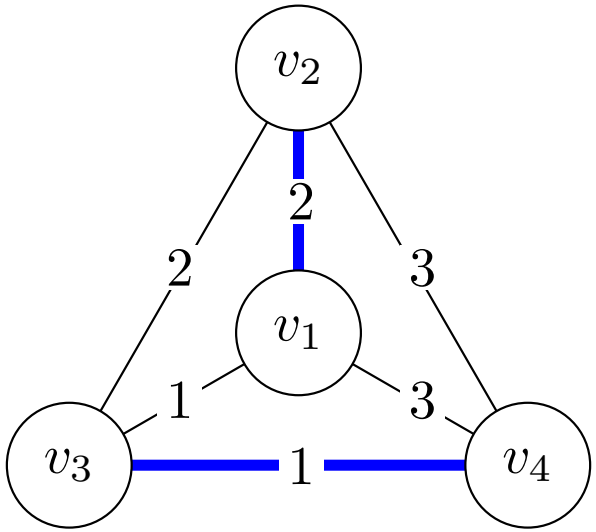
定義：最小費用完全マッチング問題 (minimum-cost perfect matching problem)

- 入力：無向グラフ $G = (V, E)$，非負辺費用関数 $c\colon E \to \mathbb{R}$
- 出力：G の完全マッチング M で，辺費用和が最小のもの (を 1 つ)
  (完全マッチングが存在しない場合,「存在しない」と出力)



定義：最小費用完全マッチング問題 (minimum-cost perfect matching problem)

- 入力：無向グラフ $G = (V, E)$，非負辺費用関数 $c\colon E \to \mathbb{R}$
- 出力：G の完全マッチング M で，辺費用和が最小のもの (を 1 つ)
  (完全マッチングが存在しない場合,「存在しない」と出力)



今日の目標

前回の内容

- 一般グラフの最小費用完全マッチング問題を解くための
  アルゴリズムを記述した
- そのアルゴリズムが停止したとき,
  正しい出力 (最小費用完全マッチング) が得られることを証明した

今回の内容

- そのアルゴリズムが多項式時間アルゴリズムであることを
  証明する

$\mathcal{C} = \{S \subseteq V \mid |S| \geq 3, |V - S| \geq 3, |S|$ が奇数 $\}$ とする

(IP')：奇カット不等式を追加した整数計画問題

$$\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} c_e x_e & & \\
\text{subject to} \quad & \sum_{e \in \delta(v)} x_e = 1 & & (\forall\, v \in V), \\
& \sum_{e \in \delta(S)} x_e \geq 1 & & (\forall\, S \in \mathcal{C}), \\
& x_e \in \{0, 1\} & & (\forall\, e \in E)
\end{aligned}$$

性質 (第 10 回講義で証明)

(IP) の許容領域 $=$ (IP') の許容領域

(LP') : (IP') の線形計画緩和

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} c_e x_e & & \\
\text{subject to} \quad & \sum_{e \in \delta(v)} x_e = 1 & & (\forall\, v \in V), \\
& \sum_{e \in \delta(S)} x_e \geq 1 & & (\forall\, S \in \mathcal{C}), \\
& x_e \geq 0 & & (\forall\, e \in E)
\end{aligned}
$$

定理 (第 11 回講義で証明) (Edmonds '65)

任意の無向グラフ $G = (V, E)$ に対して，
(LP') の最適解で，(IP) の許容解となるものが存在する

- この定理から，
  一般グラフの最小費用完全マッチング問題に対しても，
  主双対法が作れる (ような気がしてくる) ← 前回の内容

$$\boxed{\text{(IP) の最適値}} \quad \geq \quad \boxed{\text{(LP) の最適値}}$$
$$\| \qquad\qquad \wedge |$$
$$\boxed{\text{(IP') の最適値}} \quad = \quad \boxed{\text{(LP') の最適値}}$$

(LP') の双対問題

主双対法のアルゴリズムを設計するためには,
(LP') の双対問題を考える必要がある

(DLP') : (LP') の双対問題

変数は，各 $v \in V$ に対する y_v と
各 $S \in \mathcal{C}$ に対する z_S

$$\text{maximize} \quad \sum_{v \in V} y_v + \sum_{S \in \mathcal{C}} z_S$$

$$\text{subject to} \quad y_u + y_v + \sum_{\substack{S \in \mathcal{C}, \\ e \in \delta(S)}} z_S \leq c_e \qquad (\forall\, e = \{u, v\} \in E),$$

$$z_S \geq 0 \qquad (\forall\, S \in \mathcal{C})$$

アルゴリズムのアイディア：主双対法

不変条件：アルゴリズムの実行中，x, y, z は次の条件を満たし続ける

- $x \in \{0,1\}^E$
- y, z は (DLP') の許容解である
- x と y, z は相補性条件を満たす

停止条件：次の条件を満たしたら，アルゴリズムは停止する

- x は (LP') の許容解である

相補性定理の帰結

アルゴリズムが停止したとき，
x は (LP') の最適解で，y, z は (DLP') の最適解である

さらに，$x \in \{0,1\}^E$ なので，x は (IP) の最適解である

我々の目標

一般グラフの最小費用完全マッチング問題を 多項式時間で解くこと

- そのために，主双対法を使いたい
- つまり，(IP) と (DLP') を同時に解きたい
- (DLP') の許容解は $y \in \mathbb{R}^V$ と $z \in \mathbb{R}^{\mathcal{C}}$ で表される

問題点

$|\mathcal{C}|$ はグラフのサイズ $(|V| + |E|)$ に対して，指数関数的に大きい

$\therefore z$ を表現するだけで，多項式時間よりも大きな時間がかかる

問題点

$|\mathcal{C}|$ はグラフのサイズ $(|V| + |E|)$ に対して，指数関数的に大きい

$\therefore$ z を表現するだけで，多項式時間よりも大きな時間がかかる

問題点の克服法

z を格納するときに，

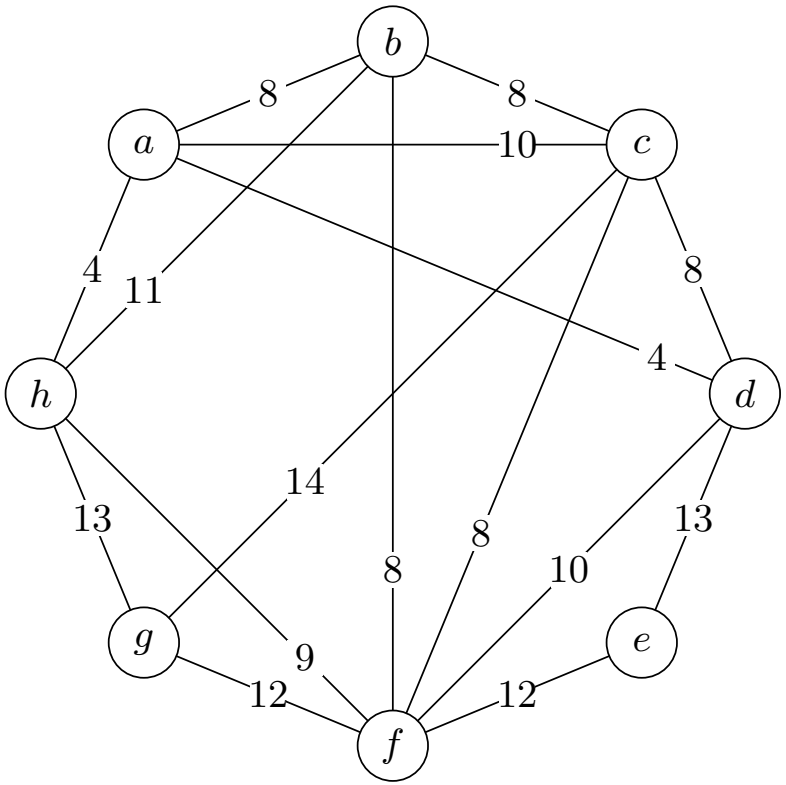
- $z_S = 0$ となる $S \in \mathcal{C}$ は無視して，
- $z_S > 0$ となる $S \in \mathcal{C}$ だけ格納する

いまから説明するアルゴリズムでは，

- このようにしてもちゃんと解けることを説明する
- アルゴリズムの動作中，$z_S > 0$ となる $S \in \mathcal{C}$ の数が
  グラフのサイズの多項式にしかならないことも保証できる

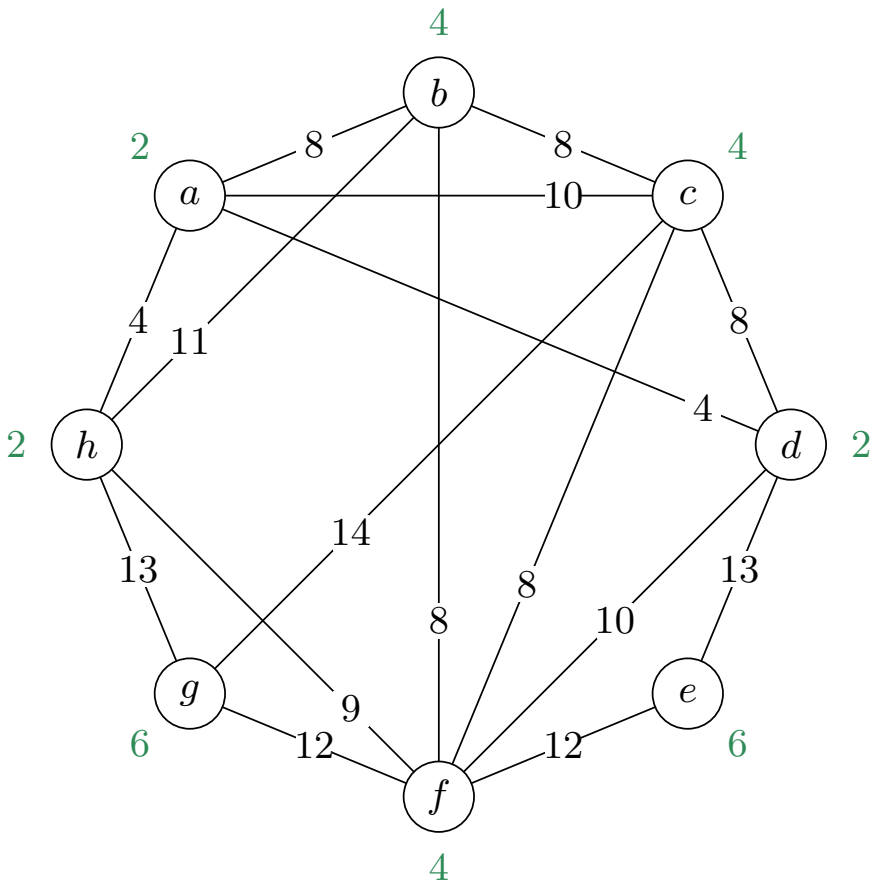
↑ 今日の内容 ↑

主双対法：例 (1)



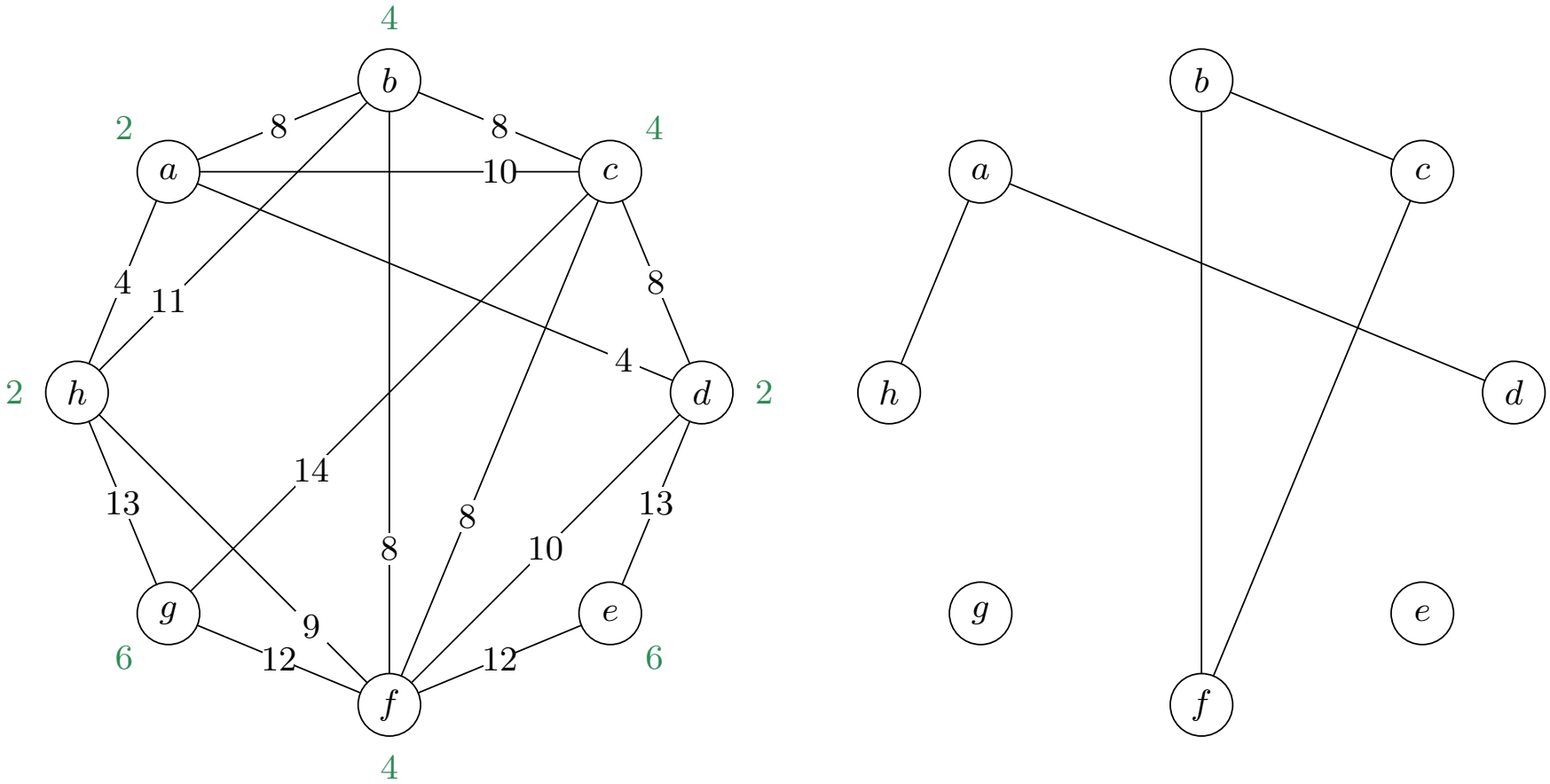
各辺に費用がついている

主双対法：例 (1)



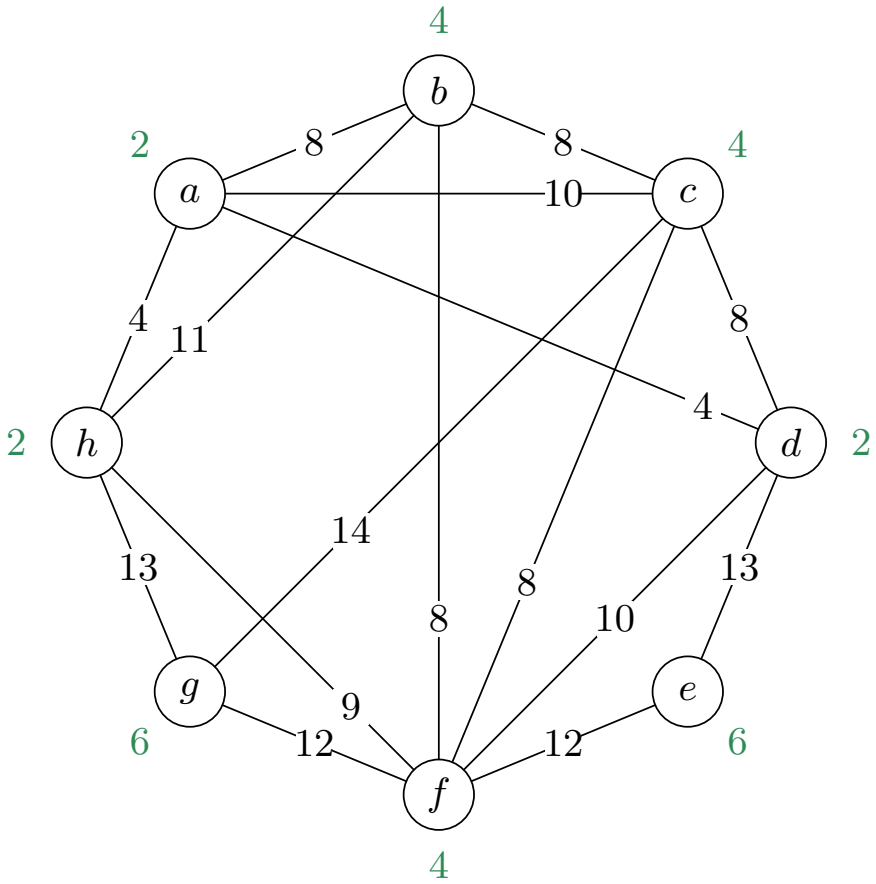
適当に双対変数 y の値を定める ($z = 0$ とする)

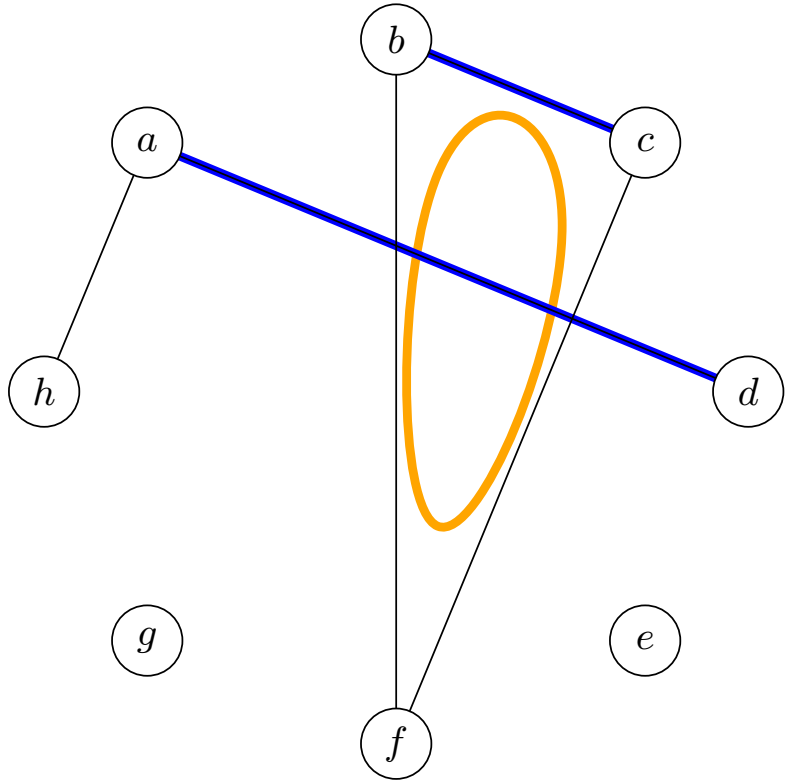
主双対法：例 (1)



$y_u + y_v = c_{u,v}$ となる辺 $\{u, v\}$ 全体から成るグラフ H を作る

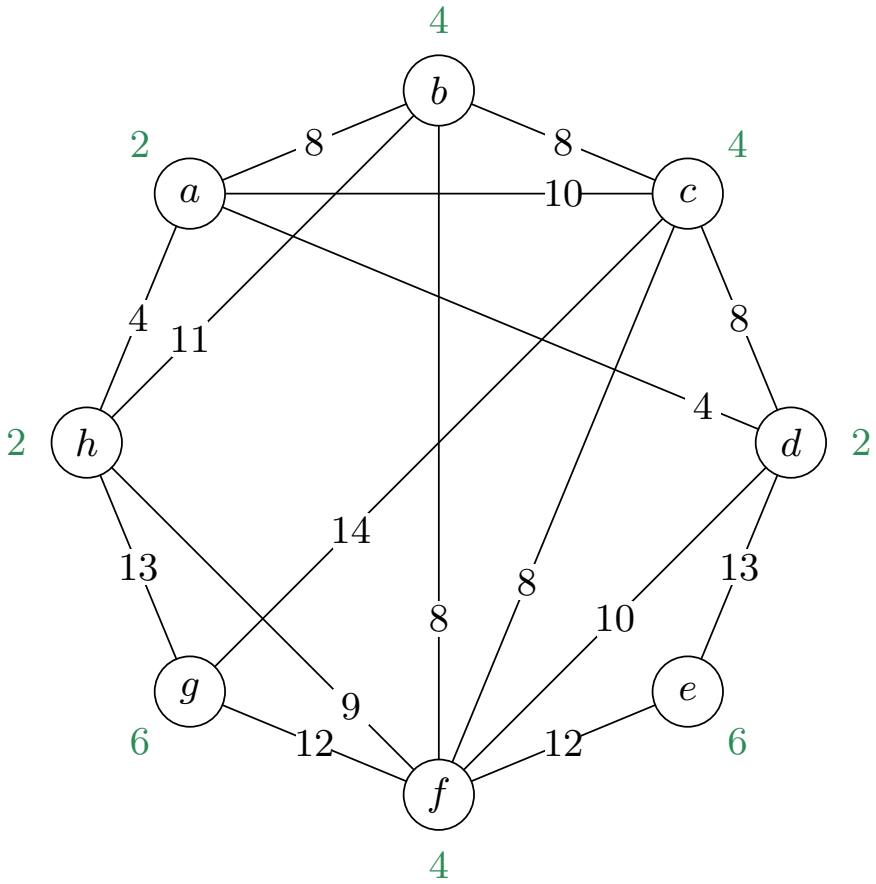
主双対法：例 (1)

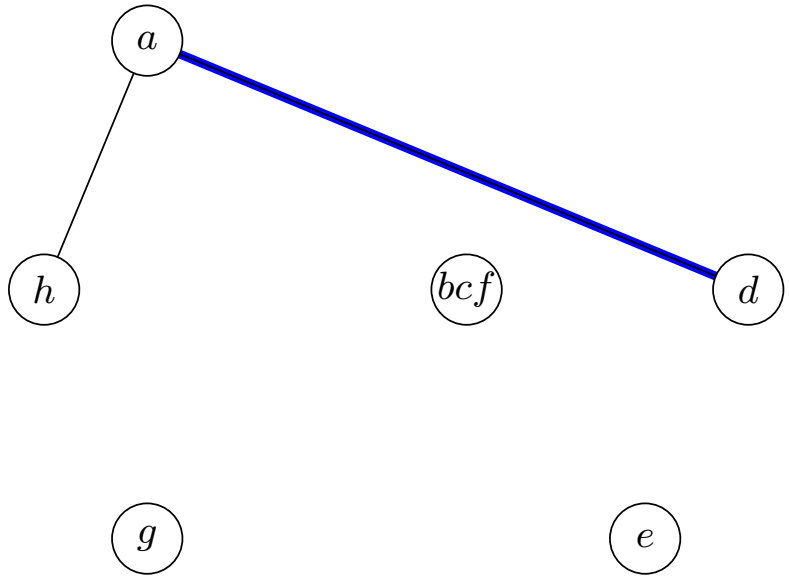




H の最大マッチング M を見つける

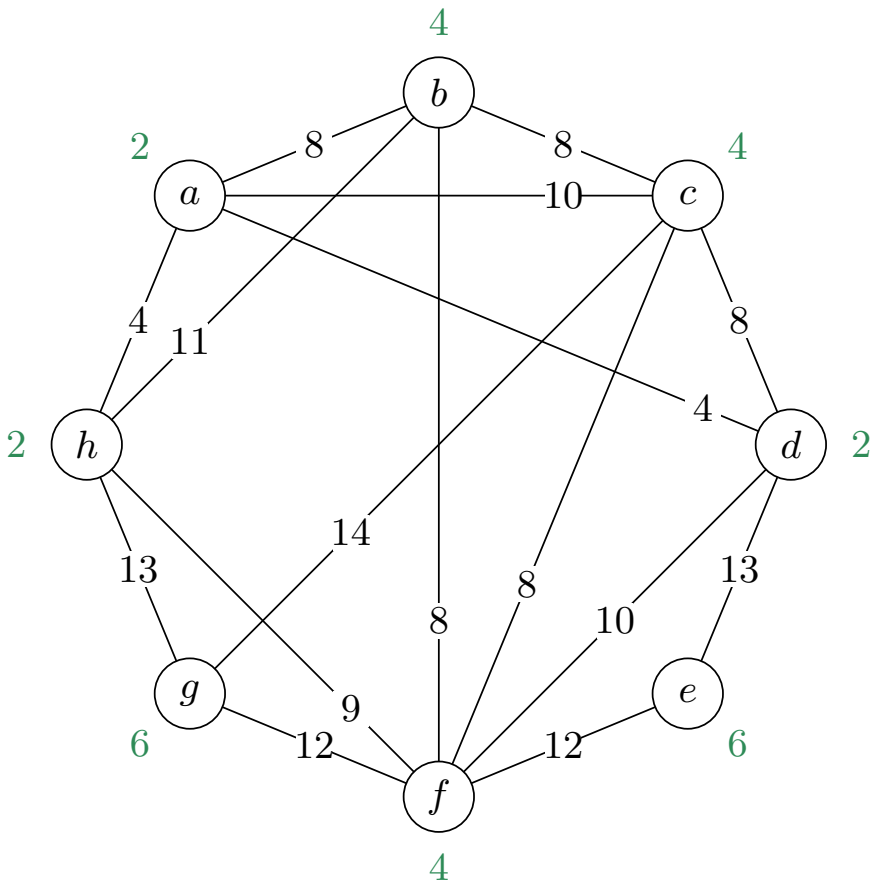
主双対法：例 (1)

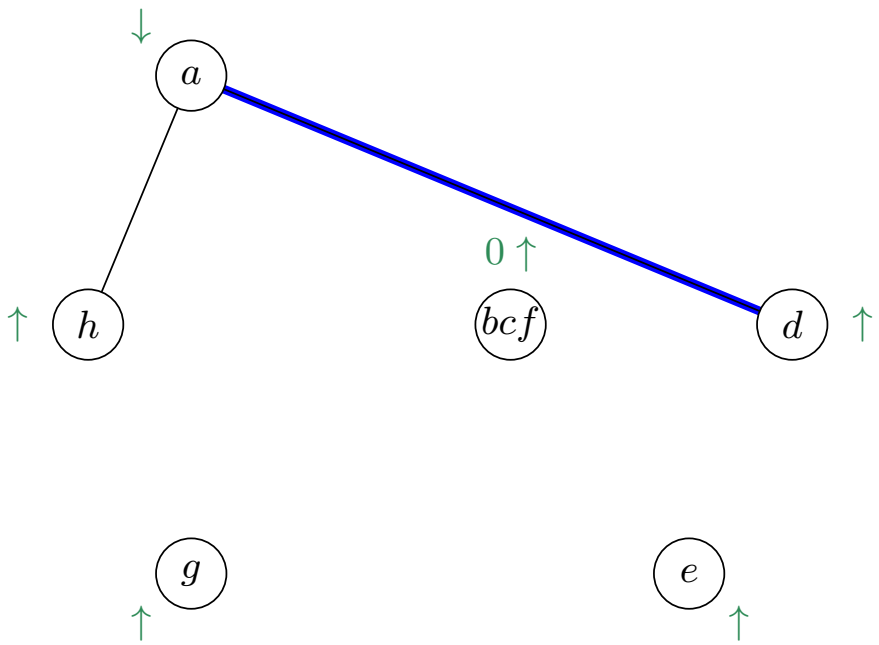




M に関する花 C がある $\Rightarrow$ C を縮約する

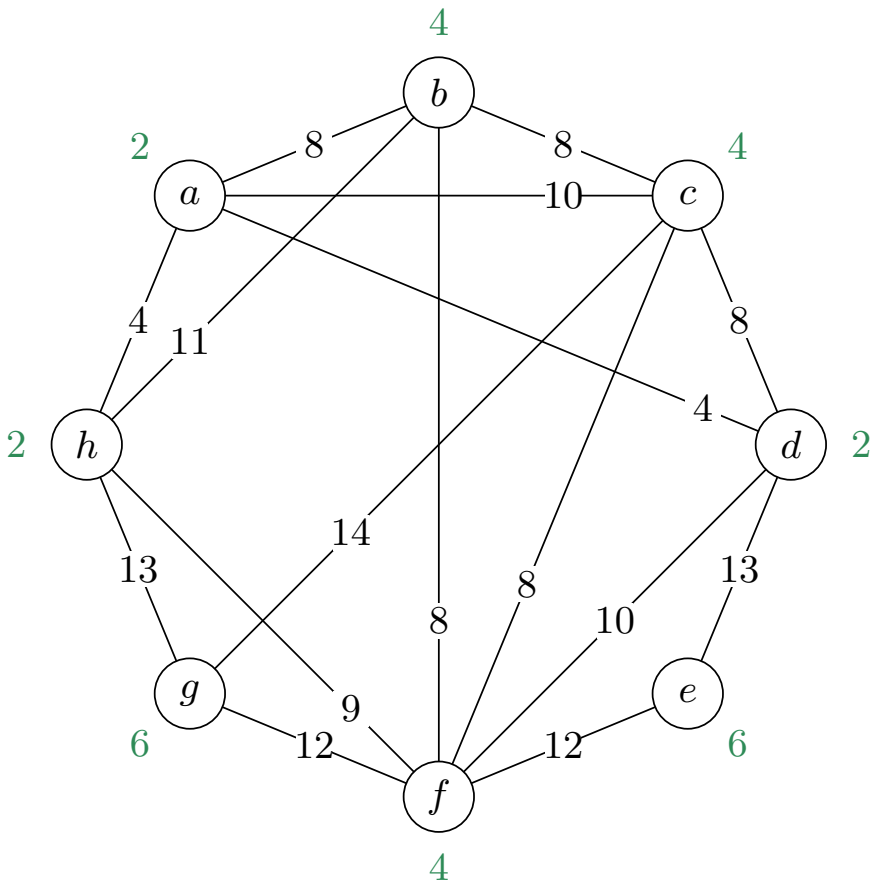
主双対法：例 (1)

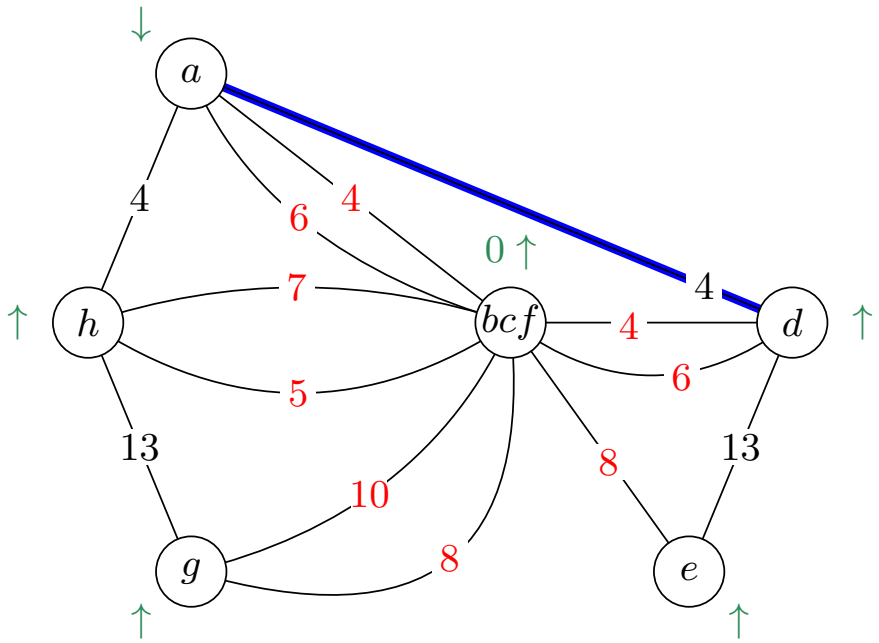




双対変数の更新を行う

主双対法：例 (1)

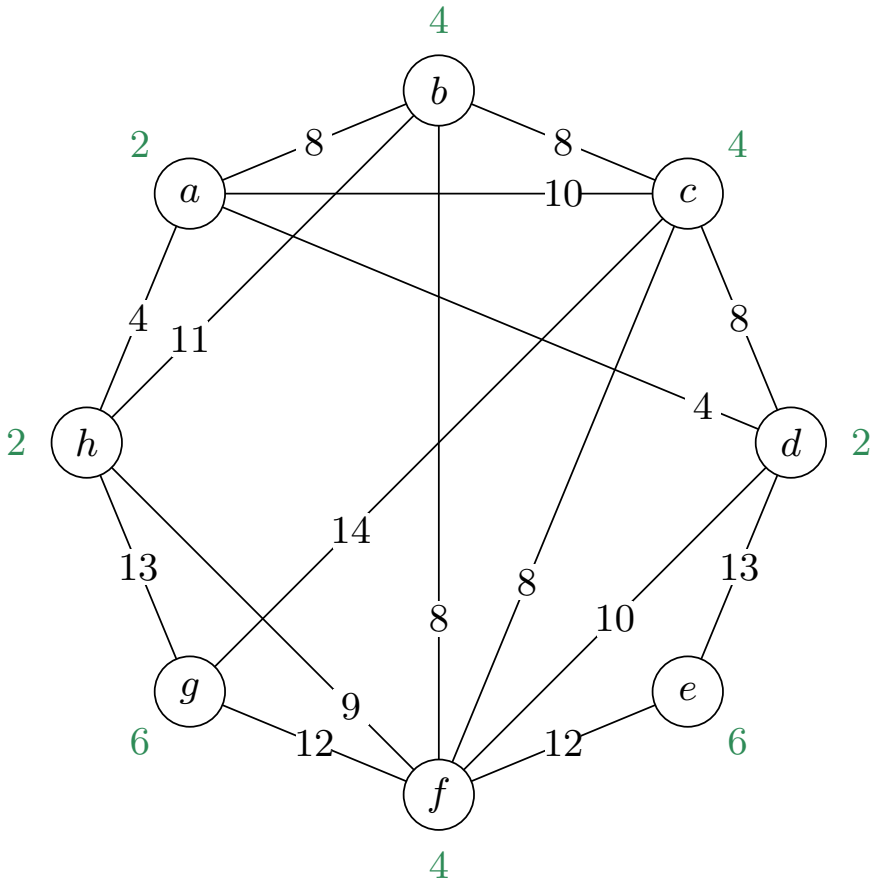


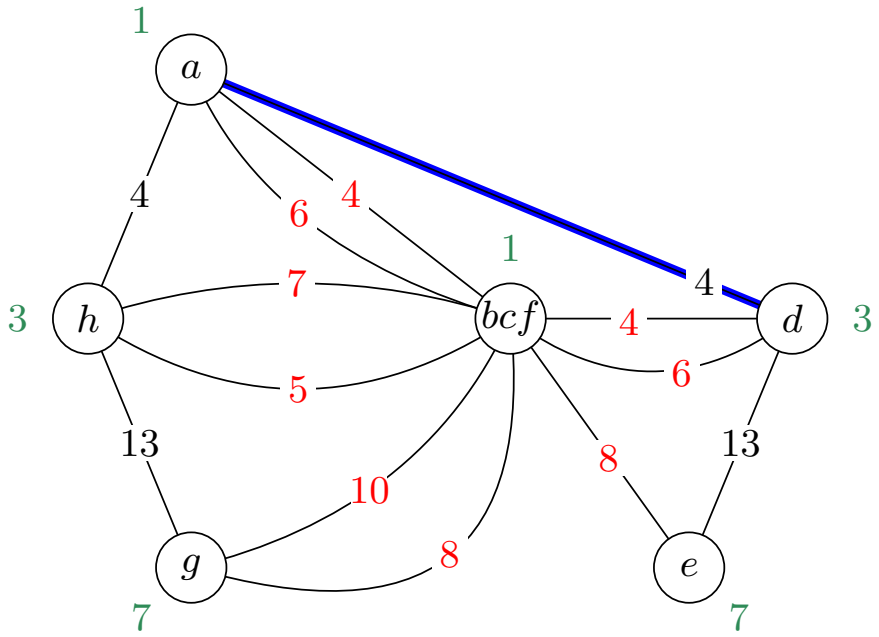


ただし，縮約後のグラフにおける費用も更新する

$c'_{u,v} := c_{u,v} - y_v$ (v は花に含まれる頂点)

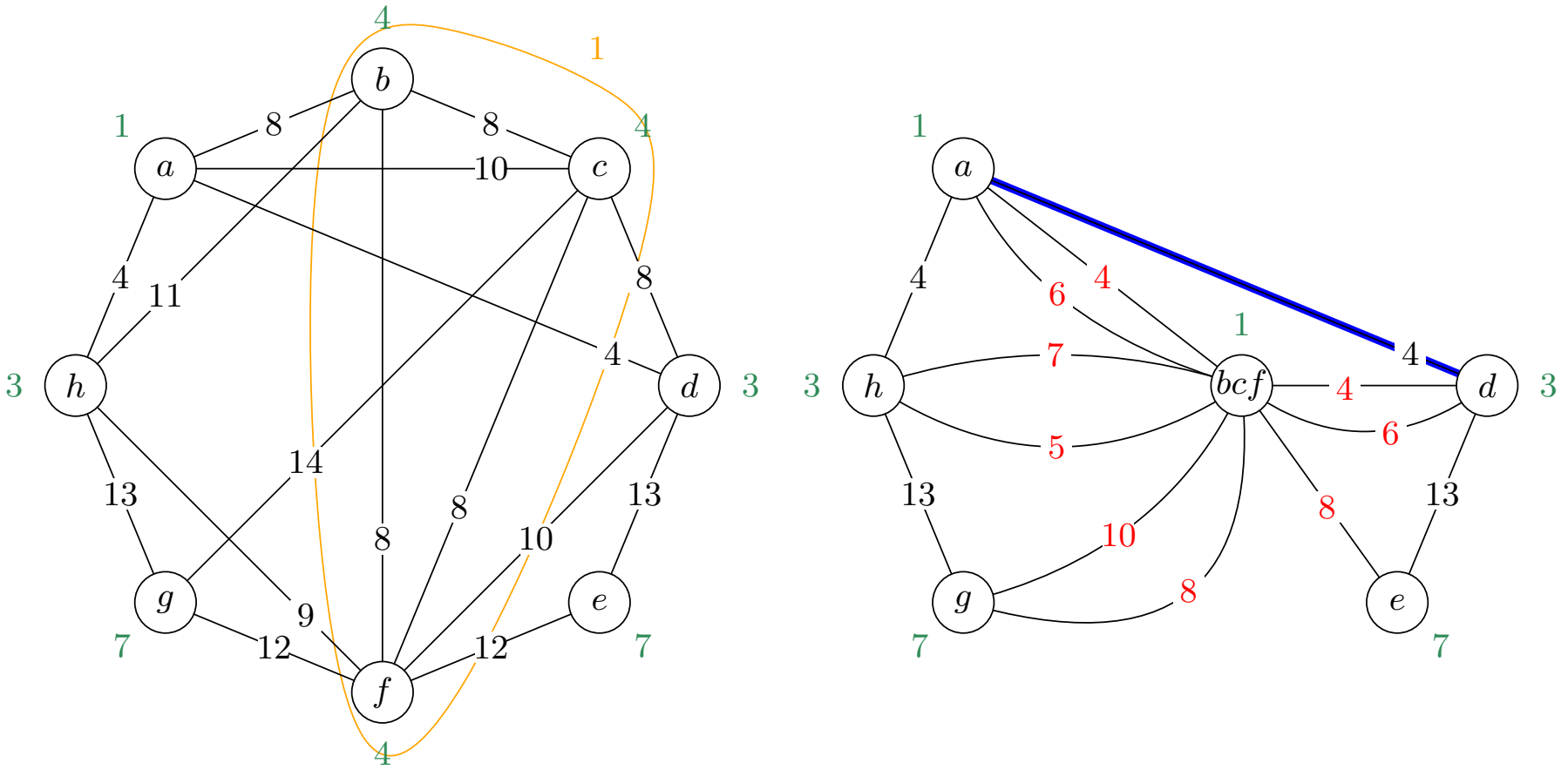
主双対法：例 (1)





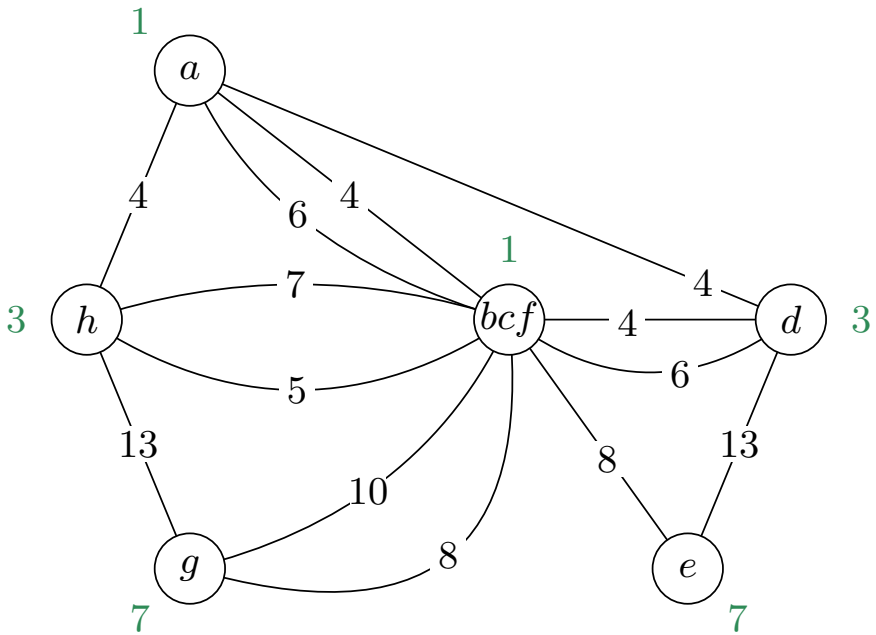
この例では，$\varepsilon = 1$

主双対法：例 (1)



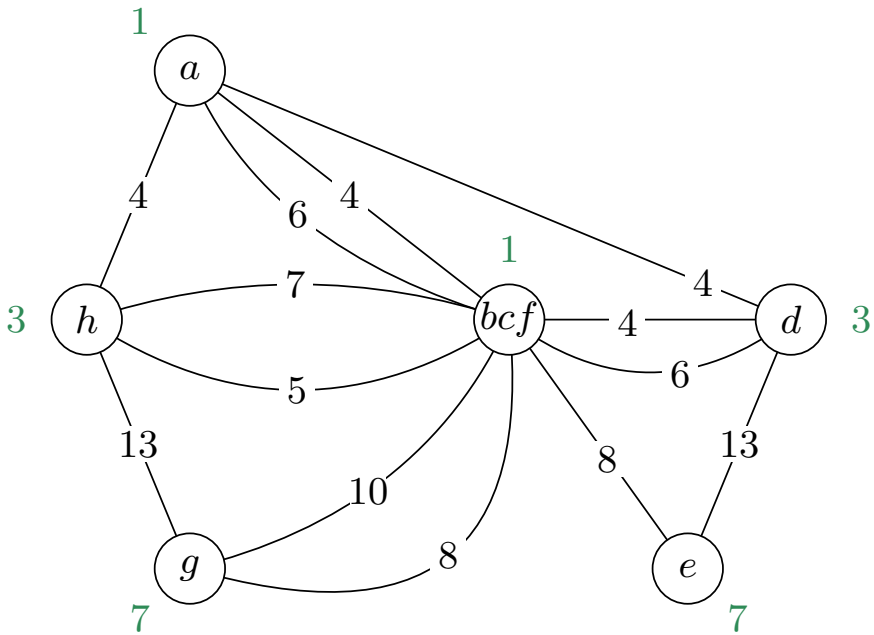
G において，双対変数は次のように解釈される (口頭説明)

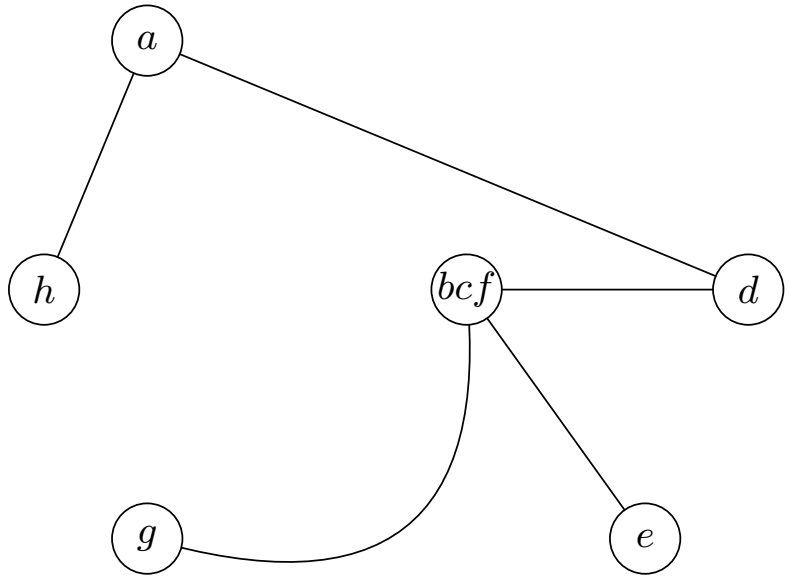
主双対法：例 (2)



このグラフを G として，各頂点 v に双対変数 y_v が与えられている

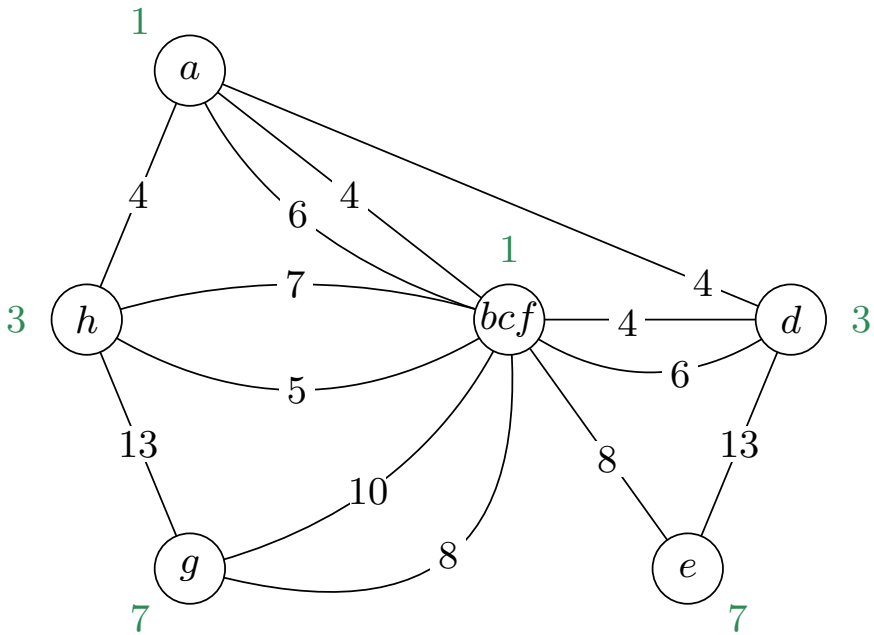
主双対法：例 (2)

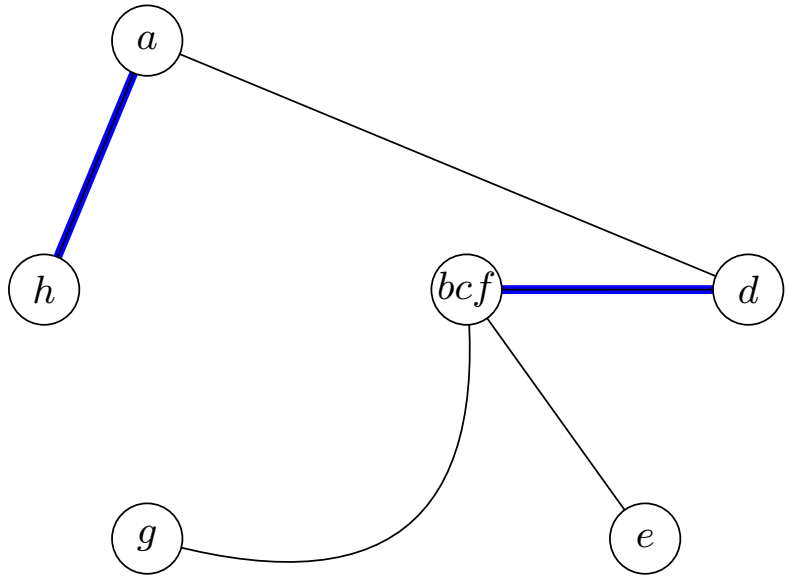




$y_u + y_v = c_{u,v}$ となる辺 $\{u, v\}$ 全体から成るグラフ H を作る

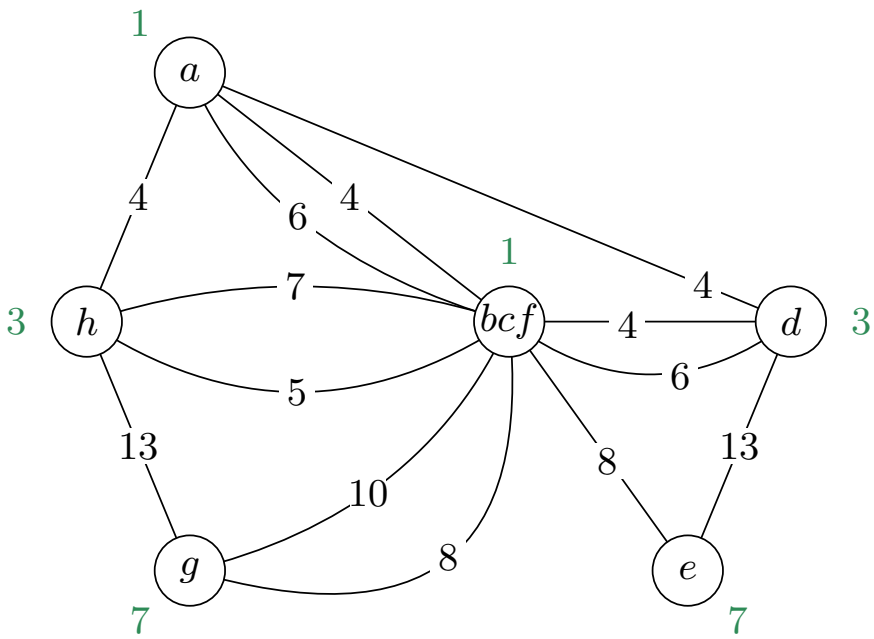
主双対法：例 (2)

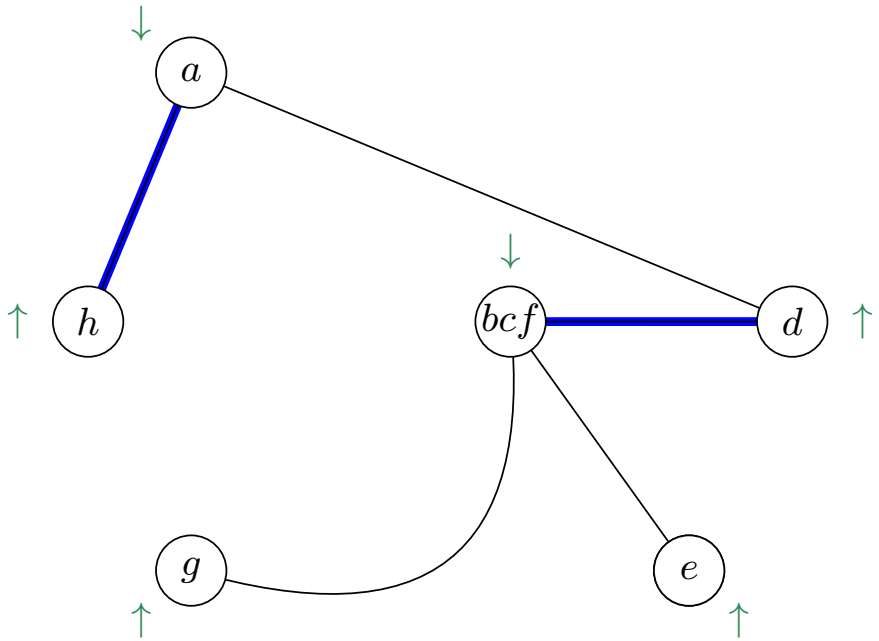




H の最大マッチングを見つける

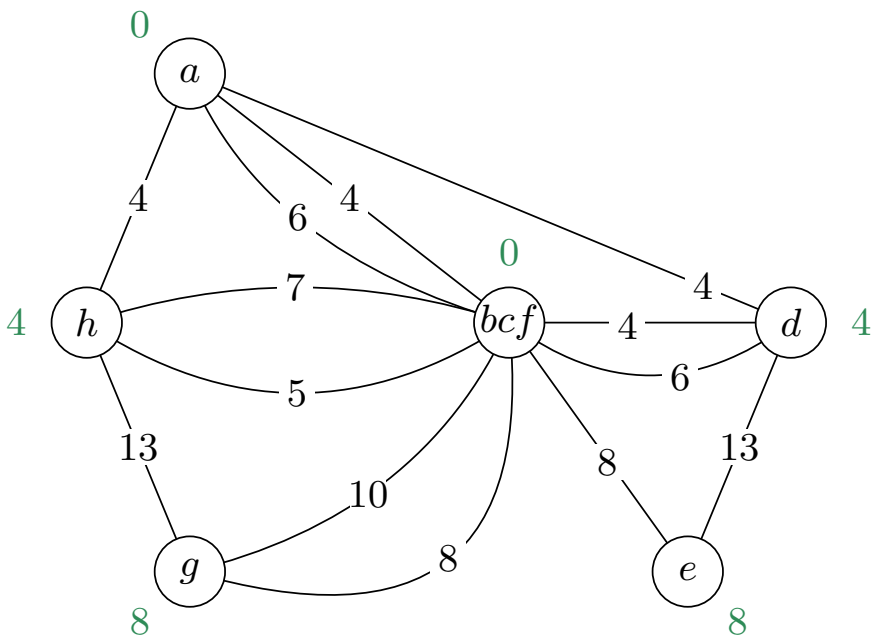
主双対法：例 (2)

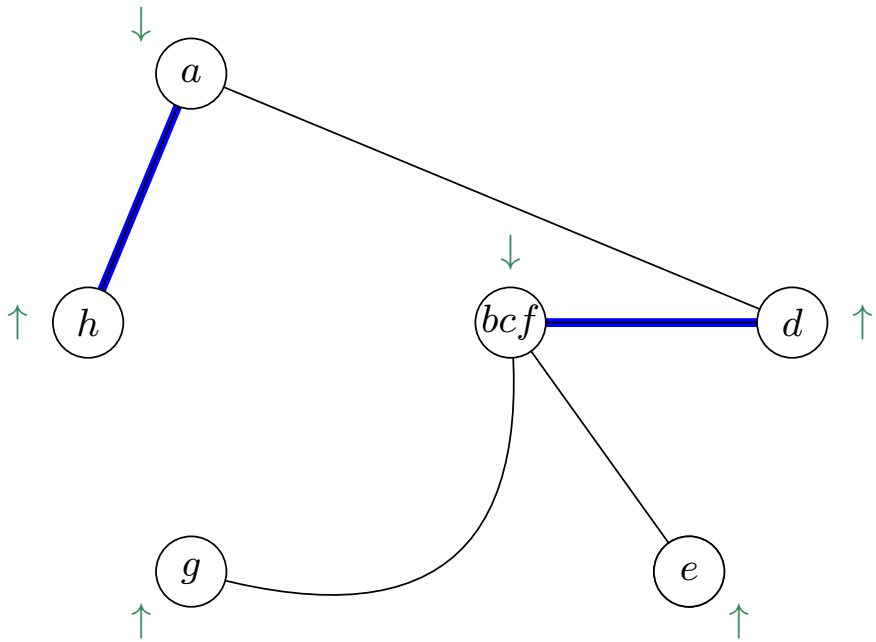




双対変数の更新を行う

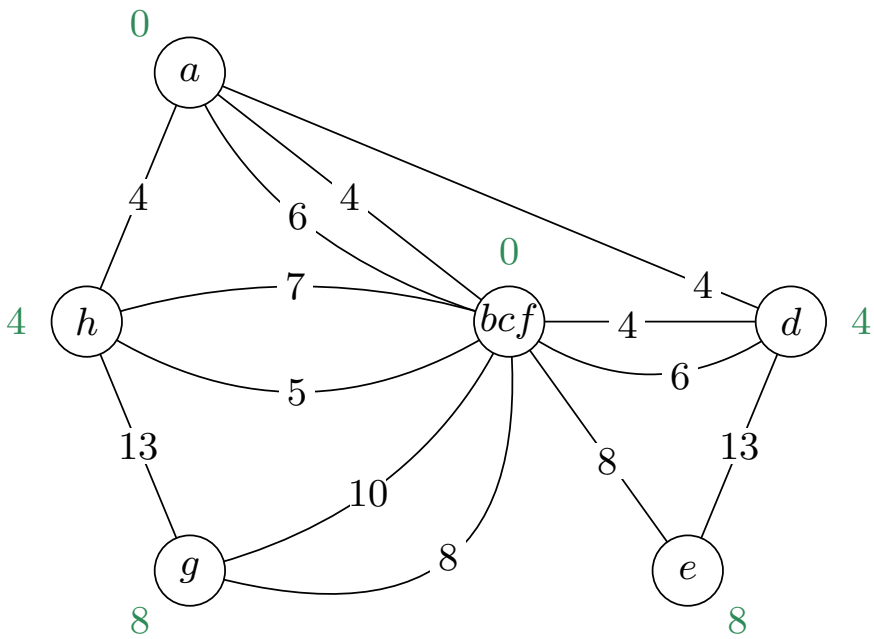
主双対法：例 (2)

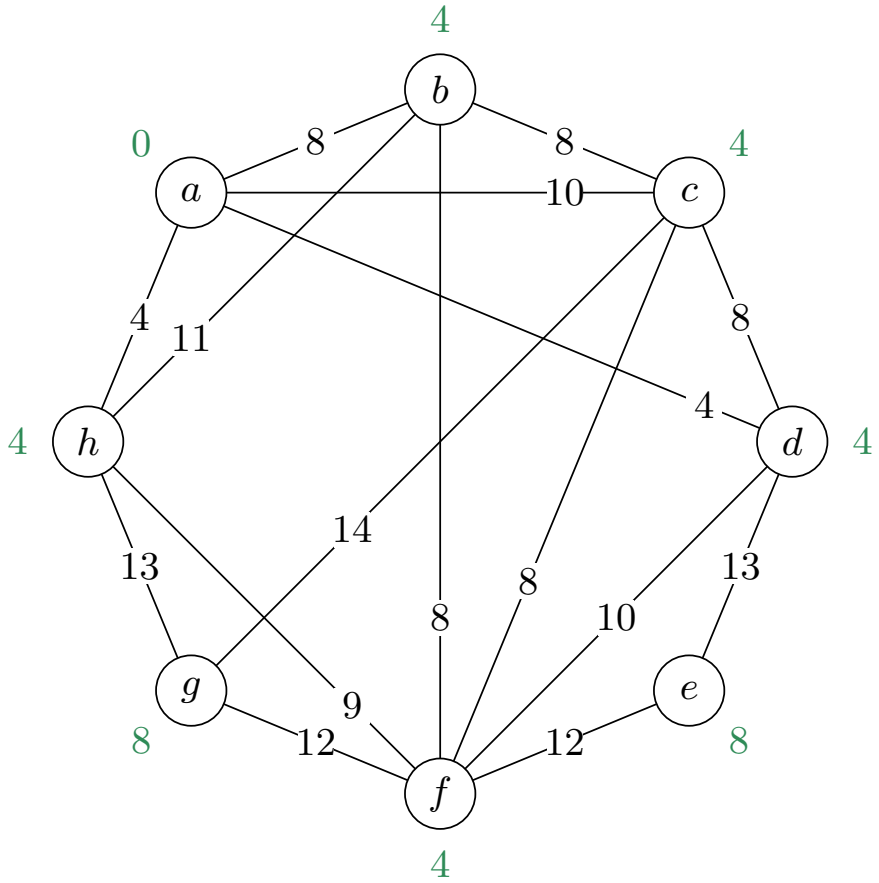




$y_{bcf} = z_{\{b,c,f\}} \geq 0$ なので，$y_{bcf} \geq 0$ であることに注意

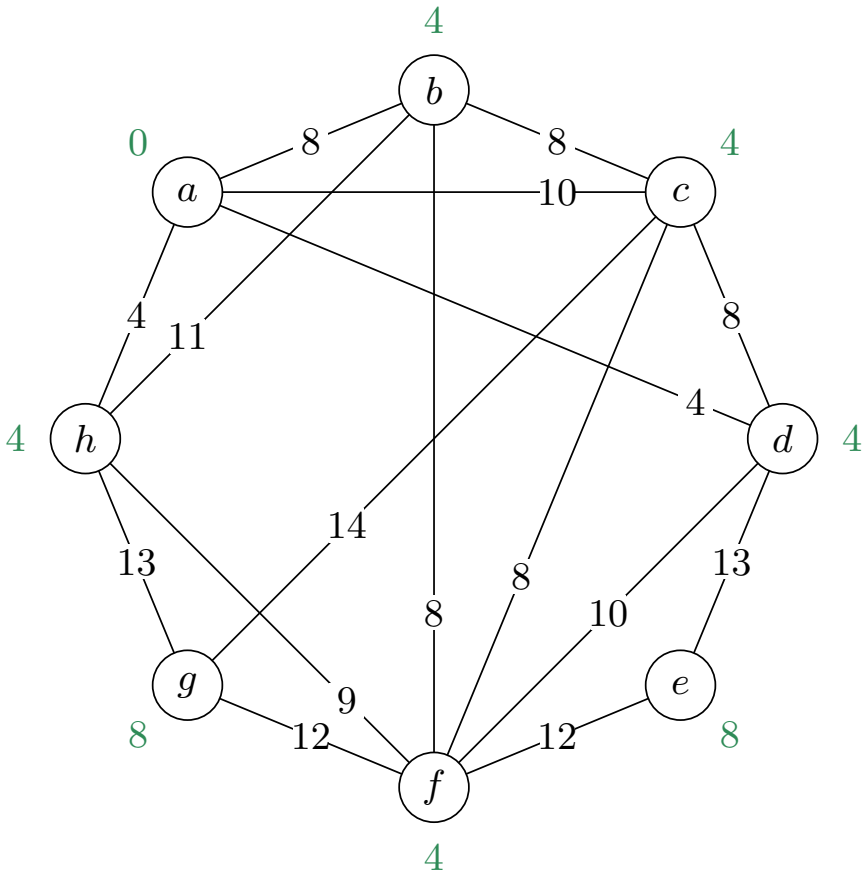
主双対法：例 (2)





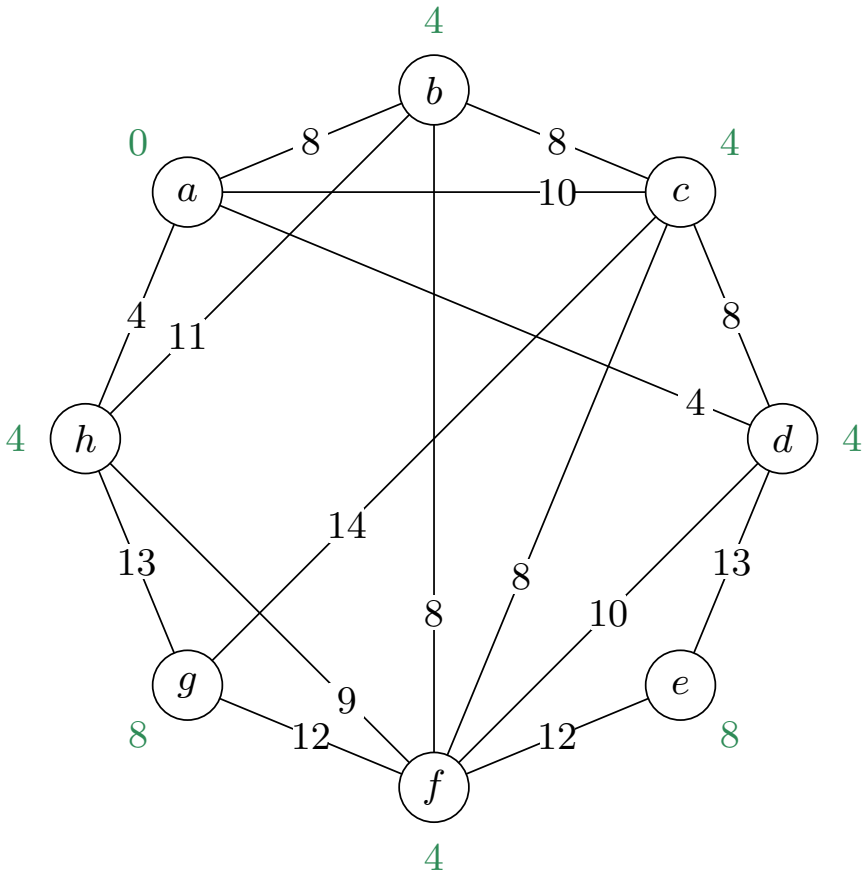
$y_{bcf} = z_{\{b,c,f\}} = 0$ なので，$\{b, c, f\}$ の縮約を戻す

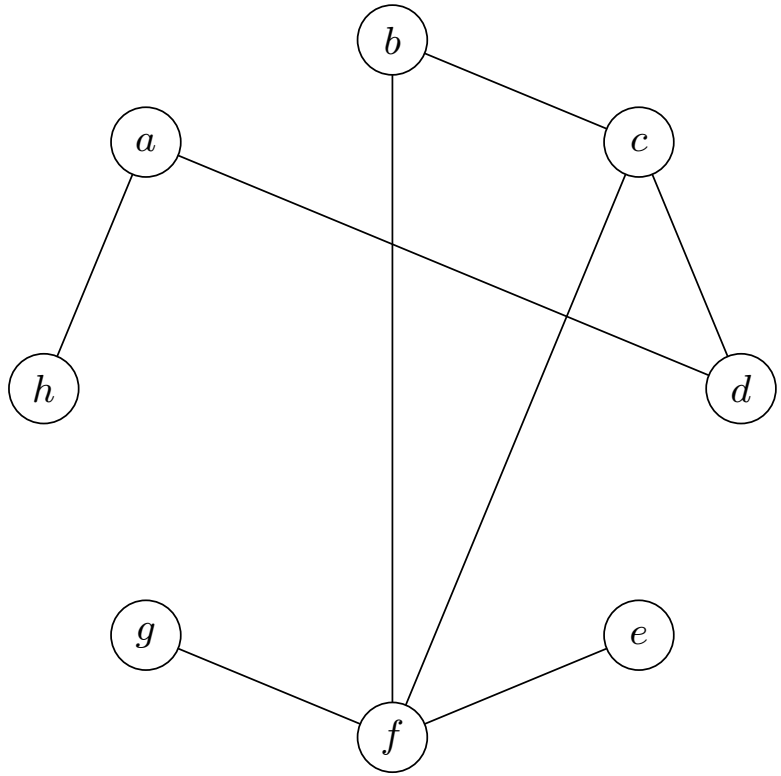
主双対法：例 (3)



このグラフを G として，各頂点 v に双対変数 y_v が与えられている

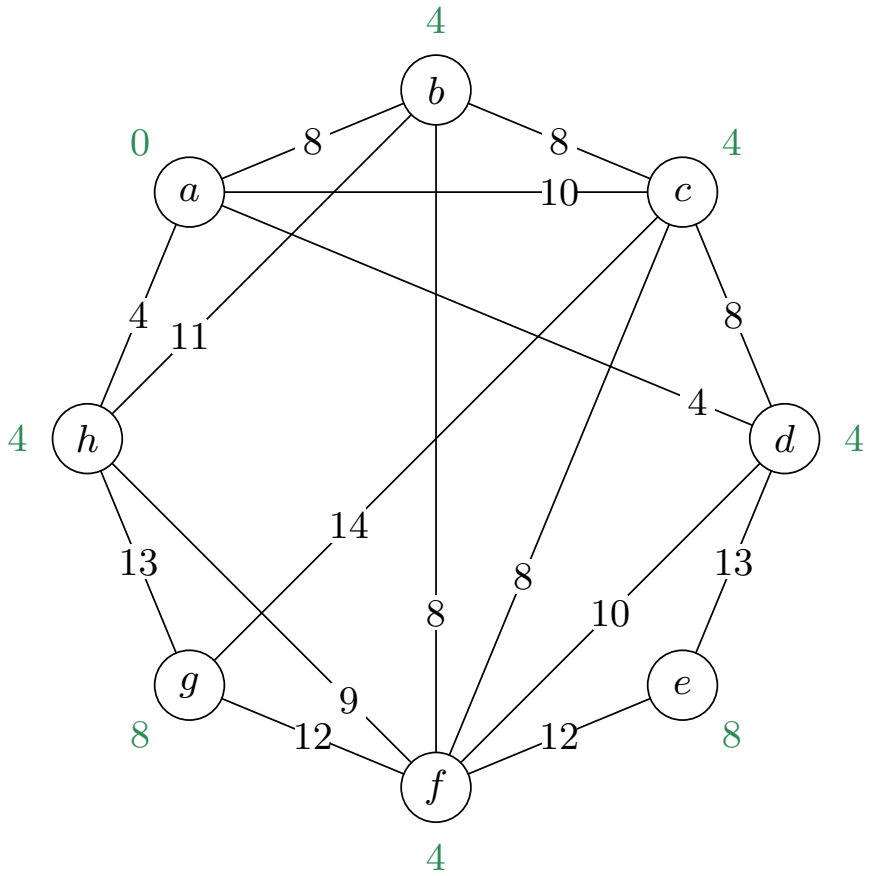
主双対法：例 (3)

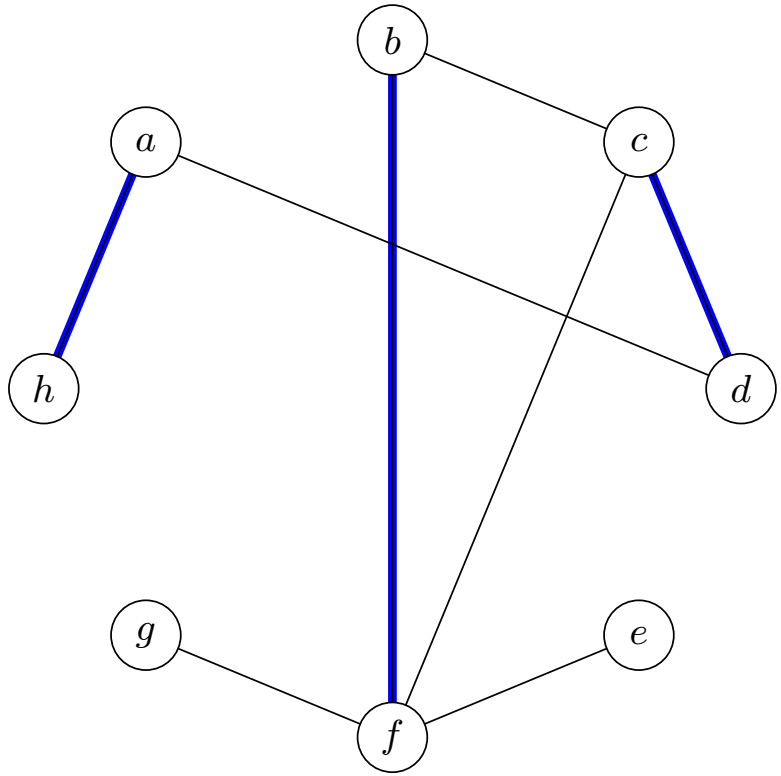




$y_u + y_v = c_{u,v}$ となる辺 $\{u, v\}$ 全体から成るグラフ H を作る

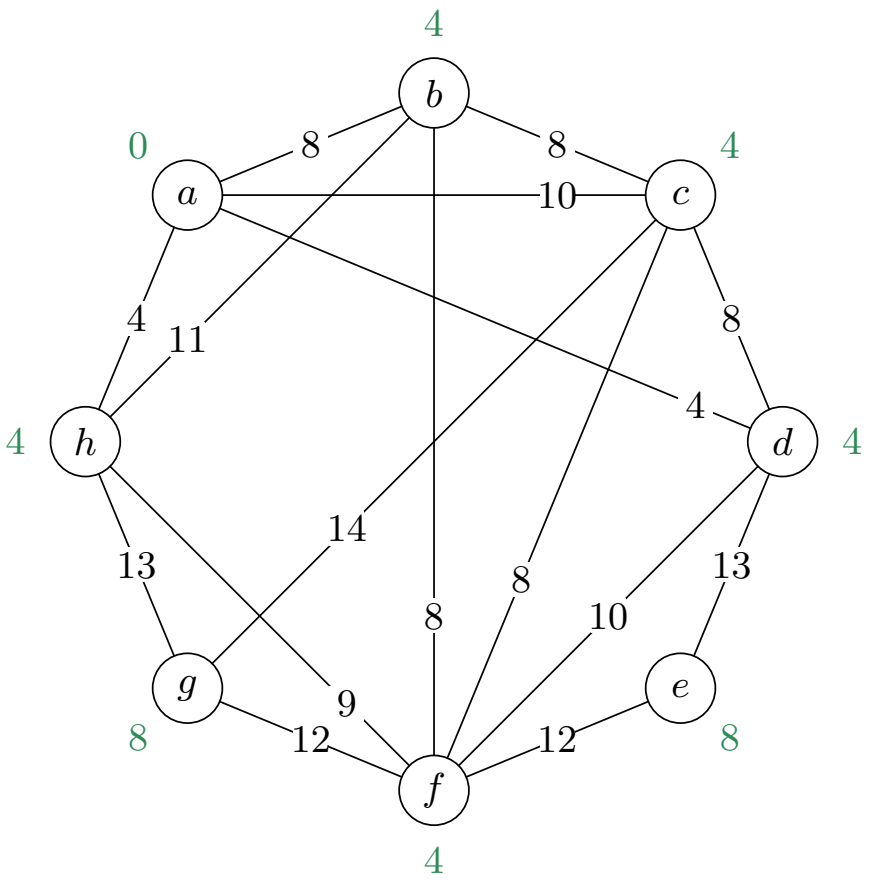
主双対法：例 (3)

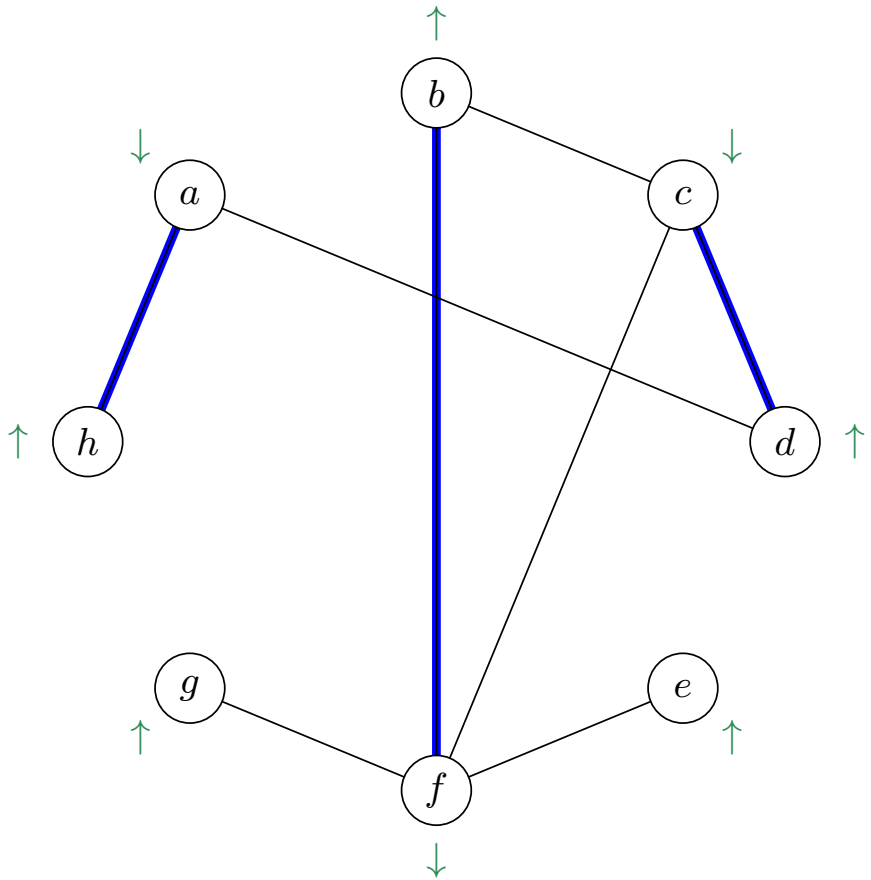




H の最大マッチングを見つける

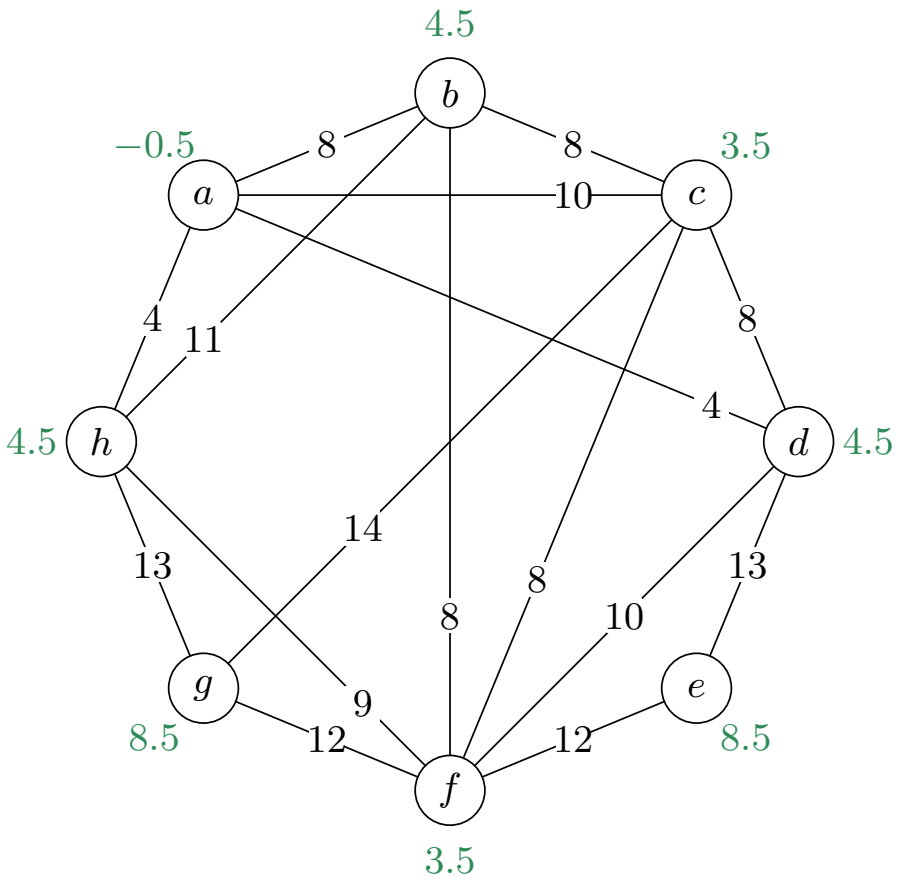
主双対法：例 (3)

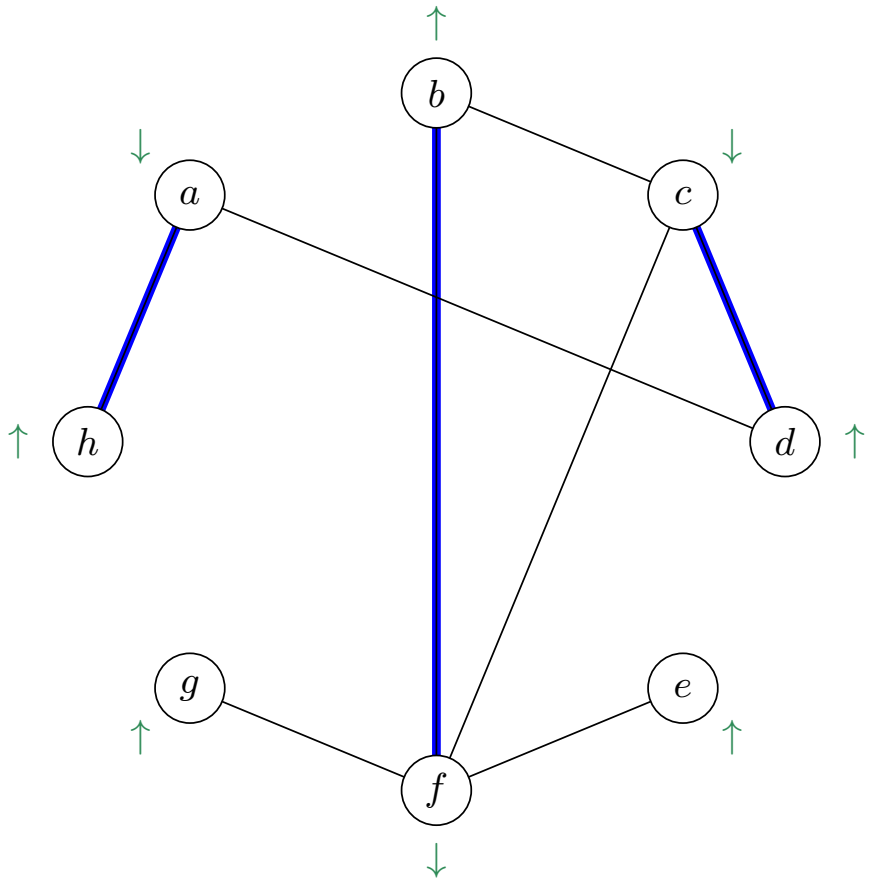




双対変数の更新を行う

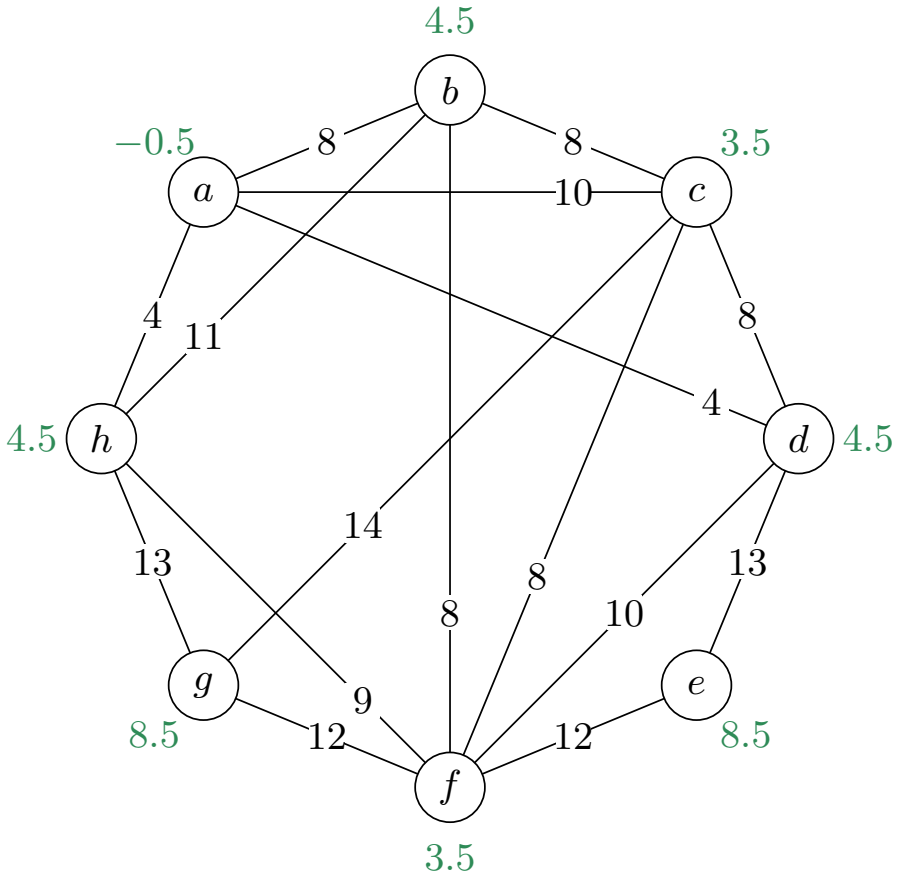
主双対法：例 (3)





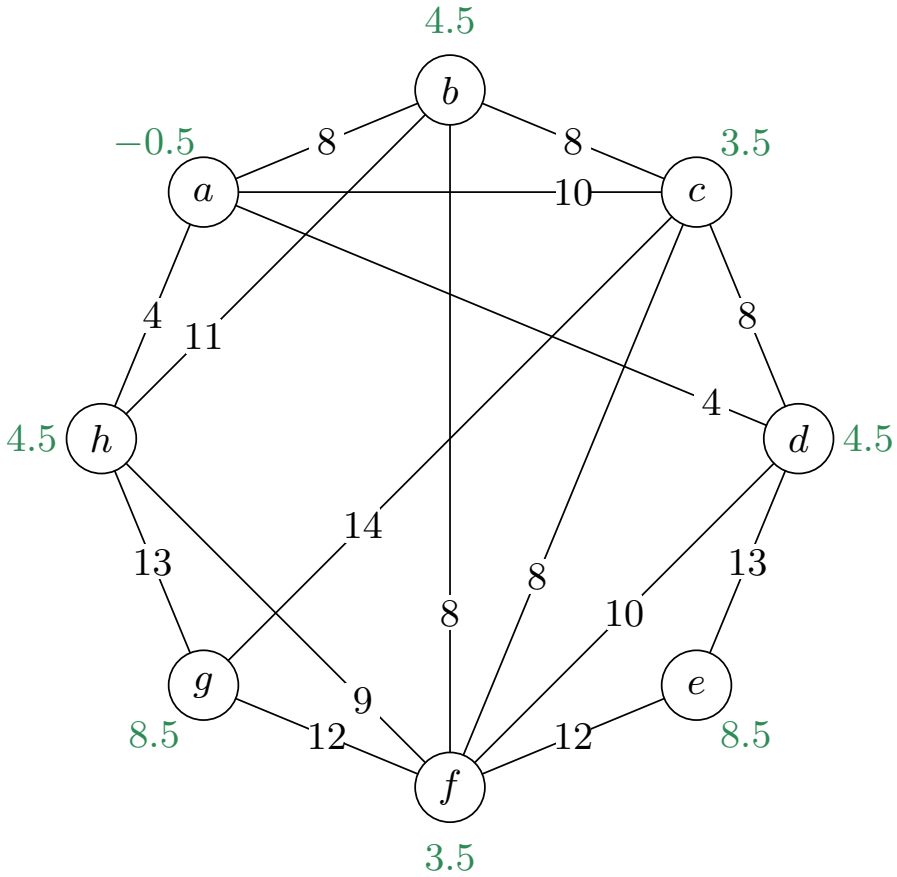
双対変数の更新を行う

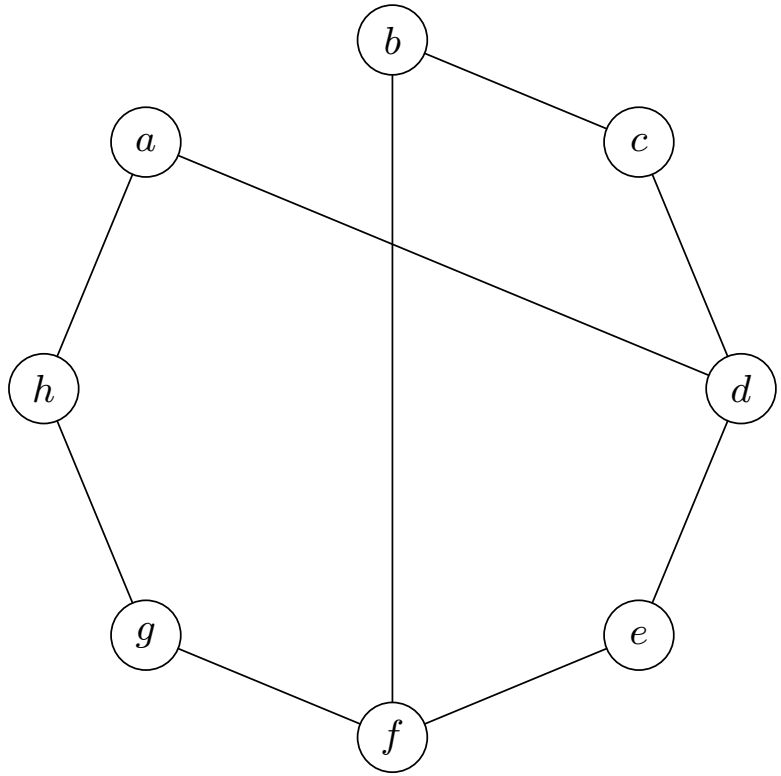
主双対法：例 (4)



このグラフを G として，各頂点 v に双対変数 y_v が与えられている

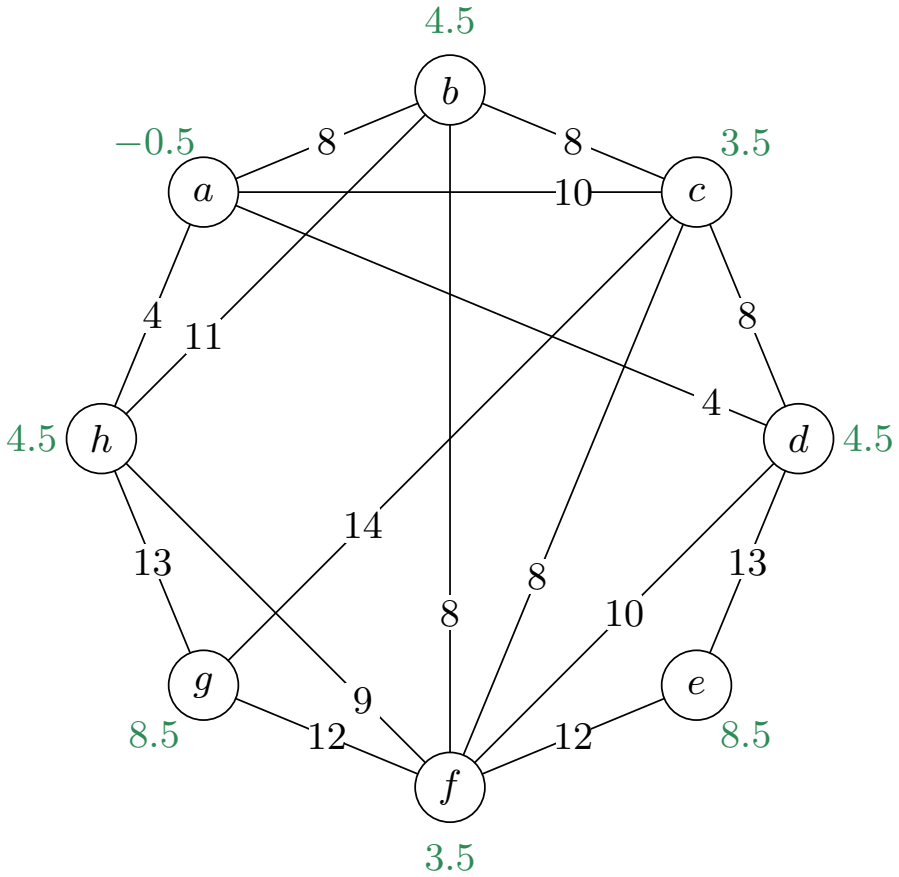
主双対法：例 (4)

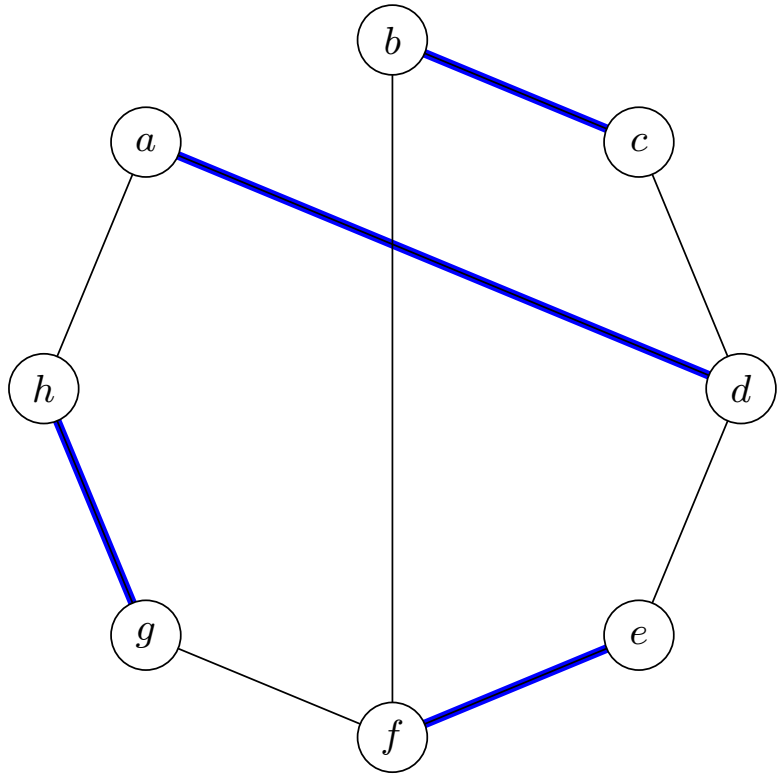




$y_u + y_v = c_{u,v}$ となる辺 $\{u, v\}$ 全体から成るグラフ H を作る

主双対法：例 (4)

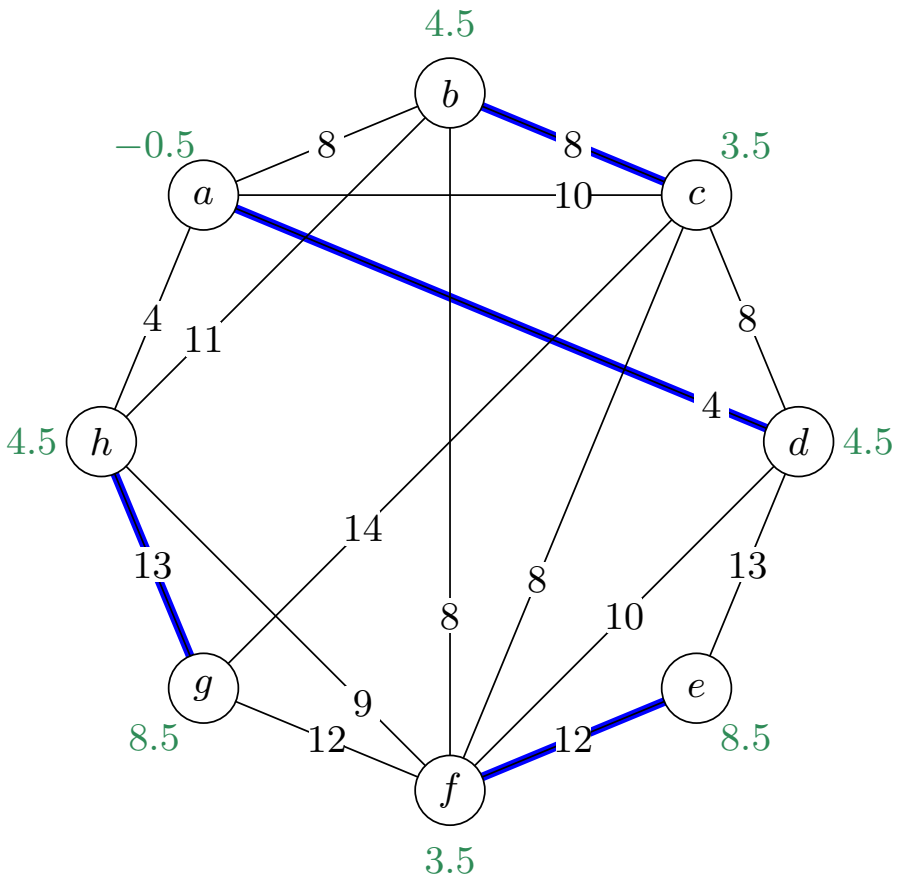




H の最大マッチングを見つける

$\rightsquigarrow$ 完全マッチングが見つかった (終了)

主双対法：例 (最終的に得られた 最小費用完全マッチング)



一般グラフの最小費用完全マッチング：主双対法 (1)

入力：グラフ $G = (V, E)$，辺費用 $c\colon E \to \mathbb{R}$
出力：G の最小費用完全マッチング $M \subseteq E$，(DLP') の最適解 y, z

主双対法

初期化

- $M := \emptyset$
- すべての $S \in \mathcal{C}$ に対して，$z_S := 0$
- すべての $v \in V$ に対して，$y_v := \dfrac{1}{2} \cdot \min\{c_{u,v} \mid \{u, v\} \in E\}$

H の構成と M の更新

1. グラフ $H = (V, E')$ を $E' = \{\{u, v\} \in E \mid y_u + y_v = c_{u,v}\}$ で構成
2. H の最大マッチングを計算して，M とする　(第 5 回講義参照)
   (同時に，M に関する花もいくつか得られる)
3. M が完全マッチング $\Rightarrow$ 終了

主双対法 (続き)

G の更新：M に関する花 C が得られているとき

1. $G := G/C$ とする
2. 新しい G において，c を次のように設定

$$c_{u,v} := \begin{cases} \min\{c_{u,w} - y_w \mid w \in C\} & (u \notin C,\ v \text{ は縮約で得られた頂点}) \\ c_{u,v} & (\text{それ以外}) \end{cases}$$

3. 新しい G に対する双対許容解 y を次のように構成

$$y_v := \begin{cases} y_v & (v \notin C) \\ 0 & (v \text{ が縮約で得られた頂点}) \end{cases}$$

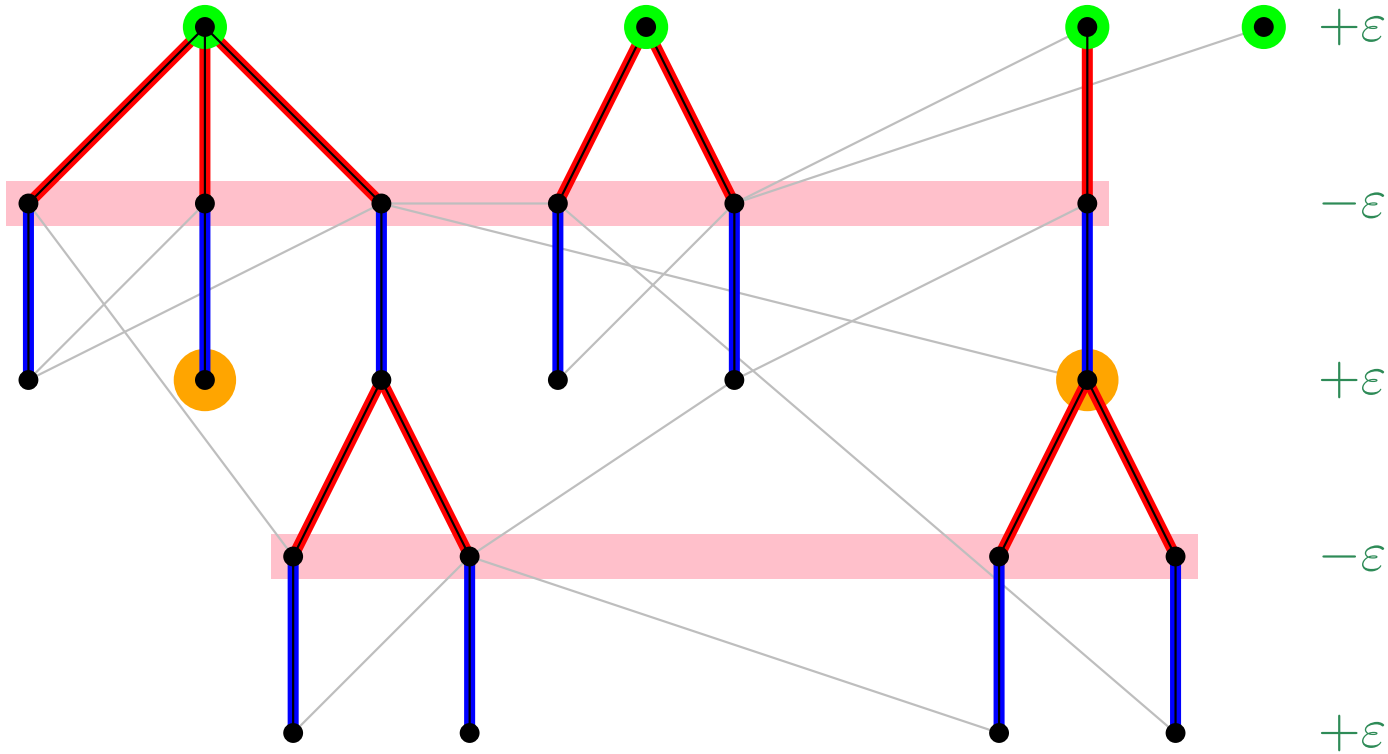
主双対法 (続き)

双対許容解の更新

1. y, z を更新 (方法は後述)
2. 縮約で得られた頂点 v に対して $y_v = 0$ であるとき
   - $G :=$ 頂点 v の縮約を解除して得られるグラフ
     (M の辺も縮約から復元，c も復元)
   - $z_C := 0$ (ただし，C を縮約して v が得られた)
3. 「H の構成と M の更新」に戻る

一般グラフの最小費用完全マッチング：主双対法 (4)

y の更新

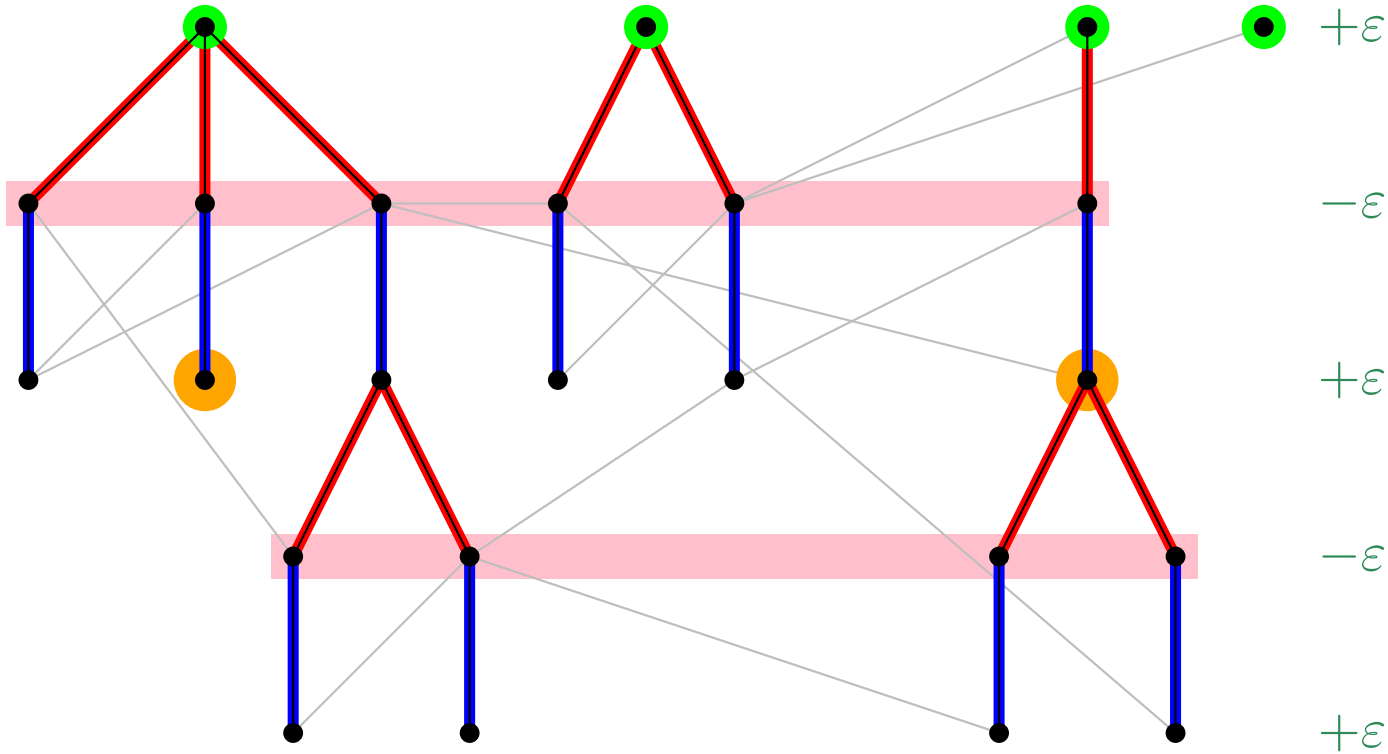


ε を定めるときの制約

- 任意の辺 $\{u, v\}$ に対して，$y_u + y_v \leq c_{u,v}$
- 縮約で得られた頂点 v に対して，$y_v \geq 0$

一般グラフの最小費用完全マッチング：主双対法 (5)

z の更新



この反復において，花 C を縮約して，頂点 v が作られたとき

- $z_C := y_v > 0$

性質：主双対法の正当性 (前回証明済)

主双対法が停止したとき，
得られる M は最小費用完全マッチングであり，
得られる y, z は (DLP') の最適解である

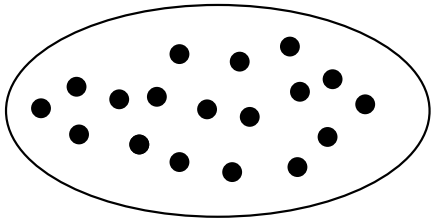
つまり，主双対法が停止すれば，主双対法の出力は正しい

目次

小目標

$z_S > 0$ を満たす $S \in \mathcal{C}$ の数が大きくないことを証明する

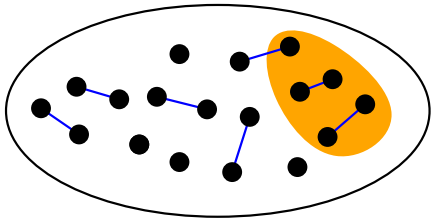
アルゴリズムの動き (イメージ)



小目標

$z_S > 0$ を満たす $S \in \mathcal{C}$ の数が大きくないことを証明する

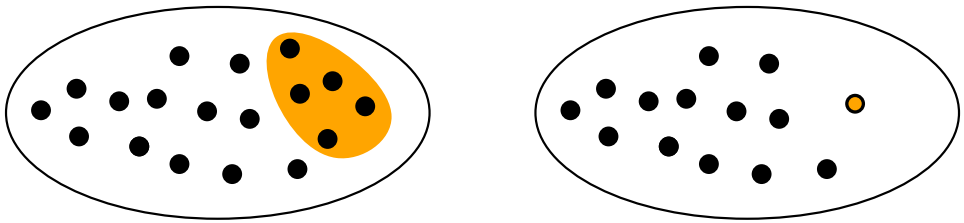
アルゴリズムの動き (イメージ)



小目標

$z_S > 0$ を満たす $S \in \mathcal{C}$ の数が大きくないことを証明する

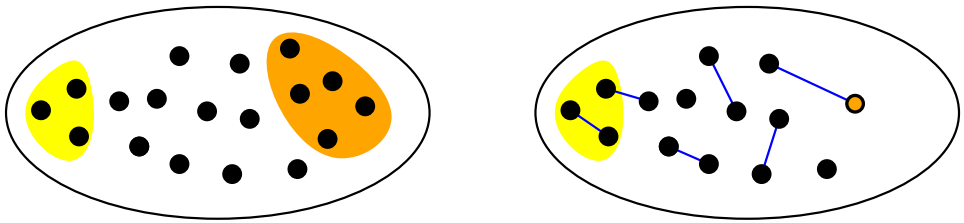
アルゴリズムの動き (イメージ)



小目標

$z_S > 0$ を満たす $S \in \mathcal{C}$ の数が大きくないことを証明する

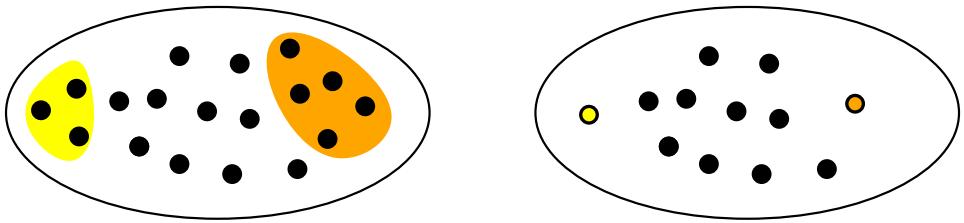
アルゴリズムの動き (イメージ)



小目標

$z_S > 0$ を満たす $S \in \mathcal{C}$ の数が大きくないことを証明する

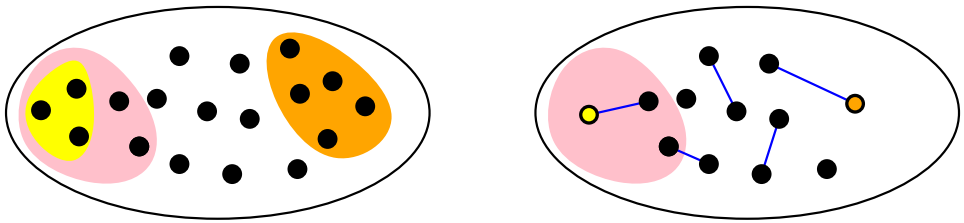
アルゴリズムの動き (イメージ)



小目標

$z_S > 0$ を満たす $S \in \mathcal{C}$ の数が大きくないことを証明する

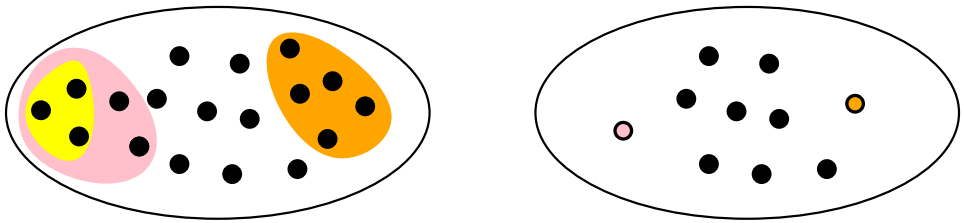
アルゴリズムの動き (イメージ)



小目標

$z_S > 0$ を満たす $S \in \mathcal{C}$ の数が大きくないことを証明する

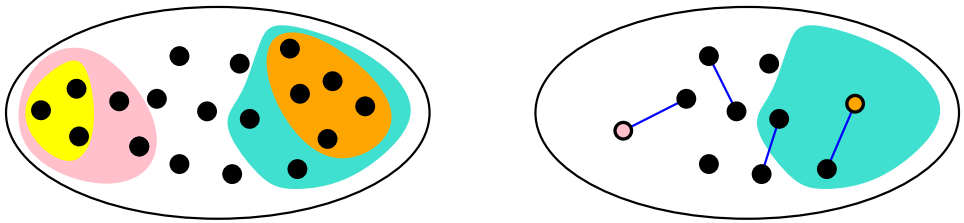
アルゴリズムの動き (イメージ)



小目標

$z_S > 0$ を満たす $S \in \mathcal{C}$ の数が大きくないことを証明する

アルゴリズムの動き (イメージ)



$z_S > 0$ を満たす S の数 (2)

アルゴリズム実行中の任意の時点において，次のように $\mathcal{C}_{>0}$ を定義

$$\mathcal{C}_{>0} = \{S \in \mathcal{C} \mid z_S > 0\}$$

(時点によって，$\mathcal{C}_{>0}$ は変化しうる)

性質：$\mathcal{C}_{>0}$ のラミナ性

アルゴリズム実行中の任意の時点において，$\mathcal{C}_{>0}$ は次を満たす

$$S, T \in \mathcal{C}_{>0} \text{ ならば，} \begin{cases} S \subseteq T & \text{であるか，} \\ S \supseteq T & \text{であるか，} \\ S \cap T = \emptyset & \text{である} \end{cases}$$

これは，アルゴリズムの動きから正しいと分かる

$\mathcal{C}_{>0}$ の満たす性質を持つ集合族はラミナ族と呼ばれる

定義：ラミナ族 (laminar family)

有限集合 V の部分集合族 $\mathcal{F} \subseteq 2^V$ がラミナ族であるとは，
次の条件を満たすこと

$$S, T \in \mathcal{F} \text{ ならば，} \begin{cases} S \subseteq T & \text{であるか，} \\ S \supseteq T & \text{であるか，} \\ S \cap T = \emptyset & \text{である} \end{cases}$$

例：$V = \{1, 2, 3, 4, 5, 6\}$, $\mathcal{F} = \{\{1\}, \{1, 2\}, \{3\}, \{3, 4\}, \{3, 4, 5\}, \{4\}\}$

1　　2　　3　　4　　5　　6

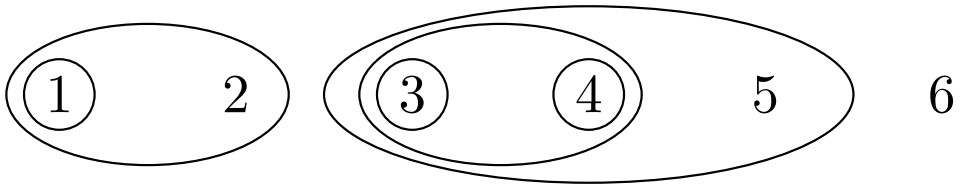
$\mathcal{C}_{>0}$ の満たす性質を持つ集合族はラミナ族と呼ばれる

定義：ラミナ族 (laminar family)

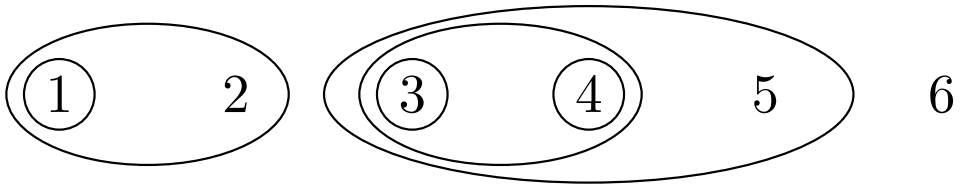
有限集合 V の部分集合族 $\mathcal{F} \subseteq 2^V$ がラミナ族であるとは，
次の条件を満たすこと

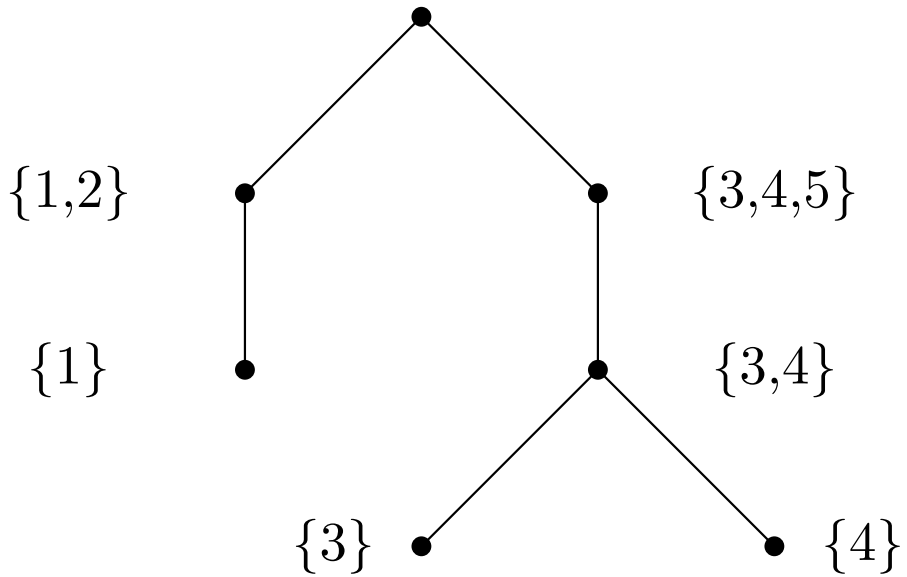
$$S, T \in \mathcal{F} \text{ ならば，} \begin{cases} S \subseteq T & \text{であるか，} \\ S \supseteq T & \text{であるか，} \\ S \cap T = \emptyset & \text{である} \end{cases}$$

例：$V = \{1, 2, 3, 4, 5, 6\}$, $\mathcal{F} = \{\{1\}, \{1, 2\}, \{3\}, \{3, 4\}, \{3, 4, 5\}, \{4\}\}$



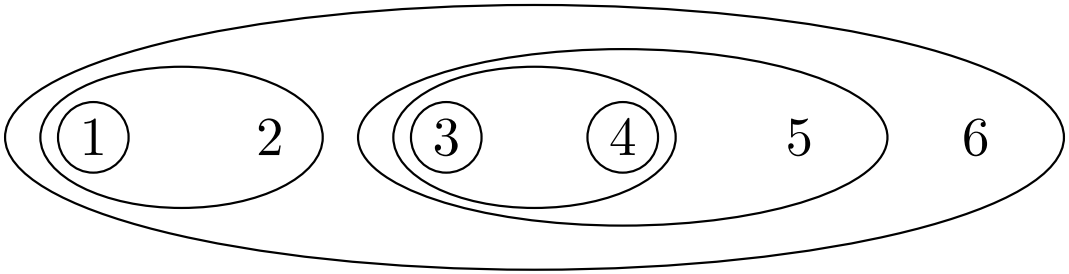
根付き木を用いたラミナ族の表現

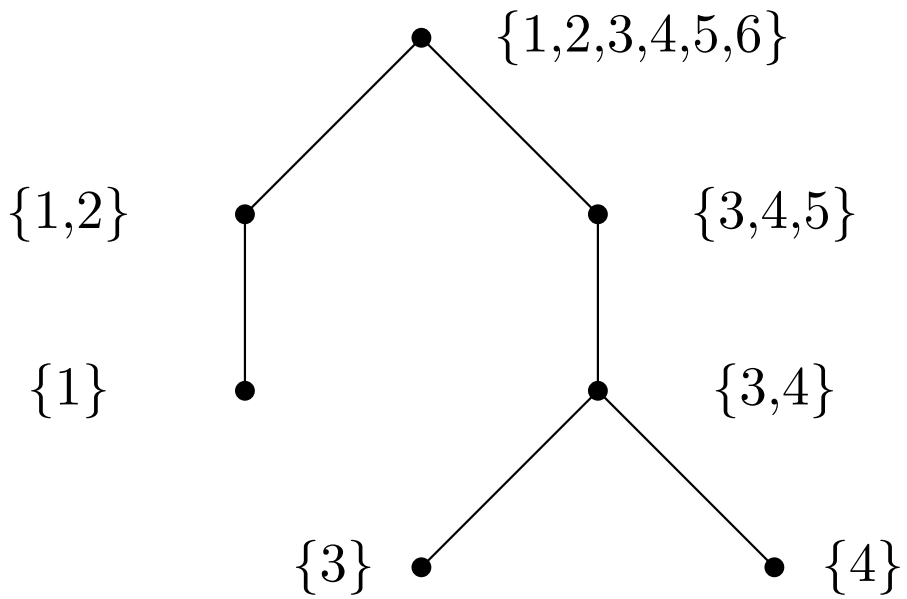




注：$\emptyset \in \mathcal{F}$ のとき，$\emptyset$ は図示されない

根付き木を用いたラミナ族の表現





注：$\emptyset \in \mathcal{F}$ のとき，$\emptyset$ は図示されない

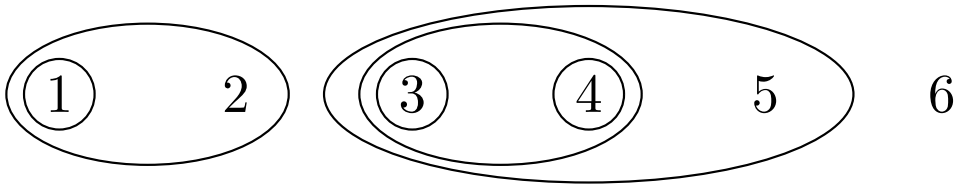
ラミナ族の要素数

有限集合 V，集合族 $\mathcal{F} \subseteq 2^V$

性質：ラミナ族の要素数の上界

$$\mathcal{F} \text{ がラミナ族} \quad \Rightarrow \quad |\mathcal{F} - \{\emptyset\}| \leq 2|V| - 1$$

例：$|V| = 6$，$|\mathcal{F} - \{\emptyset\}| = 6$，$2|V| - 1 = 11$



ラミナ族の要素数：証明 (1)

有限集合 V，集合族 $\mathcal{F} \subseteq 2^V$

性質：ラミナ族の要素数の上界

$$\mathcal{F} \text{ がラミナ族} \quad \Rightarrow \quad |\mathcal{F} - \{\emptyset\}| \leq 2|V| - 1$$

証明：$|V|$ に関する帰納法

- $|V| = 1$ のとき ($V = \{1\}$ とする)，
  ありうるラミナ族 $\mathcal{F}$ は以下の 4 つに限られる

$$\mathcal{F} = \emptyset, \quad \mathcal{F} = \{\emptyset\}, \quad \mathcal{F} = \{\emptyset, \{1\}\}, \quad \mathcal{F} = \{\{1\}\}$$

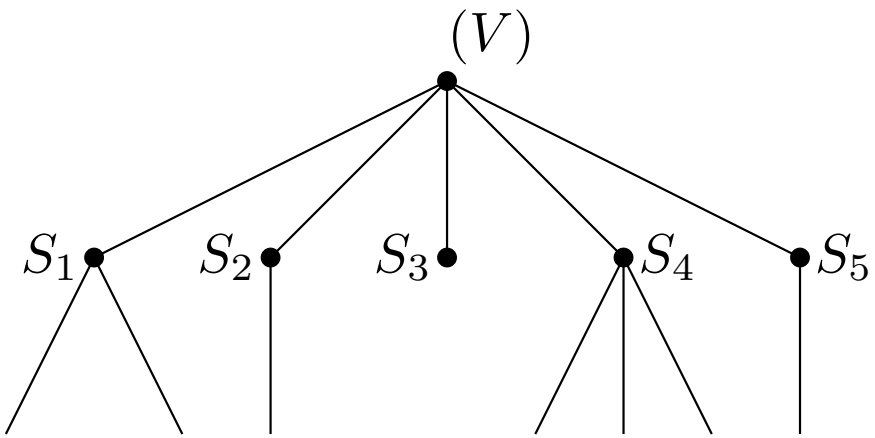
- どの場合であっても，$|\mathcal{F} - \{\emptyset\}| \leq 1 = 2|V| - 1$

ラミナ族の要素数：証明 (2)

$|V| = n \geq 2$ のとき ($|V| < n$ のときは正しいと仮定)

- ラミナ族 $\mathcal{F}$ を根付き木で表現したとき，
  根の子に対応する $\mathcal{F}$ の要素を $S_1, S_2, \ldots, S_k$ とする
- ラミナ族の定義と根付き木表現の構成法から，$S_i \cap S_j = \emptyset$ $(\forall\, i \neq j)$
- ここで，次の集合族を考える

$$\mathcal{F}_i = \{S \in \mathcal{F} \mid S \subseteq S_i\}$$

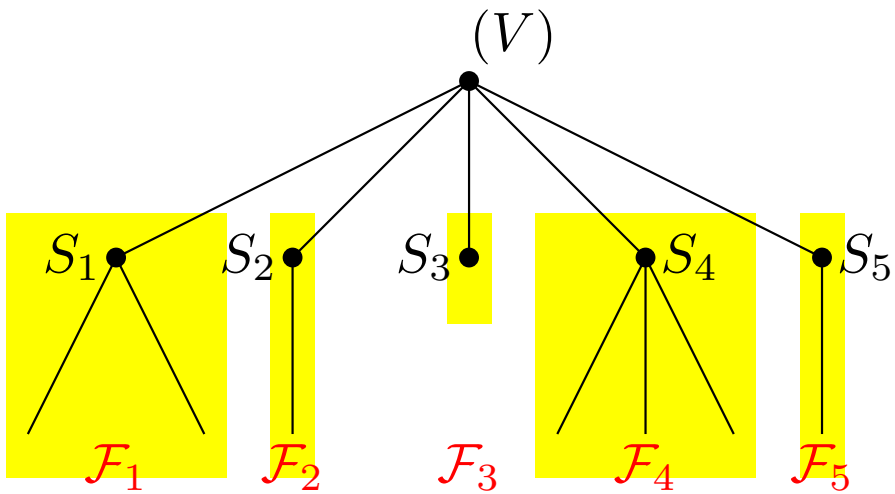


ラミナ族の要素数：証明 (2)

$|V| = n \geq 2$ のとき ($|V| < n$ のときは正しいと仮定)

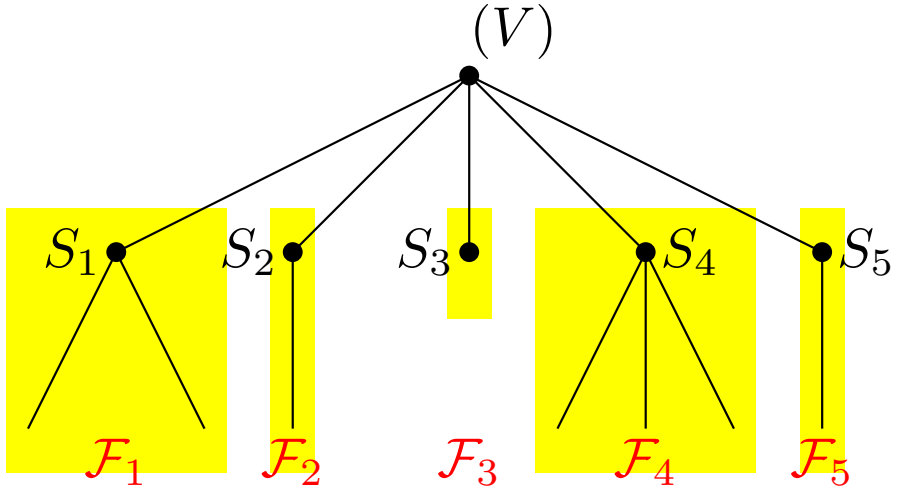
- ラミナ族 $\mathcal{F}$ を根付き木で表現したとき，
  根の子に対応する $\mathcal{F}$ の要素を $S_1, S_2, \ldots, S_k$ とする
- ラミナ族の定義と根付き木表現の構成法から，$S_i \cap S_j = \emptyset$ $(\forall\, i \neq j)$
- ここで，次の集合族を考える

$$\mathcal{F}_i = \{S \in \mathcal{F} \mid S \subseteq S_i\}$$



ラミナ族の要素数：証明 (3)

- このとき，各 $\mathcal{F}_i \subseteq 2^{S_i}$ はラミナ族で，
  各 $S \in \mathcal{F} - \{V, \emptyset\}$ は，$\mathcal{F}_1, \ldots, \mathcal{F}_k$ の中のちょうど 1 つに含まれる



- したがって，帰納法の仮定より，

$$|\mathcal{F} - \{\emptyset\}| \leq 1 + \sum_{i=1}^{k} |\mathcal{F}_i - \{\emptyset\}| \leq 1 + \sum_{i=1}^{k} (2|S_i| - 1)$$

- 場合分け $\rightarrow |V| = \sum_{i=1}^{k} |S_i|$ かどうか

ラミナ族の要素数：証明 (4)

場合 1：$|V| = \sum_{i=1}^{k} |S_i|$ のとき

- $k \geq 2$ となるので，

$$|\mathcal{F} - \{\emptyset\}| \leq 1 + \sum_{i=1}^{k} (2|S_i| - 1) \leq 1 + 2|V| - k \leq 2|V| - 1$$

場合 2：$|V| > \sum_{i=1}^{k} |S_i|$ のとき

- $\sum_{i=1}^{k} |S_i| \leq |V| - 1$ となり，$k \geq 1$ なので

$$|\mathcal{F} - \{\emptyset\}| \leq 1 + \sum_{i=1}^{k} (2|S_i| - 1) \leq 1 + (2|V| - 1) - k \leq 2|V| - 1$$

これで証明が終わる □

ラミナ族の要素数：帰結

アルゴリズム実行中の任意の時点において，次のように $\mathcal{C}_{>0}$ を定義

$$\mathcal{C}_{>0} = \{S \in \mathcal{C} \mid z_S > 0\}$$

(時点によって，$\mathcal{C}_{>0}$ は変化しうる)

まとめ

アルゴリズム実行中の任意の時点において

- $\mathcal{C}_{>0}$ はラミナ族
- $\therefore |\mathcal{C}_{>0}| \leq 2|V|$

つまり，$z_S > 0$ である $S \in \mathcal{C}$ の数は入力サイズの多項式以下

目次

主双対法の計算量 (1)

主双対法では

- 「一般グラフの最大マッチング問題」を解くアルゴリズムを繰り返し，実行する
- 双対許容解の更新を繰り返し，実行する
- 花の縮約や，縮約の解除を繰り返し，実行する

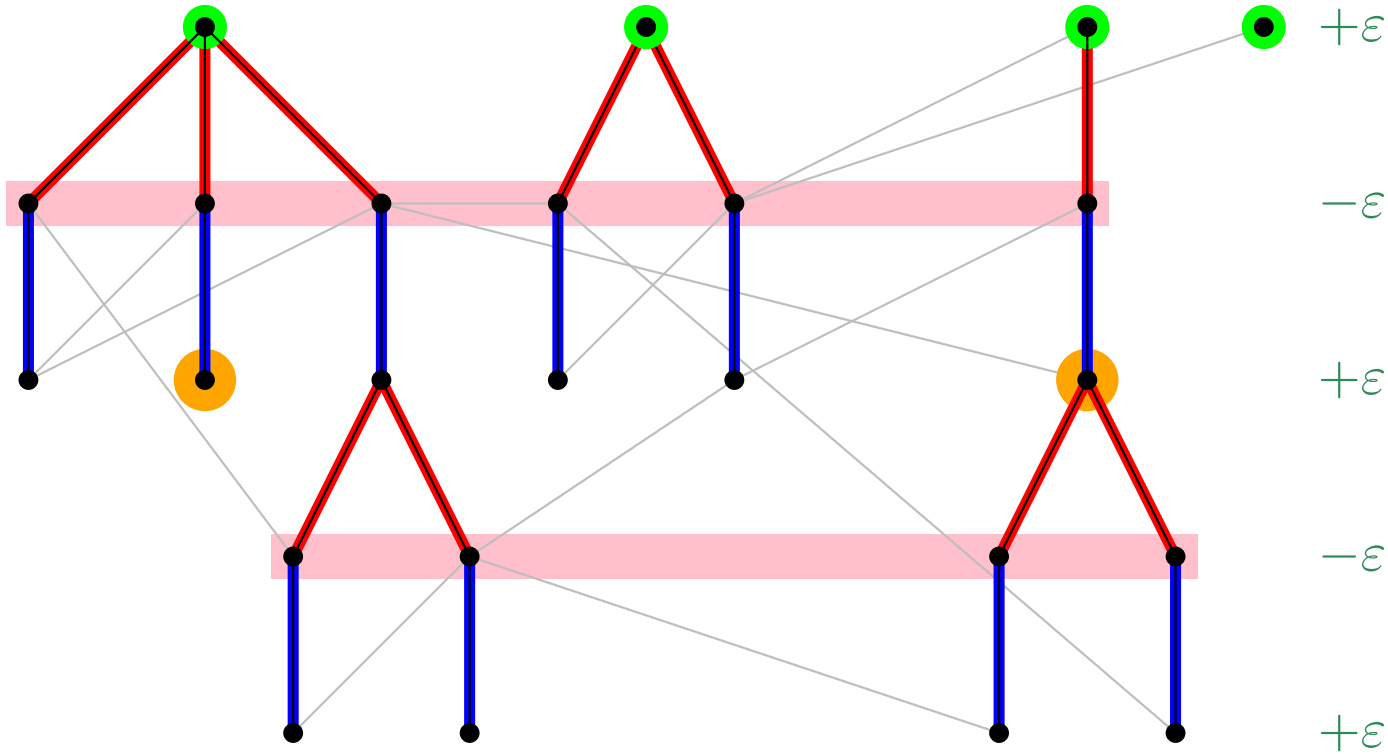
これらが何回行われるか，考える

補足 (第 5 回講義の内容，Edmonds '65)

一般グラフの最大マッチング問題は多項式時間で解ける

主双対法の計算量 (2)

y の更新

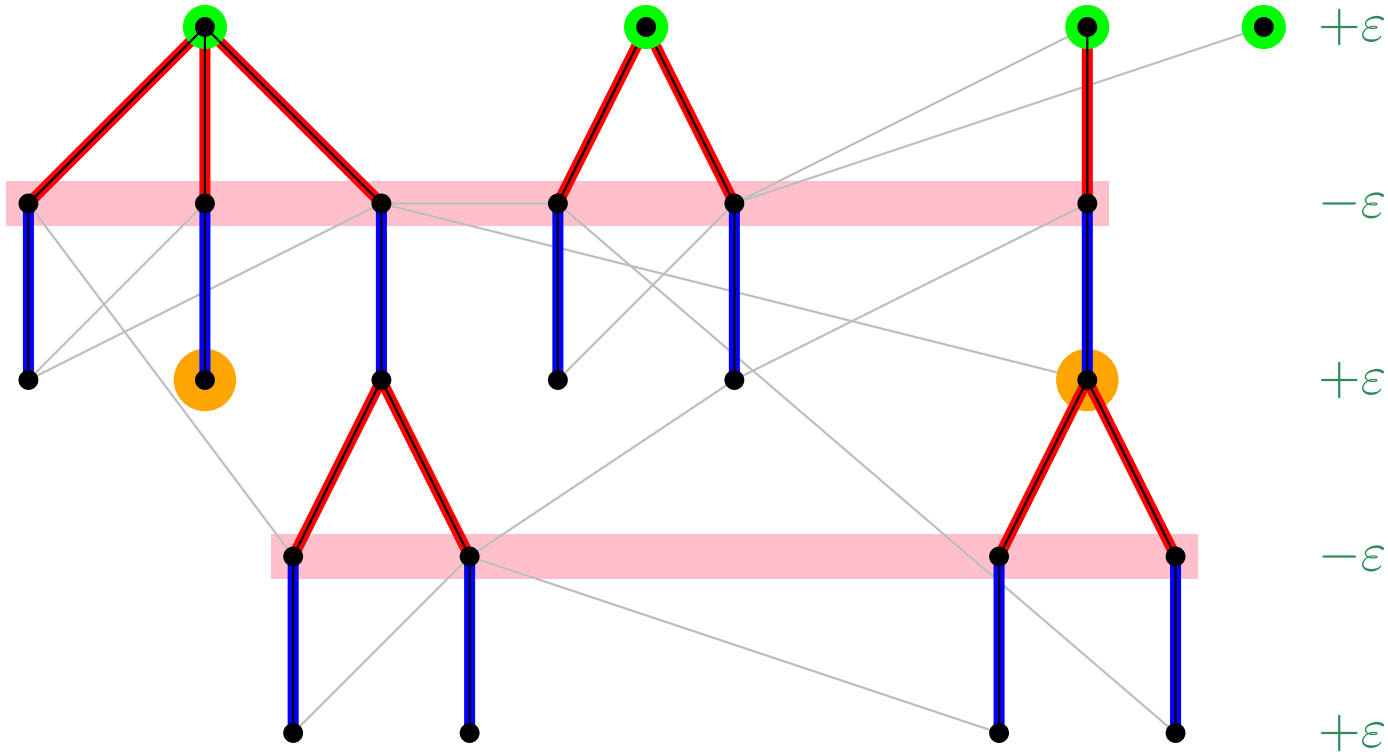


ε を定めるときの制約

- 任意の辺 $\{u, v\}$ に対して，$y_u + y_v \leq c_{u,v}$
- 縮約で得られた頂点 v に対して，$y_v \geq 0$

主双対法の計算量 (2)

y の更新

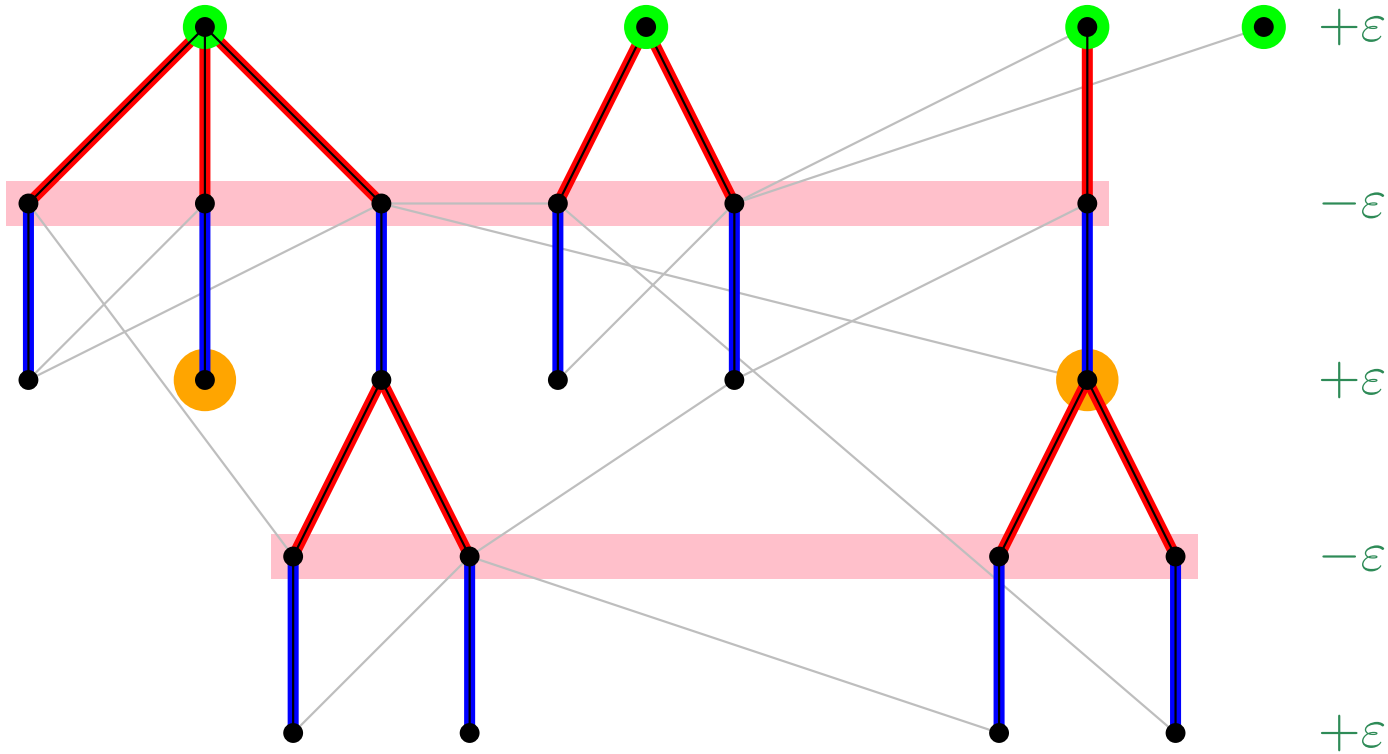


$y_u + y_v = c_{u,v}$ を満たす辺 $\{u, v\}$ ができた $\Rightarrow$

- 次の反復で，最大マッチングの要素数が 1 以上増加する

主双対法の計算量 (3)

y の更新



縮約で得られた頂点 v に対して，$y_v = 0$ となった $\Rightarrow$

- v の縮約を解除する
- しかし，高々$2|V|$ 回解除すると，縮約で得られた頂点がなくなる

主双対法の計算量 (4)

つまり，

- 高々$2|V|$ 回の縮約の解除を行うと，
  必ず最大マッチングの要素数が 1 以上増加する

したがって，

- 完全マッチングが見つかるまでの反復回数 $\leq 2|V| \cdot |V|/2 = |V|^2$

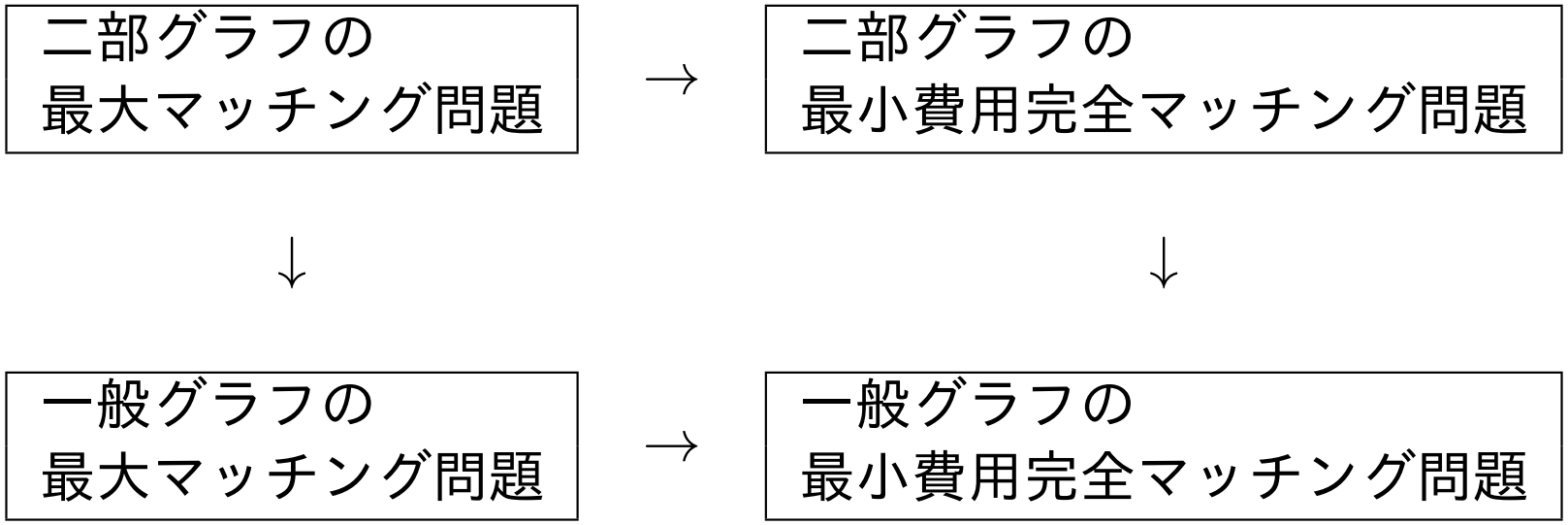
つまり，主双対法は多項式時間アルゴリズムとなる

定理：主双対法 (Edmonds '65)

一般グラフの最小費用完全マッチング問題に対する主双対法は
多項式時間で必ず最小費用完全マッチングを発見する

目次

この講義が対象とする 4 つの問題



この講義で行うこと

「これら 4 つの問題は多項式時間で解けること」の説明

重要な考え方

- 最適化における最大最小定理 (強双対性)
- 費用無し問題のアルゴリズム + 線形計画法 $\rightsquigarrow$
  費用有り問題のアルゴリズム (主双対法)

ここまでは「1960 年代」のはなし

この後，組合せ最適化の理論研究はどのように進んでいったか

- 組合せ最適化において「連続最適化の技法」を用いることは当たり前になった
  - ⇝ 多面体的組合せ論 (polyhedral combinatorics)
  - ⇝ 凸体の幾何学 (convex geometry)
  - ⇝ 拡張複雑性 (extension complexity)

- 主双対法が「近似アルゴリズムの設計」や「オンラインアルゴリズムの設計」にも適用されるようになり，これも当たり前になった

- 「花の縮約」というアイディアも当たり前となり，同様のアイディアが様々な問題の解決に使われるようになった