離散最適化基礎論

第 13 回

最小費用流問題：容量スケーリング法

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2024 年 1 月 16 日

最終更新：2024 年 1 月 20 日　09:44

スケジュール (前半)

1. 最大流と最小費用流：定義 (10/3)
2. 最大流問題：増加道法 (10/10)

＊ 休み (10/17)

3. 線形計画法の復習 (10/24)
4. 最大流問題：線形計画問題として (10/31)
5. 最大流問題：Edmonds-Karp のアルゴリズム (11/7)
6. 最大流問題：容量スケーリング法 (11/14)
7. 最大流問題：Push-Relabel 法 (概要) (11/21)
8. 最大流問題：Push-Relabel 法 (計算量評価) (11/28)

スケジュール (後半)

＊	休み	(12/5)
9.	最小費用流問題：線形計画問題として	(12/12)
10.	最小費用流問題：負閉路消去法	(12/19)
11.	最小費用流問題：正カット消去法	(12/26)
＊	休み	(1/2)
12.	最小費用流問題：逐次最短路法	(1/9)
13.	最小費用流問題：容量スケーリング法	(1/16)
14.	最小費用流問題：費用スケーリング法	(1/23)
＊	休み	(1/30)

前回までのまとめ と 今回の内容

最小費用流問題

最適性条件 ⟶ アルゴリズム

- 簡約費用最適性条件
- 負閉路最適性条件
- 正カット最適性条件

- 逐次最短路法
- 負閉路消去法
- 正カット消去法

今回の内容

- 逐次最短路法の高速化 ⇝ 容量スケーリング法
- 主双対法

[復習] 逐次最短路法：基本的な考え方

(G, u, c)　b ⟶ f　p

b-流　簡約費用最適性条件の
不等式を満たすポテンシャル

[復習] 逐次最短路法：基本的な考え方

(G, u, c)　$b^0 = 0$ ⟶ f^0　p^0

b^0-流　簡約費用最適性条件の
不等式を満たすポテンシャル

(G, u, c)　b ⟶ f　p

b-流　簡約費用最適性条件の
不等式を満たすポテンシャル

[復習] 逐次最短路法：基本的な考え方

(G, u, c)　$b^0 = 0$ ⟶ f^0　p^0

b^0-流　簡約費用最適性条件の
不等式を満たすポテンシャル

(G, u, c)　b^1 ⟶ f^1　p^1

b^1-流　簡約費用最適性条件の
不等式を満たすポテンシャル

(G, u, c)　b ⟶ f　p

b-流　簡約費用最適性条件の
不等式を満たすポテンシャル

[復習] 逐次最短路法：基本的な考え方

(G, u, c) $b^0 = 0$ → f^0 p^0

b^0-流　簡約費用最適性条件の不等式を満たすポテンシャル

(G, u, c) b^1 → f^1 p^1

b^1-流　簡約費用最適性条件の不等式を満たすポテンシャル

(G, u, c) b^2 → f^2 p^2

b^2-流　簡約費用最適性条件の不等式を満たすポテンシャル

(G, u, c) b → f p

b-流　簡約費用最適性条件の不等式を満たすポテンシャル

[復習] 逐次最短路法：基本的な考え方

(G, u, c) $b^0 = 0$ → f^0 p^0

b^0-流　簡約費用最適性条件の不等式を満たすポテンシャル

(G, u, c) b^1 → f^1 p^1

b^1-流　簡約費用最適性条件の不等式を満たすポテンシャル

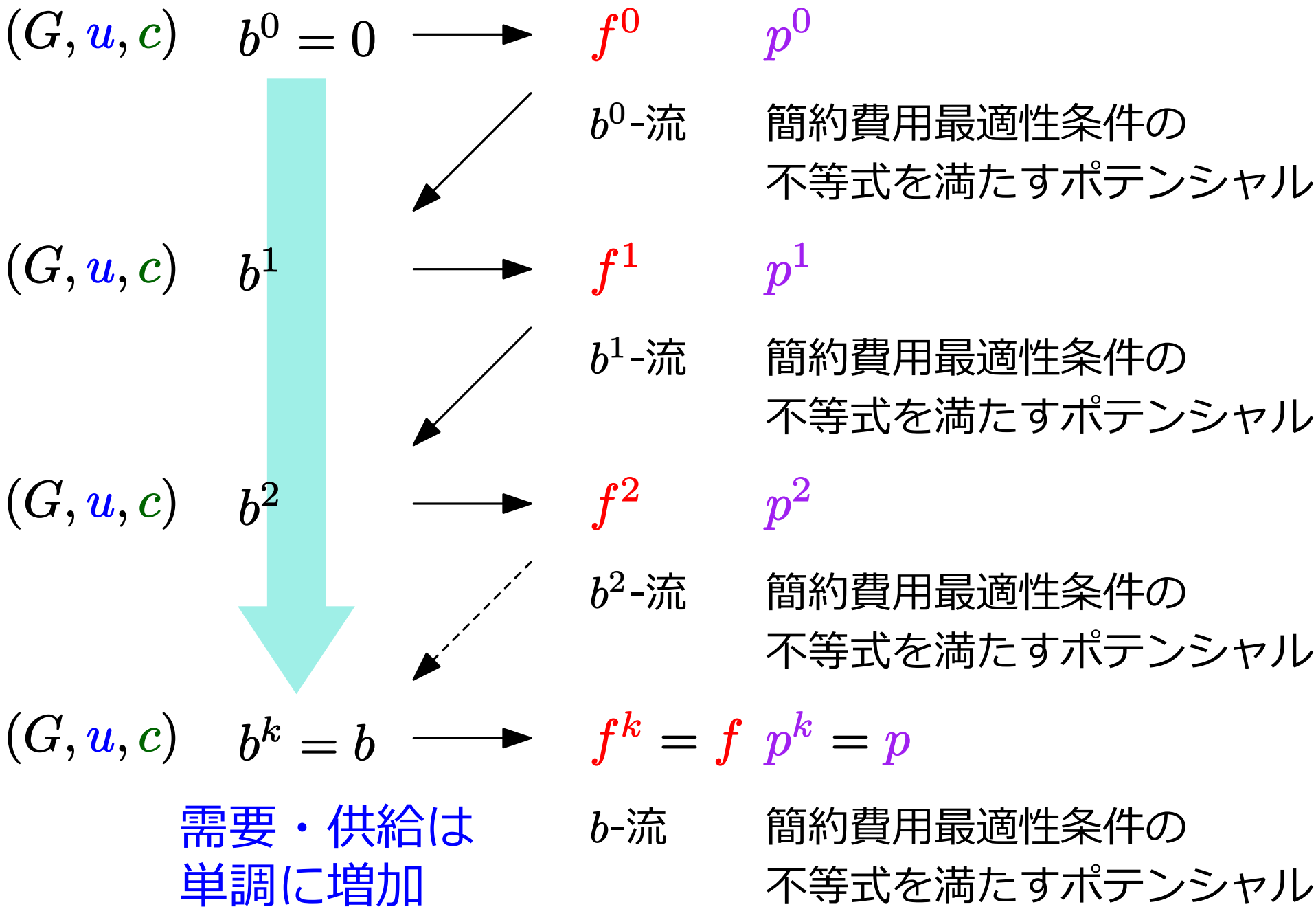
(G, u, c) b^2 → f^2 p^2

b^2-流　簡約費用最適性条件の不等式を満たすポテンシャル

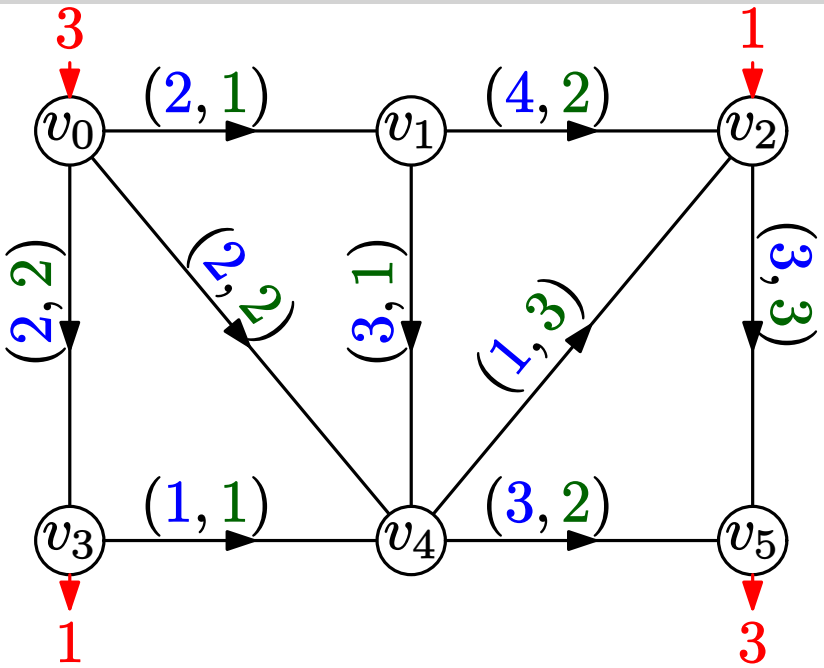
(G, u, c) $b^k = b$ → $f^k = f$ $p^k = p$

b-流　簡約費用最適性条件の不等式を満たすポテンシャル

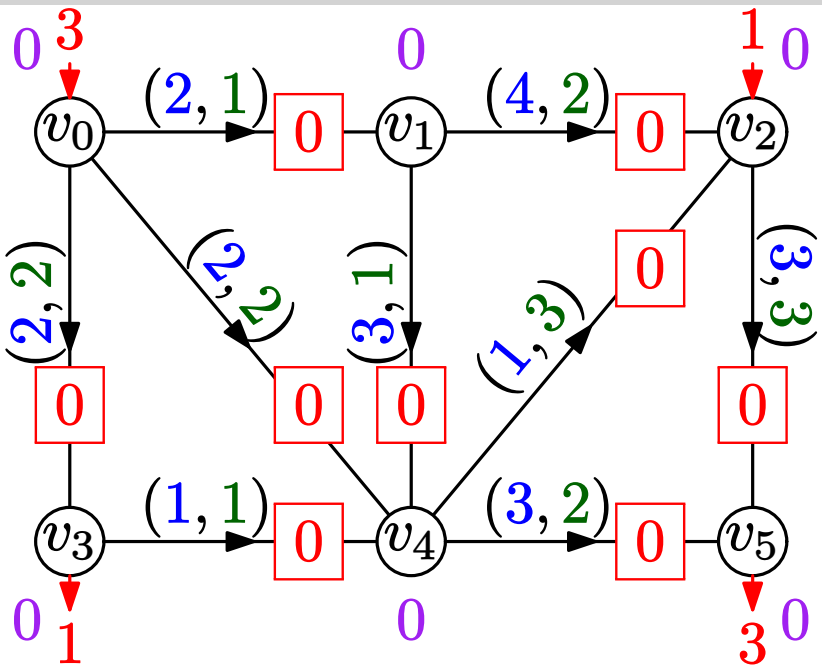
[復習] 逐次最短路法：基本的な考え方



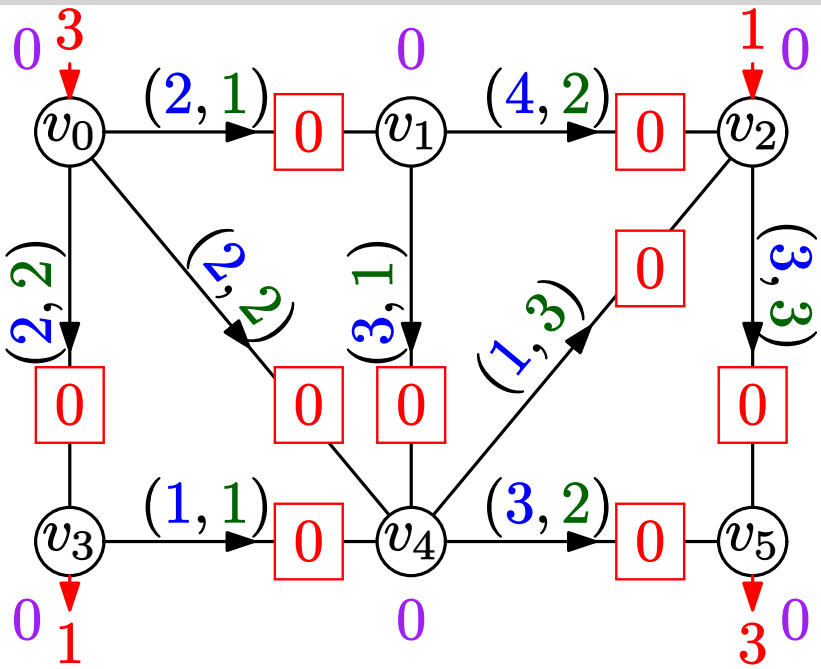
[復習] 逐次最短路法：例 (1/4)

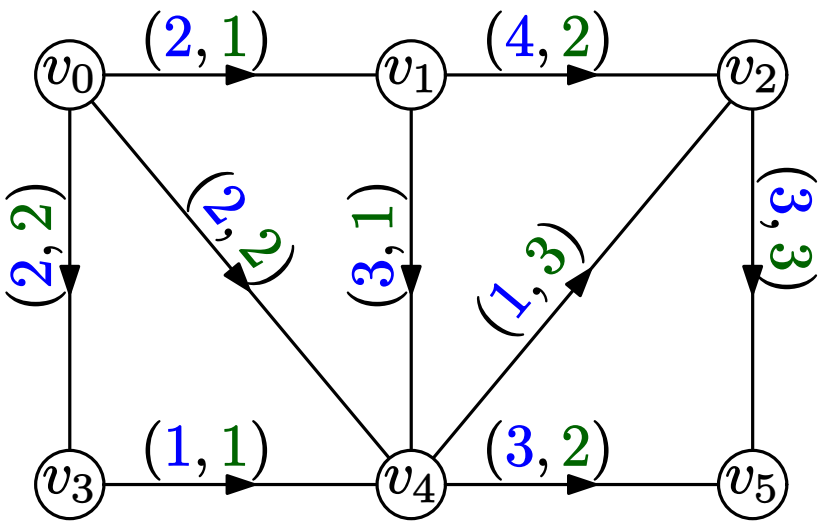


[復習] 逐次最短路法：例 (1/4)

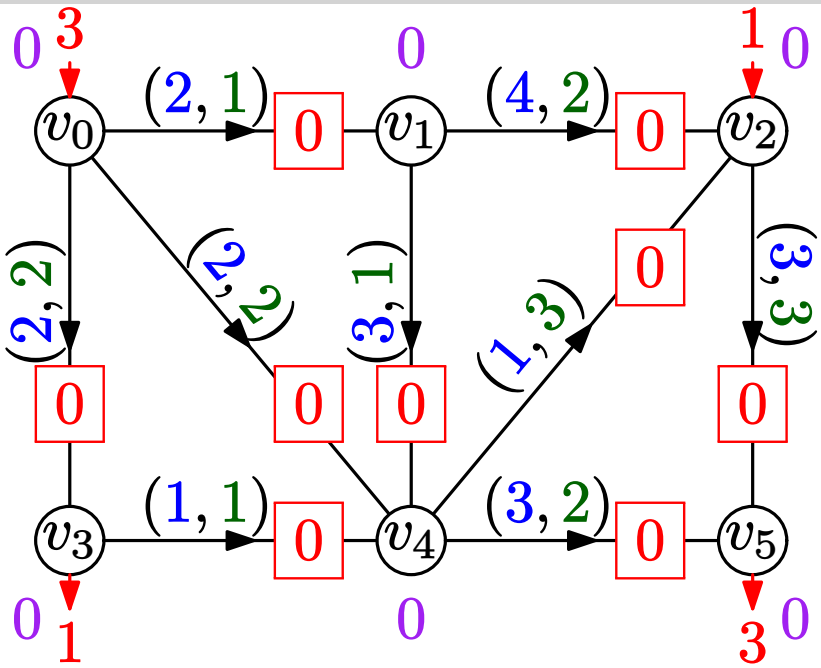


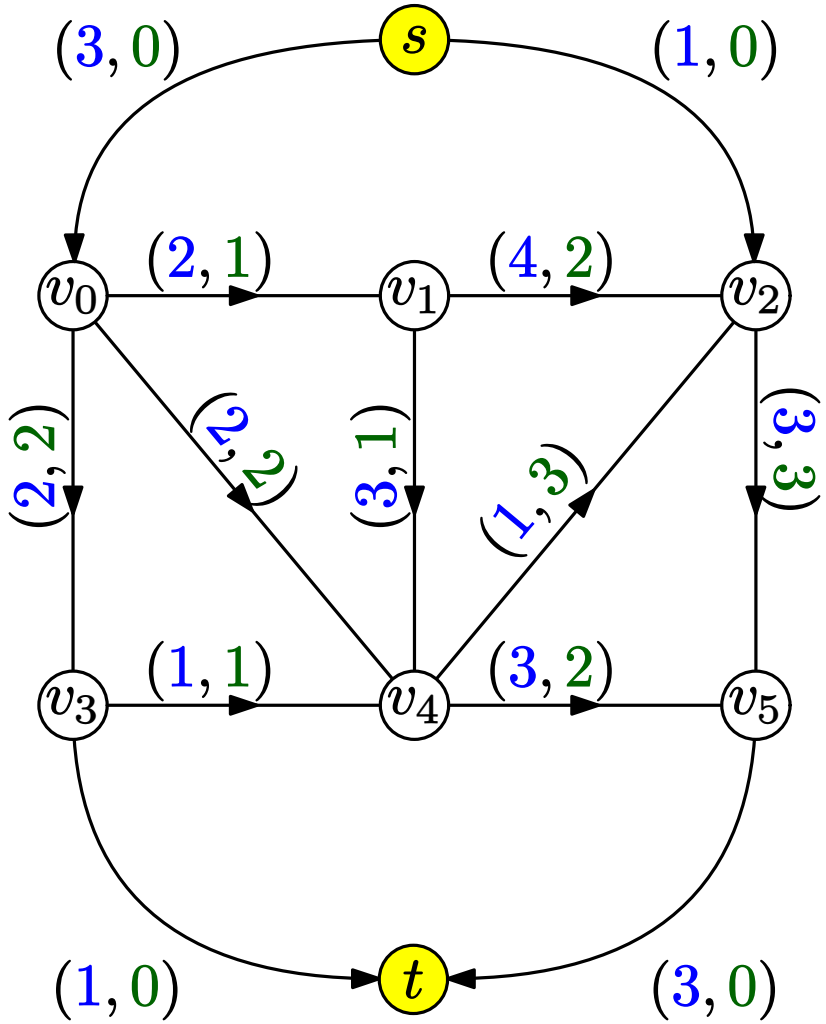
[復習] 逐次最短路法：例 (1/4)

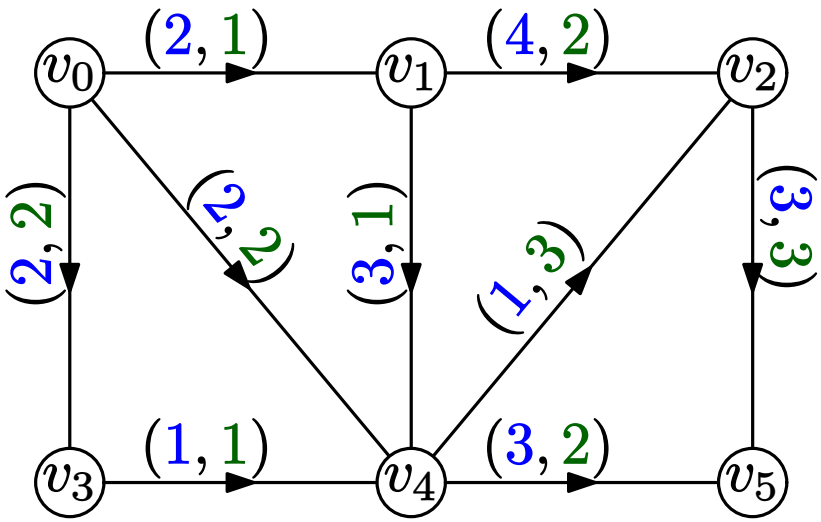




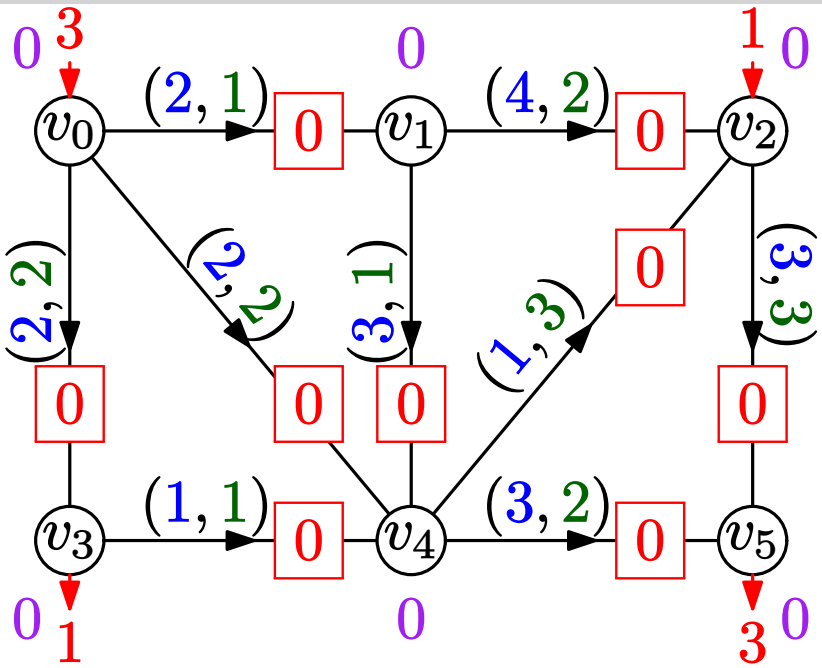
[復習] 逐次最短路法：例 (1/4)

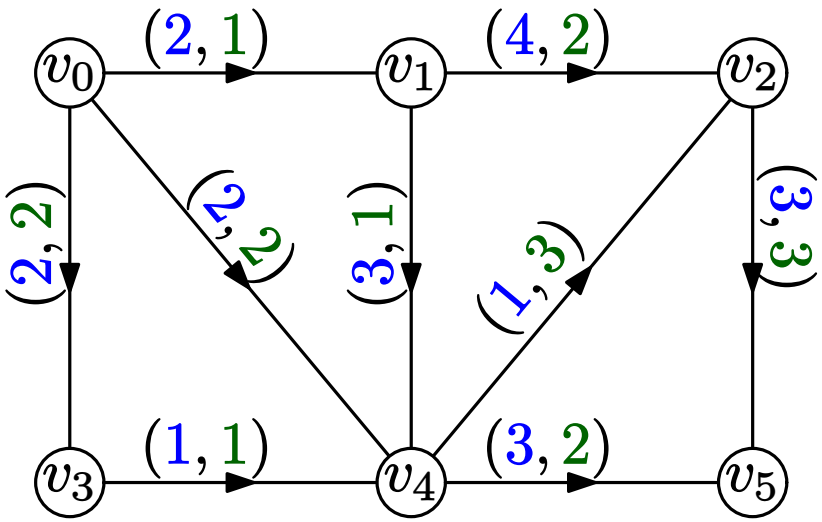


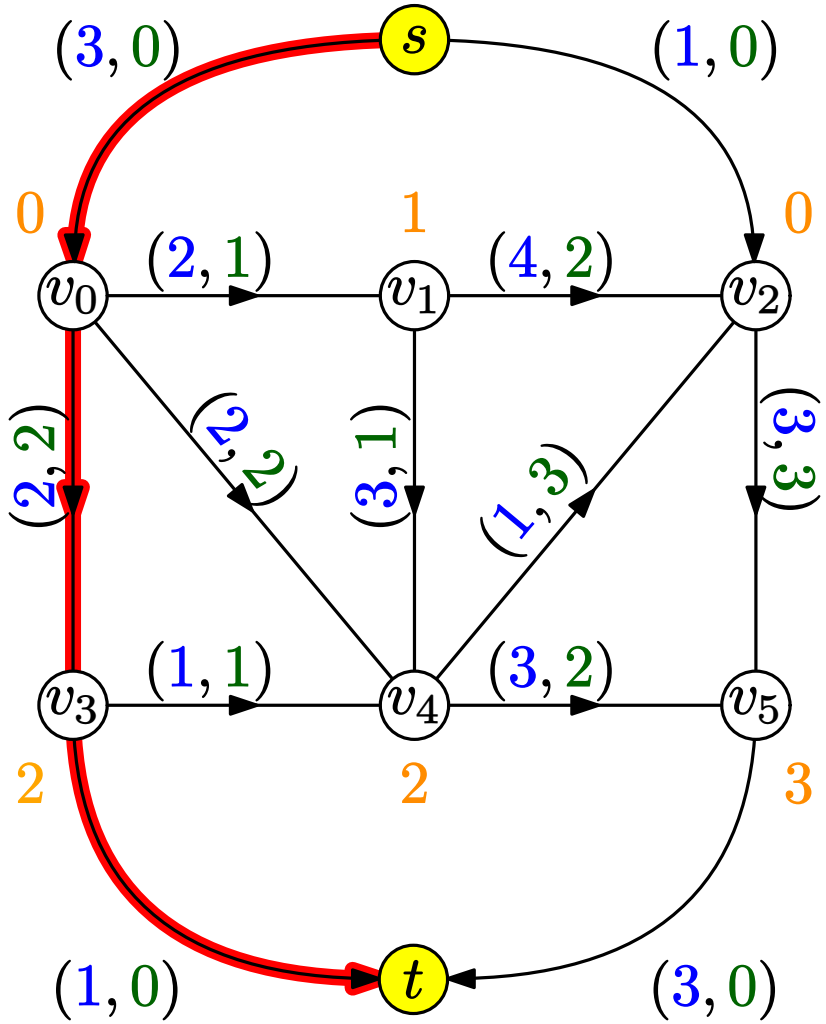




[復習] 逐次最短路法：例 (1/4)

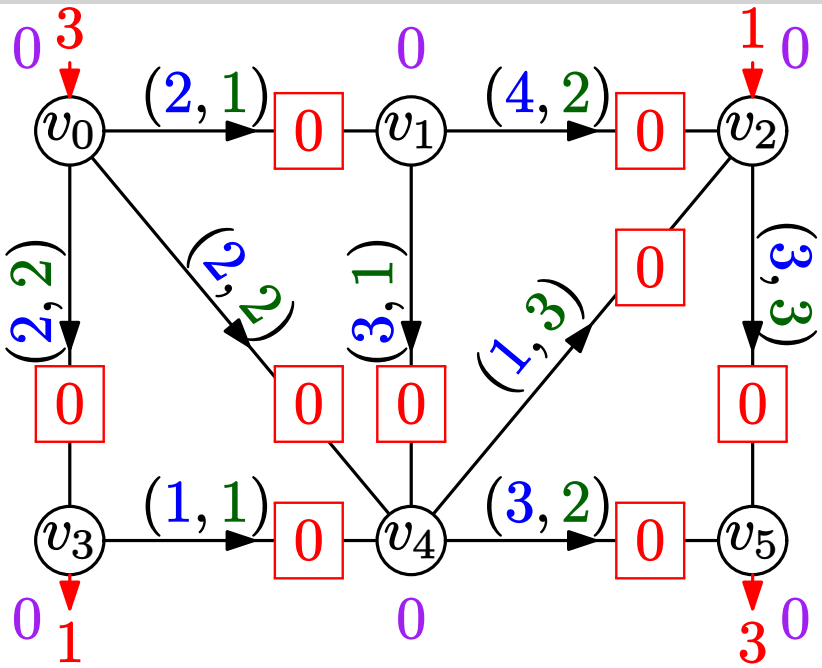


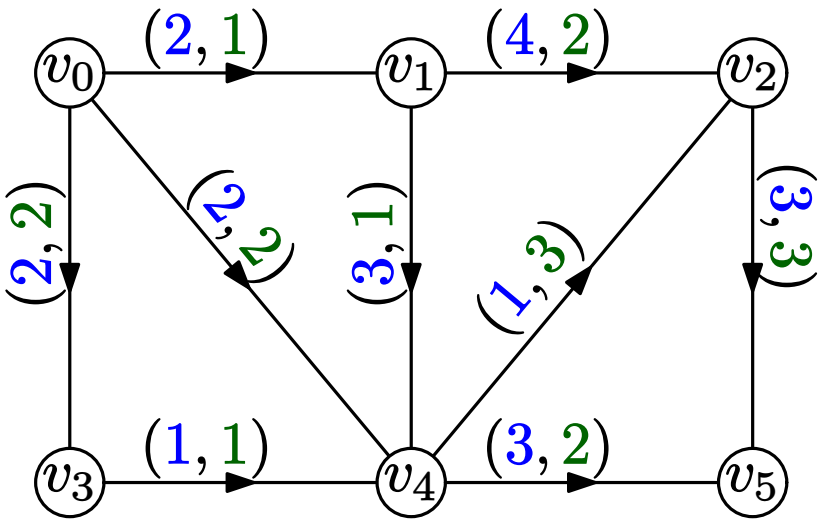


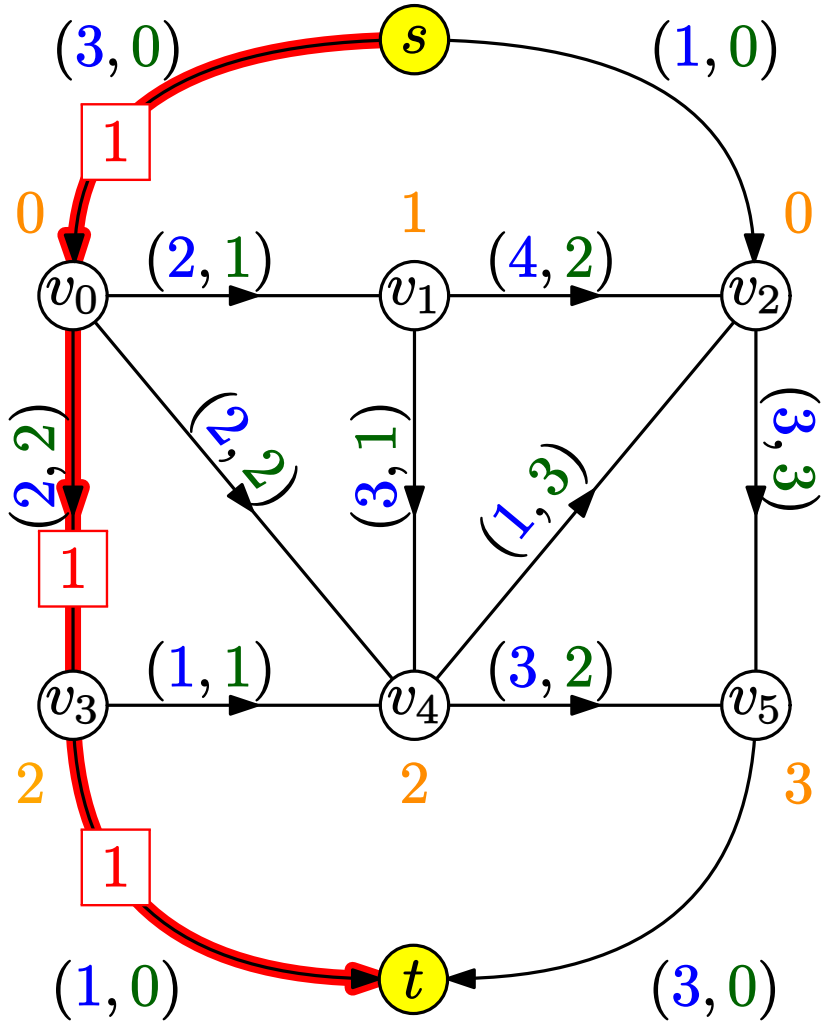


簡約費用を距離とする最短 s-t 路

[復習] 逐次最短路法：例 (1/4)



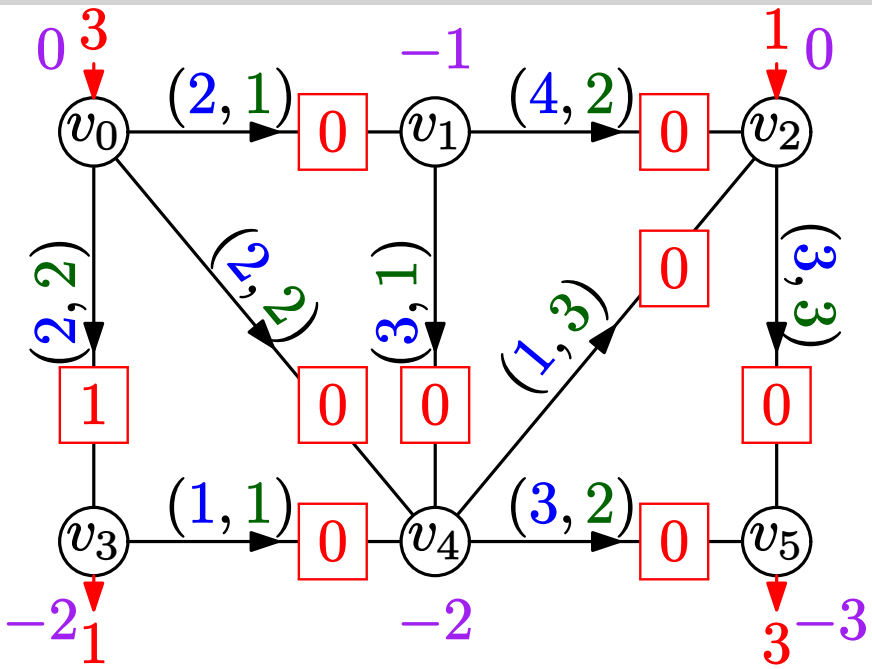




簡約費用を距離とする最短 s-t 路

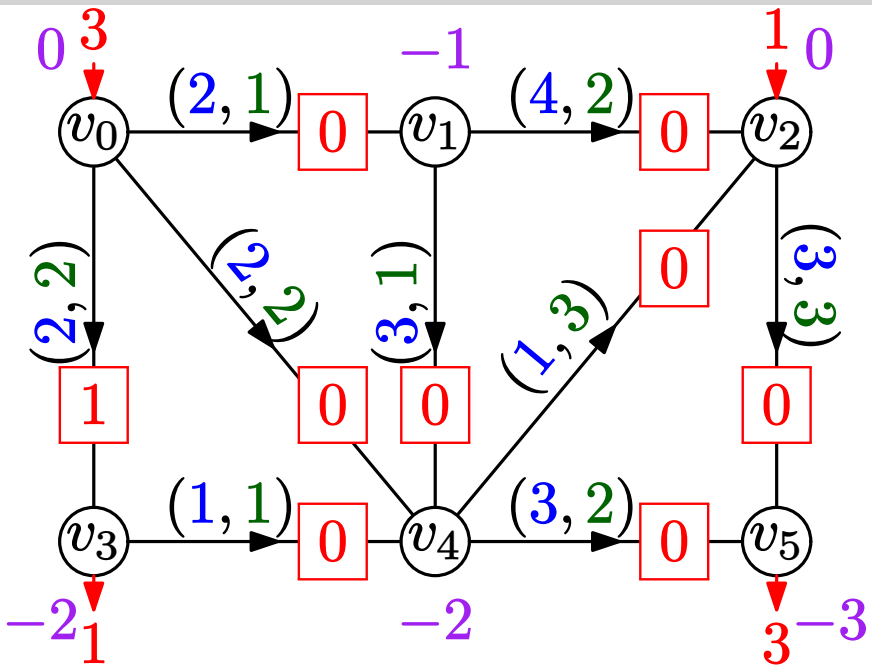
流せるだけ流す

[復習] 逐次最短路法：例 (2/4)

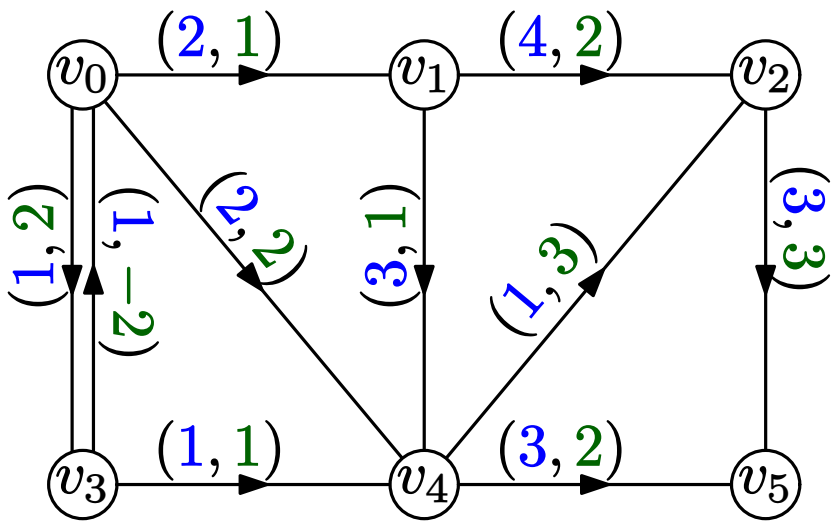


流とポテンシャルの更新

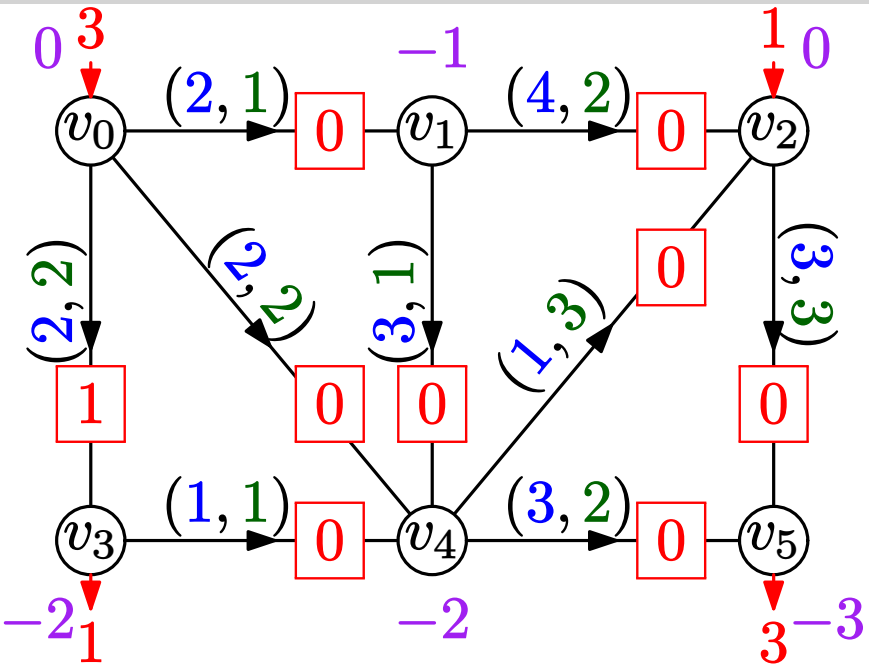
[復習] 逐次最短路法：例 (2/4)



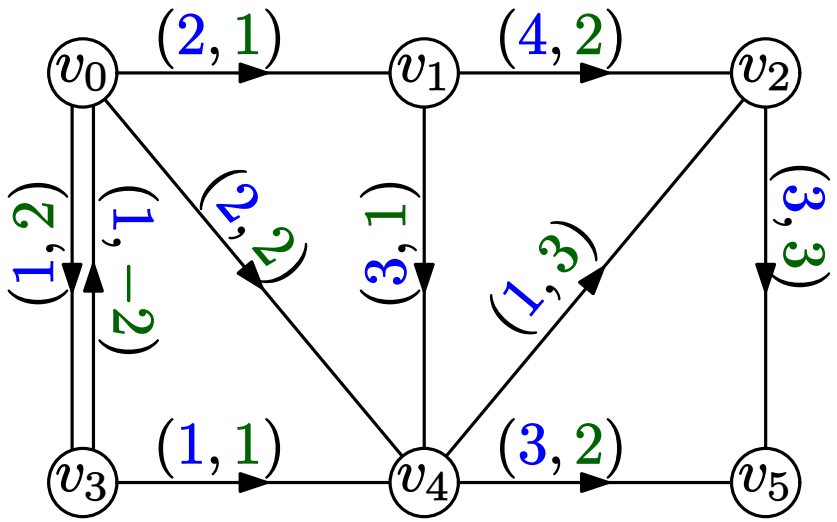
流とポテンシャルの更新

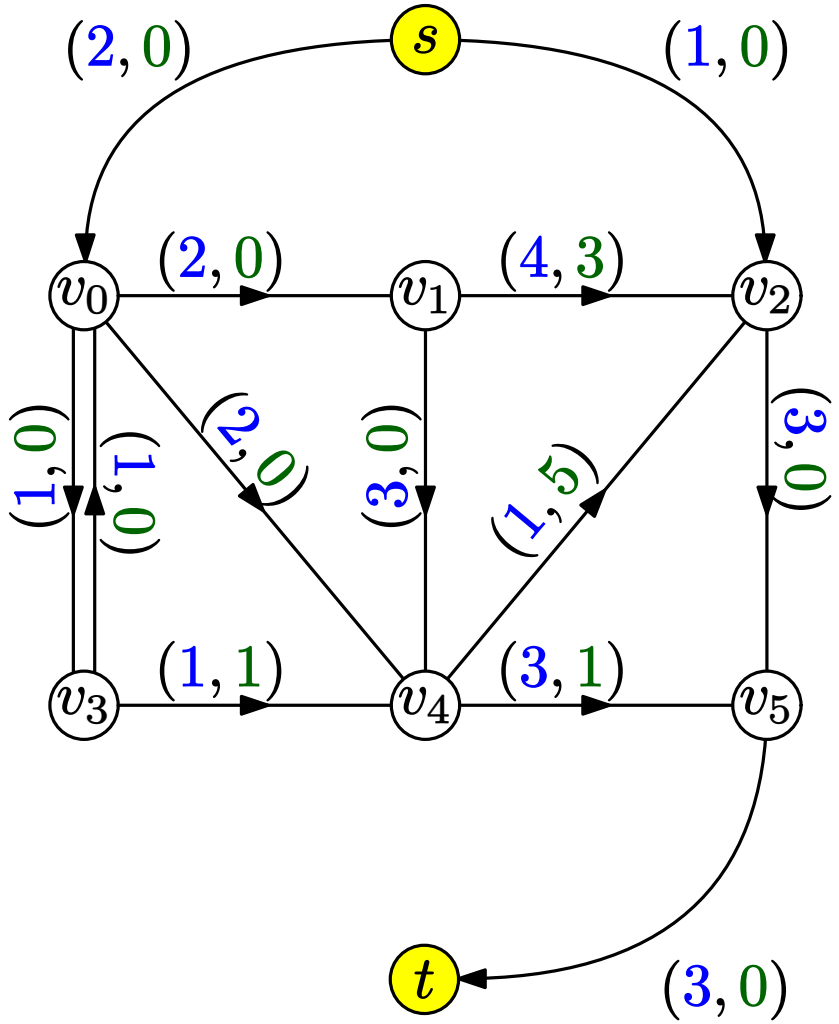


[復習] 逐次最短路法：例 (2/4)

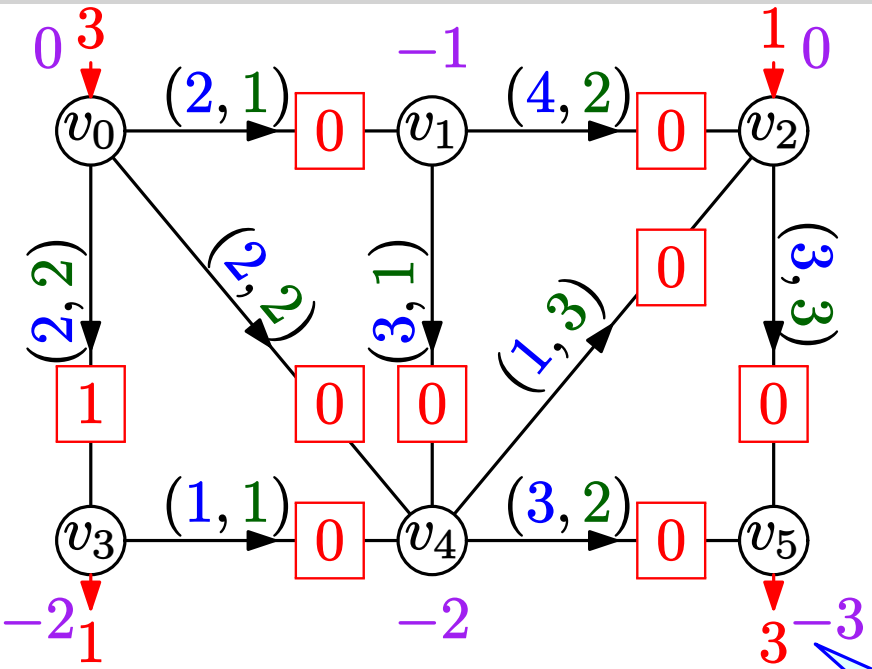


流とポテンシャルの更新

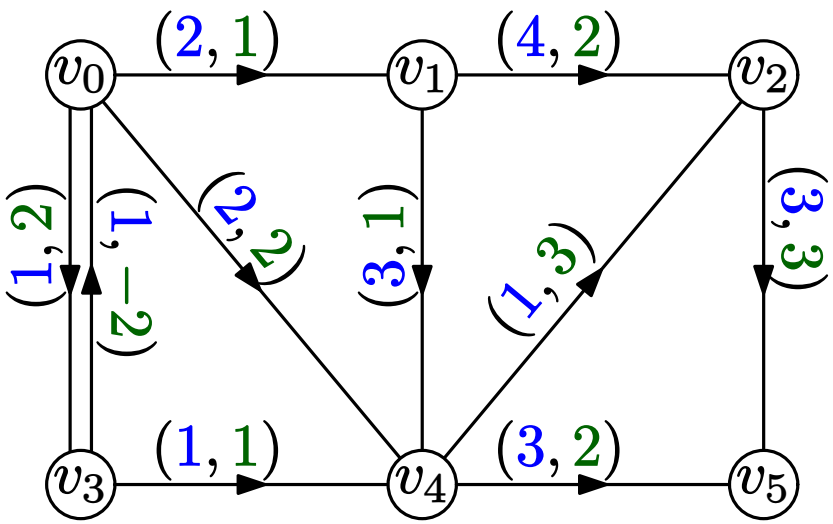


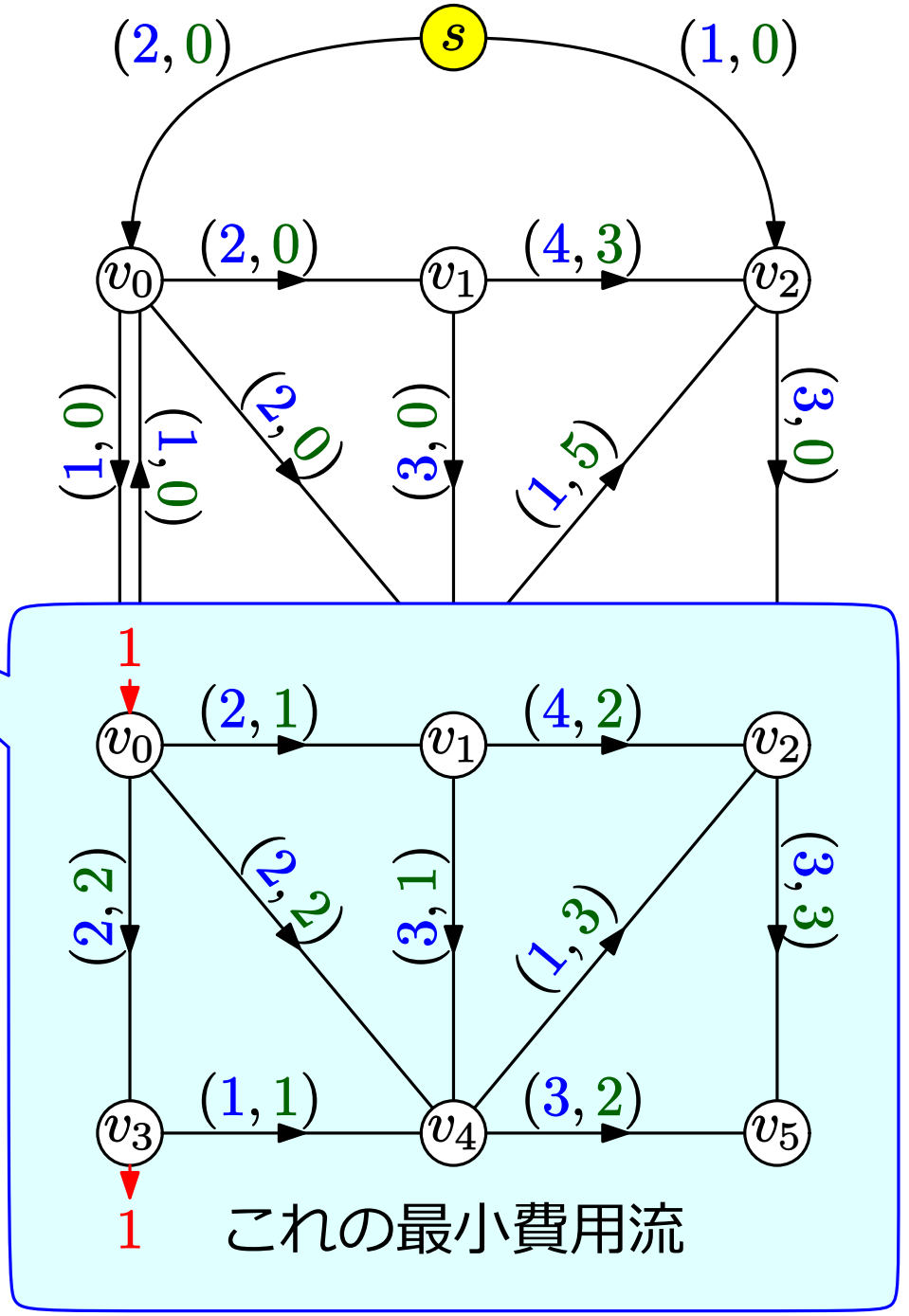


[復習] 逐次最短路法：例 (2/4)

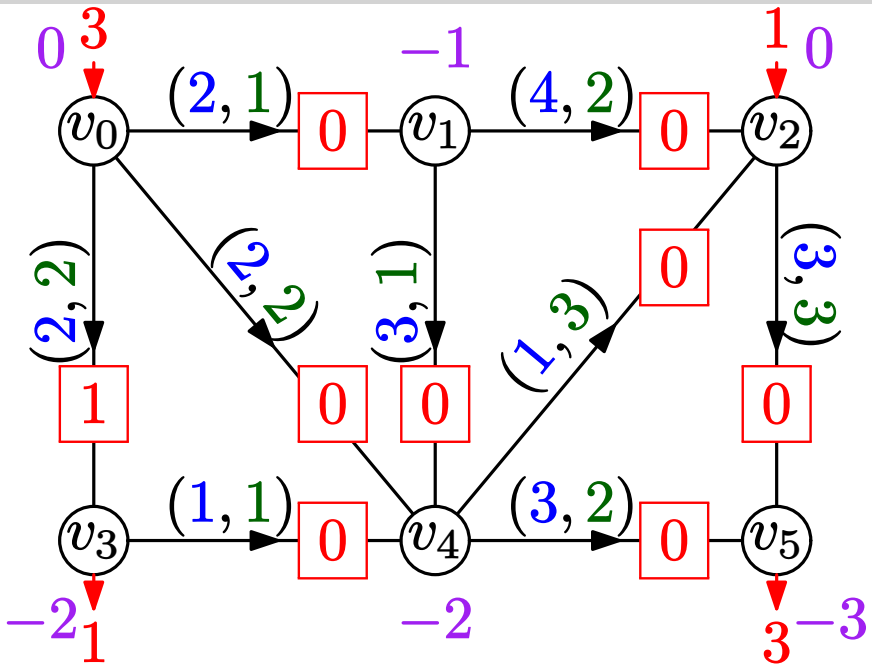


流とポテンシャルの更新

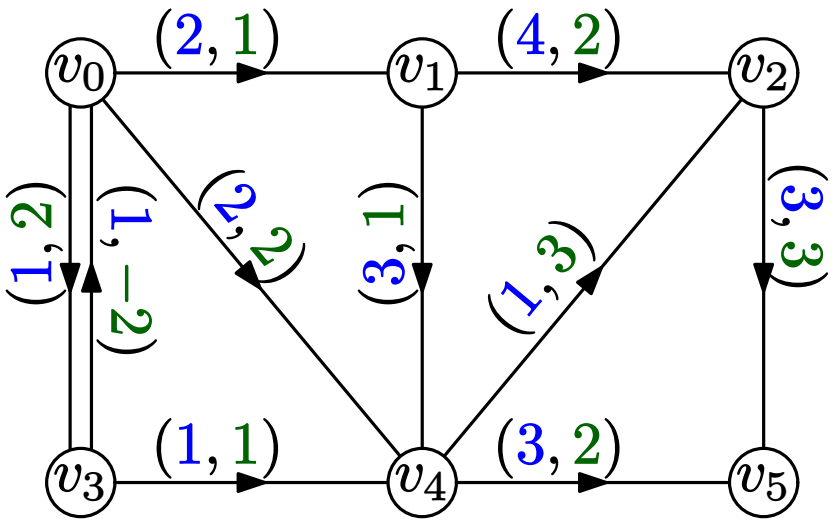


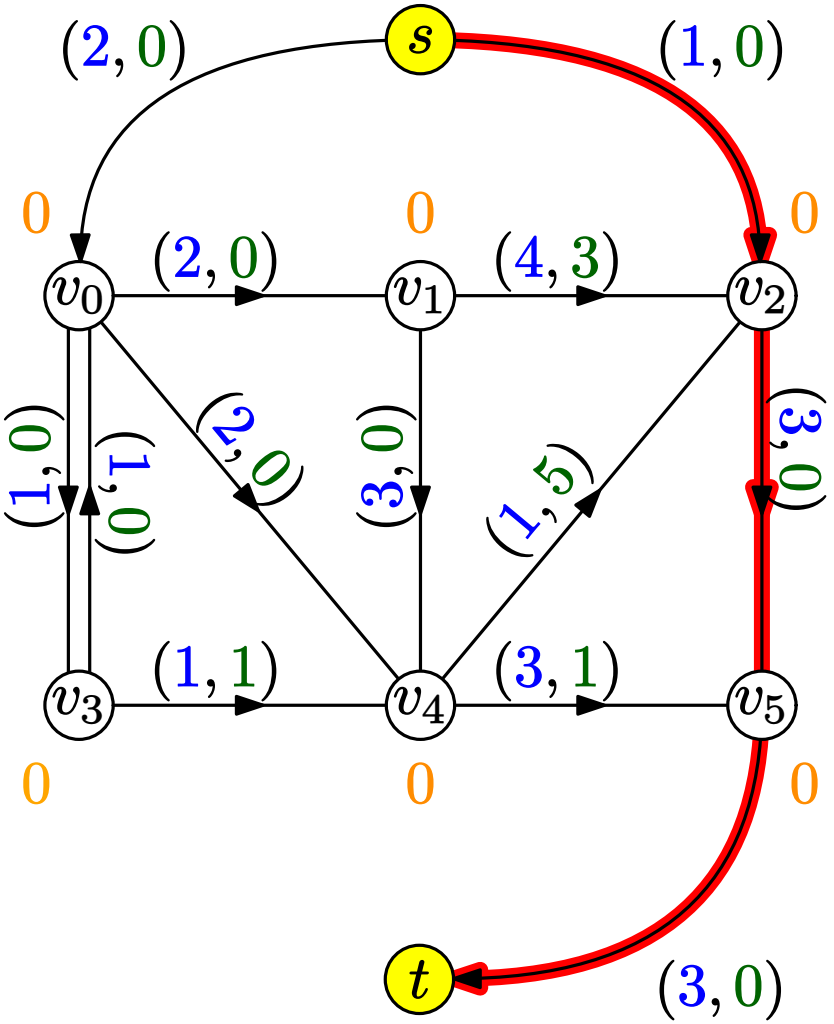


[復習] 逐次最短路法：例 (2/4)



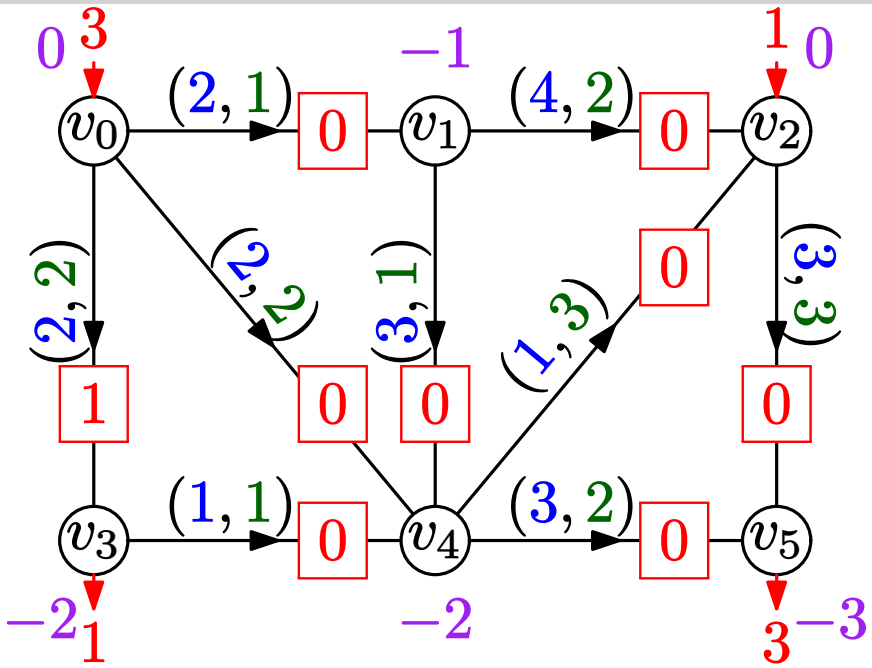
流とポテンシャルの更新



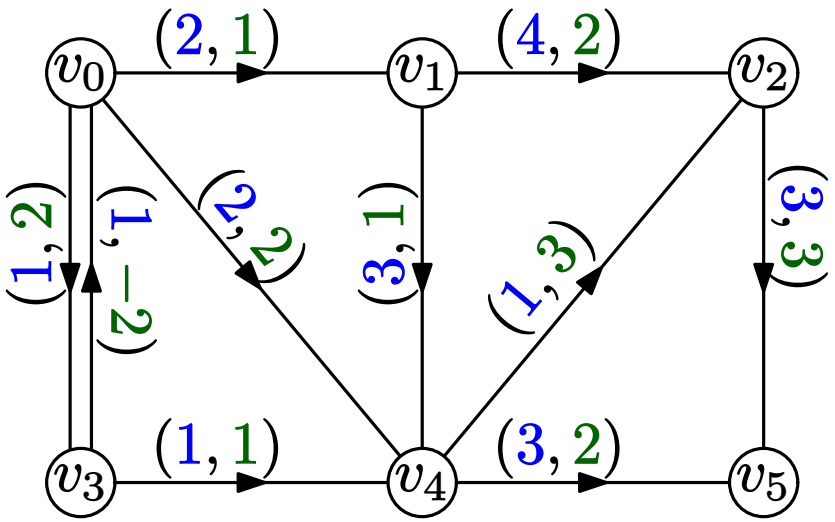


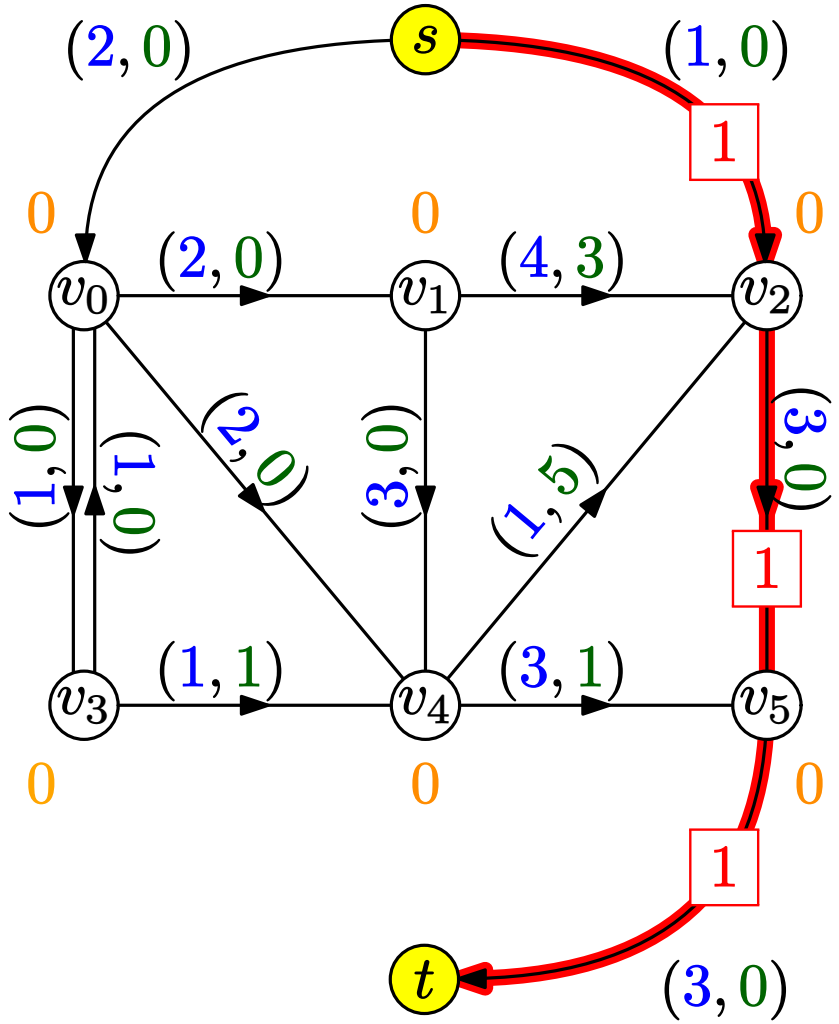
簡約費用を距離とする最短 s-t 路

[復習] 逐次最短路法：例 (2/4)



流とポテンシャルの更新

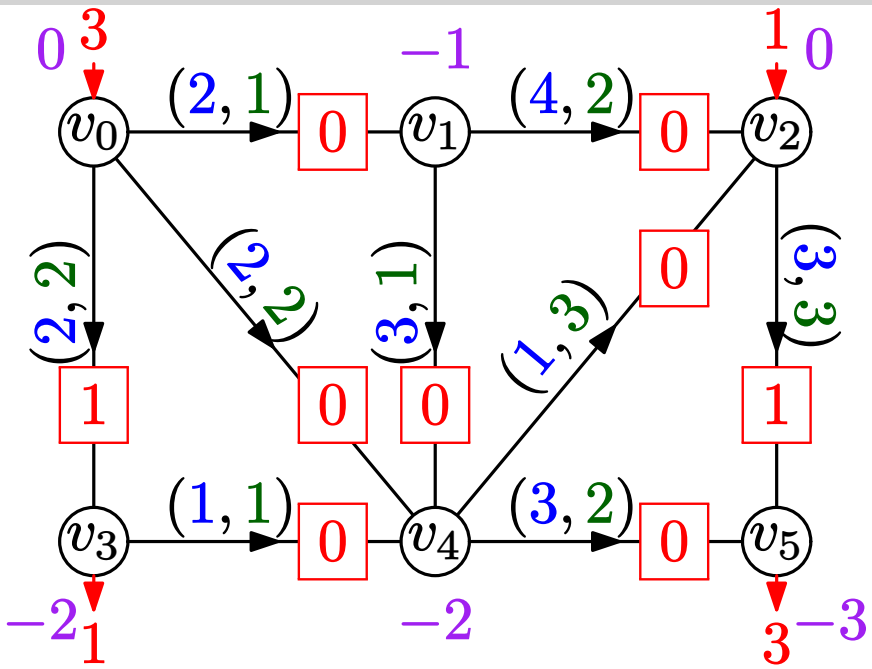




簡約費用を距離とする最短 s-t 路

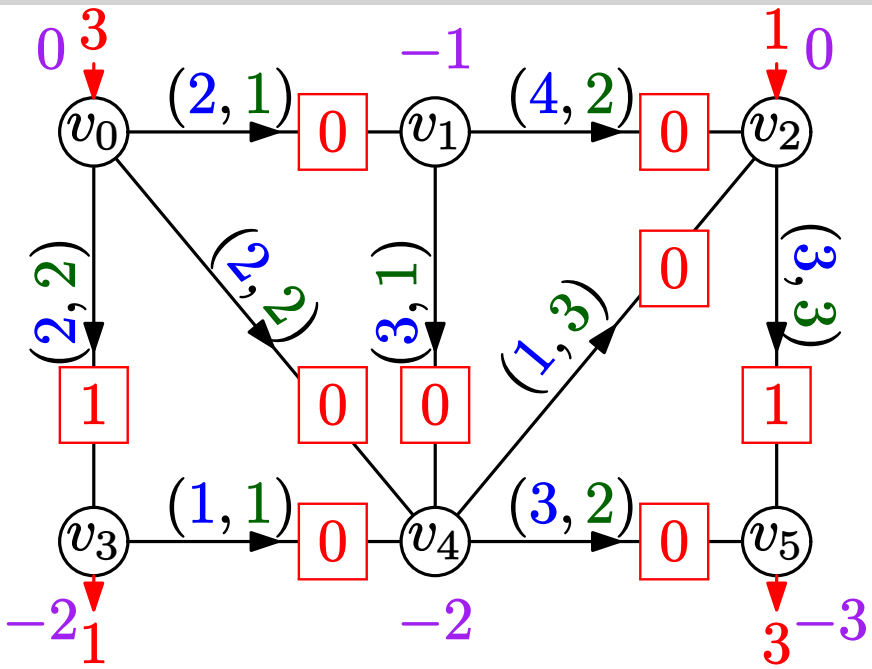
流せるだけ流す

[復習] 逐次最短路法：例 (3/4)

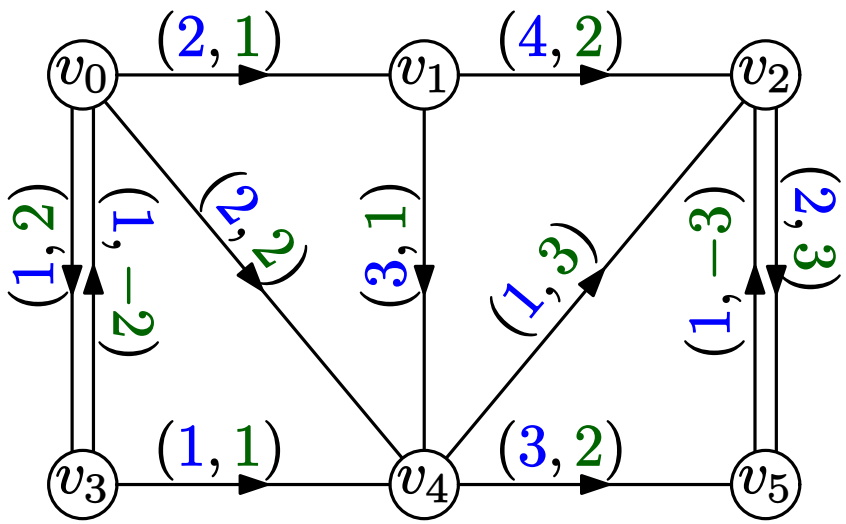


流とポテンシャルの更新

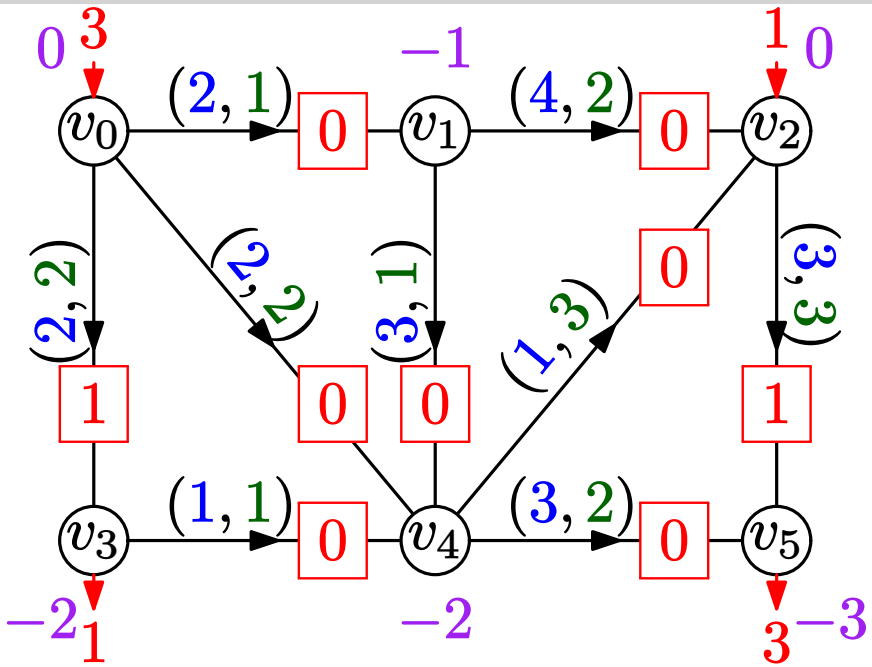
[復習] 逐次最短路法：例 (3/4)



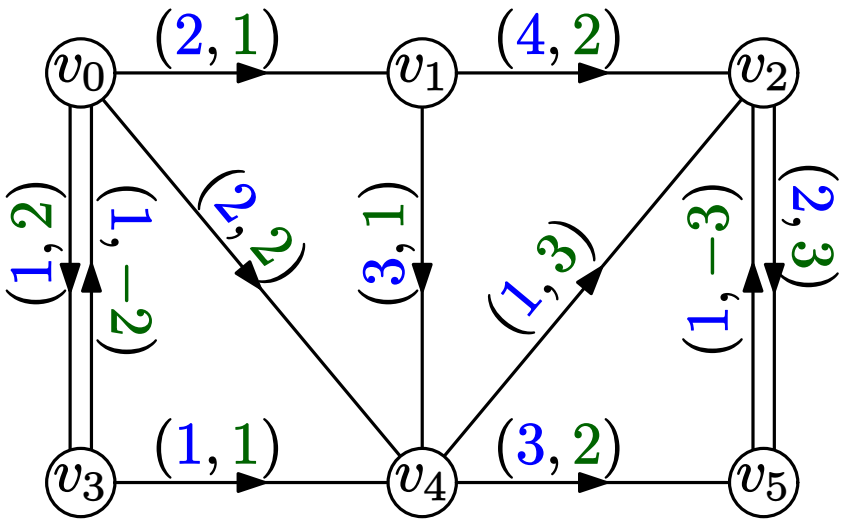
流とポテンシャルの更新

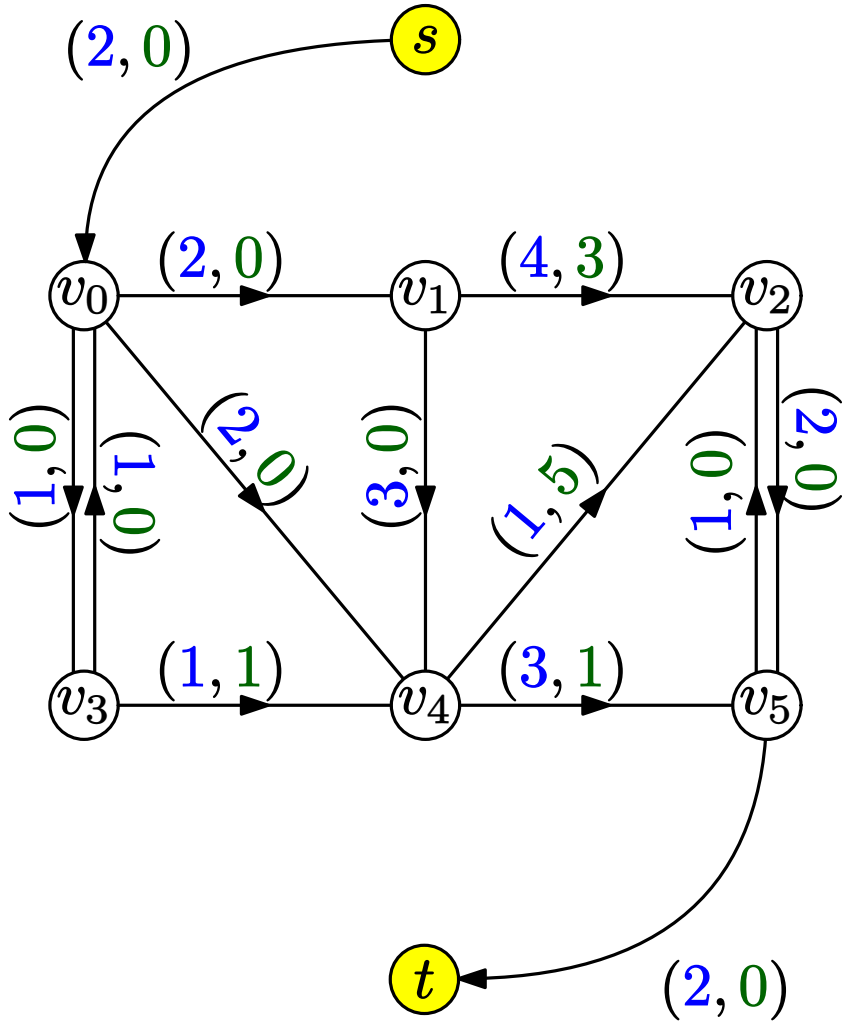


[復習] 逐次最短路法：例 (3/4)

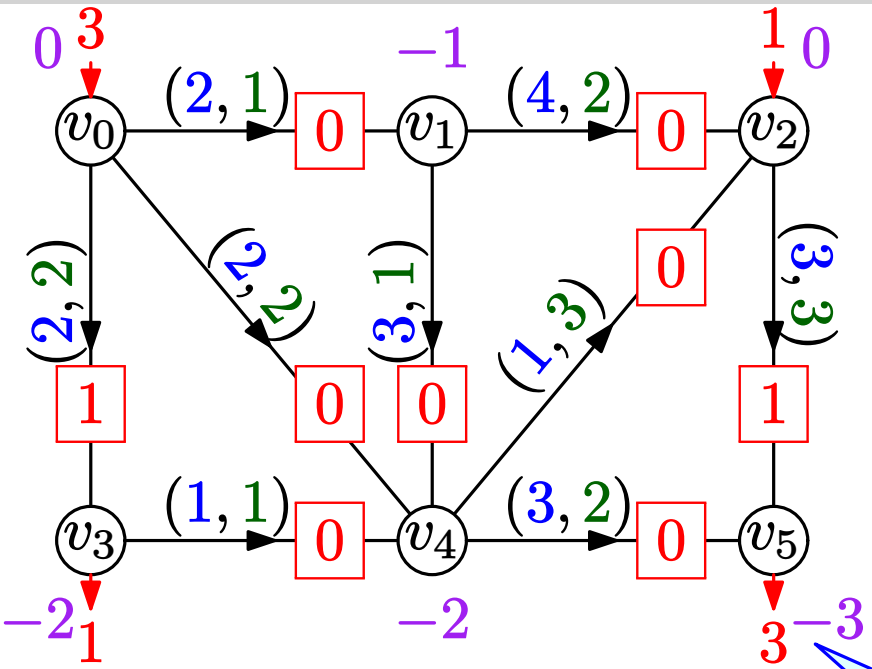


流とポテンシャルの更新

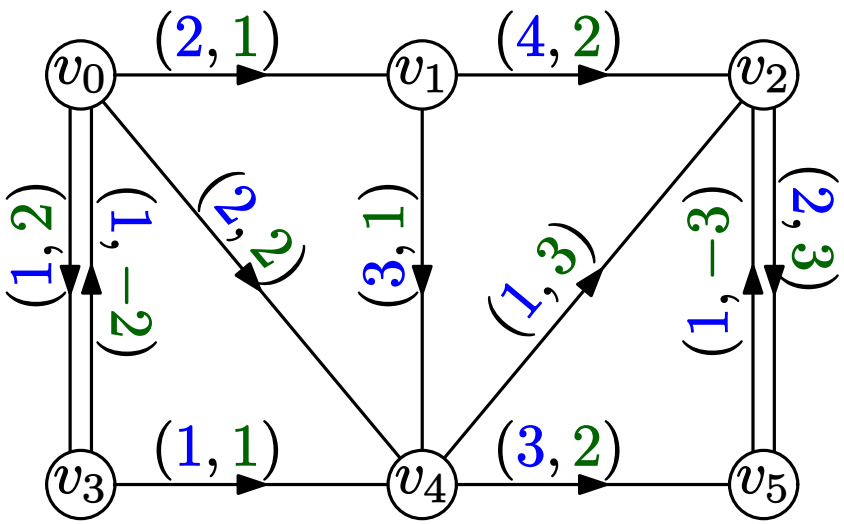


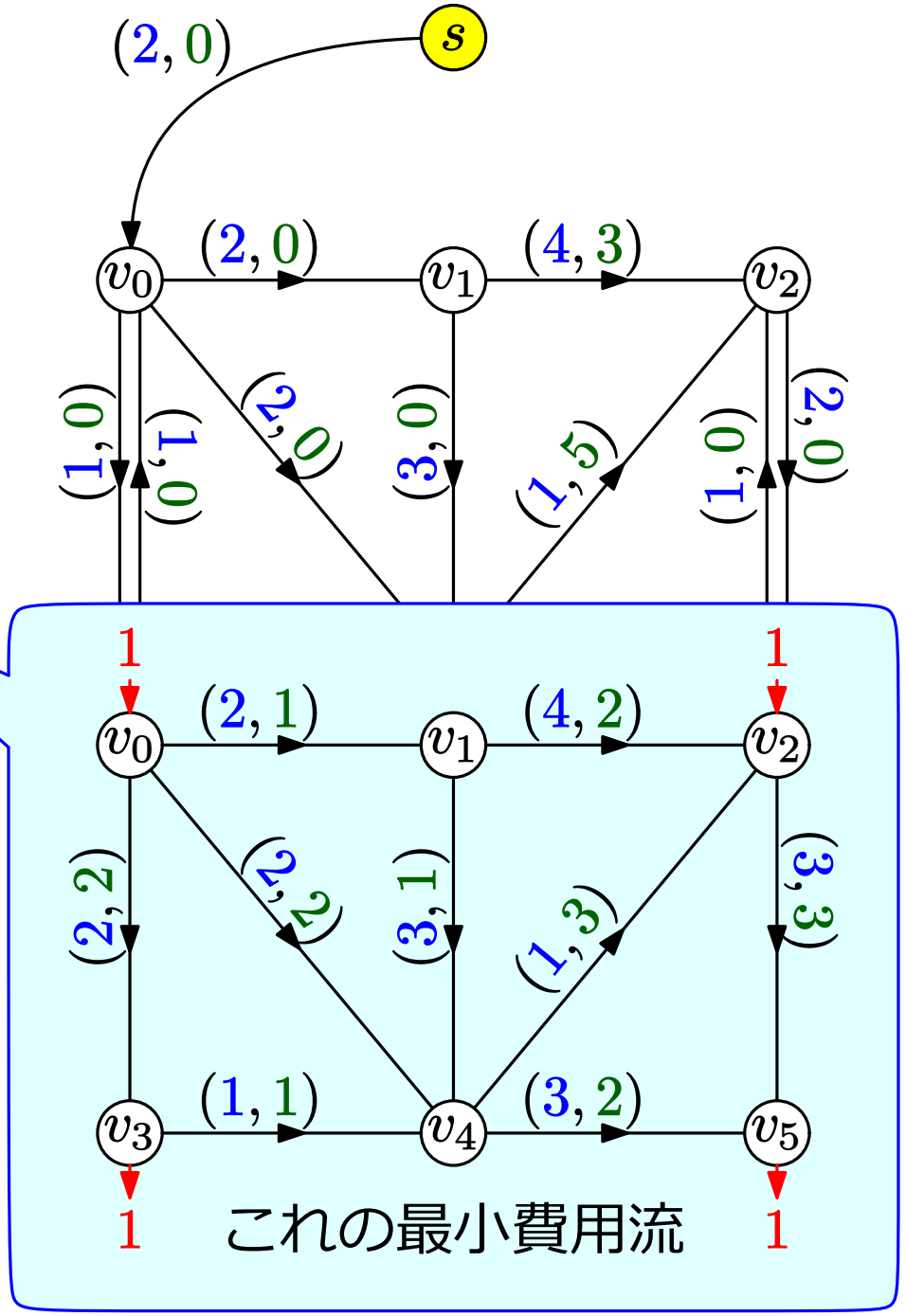


[復習] 逐次最短路法：例 (3/4)

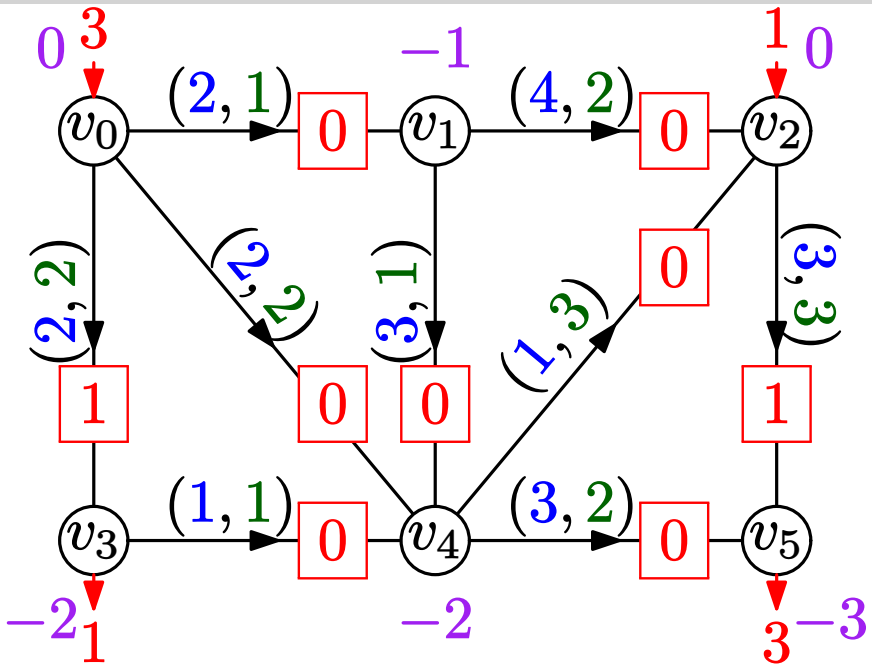


流とポテンシャルの更新

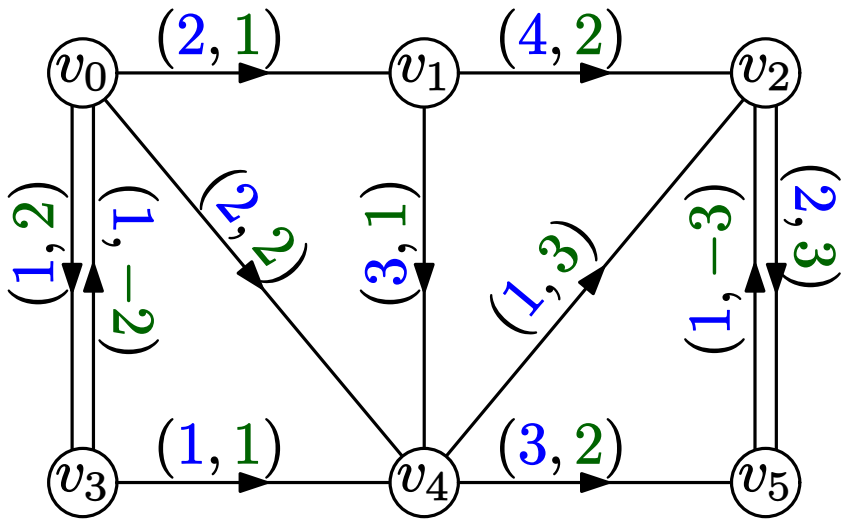


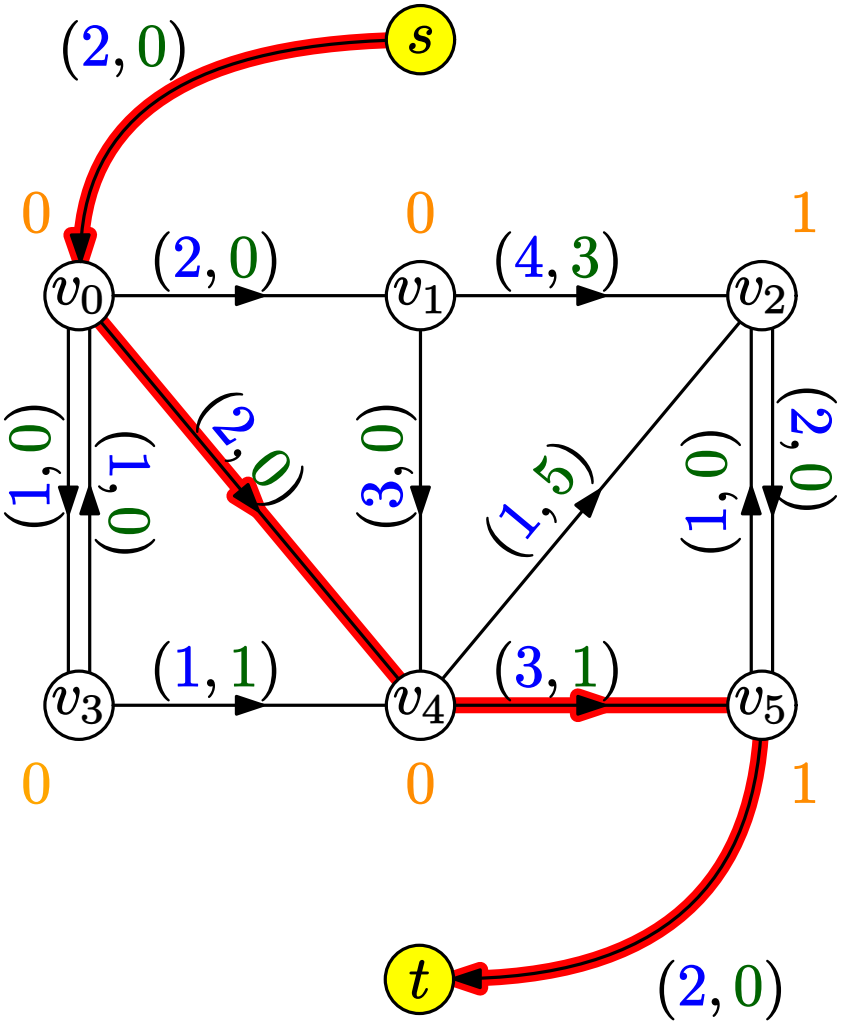


[復習] 逐次最短路法：例 (3/4)



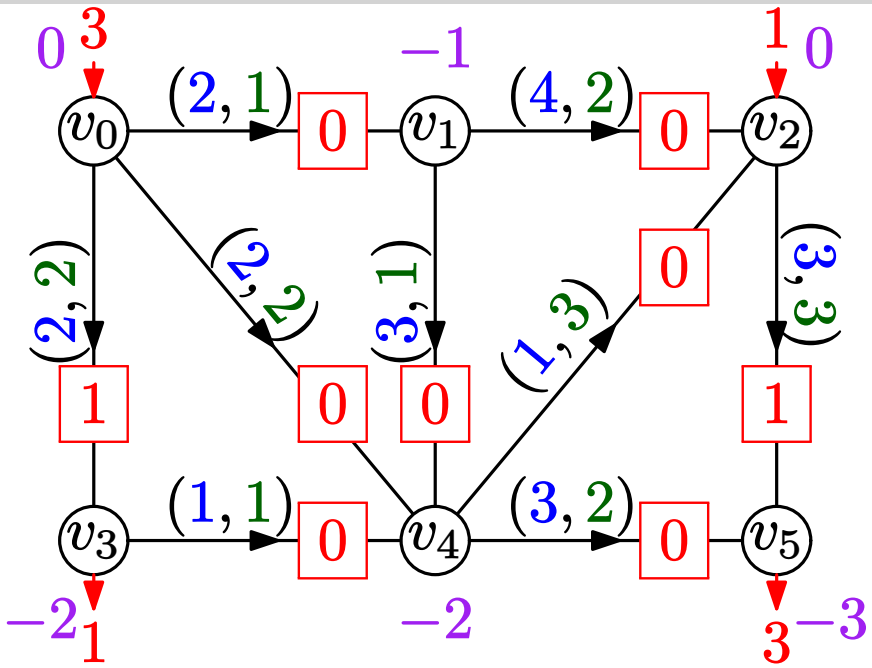
流とポテンシャルの更新



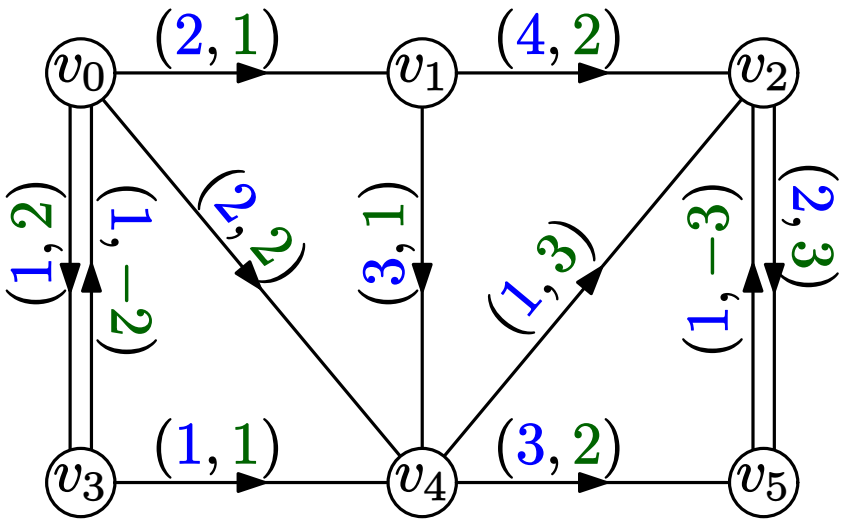


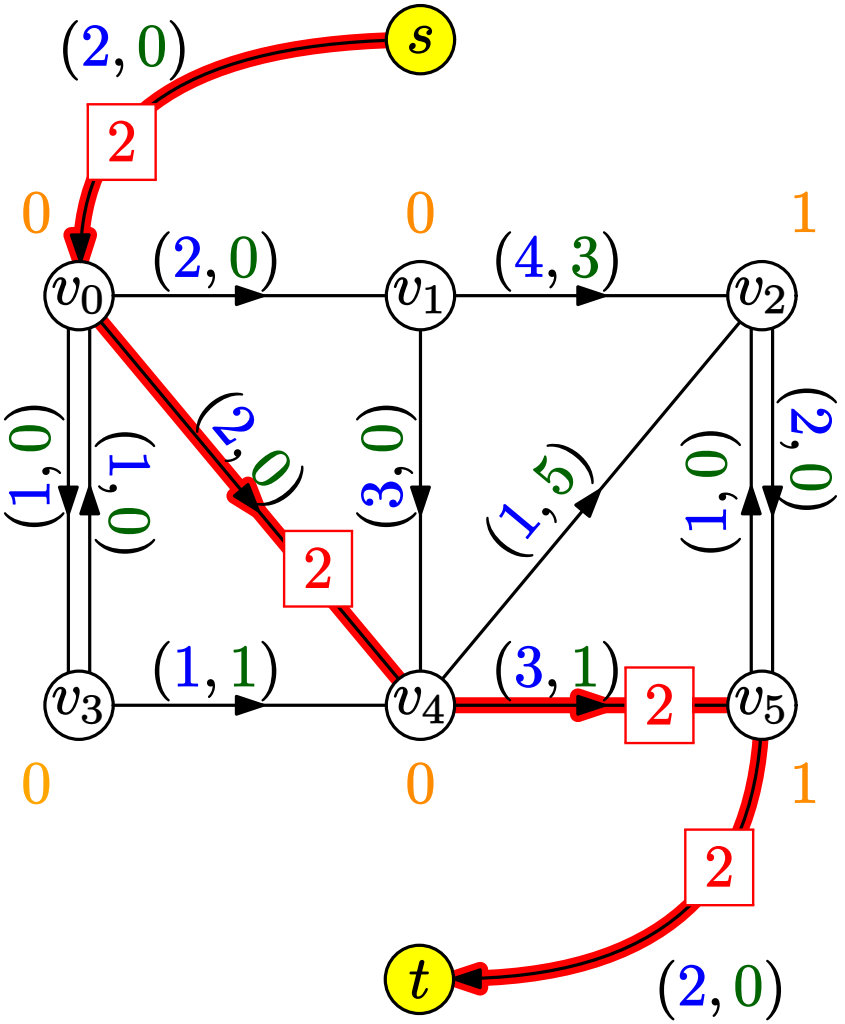
簡約費用を距離とする最短 s-t 路

[復習] 逐次最短路法：例 (3/4)



流とポテンシャルの更新

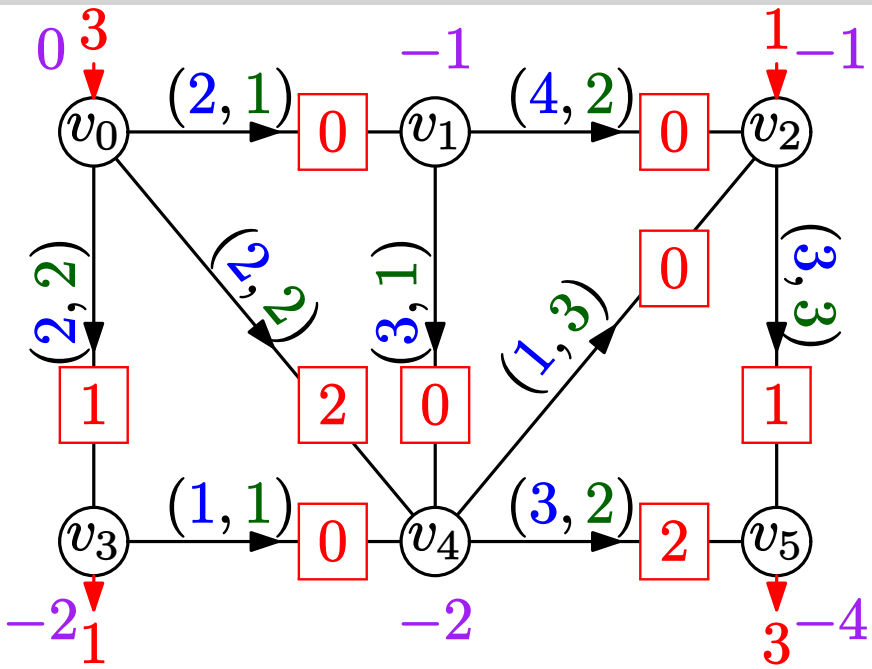




簡約費用を距離とする最短 s-t 路

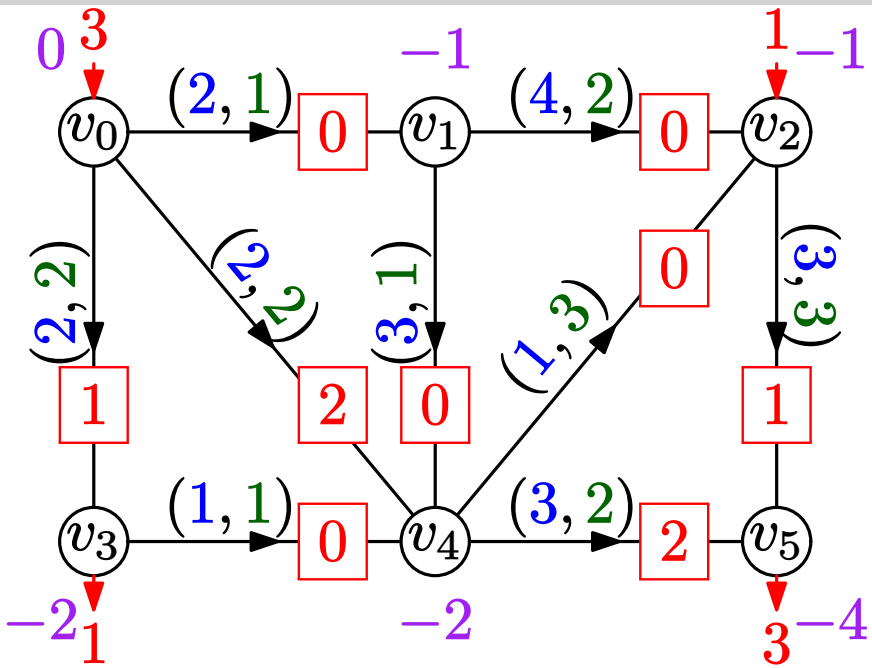
流せるだけ流す

[復習] 逐次最短路法：例 (4/4)

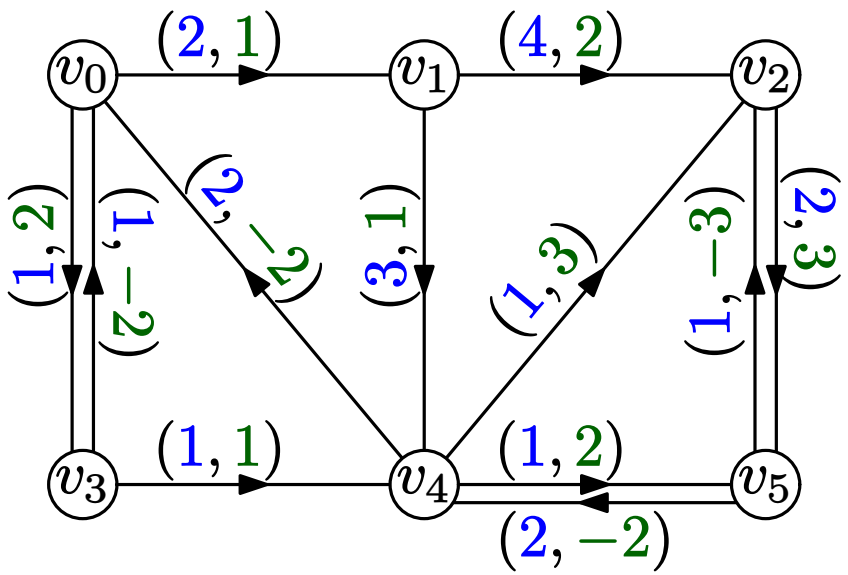


流とポテンシャルの更新

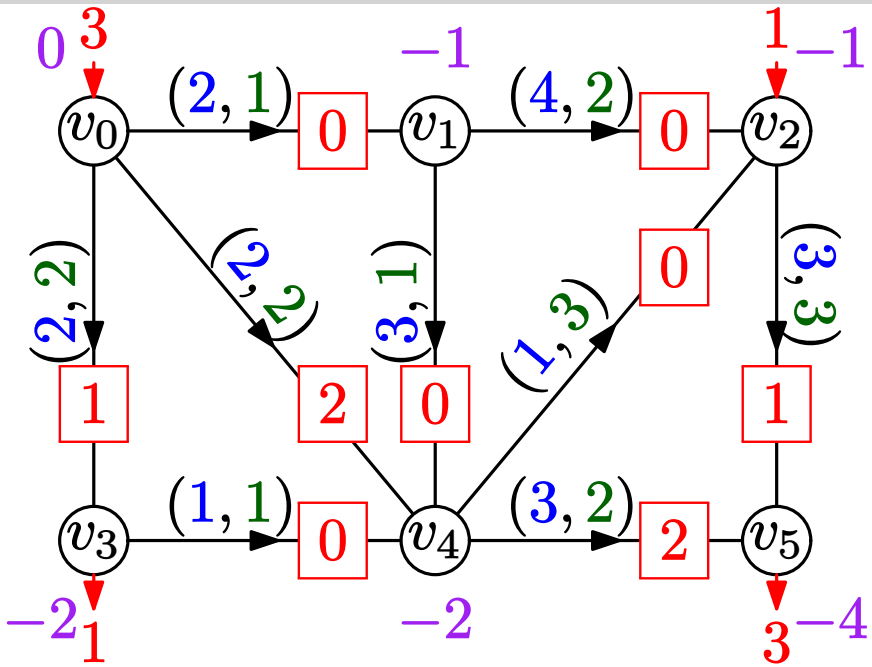
[復習] 逐次最短路法：例 (4/4)



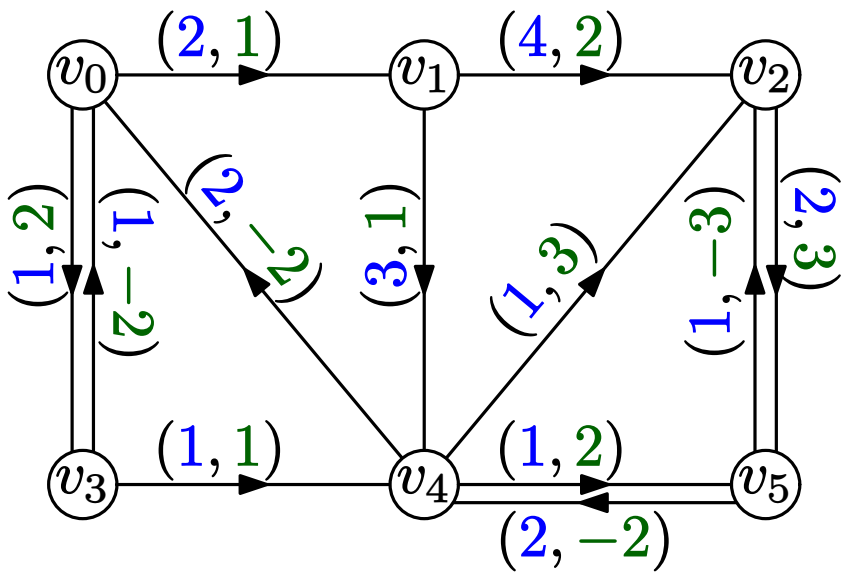
流とポテンシャルの更新

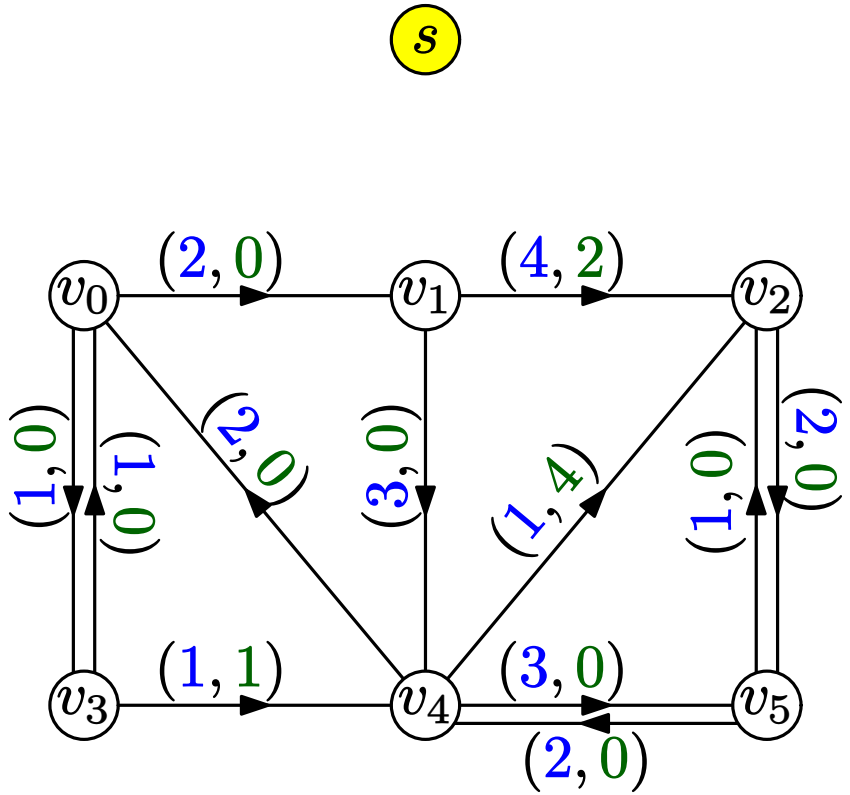


[復習] 逐次最短路法：例 (4/4)

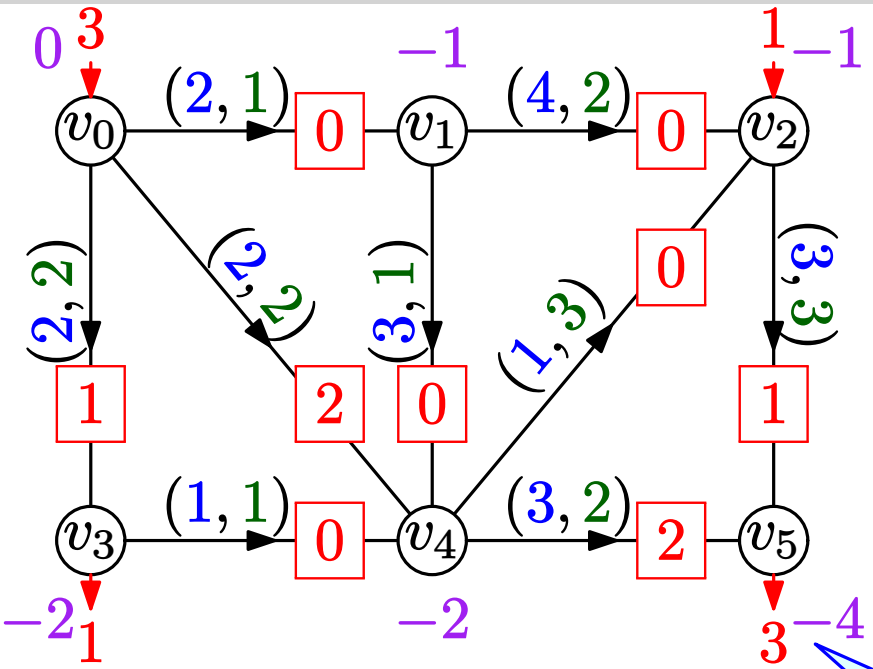


流とポテンシャルの更新

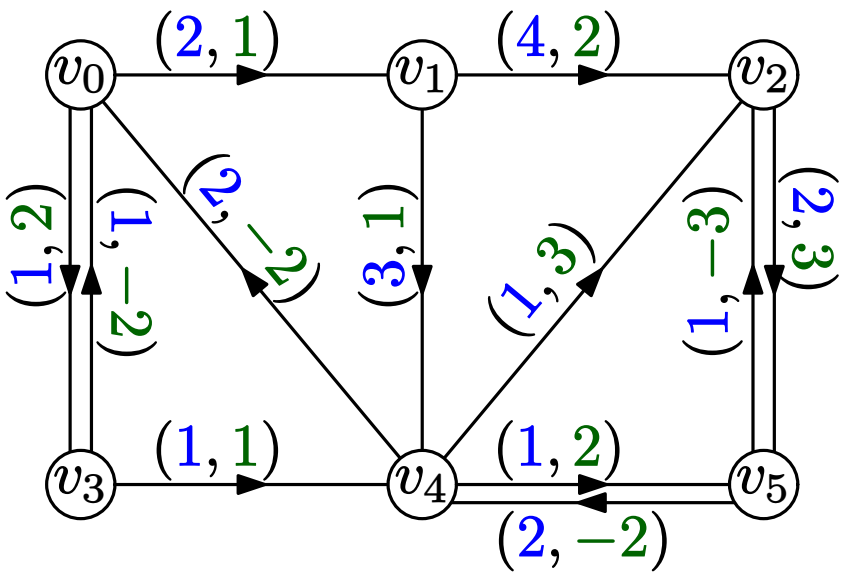


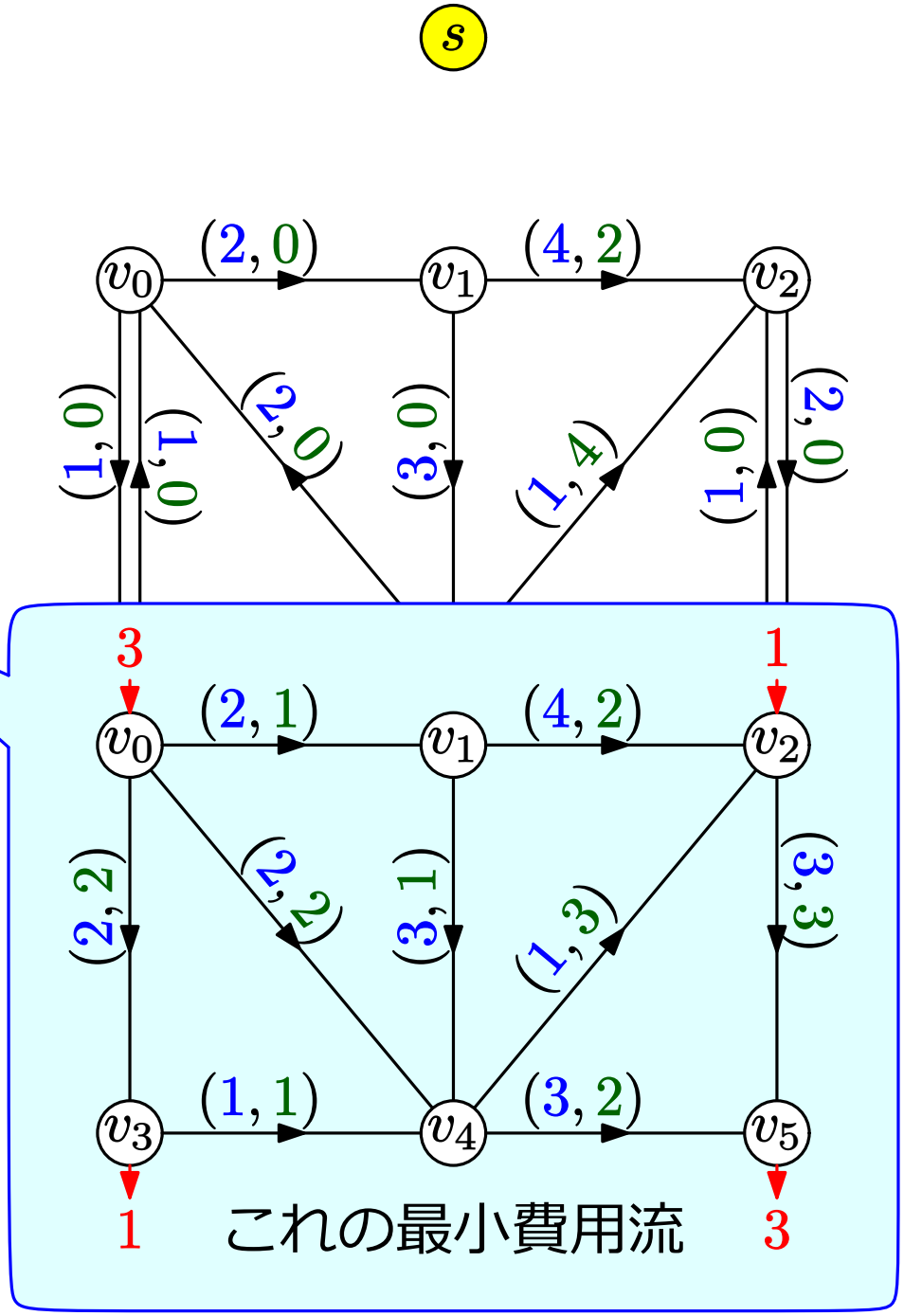


[復習] 逐次最短路法：例 (4/4)

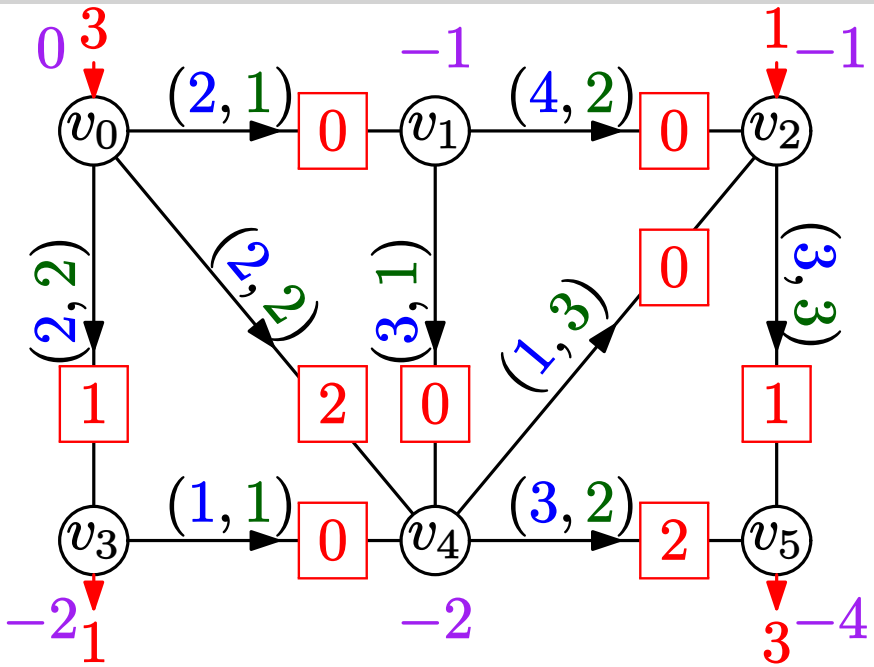


流とポテンシャルの更新

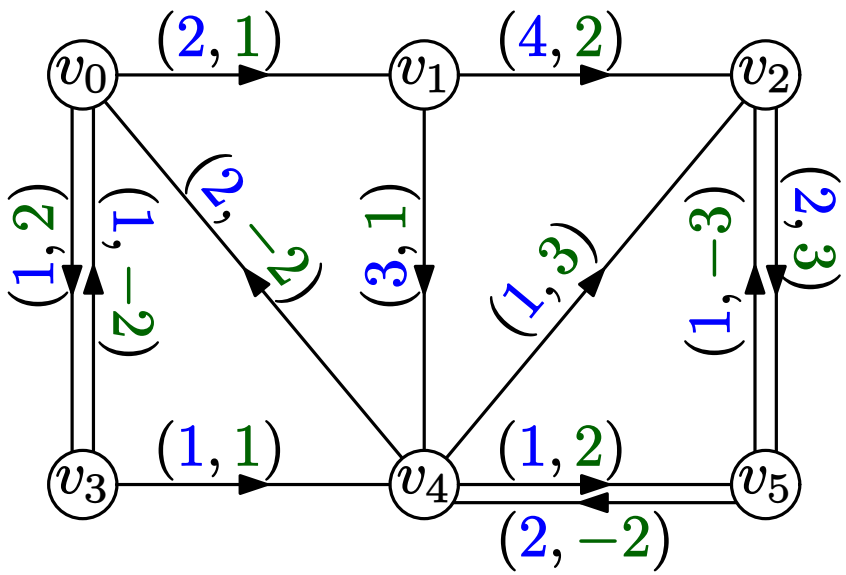


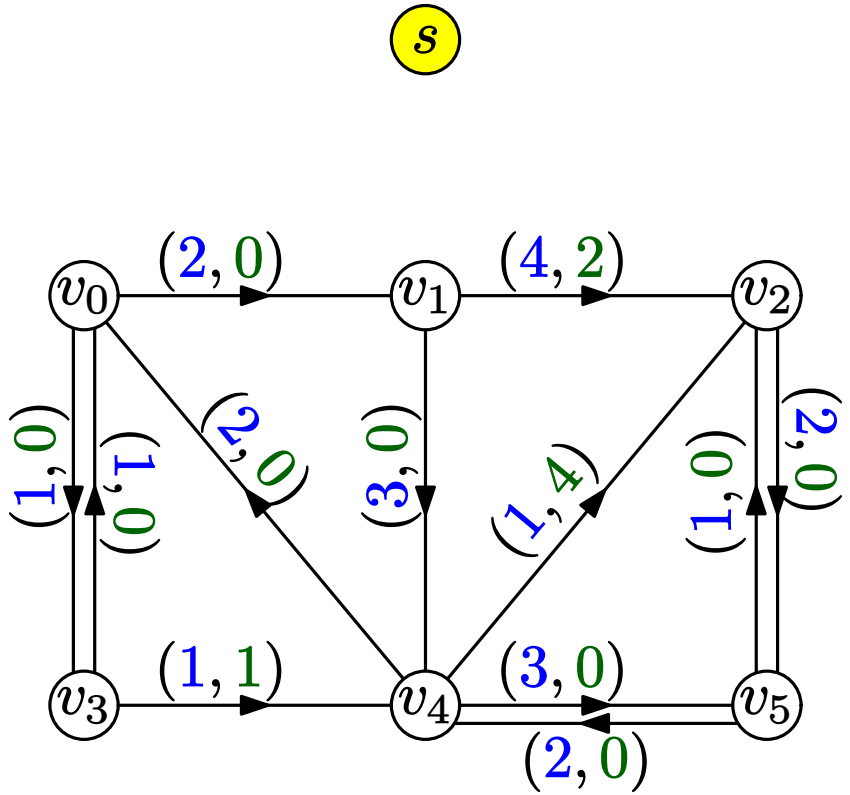


[復習] 逐次最短路法：例 (4/4)



流とポテンシャルの更新





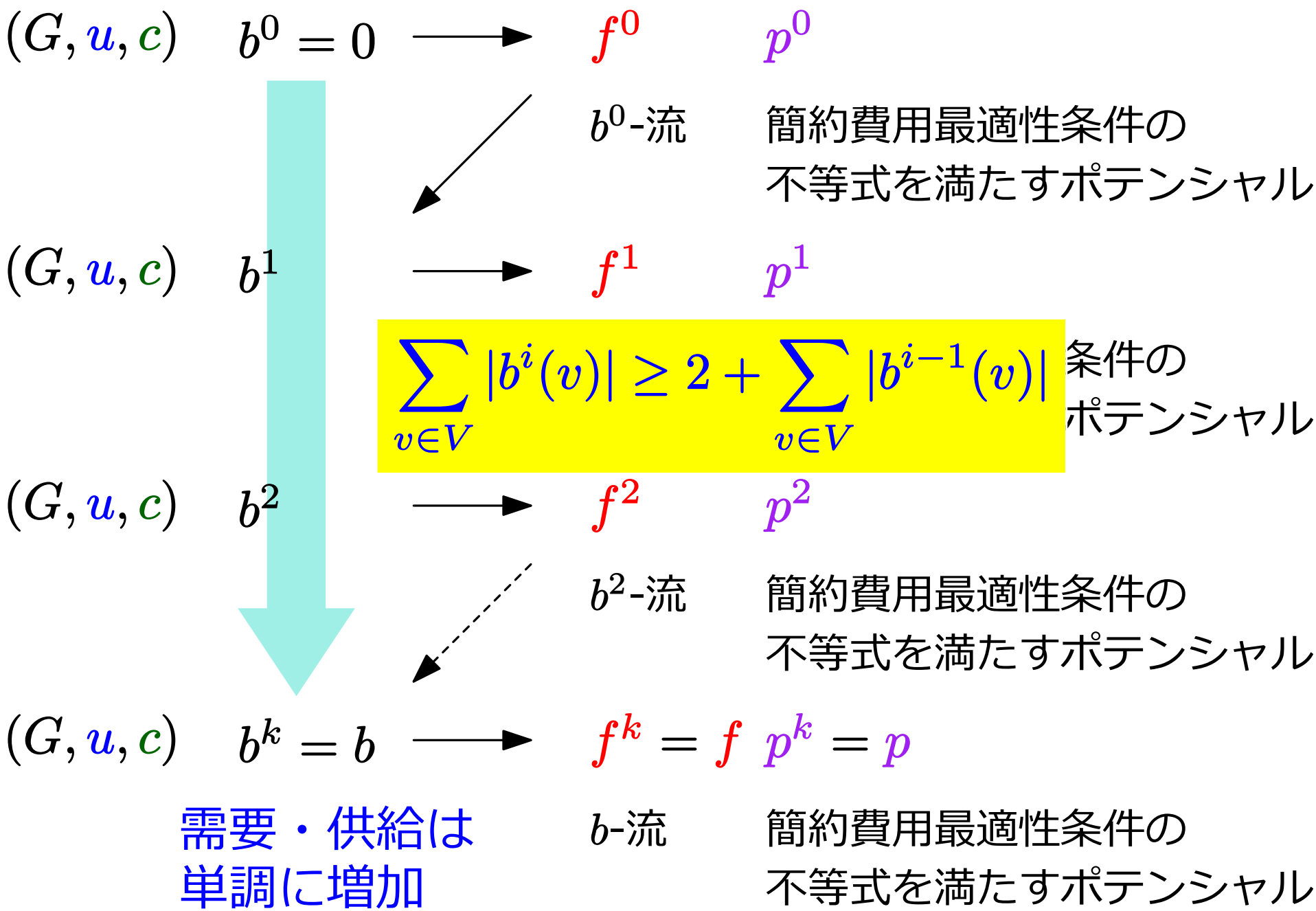
最小費用 b-流が得られた

[復習] 逐次最短路法

アルゴリズム：逐次最短路法

- 初期化：流 $f = 0$，ポテンシャル $p = 0$
- 反復：f が b-流になるまで次を実行
  1. 補助ネットワークを作成する
  2. 費用を簡約費用に変更し，頂点 s, t を追加する
  3. まだ供給のある頂点に s から弧を作り，
     容量は残っている供給，費用は 0 とする
  4. まだ需要のある頂点から t に弧を作り，
     容量は残っている需要，費用は 0 とする
  5. 各頂点 v に対して，次を計算
     $d_v =$ 簡約費用を距離とする最短 s-v 長
  6. 最短 s-t 路に沿って流せるだけ流し，f を更新する
  7. $p_v = p_v - d_v$ とポテンシャルを更新する

[復習] 逐次最短路法の計算量 (1/1)



[復習] 逐次最短路法の計算量 (1/1)

(G, u, c) $b^0 = 0$ → f^0 p^0

b^0-流　簡約費用最適性条件の不等式を満たすポテンシャル

(G, u, c) b^1 → f^1 p^1

$$\sum_{v \in V} |b^i(v)| \geq 2 + \sum_{v \in V} |b^{i-1}(v)|$$

条件のポテンシャル

(G, u, c) b^2 → f^2 p^2

$$\therefore \quad \text{反復回数} \leq \frac{1}{2} \sum_{v \in V} |b(v)| = O(nB)$$

ヤル

(G, u, c) $b^k = b$ → $f^k = f$ $p^k = p$

需要・供給は単調に増加

b-流　簡約費用最適性条件の不等式を満たすポテンシャル

[復習] 逐次最短路法の計算量 (2/2)

性質：逐次最短路法の計算量

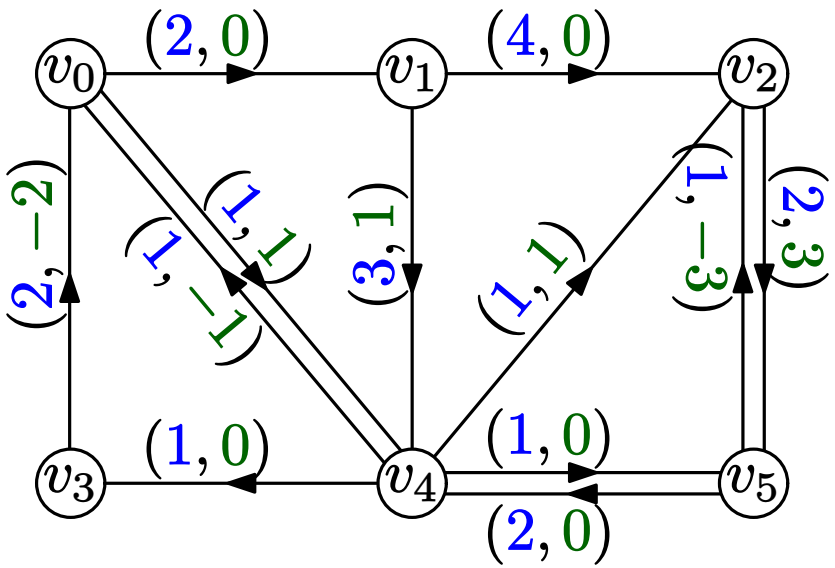
容量 u と供給/需要 b が整数 $\Rightarrow$
逐次最短路法は $O(n(m + n\log n)B)$ 時間で停止する

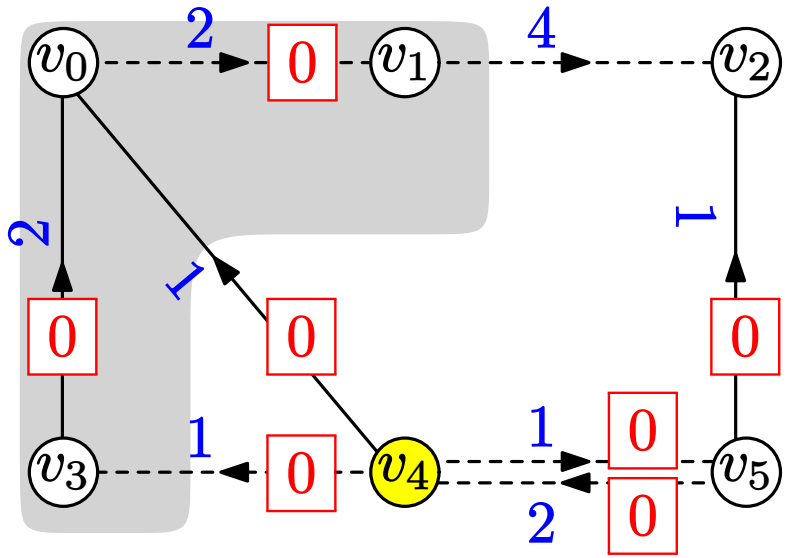
注：

- この計算量は 擬多項式時間
- 費用 c は整数でなくてもよい
  (c.f. Dijkstra 法は強多項式時間アルゴリズム)

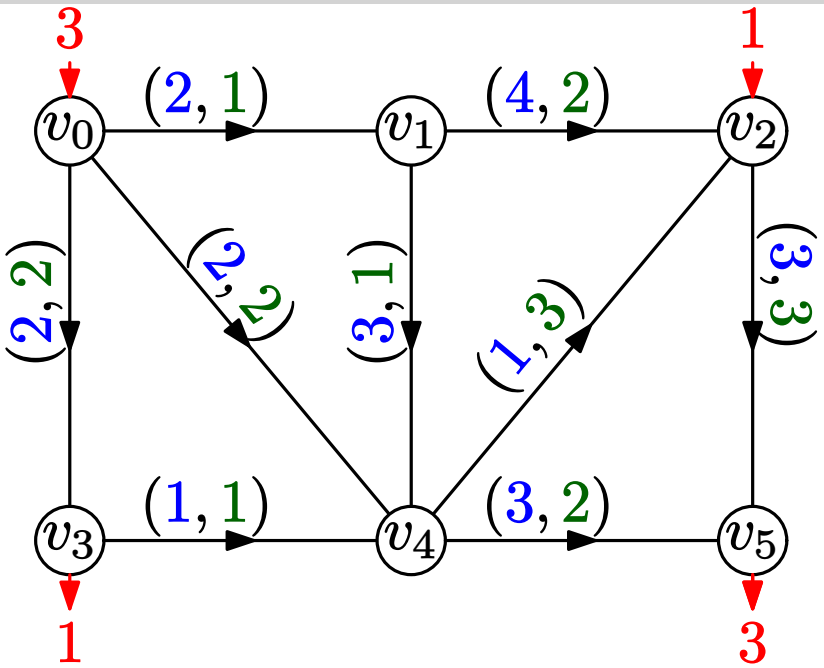
本日の内容

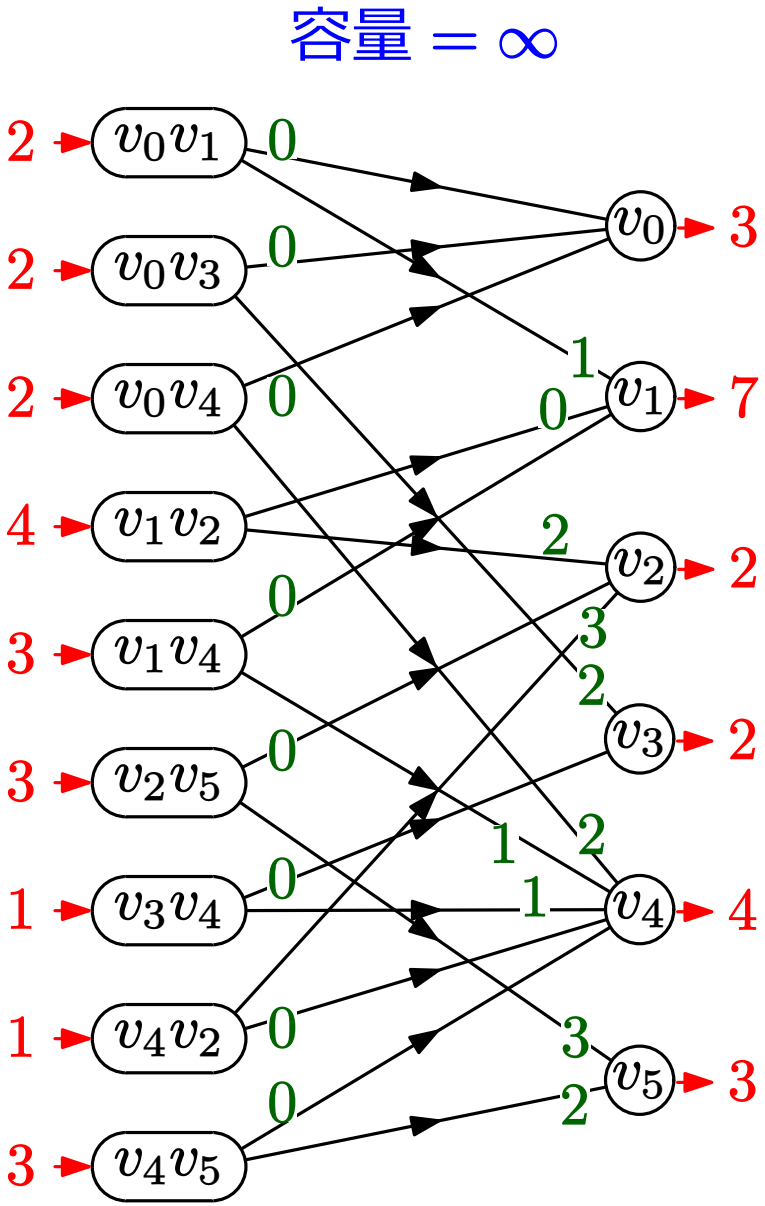
1. **容量スケーリング法：準備**
2. 容量スケーリング法：概要
3. 容量スケーリング法：計算量
4. 主双対法



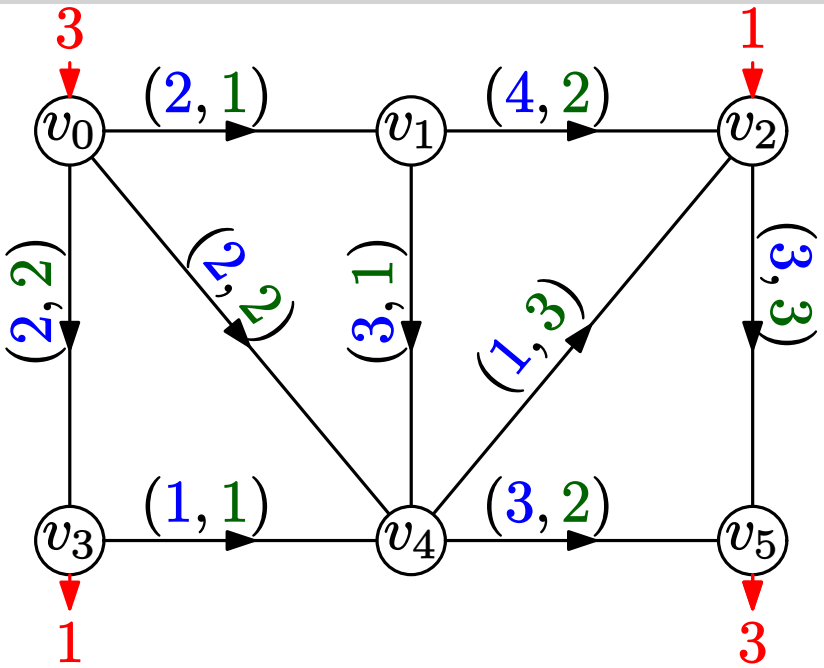


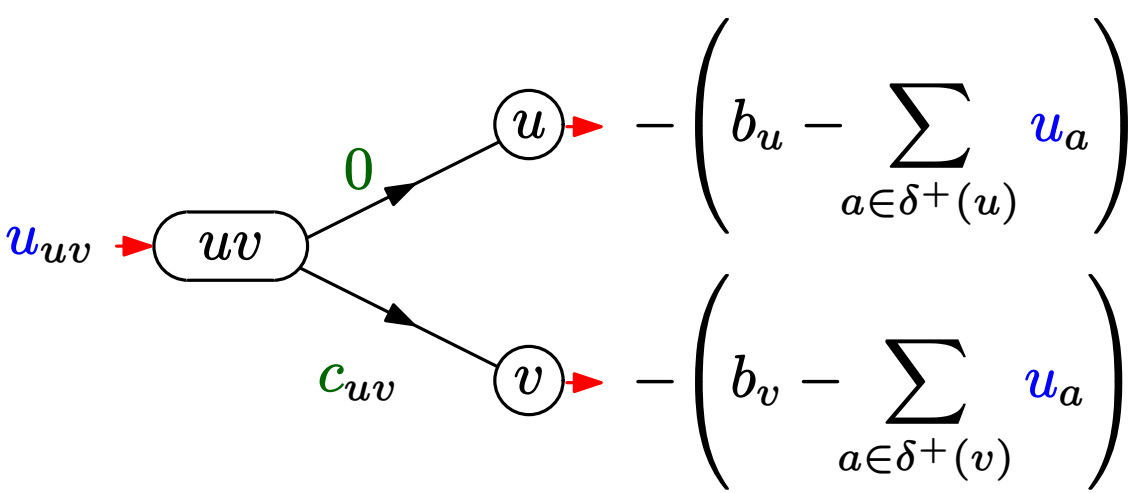
容量 ∞ の場合に変換：例



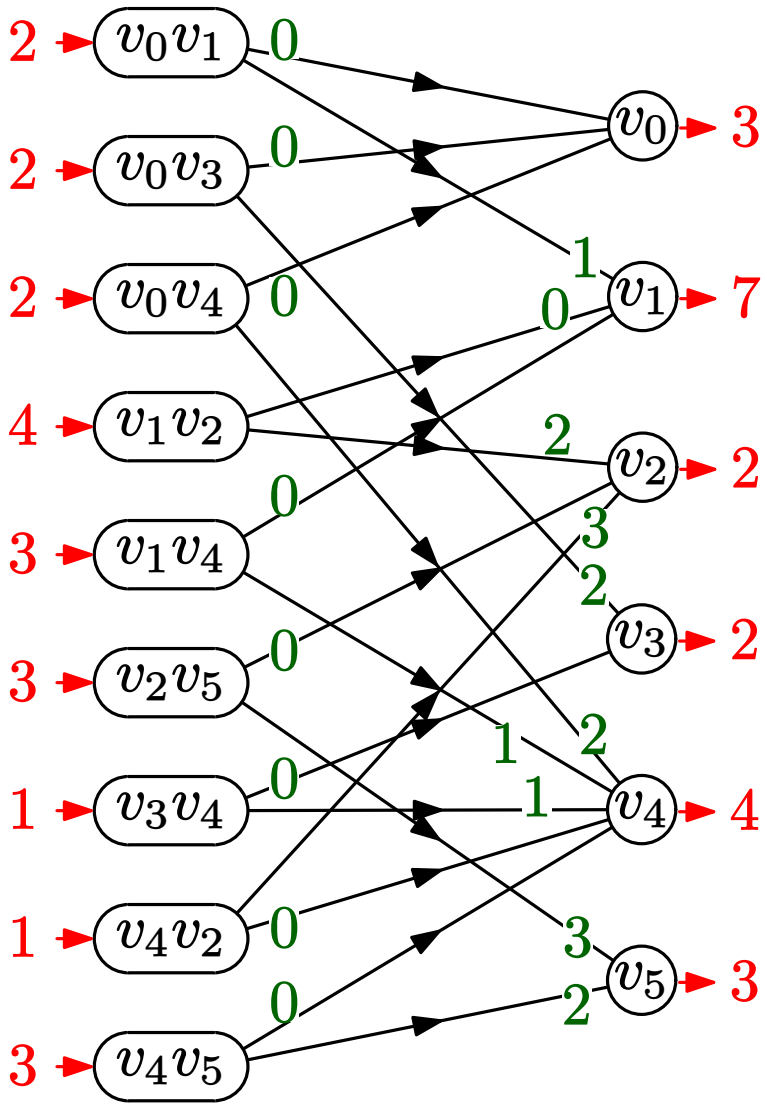


容量 ∞ の場合に変換：例

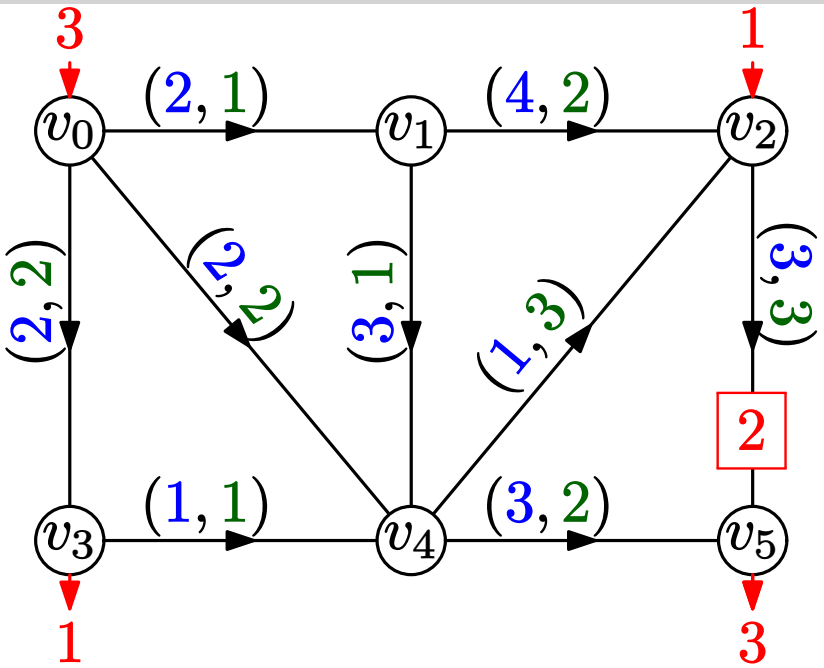


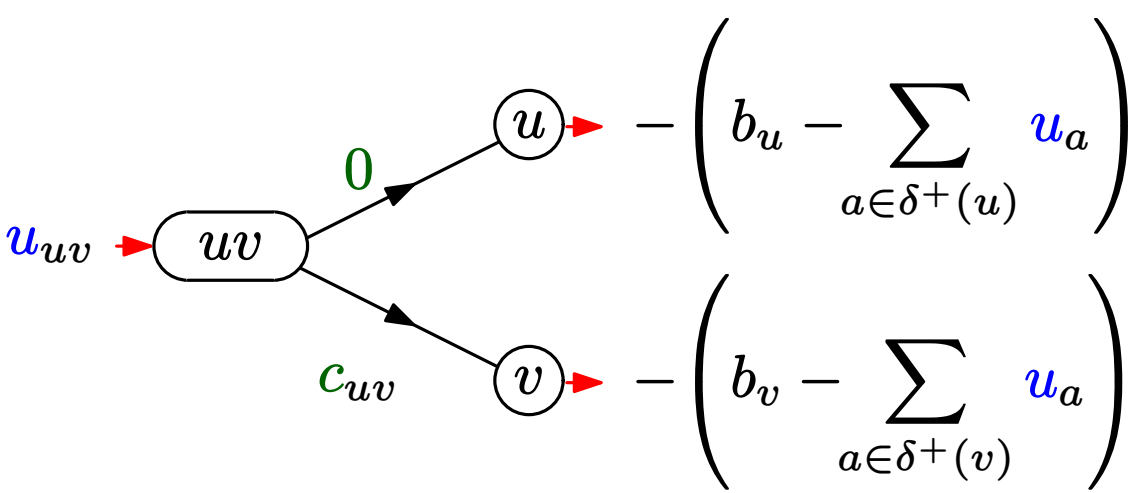


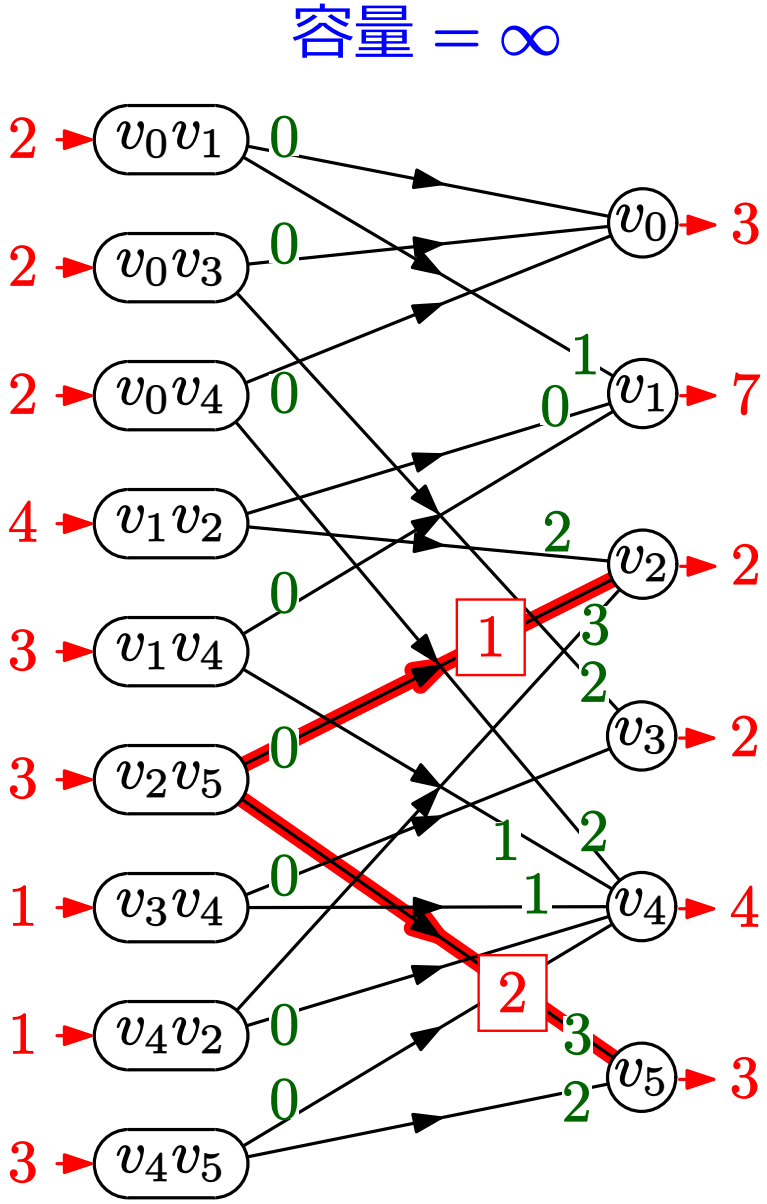
容量 $= \infty$



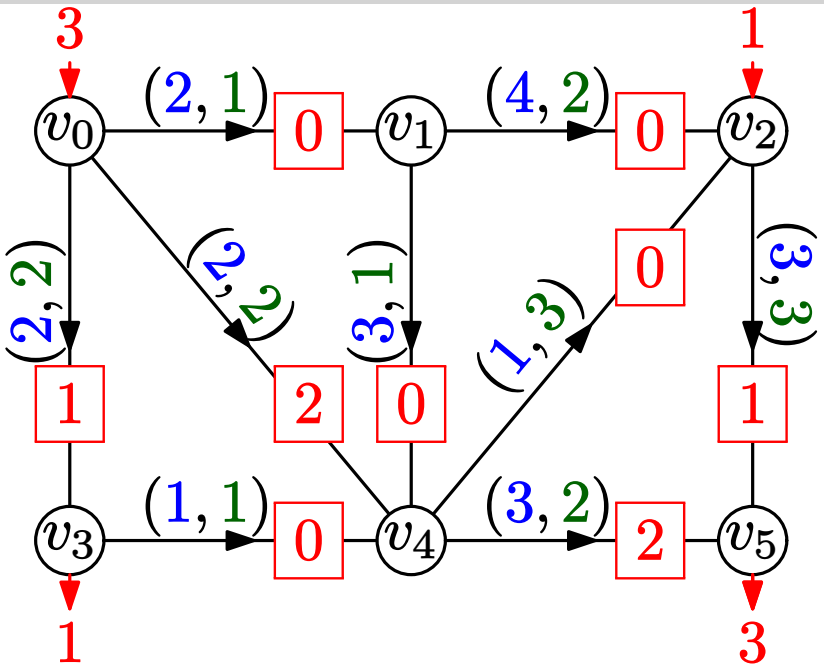
容量 ∞ の場合に変換：例

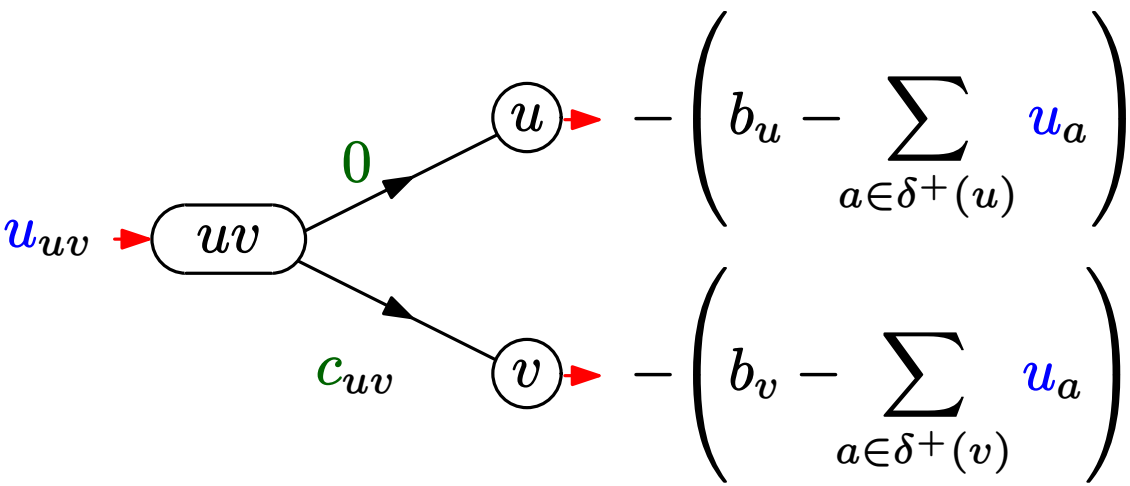


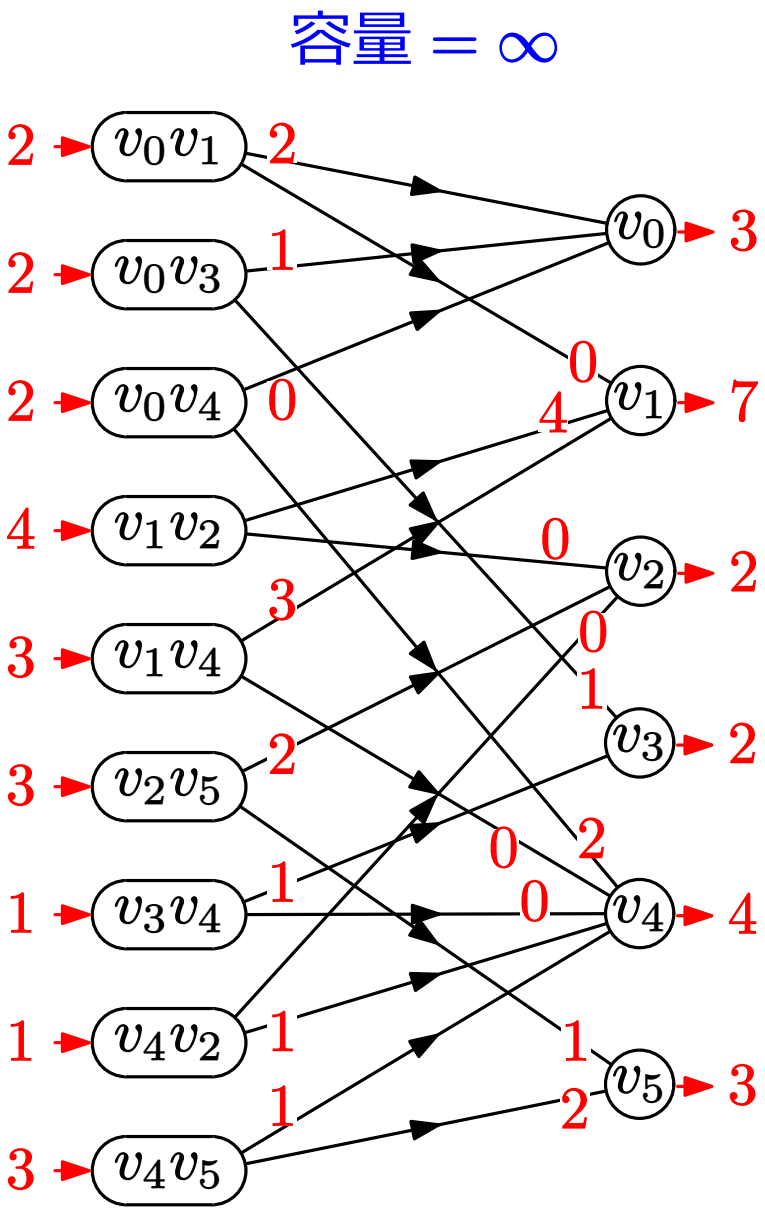




容量 ∞ の場合に変換：例







容量 ∞ の問題に変換 (1/2)

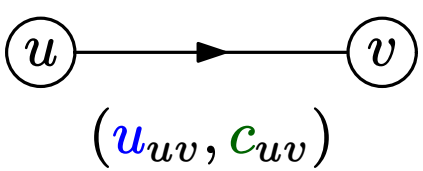
$(G=(V,A),u,c),\ b$ ▶▶▶ $(G'=(A\cup V,E),u',c'),\ b'$

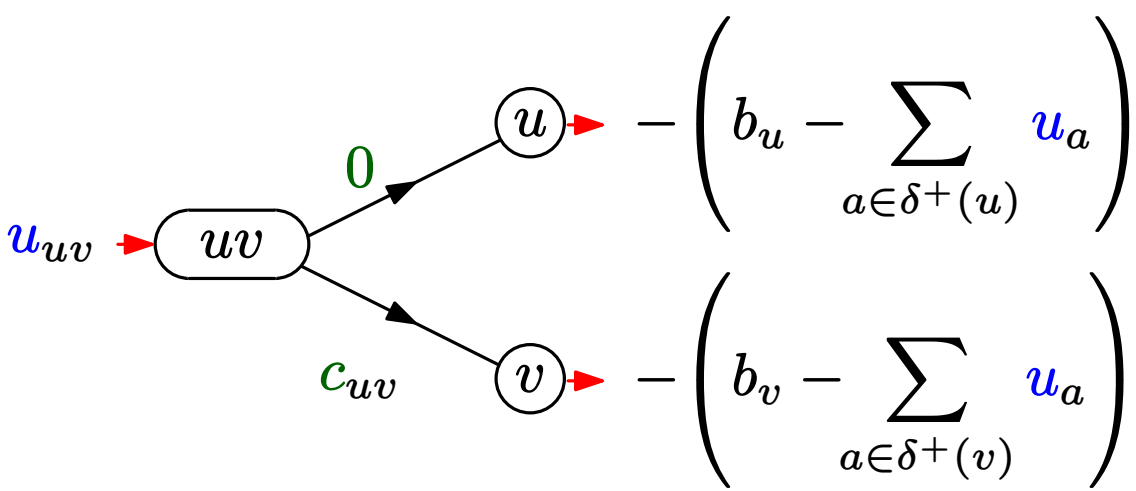
$$E=\{(a,v)\mid v\in V, a\in\delta^+(v)\cup\delta^-(v)\}$$

$$u'(a,v)=\infty$$

$$c'(a,v)=\begin{cases}0 & (a\in\delta^+(v)),\\ c(a) & (a\in\delta^-(v))\end{cases}$$

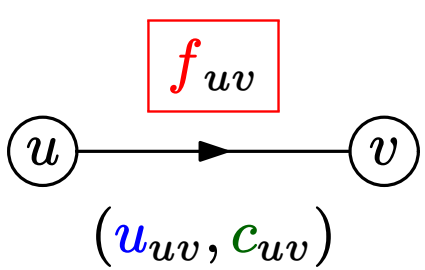
$$b'(a)=u(a),\ b'(v)=b(v)-\sum_{a\in\delta^+(v)}u(a)$$

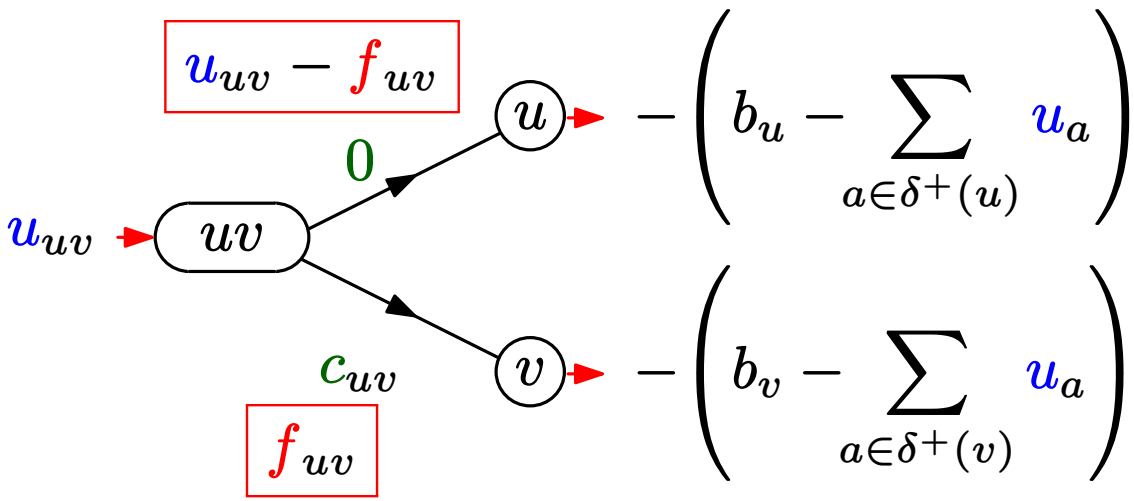




容量 ∞ の問題に変換 (2/2)

$(G=(V,A),u,c),\ b$ ▶▶▶ $(G'=(A\cup V,E),u',c'),\ b'$





容量 ∞ の問題に変換 (2/2)

$(G=(V,A),u,c),\ b$ ▶▶▶ $(G'=(A\cup V,E),u',c'),\ b'$

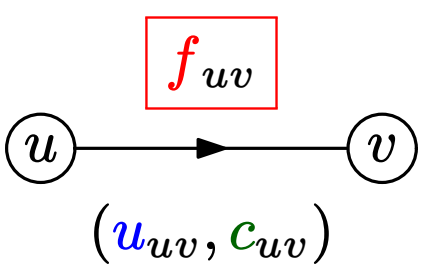
f : b-流

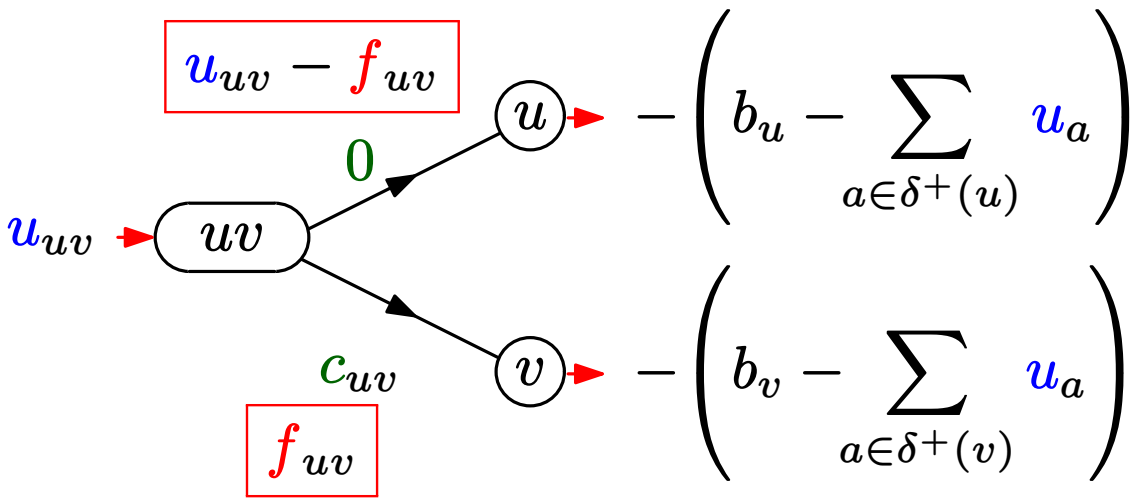
$$f'^{+}(v)-f'^{-}(v)=-\sum_{a\in\delta^{+}(v)}f'(a)-\sum_{a\in\delta^{-}(v)}f'(a)$$

$$=-\sum_{a\in\delta^{+}(v)}(u(a)-f(a))-\sum_{a\in\delta^{-}(v)}f(a)$$

$$=f^{+}(v)-f^{-}(v)-\sum_{a\in\delta^{+}(v)}u(a)$$

$$=b(v)-\sum_{a\in\delta^{+}(v)}u(a)=b'(v)$$





容量 ∞ の問題に変換：まとめ

性質：容量 ∞ の問題への変換

最小費用流問題において
頂点数 n，弧数 m の入力は
頂点数 $n+m$，弧数 $2m$，容量が∞である同値の入力に
$O(m+n)$ 時間で変換できる

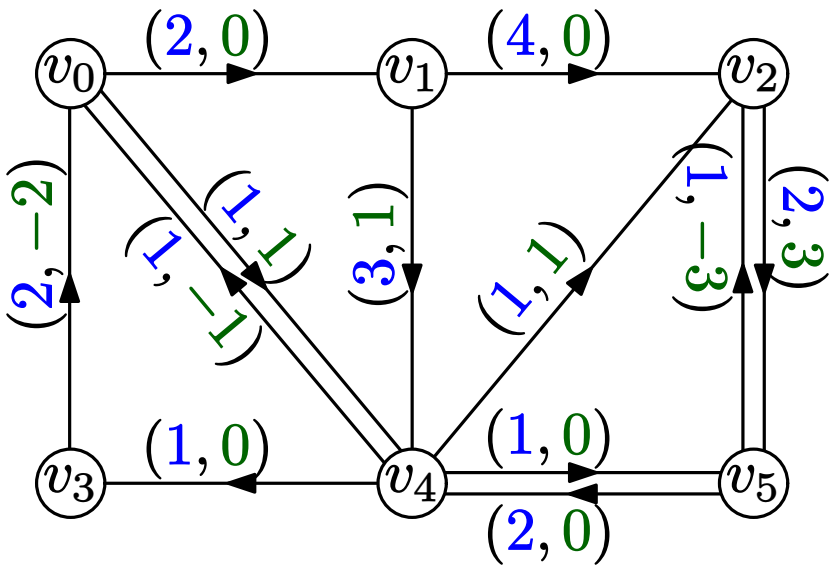
注：∞ = とてつもなく大きな整数

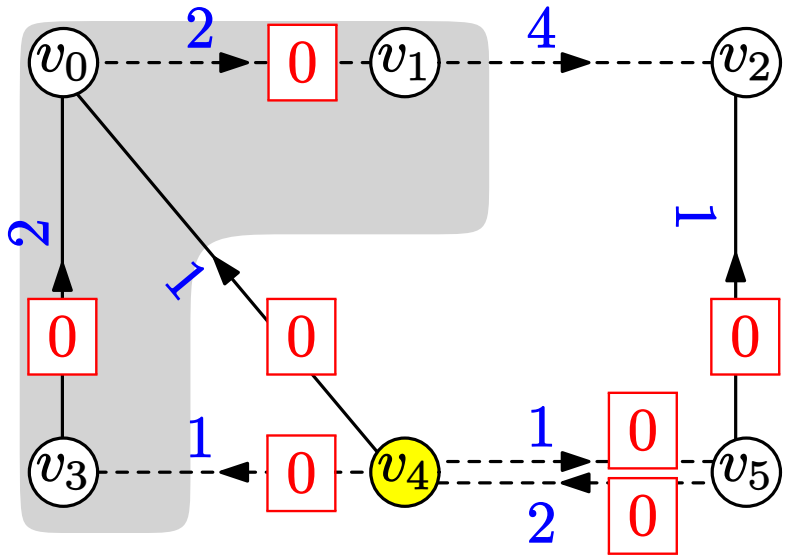
特に，$\max_{v \in V} |b'(v)| \leq \max_{v \in V} |b(v)| + m \max_{a \in A} u(a) = B + mU$

容量スケーリング法 を 容量 ∞ の入力に対して適用する

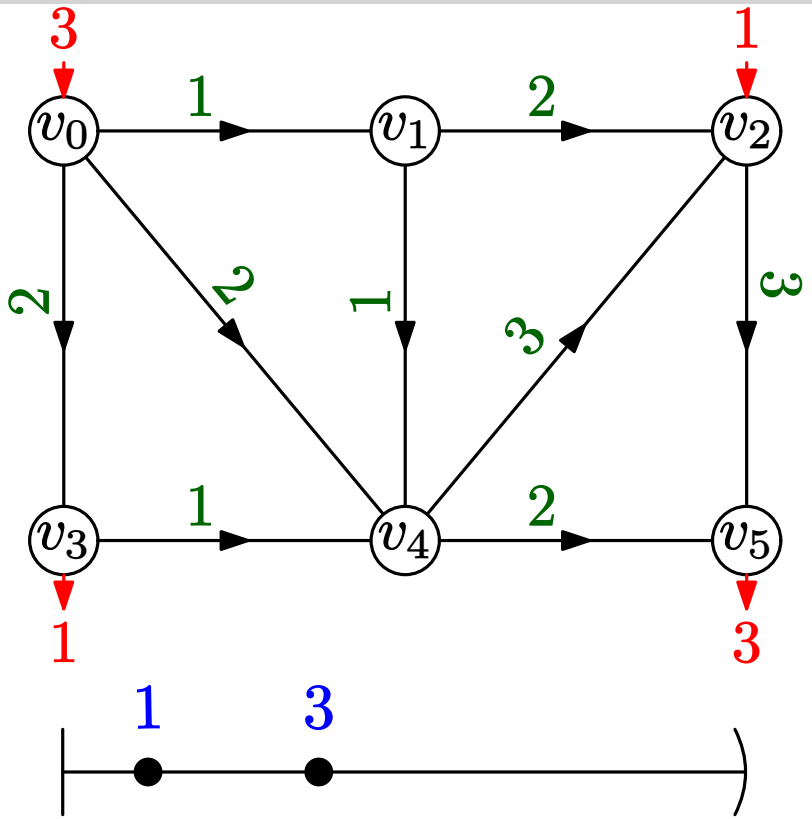
本日の内容

1. 容量スケーリング法：準備
2. **容量スケーリング法：概要**
3. 容量スケーリング法：計算量
4. 主双対法

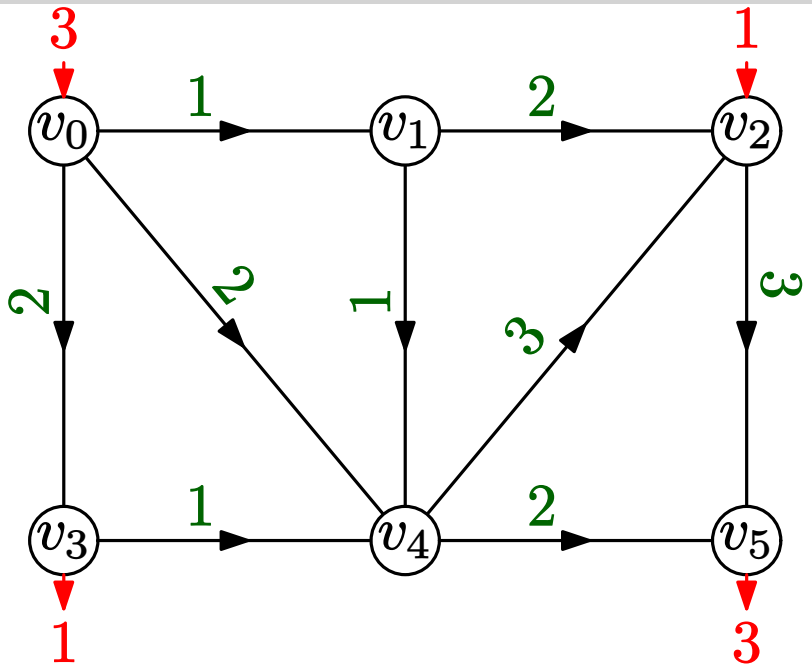


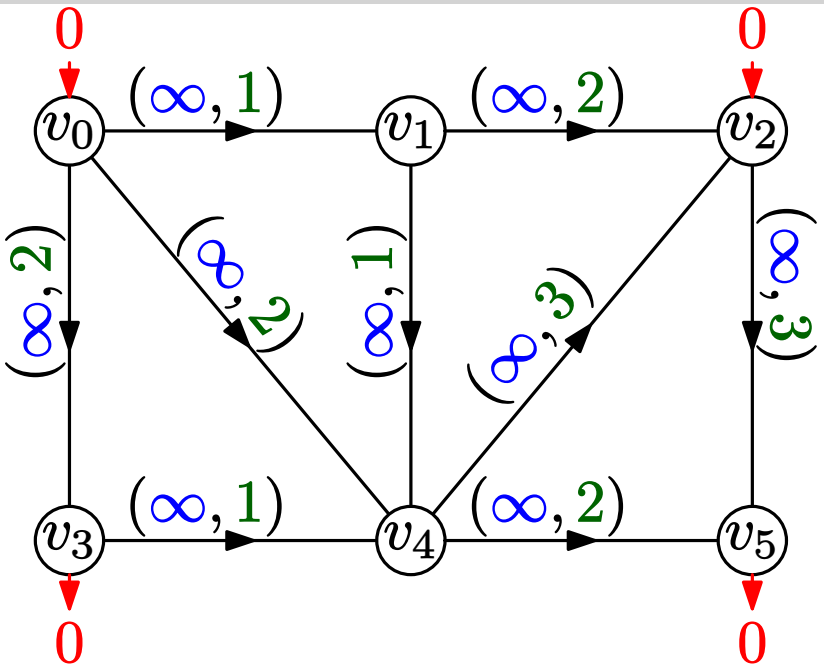


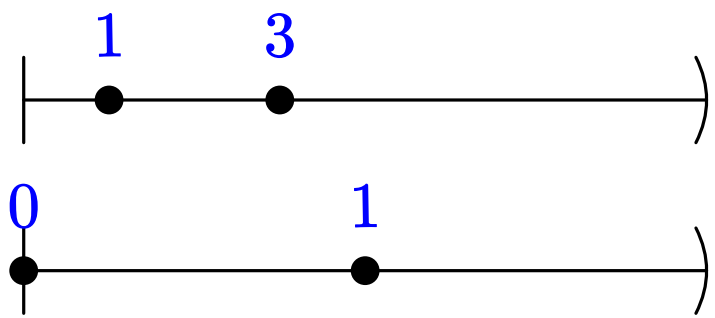
容量スケーリング法：例 (1/5)



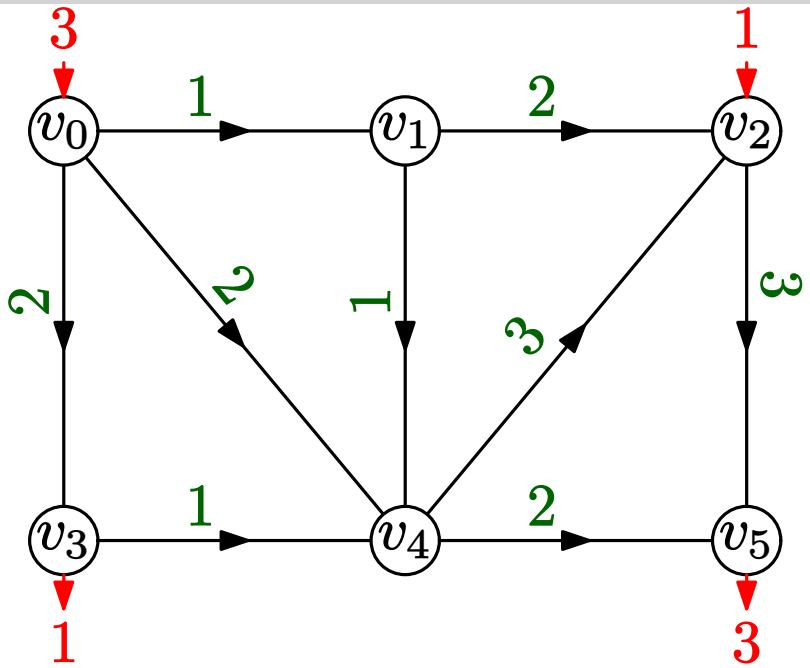
容量スケーリング法：例 (1/5)

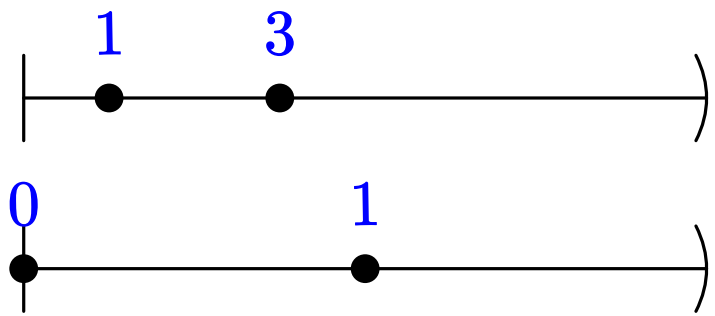


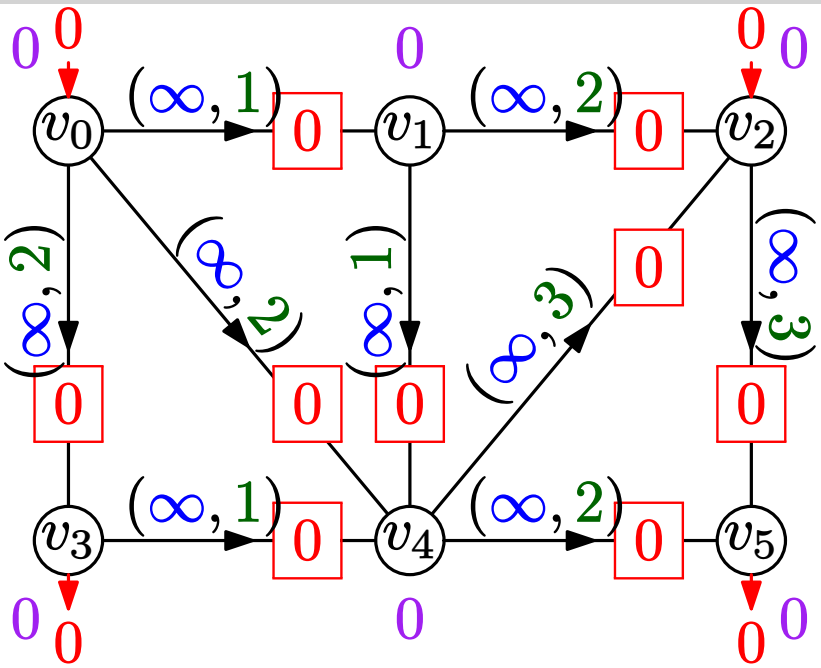




容量スケーリング法：例 (1/5)

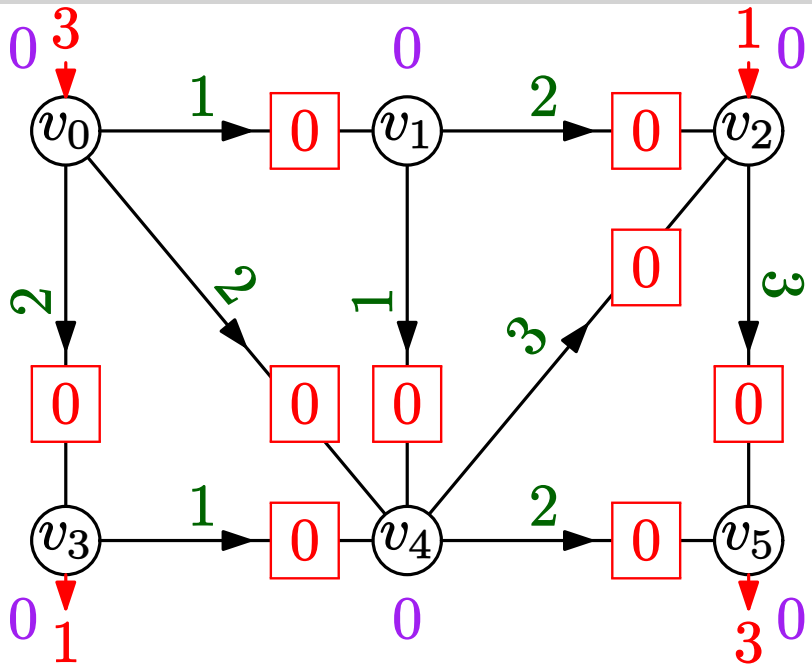


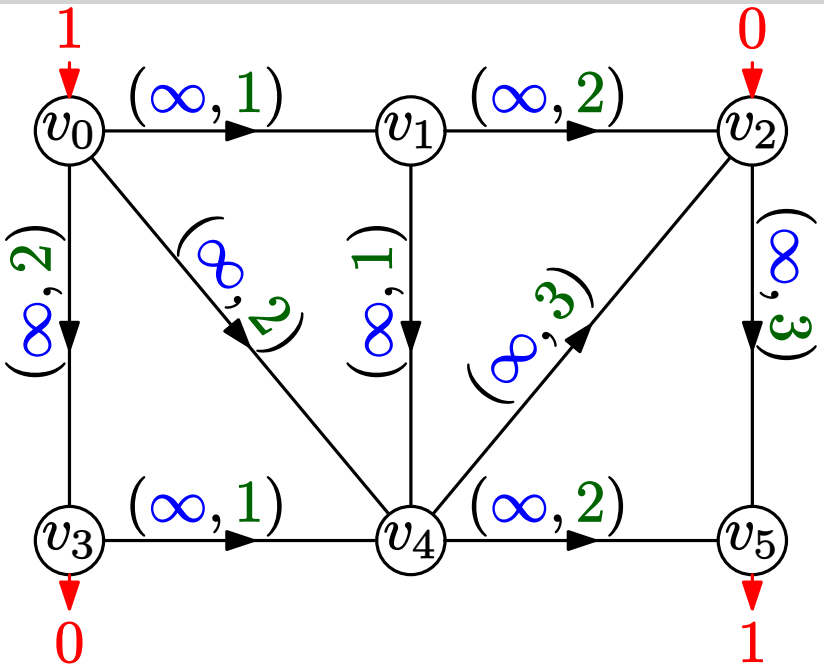


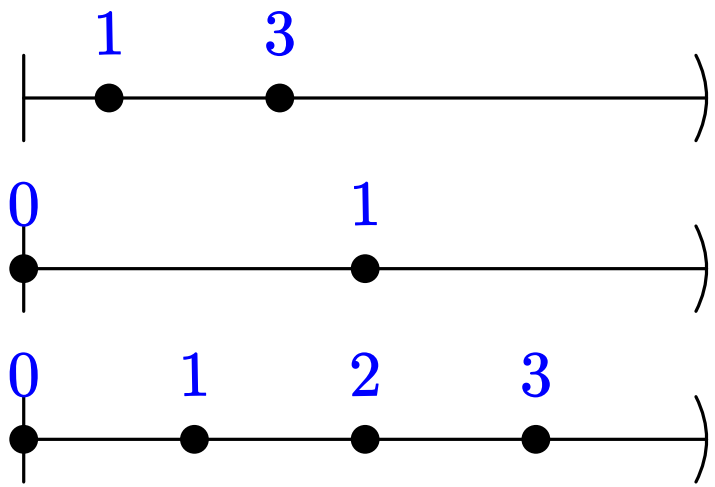


最短路の計算

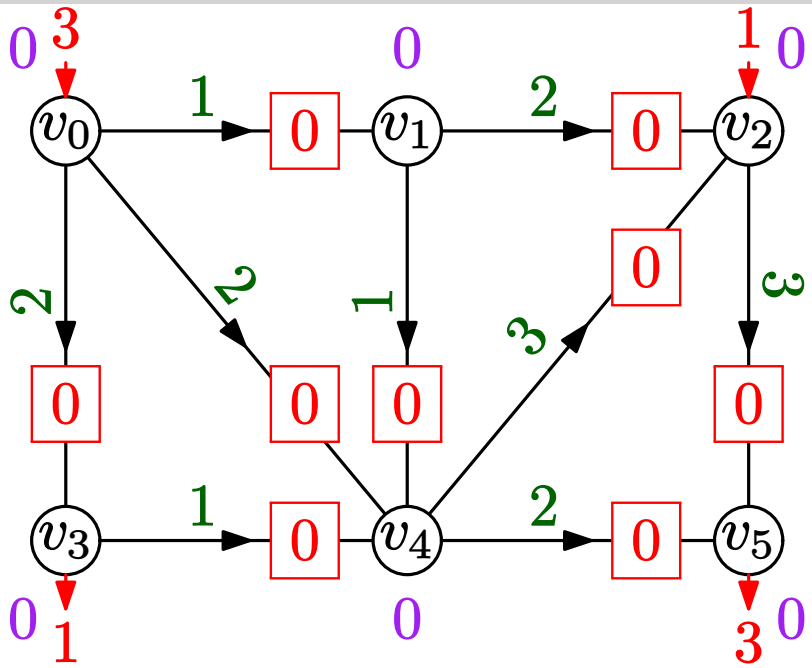
容量スケーリング法：例 (2/5)

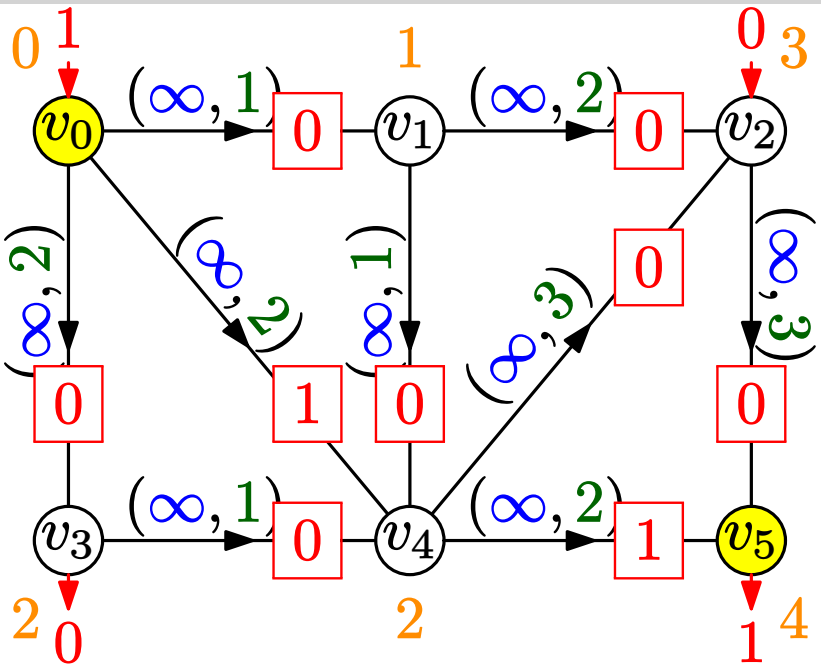




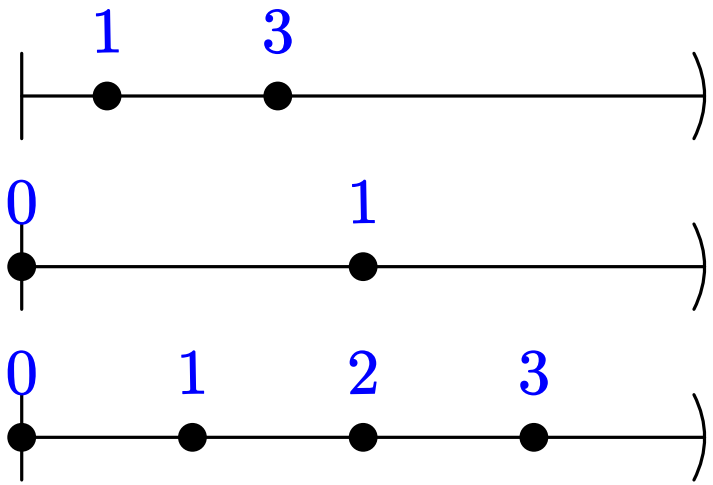


容量スケーリング法：例 (2/5)

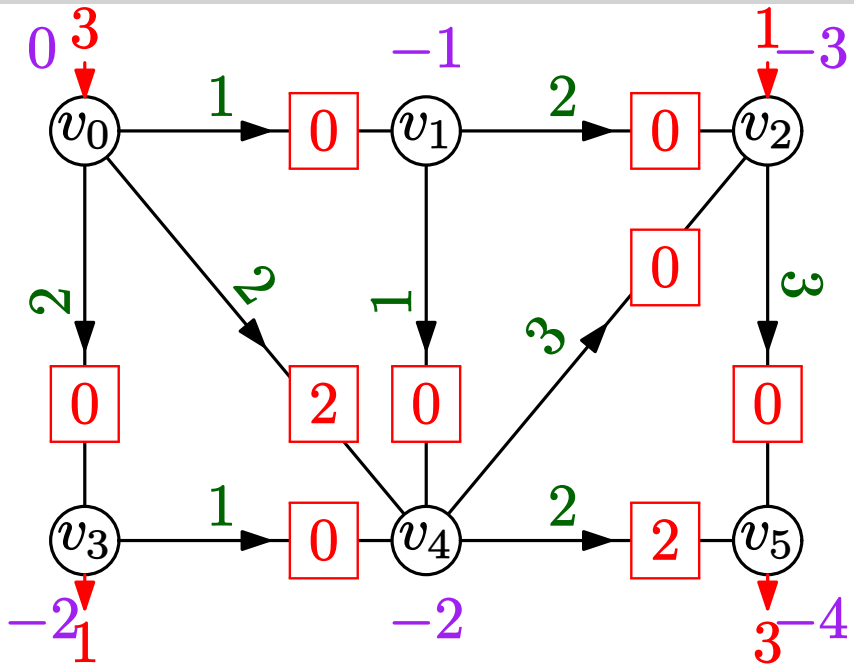


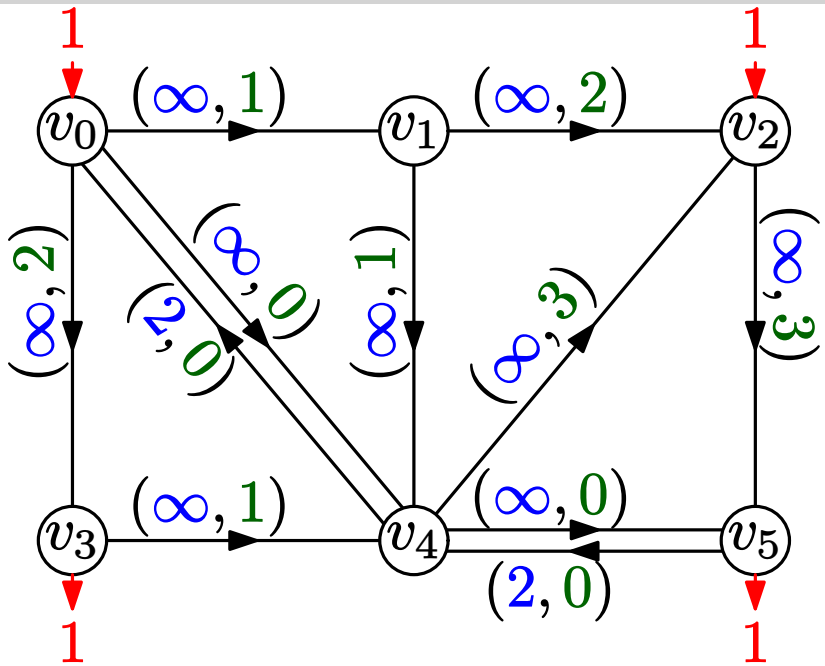


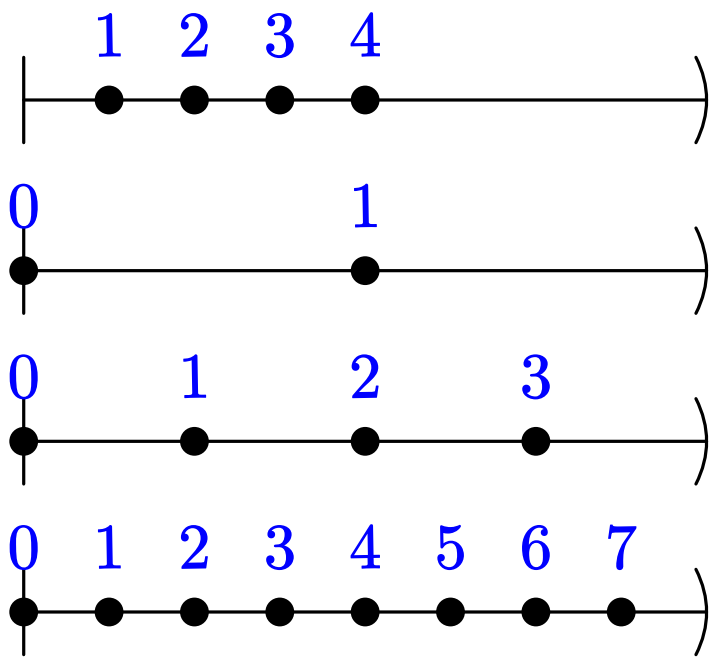
最短路の計算



容量スケーリング法：例 (3/5)

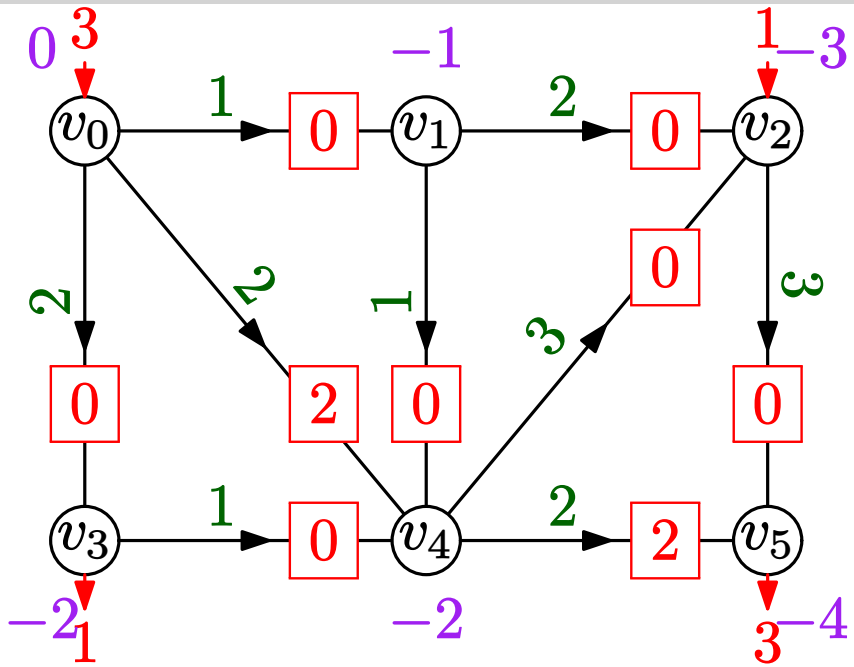


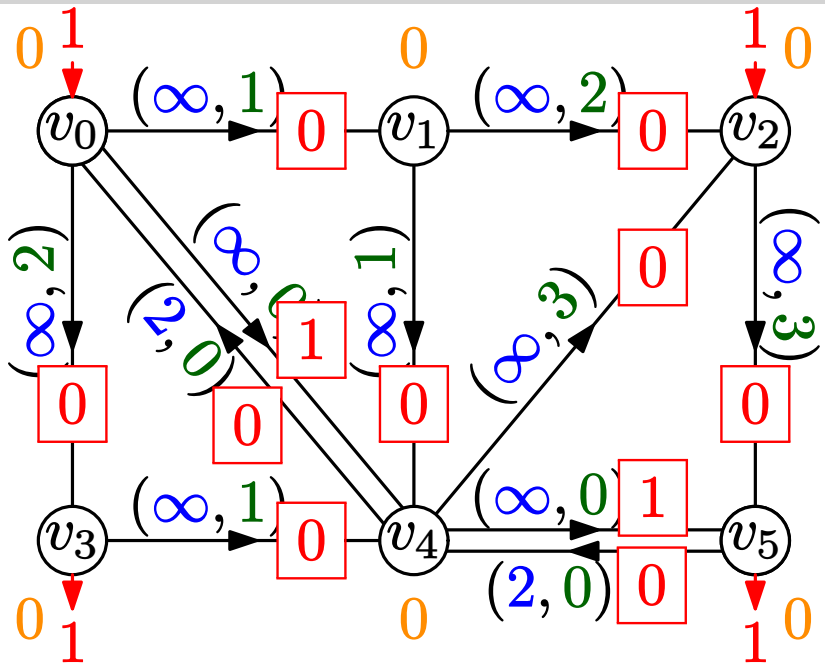




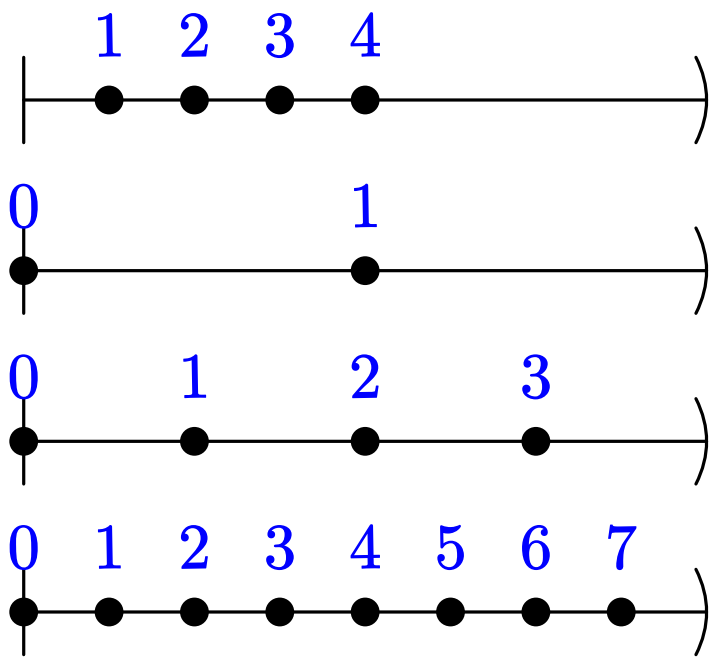
満たされている需要・供給 = 2

容量スケーリング法：例 (3/5)



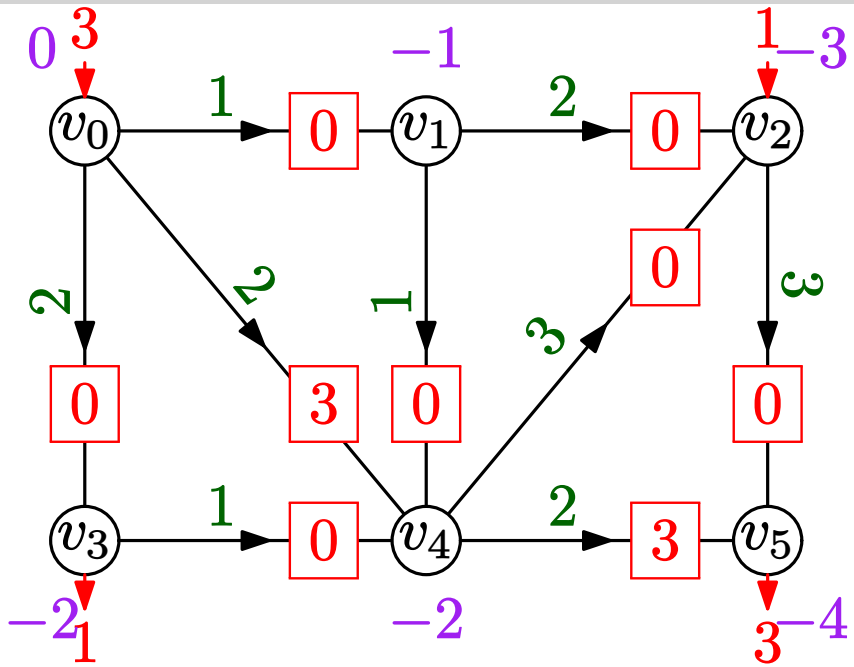


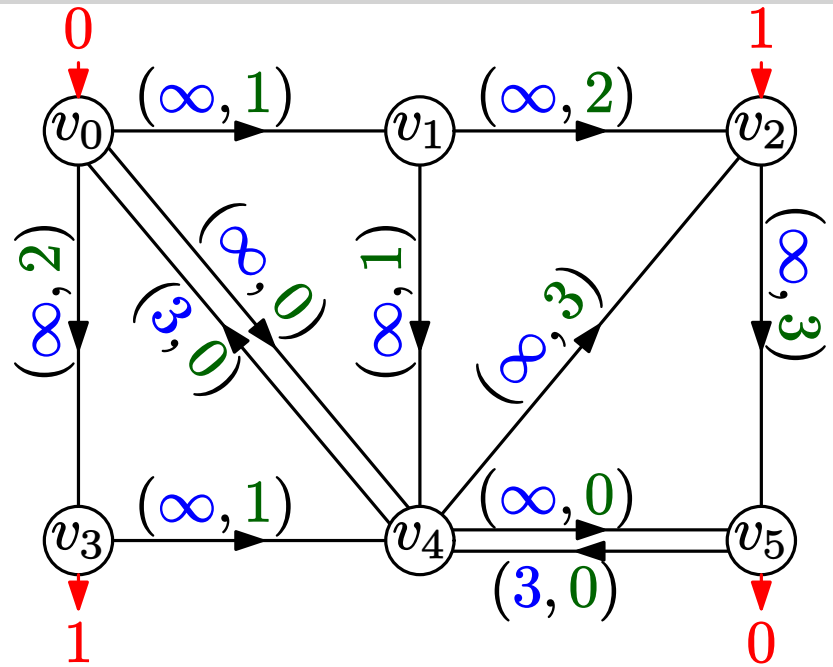
最短路の計算

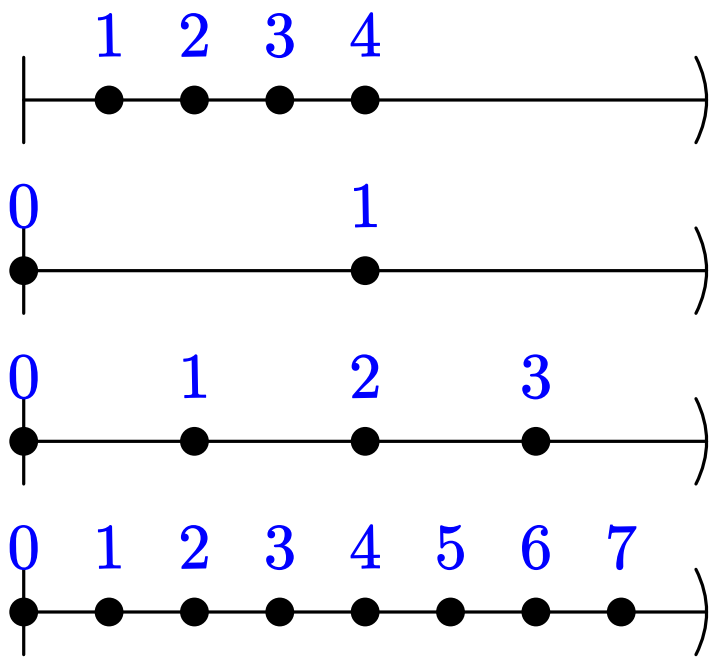


満たされている需要・供給 = 2

容量スケーリング法：例 (4/5)

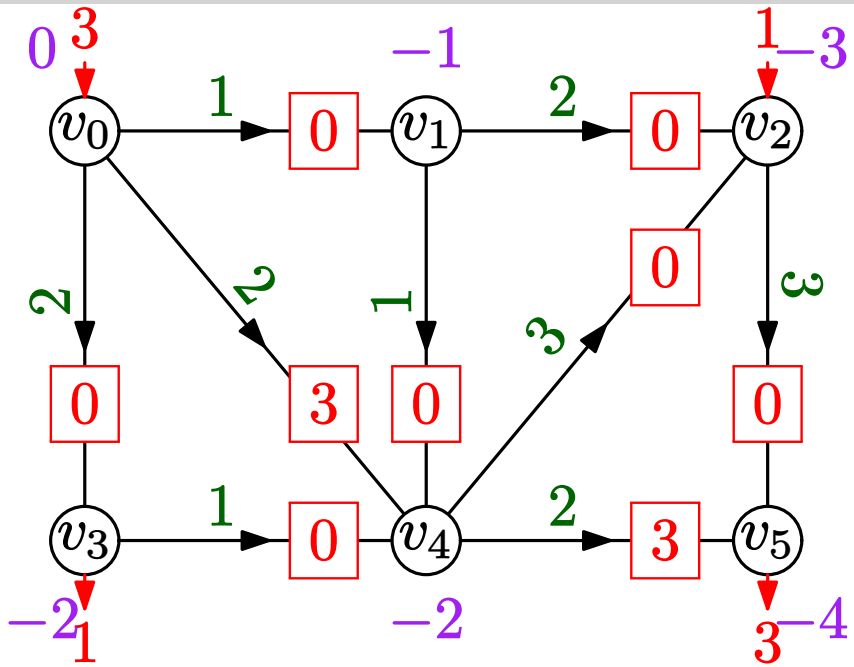


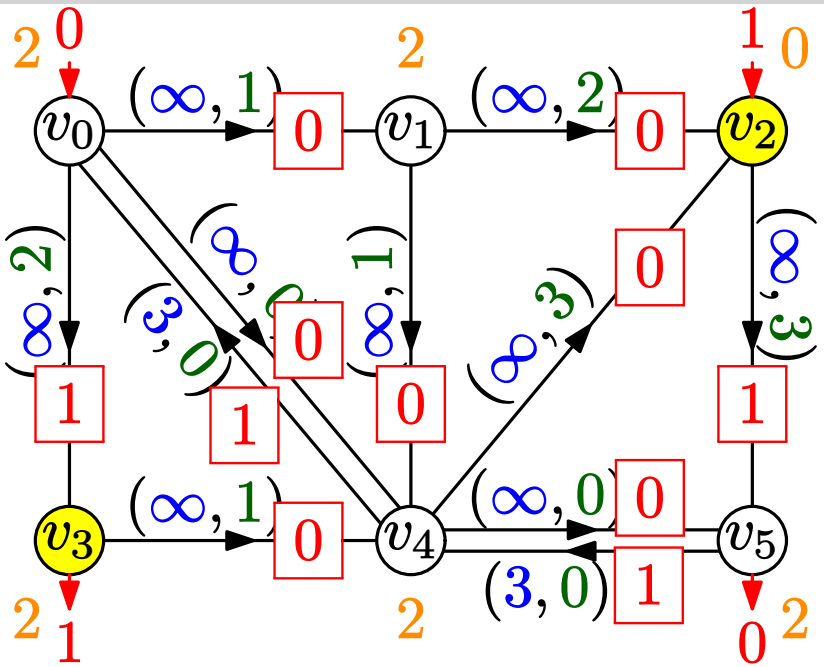


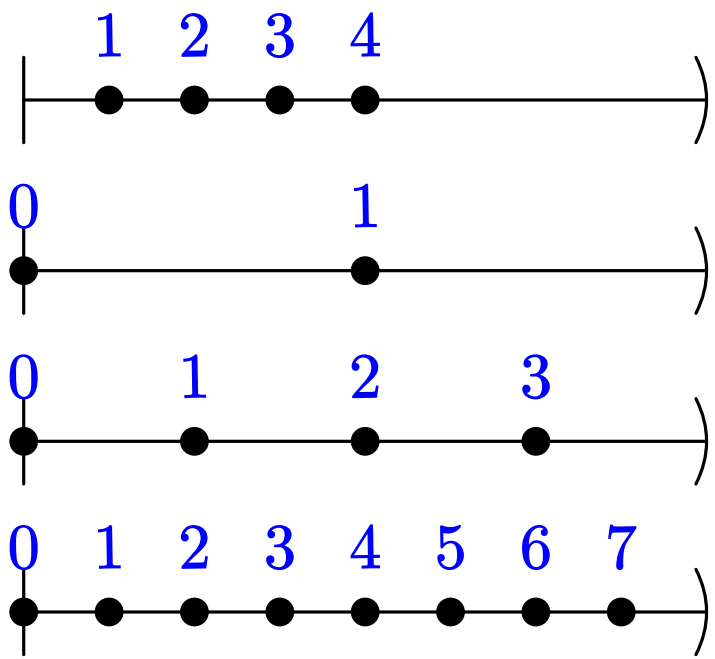


満たされている需要・供給 = 3

容量スケーリング法：例 (4/5)

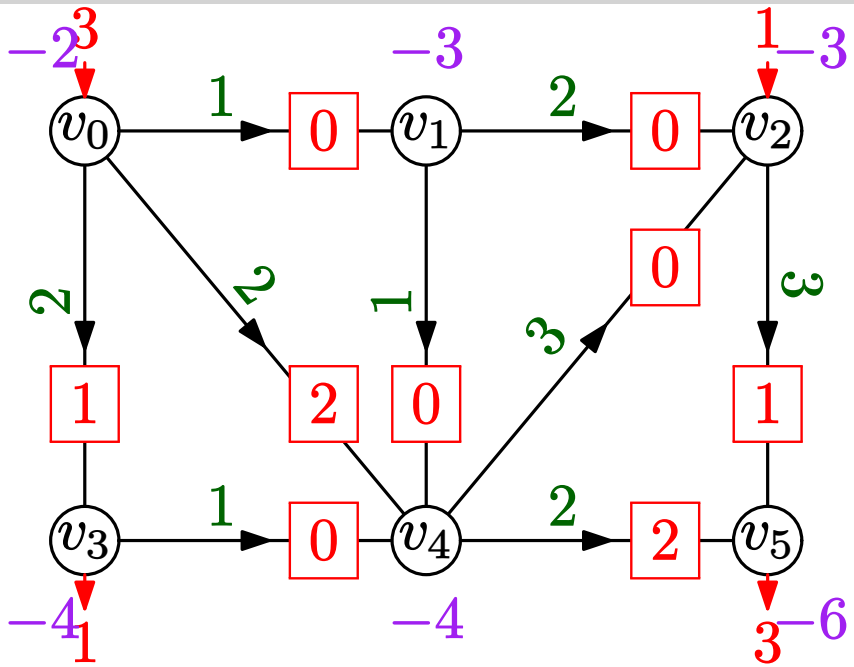


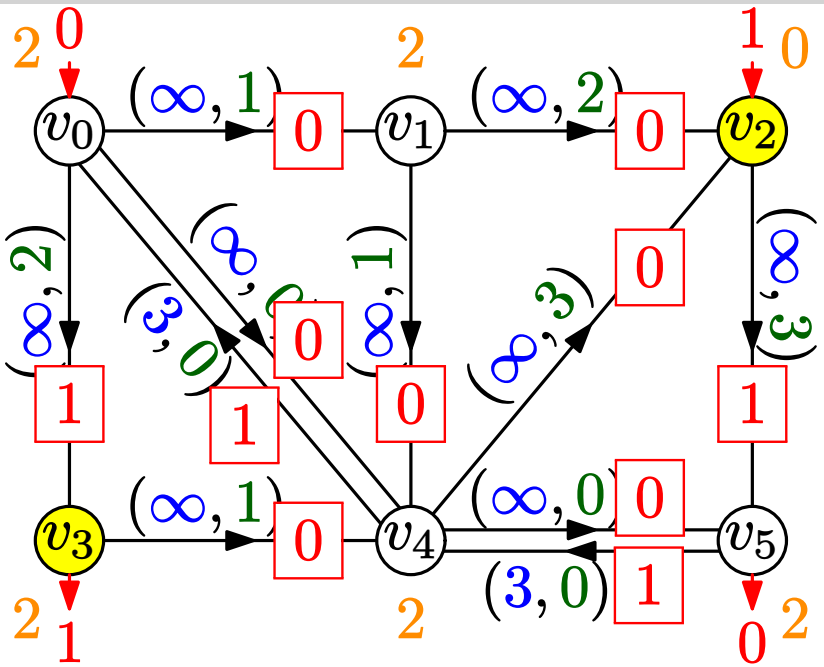


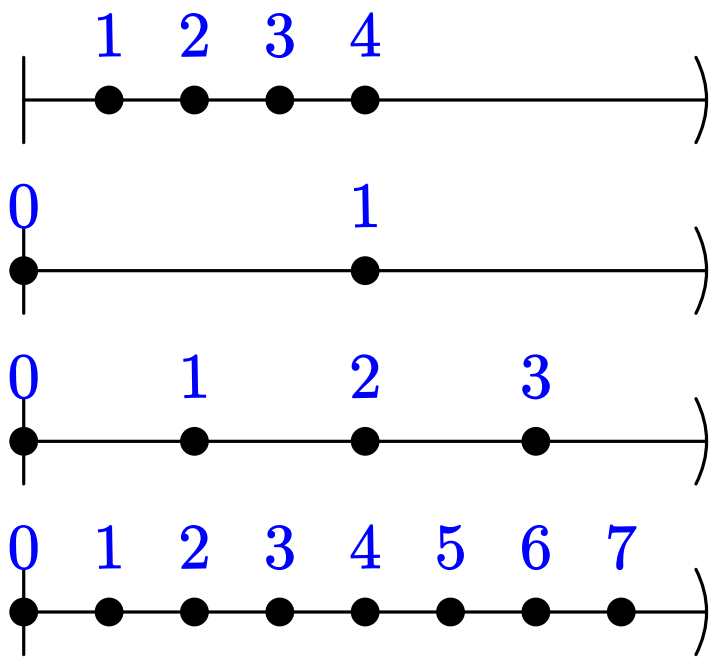


満たされている需要・供給 = 3

容量スケーリング法：例 (4/5)

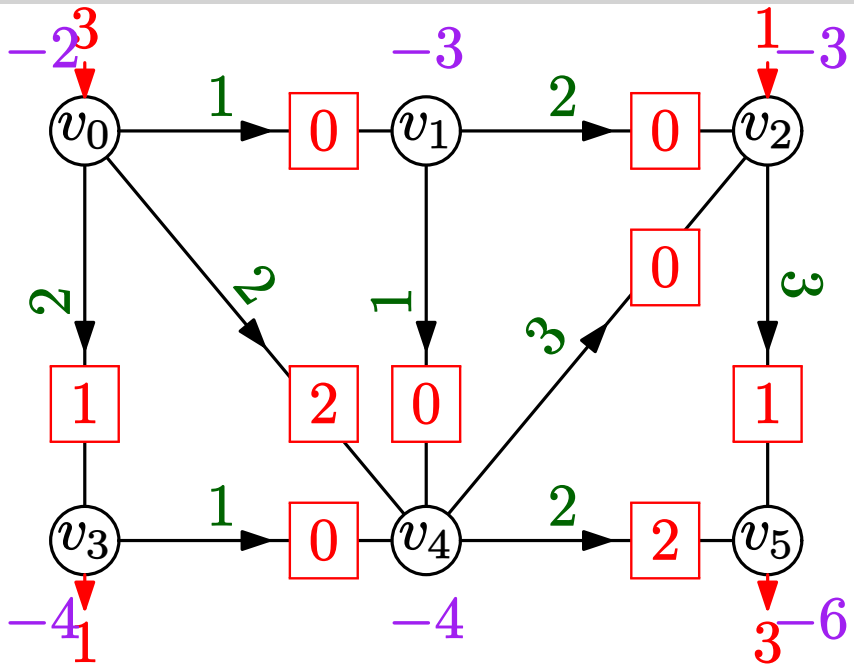


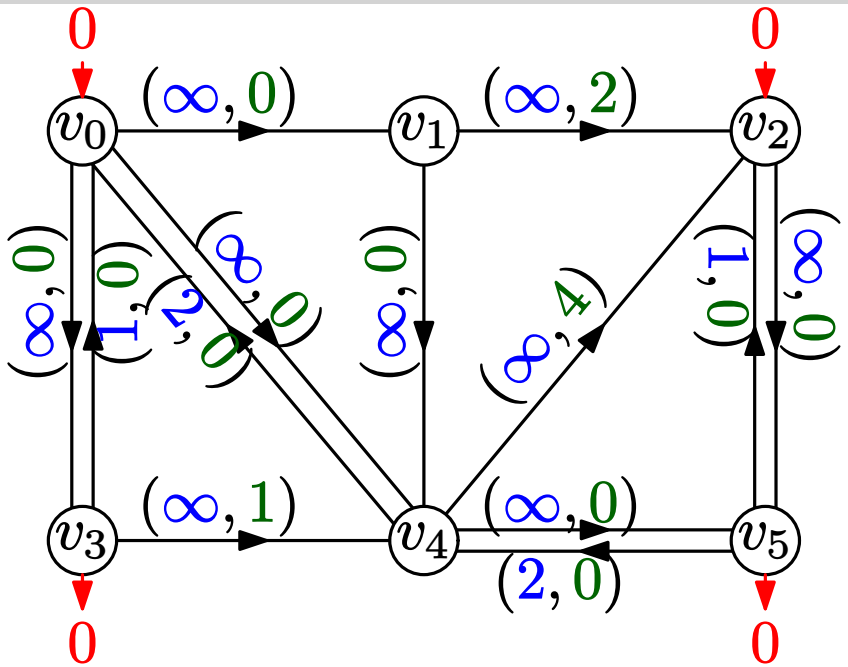


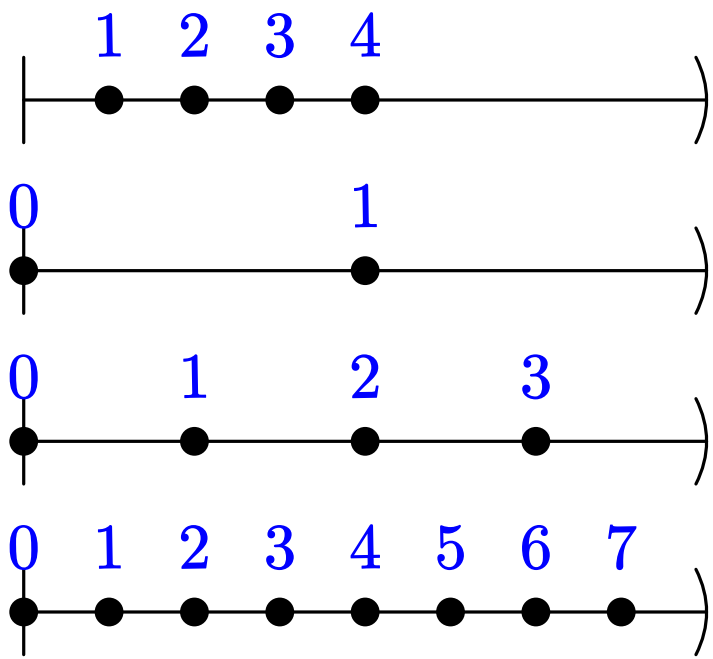


満たされている需要・供給 = 3

容量スケーリング法：例 (5/5)







満たされている需要・供給 $= 4$

最小費用 b-流！

容量スケーリング法

アルゴリズム：容量スケーリング法

- 初期化：$f_0 = 0$, $p_0 = 0$, $K = \lfloor 1 + \log_2 B \rfloor$
- 反復：$k = 1, 2, \ldots, K$ に対して順に以下を実行
  1. $b_k(v) = \mathrm{sgn}(b(v)) \lfloor |b(v)|/2^{K-k} \rfloor$ $(\forall\, v \in V)$ とする
  2. $2f_{k-1}$ を初期流，p_{k-1} を初期ポテンシャルとする
  3. 供給/需要が残る頂点 u, v のある限り，以下を実行
     (a) 補助ネットワーク (費用は簡約費用) で
     最短 u-v 路を見つける
     (b) 最短路に沿って流 f_k とポテンシャル p_k の更新
- 出力：f_K

補足：$B = \max_{v \in V} |b(v)|$

容量スケーリング法

アルゴリズム：容量スケーリング法

- 初期化：$f_0 = 0$, $p_0 = 0$, $K = \lfloor 1 + \log_2 B \rfloor$
- 反復：$k = 1, 2, \ldots, K$ に対して順に以下を実行
  1. $b_k(v) = \mathrm{sgn}(b(v)) \lfloor |b(v)|/2^{K-k} \rfloor$ $(\forall\, v \in V)$ とする
  2. $2f_{k-1}$ を初期流，p_{k-1} を初期ポテンシャルとする
  3. 供給/需要が残る頂点 u, v のある限り，以下を実行
     (a) 補助ネットワーク (費用は簡約費用) で
     最短 u-v 路を見つける
     (b) 最短路に沿って流 f_k とポテンシャル p_k の更新
- 出力：f_K

逐次最短路法の動き

補足：$B = \max_{v \in V} |b(v)|$

容量スケーリング法：正当性

性質：容量スケーリング法の正当性

容量スケーリング法が停止する ⇒
出力 f_K はネットワーク (G, c) の最小費用 b-流である

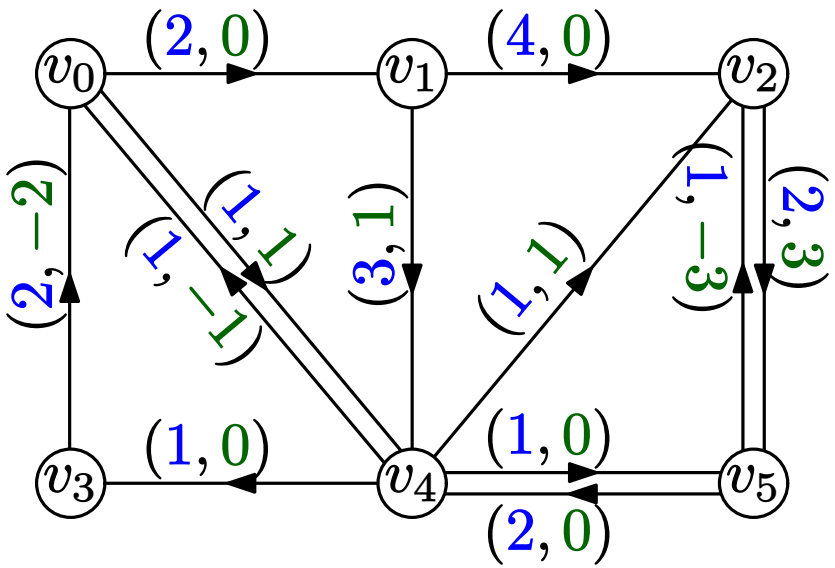
証明：$b_K(v) = \mathrm{sgn}(b(v)) \lfloor |b(v)|/2^{K-K} \rfloor = b(v)$

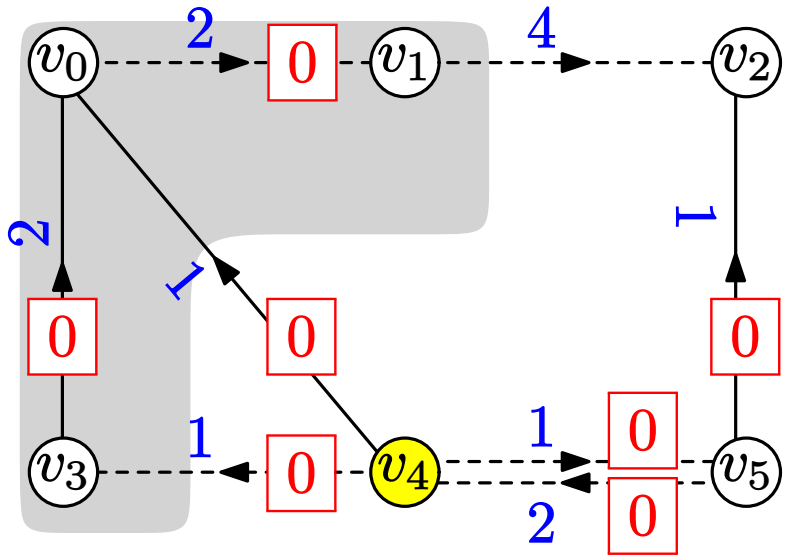
- $\therefore f_K$ は (G, c) の最小費用 b-流

□

本日の内容

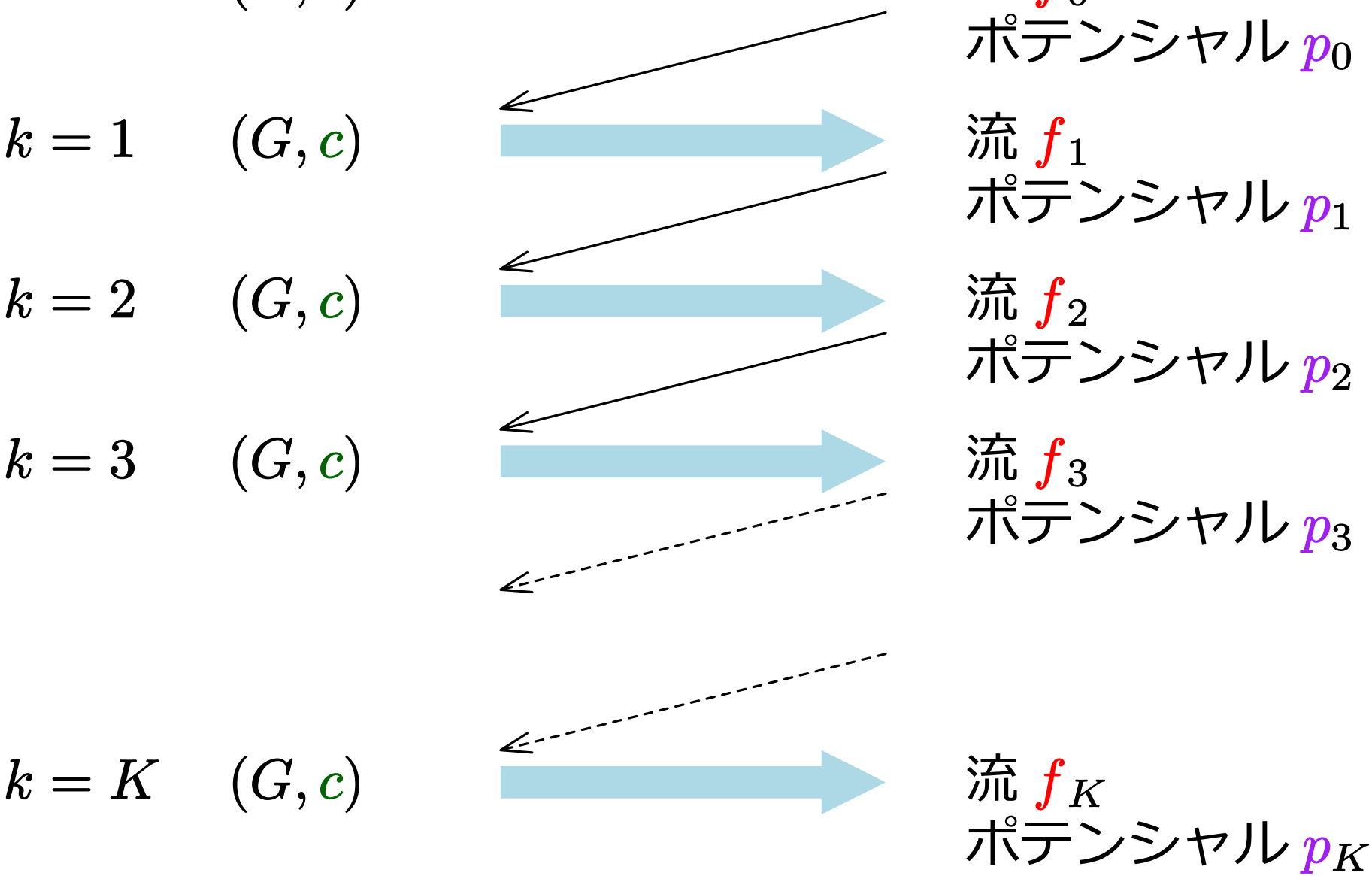
1. 容量スケーリング法：準備
2. 容量スケーリング法：概要
3. **容量スケーリング法：計算量**
4. 主双対法





容量スケーリング法：計算量に向けて

逐次最短路法



容量スケーリング法：確認事項

$k = i$ (G, c) ⇒ 流 f_i，ポテンシャル p_i

$k = i+1$ (G, c) ⇒ 流 f_{i+1}，ポテンシャル p_{i+1}

確認事項

1. p_i が (G, c) と $2f_i$ に対して，
   簡約費用最適性条件の不等式を満たす
2. 第 i 反復終了時に満たされる
   (スケーリング後の) 供給・需要 $s_i \geq \frac{1}{2^{K-i+1}} \sum_{v \in V} |b(v)| - n$

確認事項 1

確認事項

1. p_i が (G, c) と $2f_i$ に対して，
   簡約費用最適性条件の不等式を満たす

$\overset{\text{定義}}{\iff}$ 任意の $uv \in A_{2f_i}$ に対して，$(c_{2f_i})_{uv} - p_u + p_v \geq 0$

確認事項 1

確認事項

1. p_i が (G, c) と $2f_i$ に対して，
   簡約費用最適性条件の不等式を満たす

$\overset{\text{定義}}{\iff}$ 任意の $uv \in A_{2f_i}$ に対して，$(c_{2f_i})_{uv} - p_u + p_v \geq 0$

使える性質：p_i は (G, c) と f_i に対して，
簡約費用最適性条件の不等式を満たす

$\overset{\text{定義}}{\iff}$ 任意の $uv \in A_{f_i}$ に対して，$(c_{f_i})_{uv} - p_u + p_v \geq 0$

確認事項 1

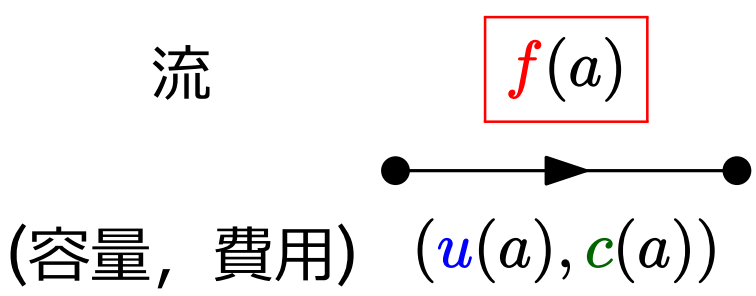
確認事項

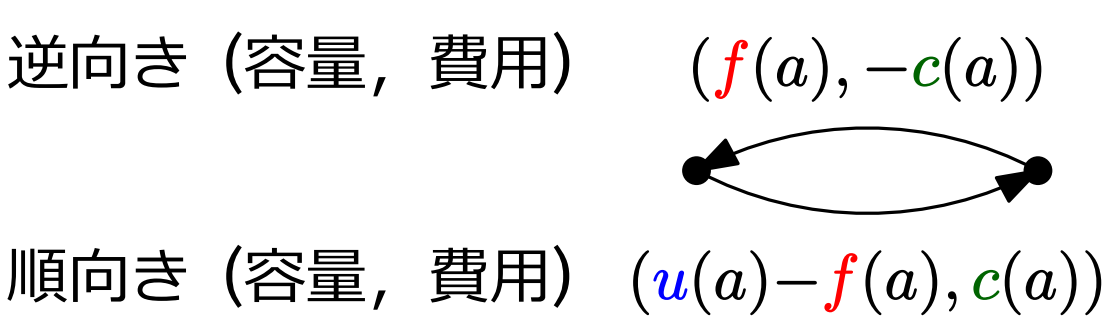
1. p_i が (G, c) と $2f_i$ に対して，
   簡約費用最適性条件の不等式を満たす

$\overset{\text{定義}}{\iff}$ 任意の $uv \in A_{2f_i}$ に対して，$(c_{2f_i})_{uv} - p_u + p_v \geq 0$

使える性質：p_i は (G, c) と f_i に対して，
簡約費用最適性条件の不等式を満たす

$\overset{\text{定義}}{\iff}$ 任意の $uv \in A_{f_i}$ に対して，$(c_{f_i})_{uv} - p_u + p_v \geq 0$





確認事項 1

確認事項

1. p_i が (G, c) と $2f_i$ に対して，
   簡約費用最適性条件の不等式を満たす

$\overset{\text{定義}}{\iff}$ 任意の $uv \in A_{2f_i}$ に対して，$(c_{2f_i})_{uv} - p_u + p_v \geq 0$

使える性質：p_i は (G, c) と f_i に対して，
簡約費用最適性条件の不等式を満たす

$\overset{\text{定義}}{\iff}$ 任意の $uv \in A_{f_i}$ に対して，$(c_{f_i})_{uv} - p_u + p_v \geq 0$

証明：容量が ∞ であるため，$A_{f_i} = A_{2f_i}$ であり，
かつ，任意の $uv \in A_{f_i}$ に対して，$(c_{f_i})_{uv} = (c_{2f_i})_{uv}$

$\therefore$ 任意の $uv \in A_{2f_i}$ に対して，$(c_{2f_i})_{uv} - p_u + p_v \geq 0$ □

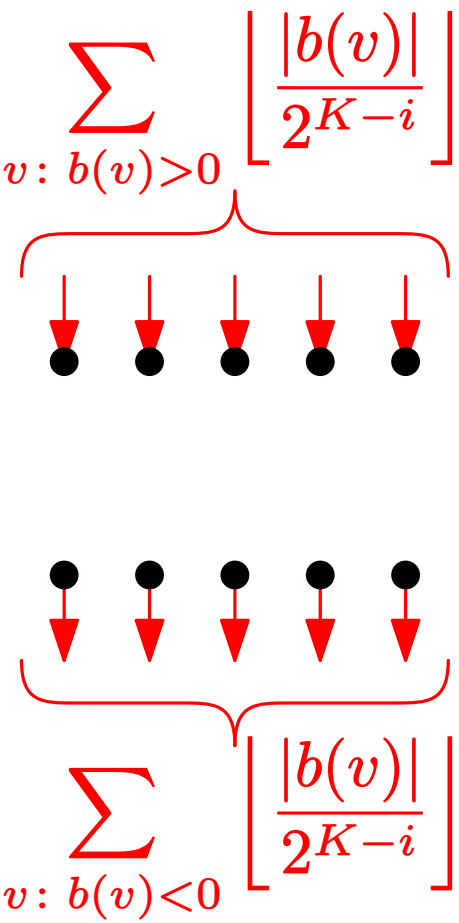
確認事項 2

確認事項

2. 第 i 反復終了時に満たされる
   (スケーリング後の) 供給・需要 $s_i \geq \frac{1}{2^{K-i+1}} \sum_{v \in V} |b(v)| - n$

証明：

$$s_i = \min \left\{ \sum_{v:\ b(v)>0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor, \sum_{v:\ b(v)<0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor \right\}$$



確認事項 2

確認事項

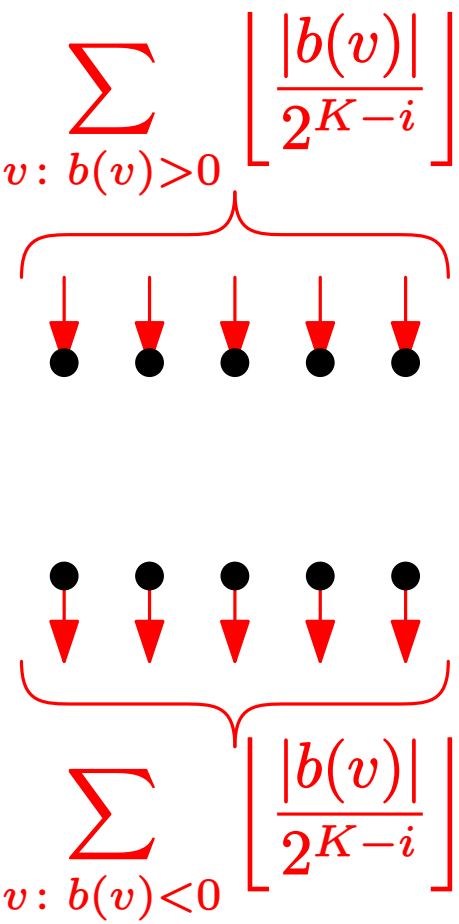
2. 第 i 反復終了時に満たされる
   (スケーリング後の) 供給・需要 $s_i \geq \frac{1}{2^{K-i+1}} \sum_{v \in V} |b(v)| - n$

証明：

$$s_i = \min \left\{ \sum_{v:\ b(v)>0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor, \sum_{v:\ b(v)<0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor \right\}$$

$$\sum_{v:\ b(v)>0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor \geq \sum_{v:\ b(v)>0} \left(\frac{|b(v)|}{2^{K-i}} - 1 \right)$$

$$\geq \frac{1}{2^{K-i}} \sum_{v:\ b(v)>0} |b(v)| - n$$



確認事項 2

確認事項

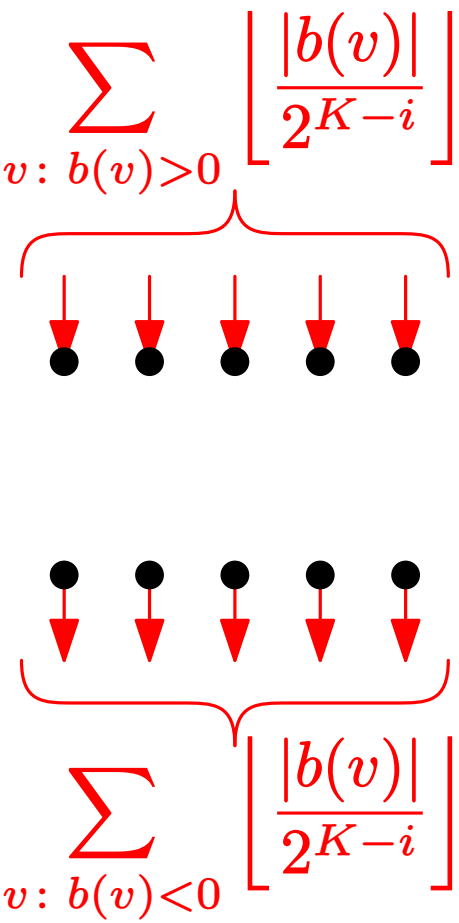
2. 第 i 反復終了時に満たされる
   (スケーリング後の) 供給・需要 $s_i \geq \frac{1}{2^{K-i+1}} \sum_{v \in V} |b(v)| - n$

証明：

$$s_i = \min \left\{ \sum_{v:\ b(v)>0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor, \sum_{v:\ b(v)<0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor \right\}$$

$$\sum_{v:\ b(v)<0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor \geq \sum_{v:\ b(v)<0} \left(\frac{|b(v)|}{2^{K-i}} - 1 \right)$$

$$\geq \frac{1}{2^{K-i}} \sum_{v:\ b(v)<0} |b(v)| - n$$



確認事項 2

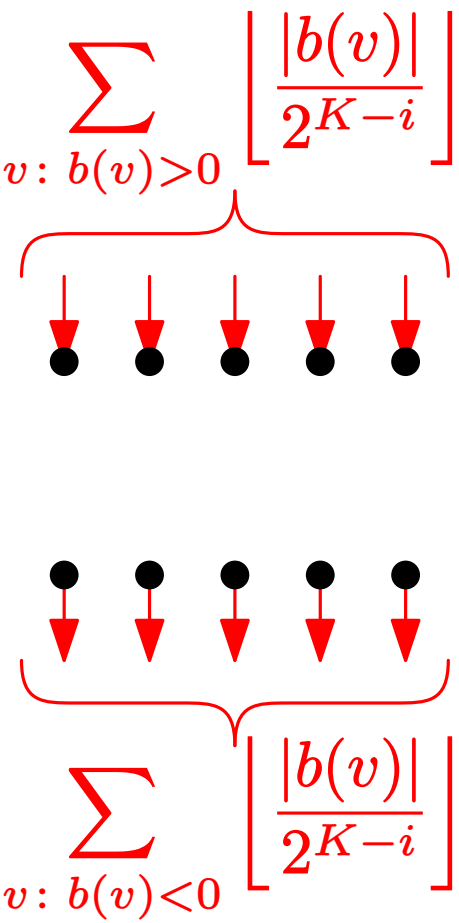
確認事項

2. 第 i 反復終了時に満たされる
   (スケーリング後の) 供給・需要 $s_i \geq \frac{1}{2^{K-i+1}} \sum_{v \in V} |b(v)| - n$

証明：

$$s_i = \min \left\{ \sum_{v:\ b(v)>0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor, \sum_{v:\ b(v)<0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor \right\}$$

$$\geq \frac{1}{2^{K-i}} \min \left\{ \sum_{v:\ b(v)>0} |b(v)|, \sum_{v:\ b(v)<0} |b(v)| \right\} - n$$



確認事項 2

確認事項

2. 第 i 反復終了時に満たされる
   (スケーリング後の) 供給・需要 $s_i \geq \dfrac{1}{2^{K-i+1}} \sum_{v \in V} |b(v)| - n$

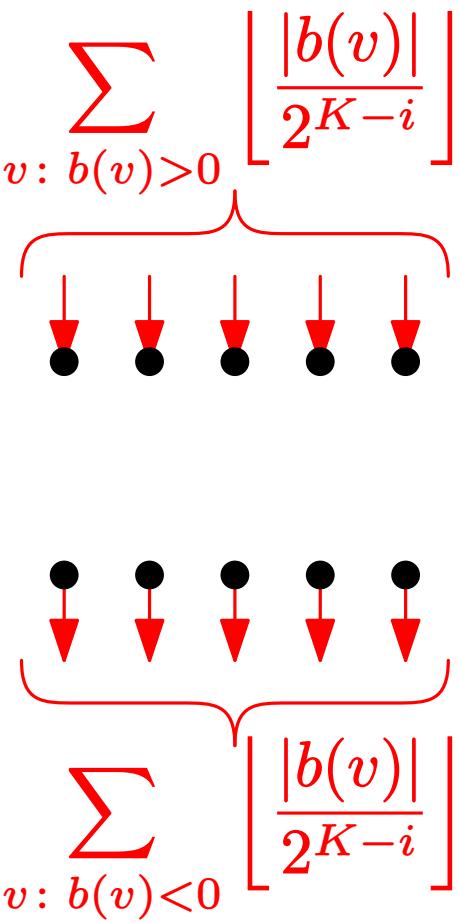
証明：

$$s_i = \min \left\{ \sum_{v:\ b(v)>0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor, \sum_{v:\ b(v)<0} \left\lfloor \frac{|b(v)|}{2^{K-i}} \right\rfloor \right\}$$

$$\geq \frac{1}{2^{K-i}} \min \left\{ \sum_{v:\ b(v)>0} |b(v)|, \sum_{v:\ b(v)<0} |b(v)| \right\} - n$$

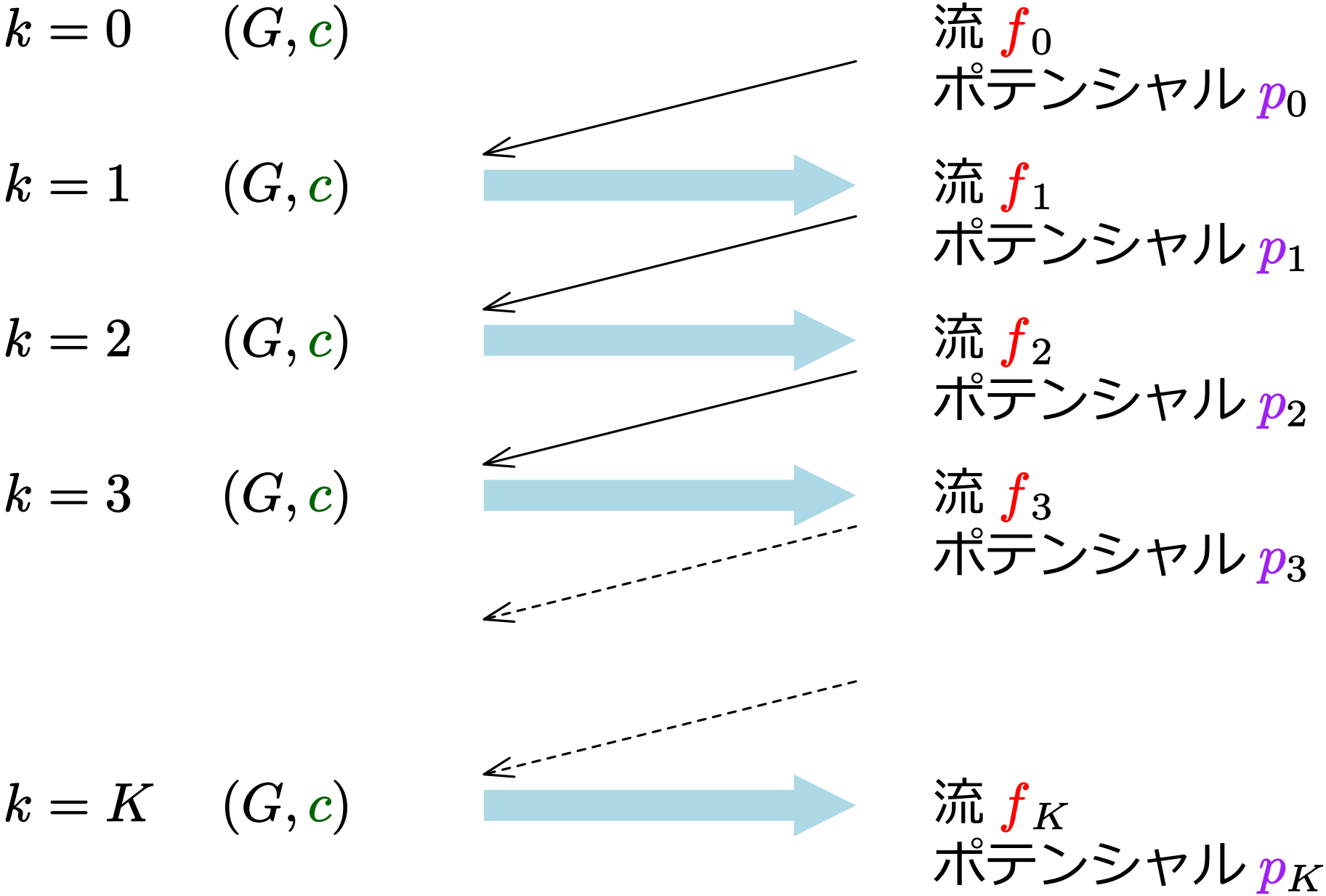
$$= \frac{1}{2^{K-i}} \cdot \frac{1}{2} \sum_{v \in V} |b(v)| - n$$

□

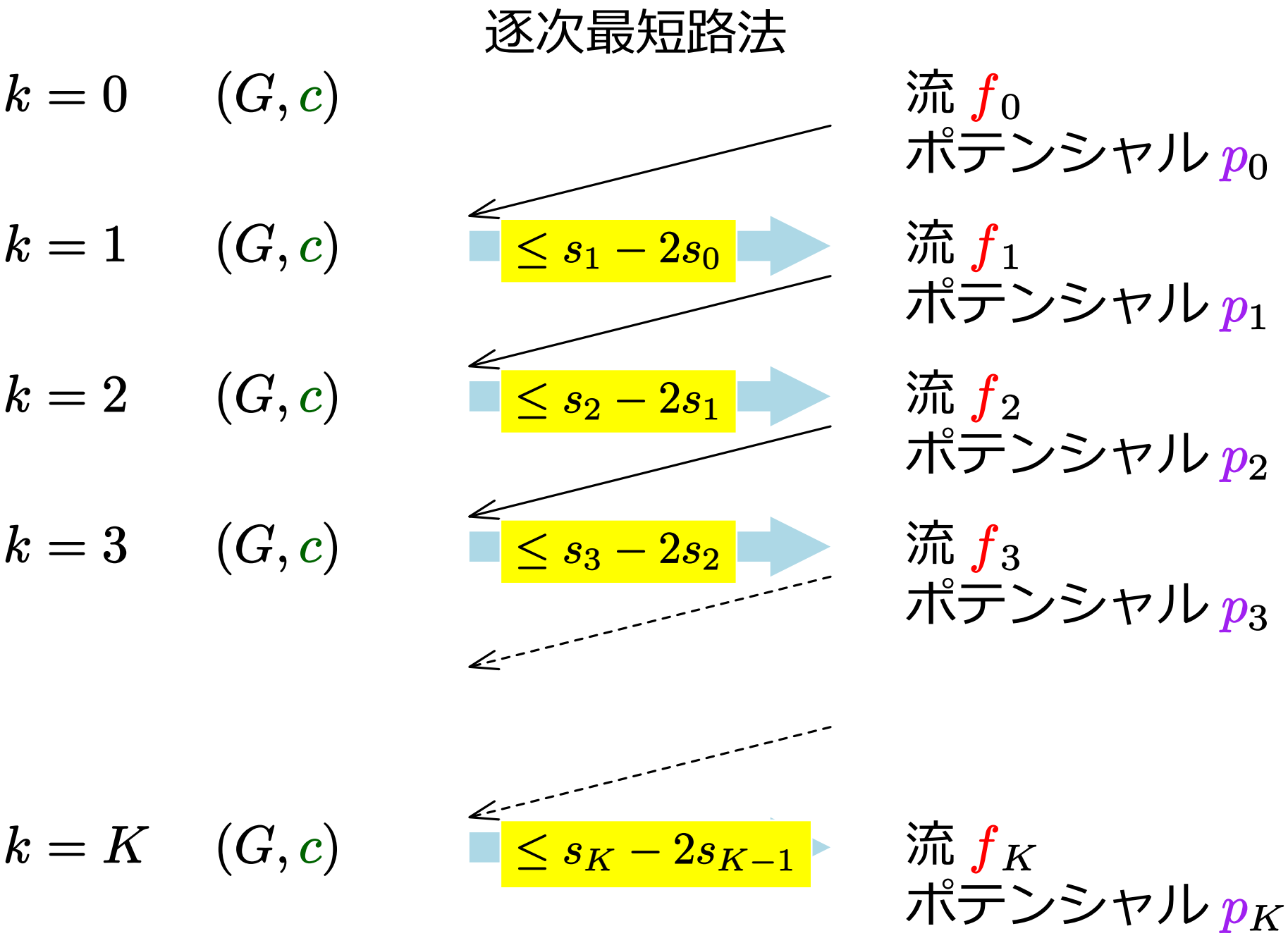


容量スケーリング法：計算量

逐次最短路法



容量スケーリング法：計算量



容量スケーリング法：計算量の算定

最短路計算を行う回数

$$
\begin{aligned}
&\leq \sum_{i=1}^{K} (s_i - 2s_{i-1}) \\
&= s_K - \sum_{i=0}^{K-1} s_i \\
&\leq \frac{1}{2} \sum_{v \in V} |b(v)| - \sum_{i=0}^{K-1} \left(\frac{1}{2^{K-i+1}} \sum_{v \in V} |b(v)| - n \right) \\
&= \frac{1}{2^K} \cdot \frac{1}{2} \sum_{v \in V} |b(v)| + nK = O(n \log B)
\end{aligned}
$$

最短路計算 1 回の計算量 (Dijkstra 法)

$$= O(m + n \log n)$$

容量スケーリング法：計算量の算定 (続)

全体の計算量

$$= O((m' + n' \log n') \cdot n' \log B')$$

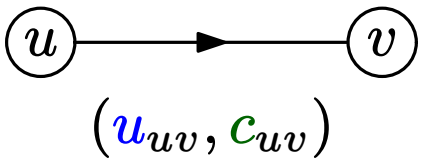
$$= O(m^2 (\log n)(\log(B + mU)))$$

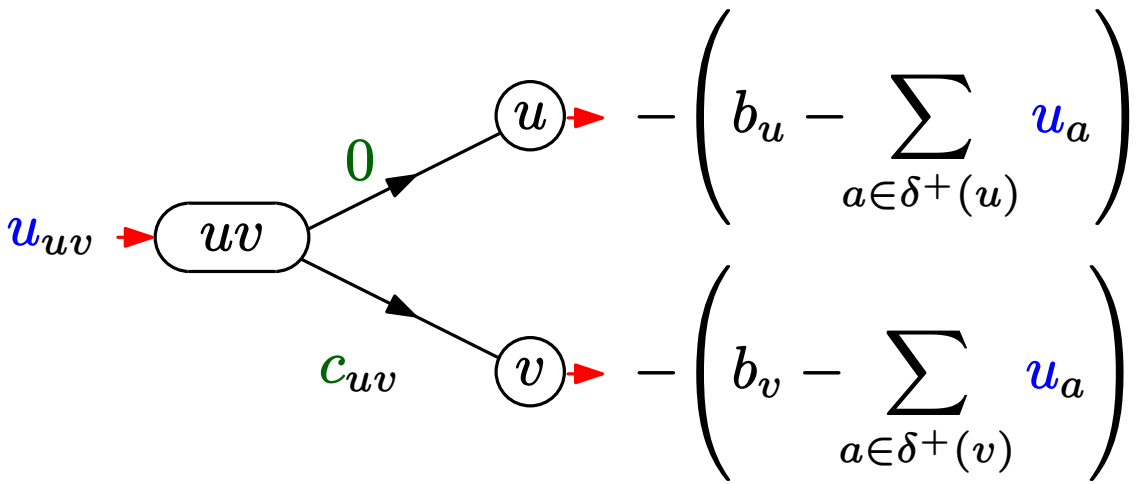
変換後のグラフにおいて

頂点数 $n' = m + n$

弧数 $m' = 2m$

$B' \leq B + mU$





計算量：まとめ

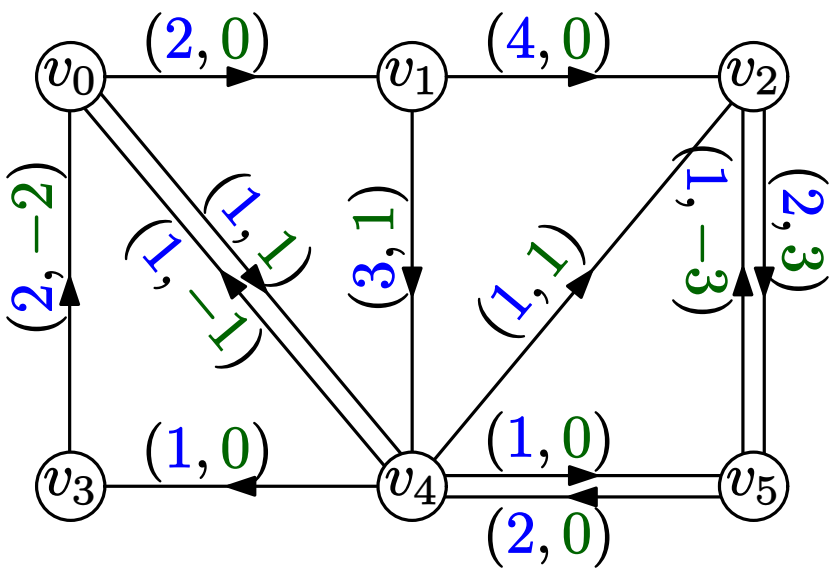
性質：容量スケーリング法の計算量

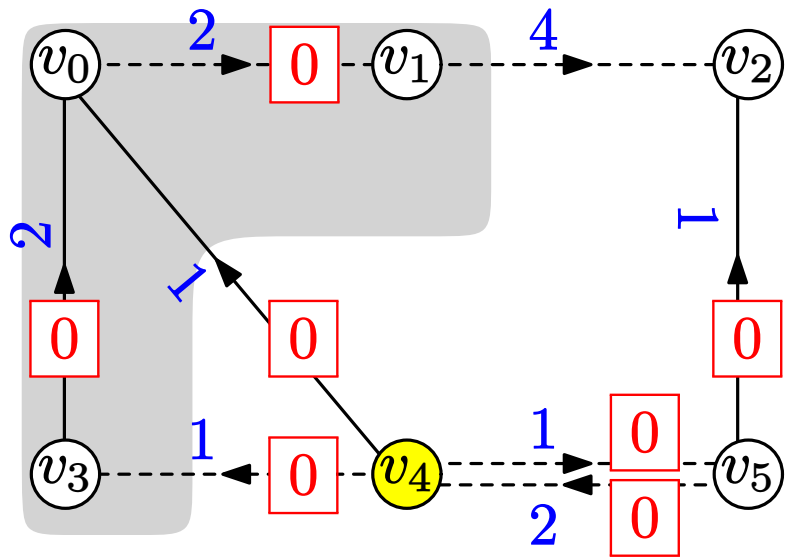
容量と供給・需要がすべて整数であるとき，
容量スケーリング法の計算量は
$O(m^2(\log n)(\log(B + mU)))$

これは弱多項式時間の計算量

本日の内容

1. 容量スケーリング法：準備
2. 容量スケーリング法：概要
3. 容量スケーリング法：計算量
4. **主双対法**





主双対法：考え方

主双対法に対する視点

主許容性
双対許容性 を満たしたまま **相補性** を満たしに行く

使う最適性条件：簡約費用最適性条件

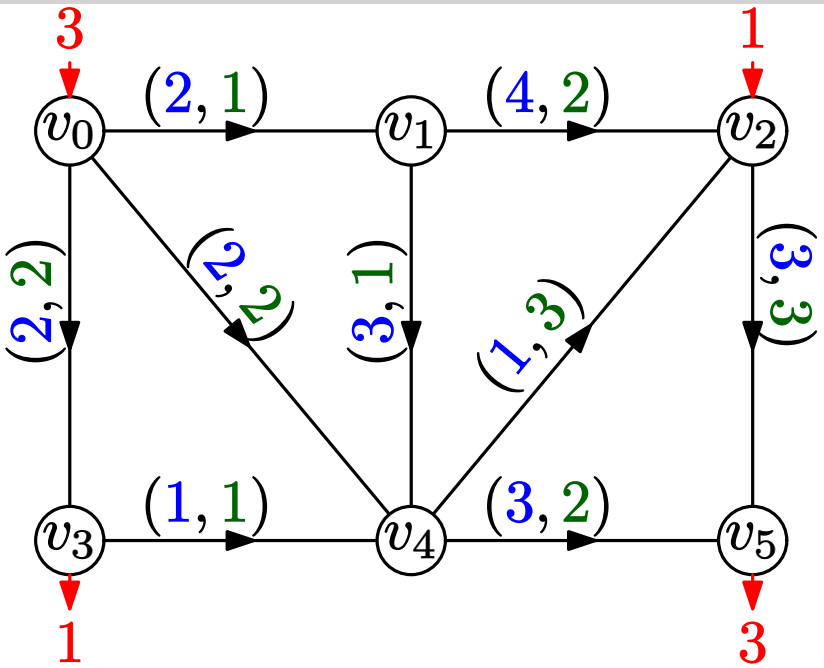
負閉路消去法に対する視点

主許容性
相補性 を満たしたまま **双対許容性** を満たしに行く

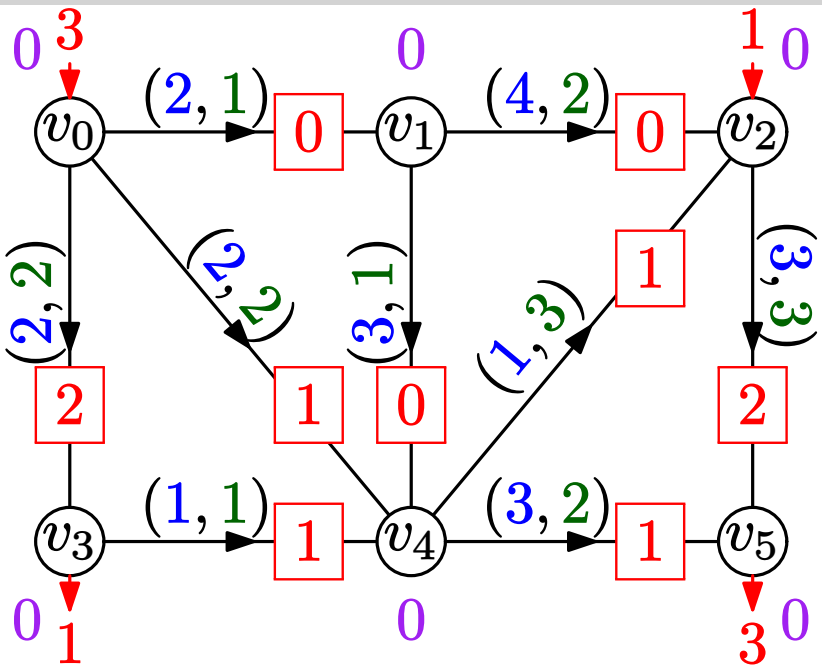
正カット消去法，逐次最短路法に対する視点

双対許容性
相補性 を満たしたまま **主許容性** を満たしに行く

主双对法：例 (1/8)

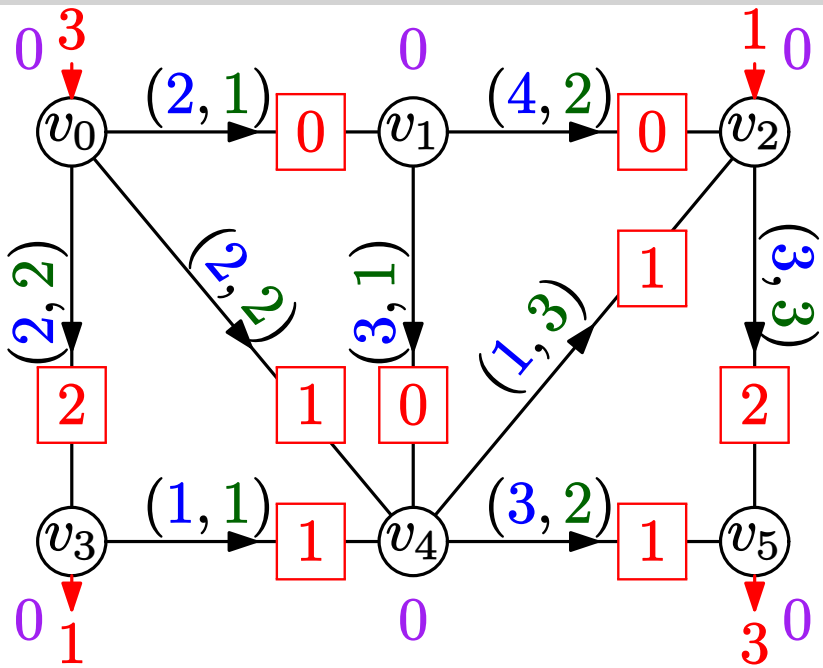


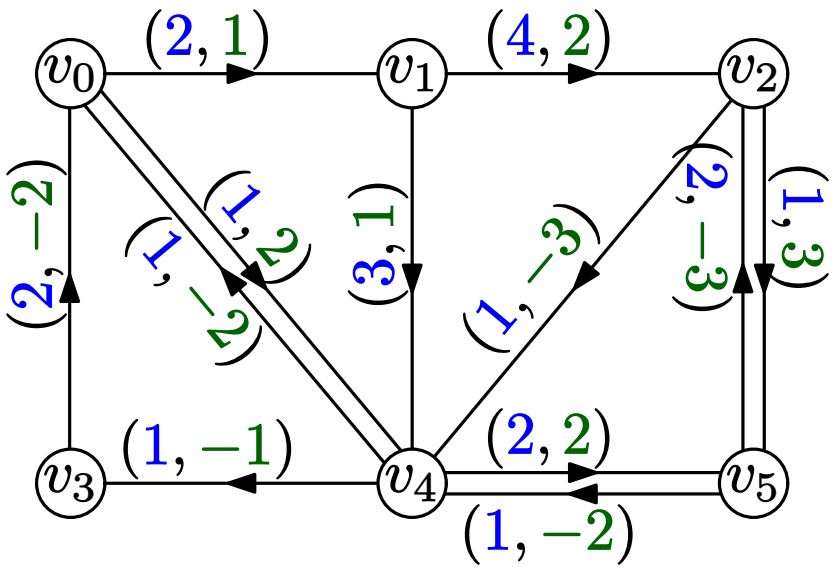
主双対法：例 (1/8)



初期化：b-流 f，
ポテンシャル $p = 0$

主双対法：例 (1/8)

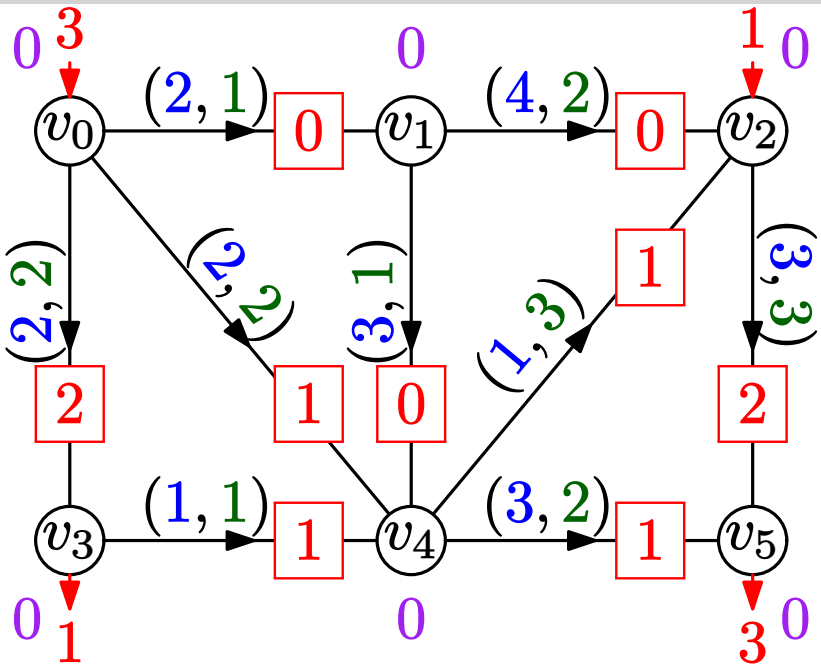


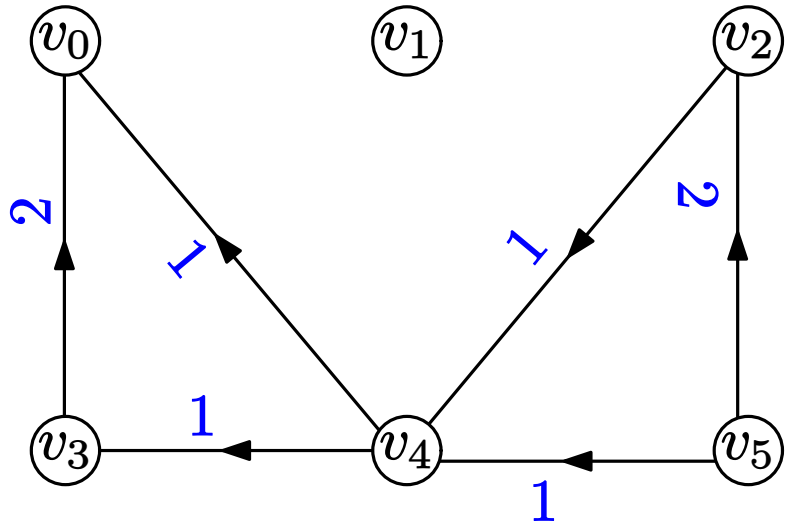


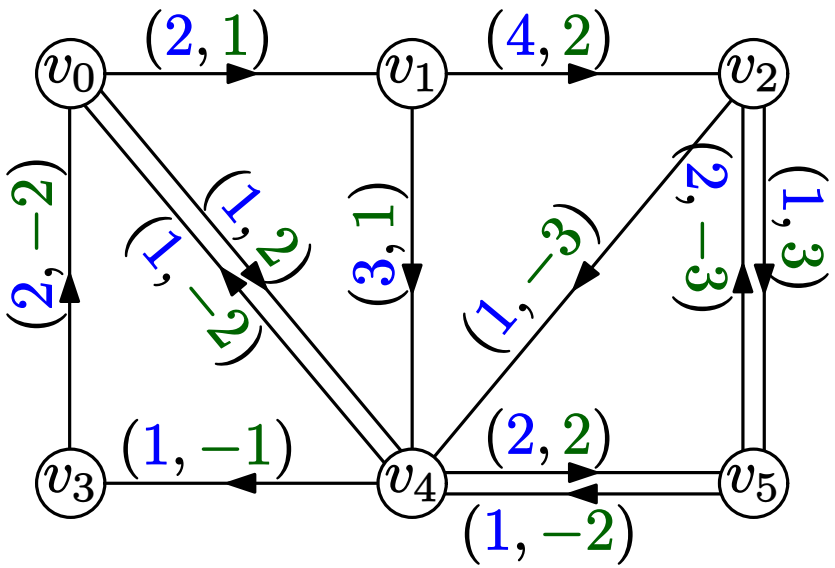
初期化：b-流 f，
ポテンシャル $p = 0$

補助ネットワーク
(費用は簡約費用)

主双対法：例 (1/8)



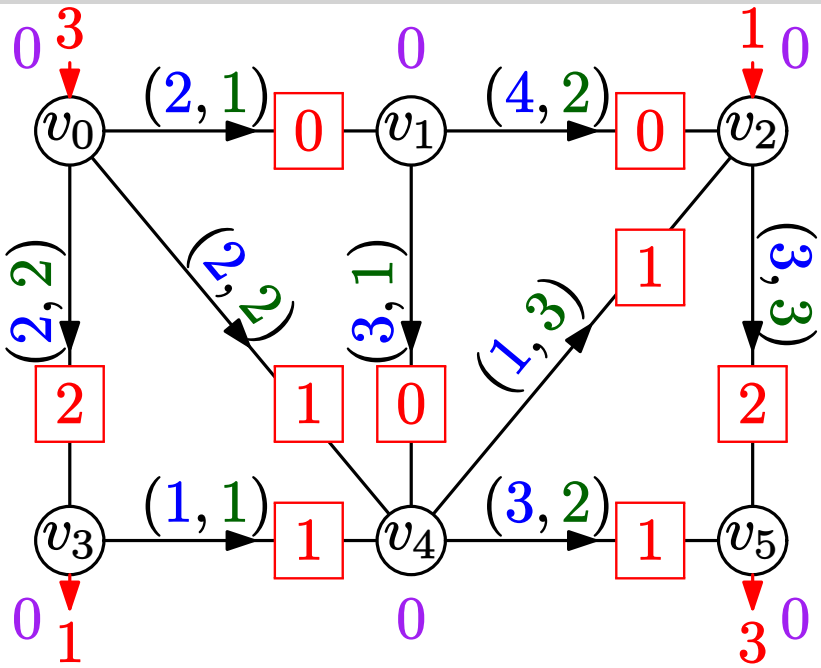


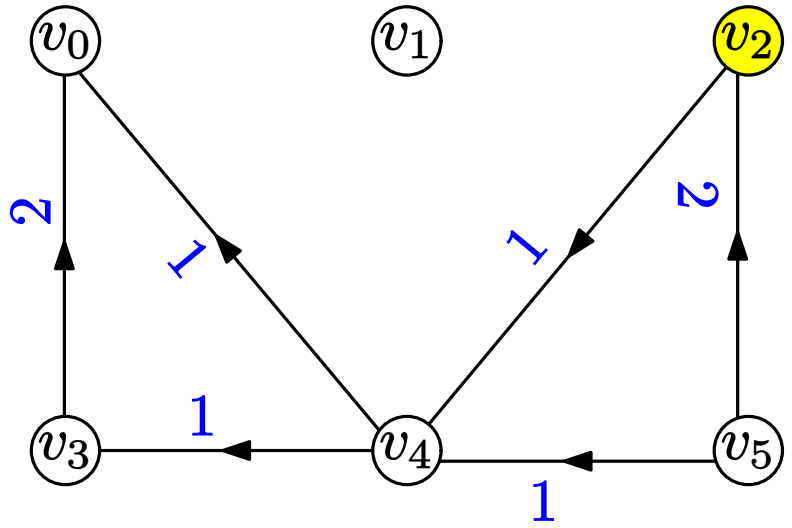


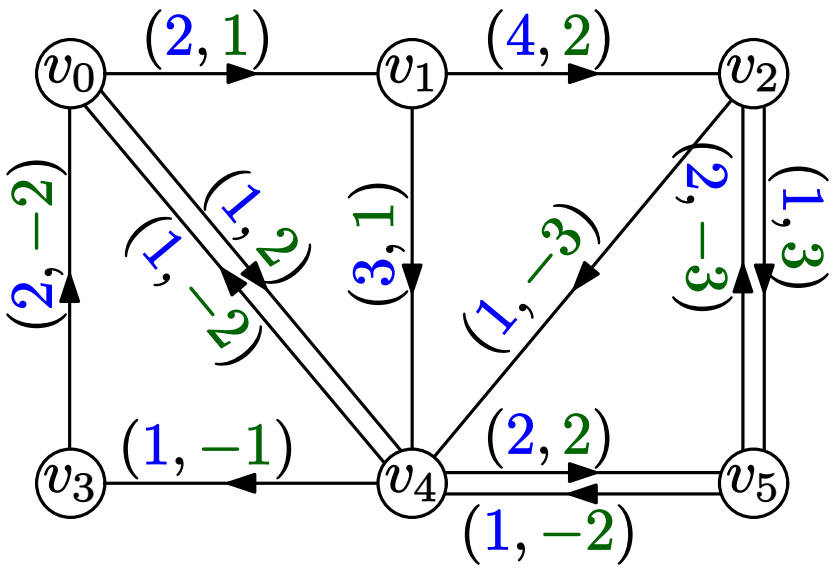
初期化：b-流 f，
ポテンシャル $p = 0$

補助ネットワーク
(費用は簡約費用)

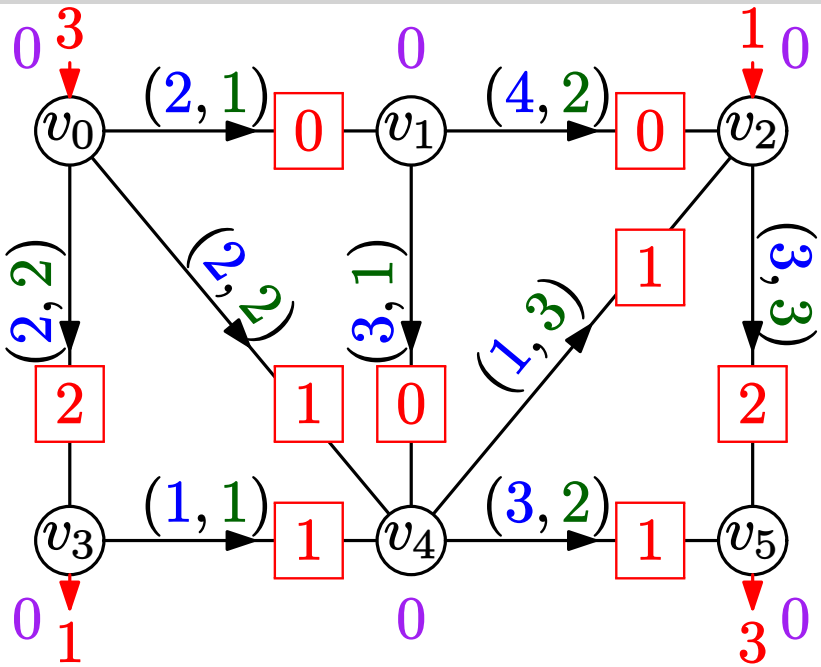
主双对法：例 (2/8)

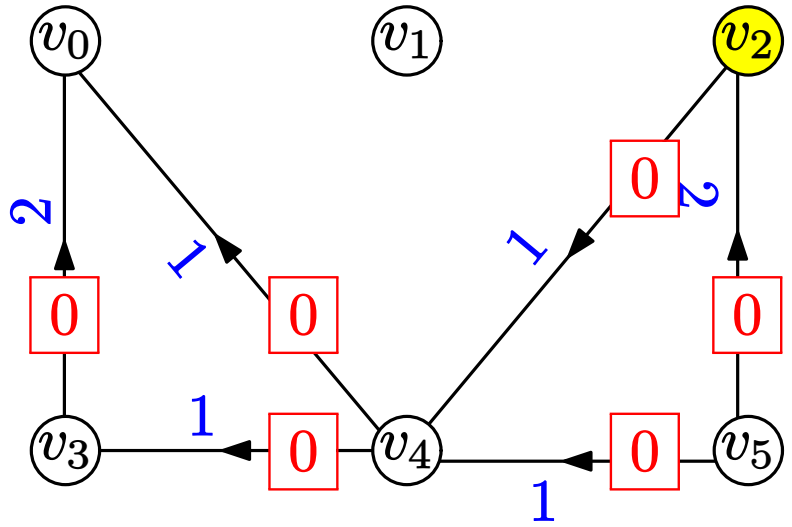

0 3
0
1 0
v_0
v_1
v_2
(2, 1)
(4, 2)
0
0
(2, 2)
(2, 2)
(3, 1)
(3, 3)
1
(1, 3)
2
1
0
2
(1, 1)
(3, 2)
v_3
v_4
v_5
1
1
0 1
0
3 0

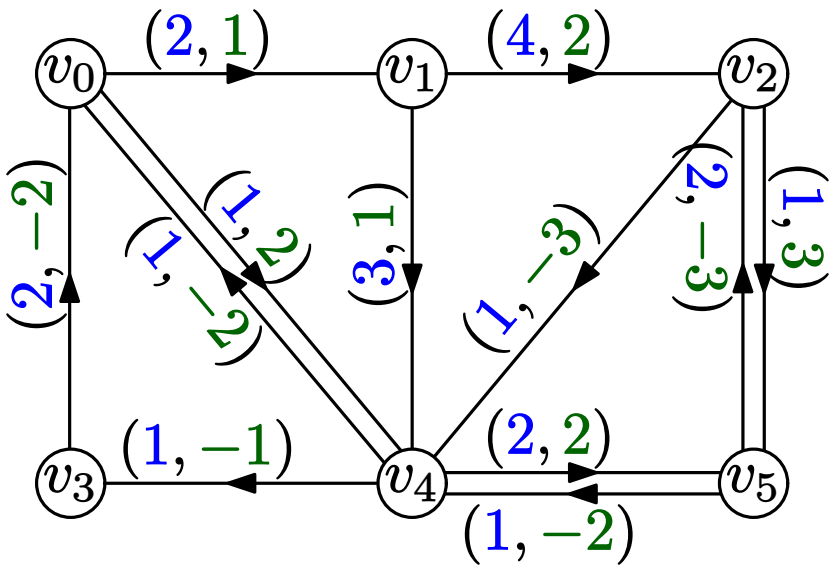

v_0
v_1
v_2
2
1
1
2
v_3
1
v_4
v_5
1


(2, 1)
(4, 2)
v_0
v_1
v_2
(2, −2)
(1, 2)
(1, −2)
(3, 1)
(1, −3)
(2, −3)
(1, 3)
(1, −1)
(2, 2)
v_3
v_4
v_5
(1, −2)

主双対法：例 (2/8)

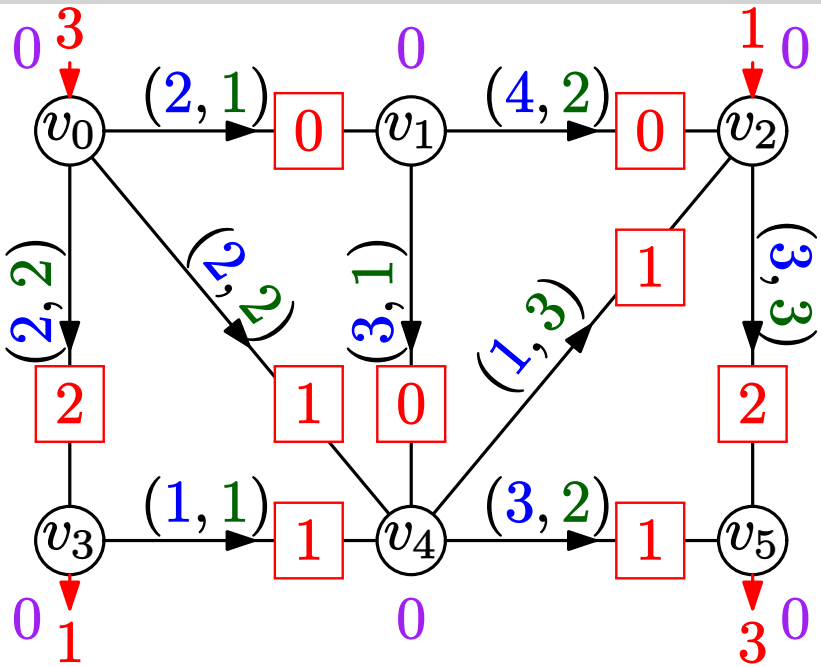


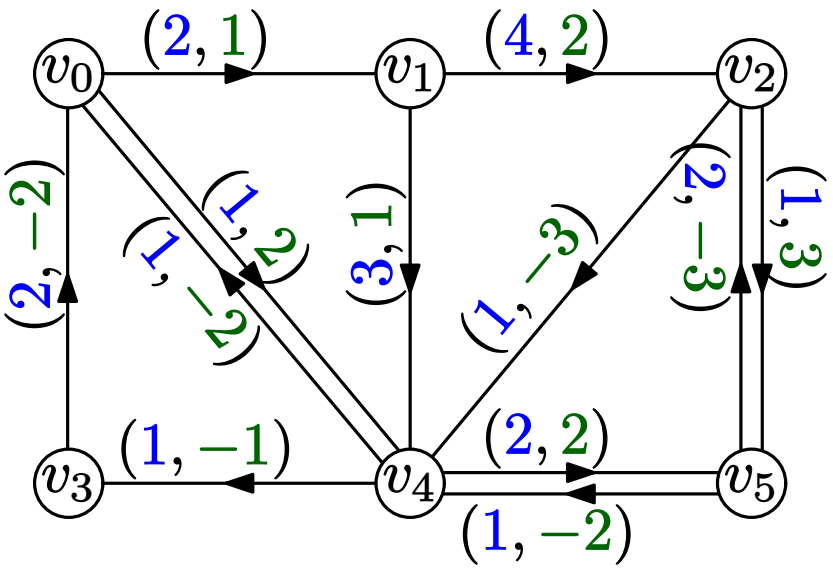


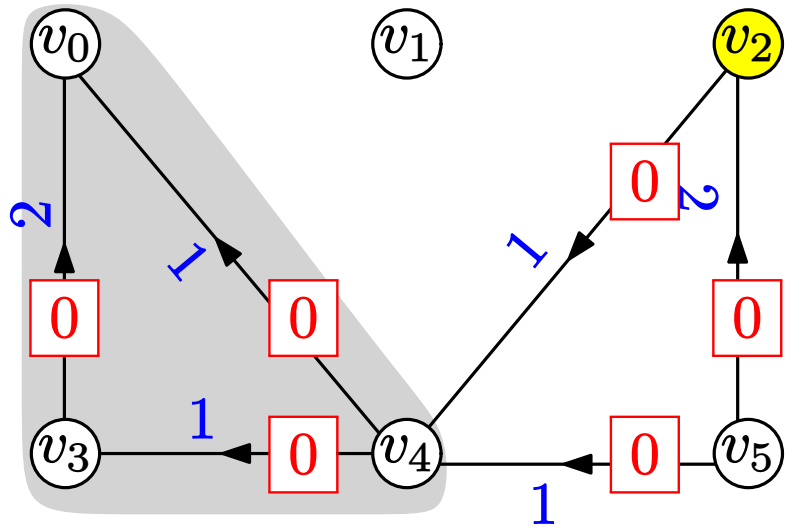


v_2-v_2 流で，
v_2 への流入量を最大に
するものを見つける

主双対法：例 (2/8)



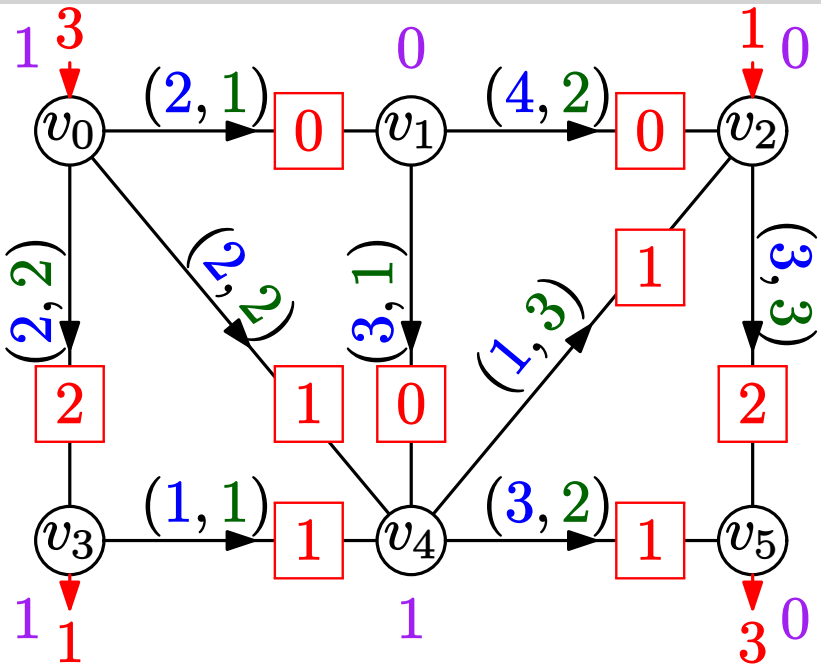


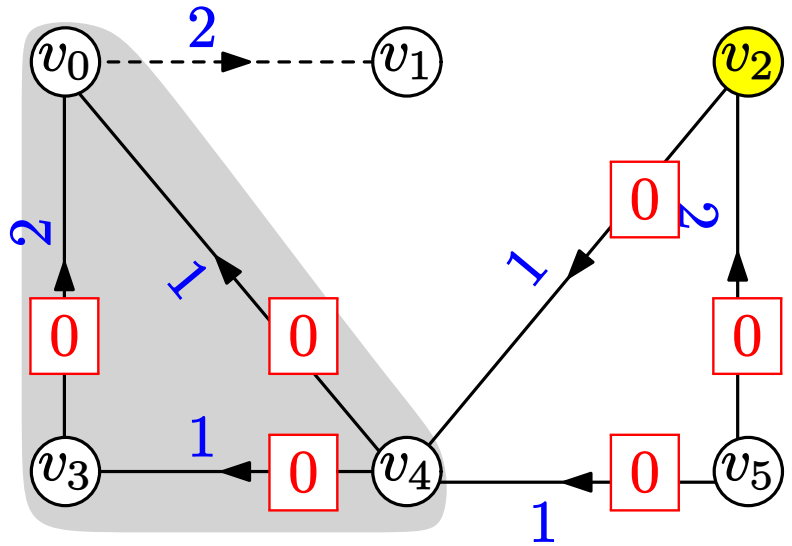


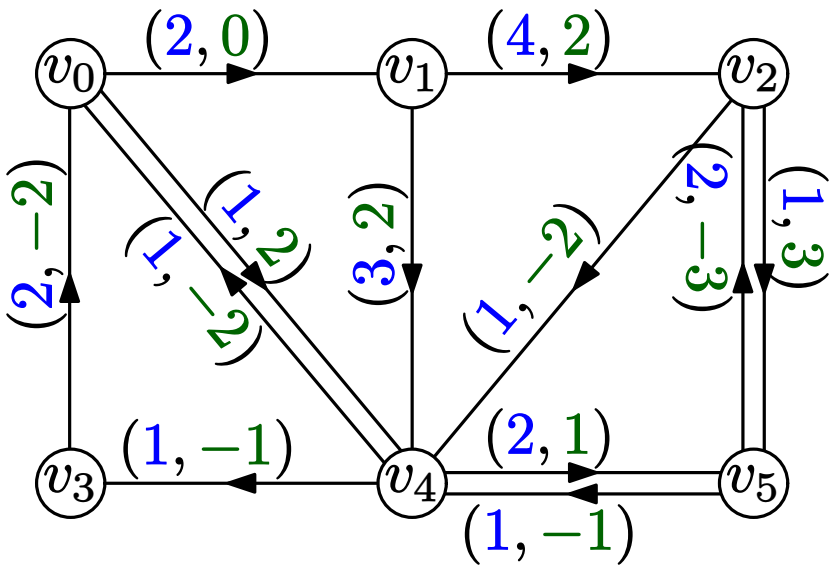
v_2-v_2 流で，
v_2 への流入量を最大に
するものを見つける

v_2 から到達できる (v_2 以外の)
頂点の集合 W を定める

主双対法：例 (2/8)





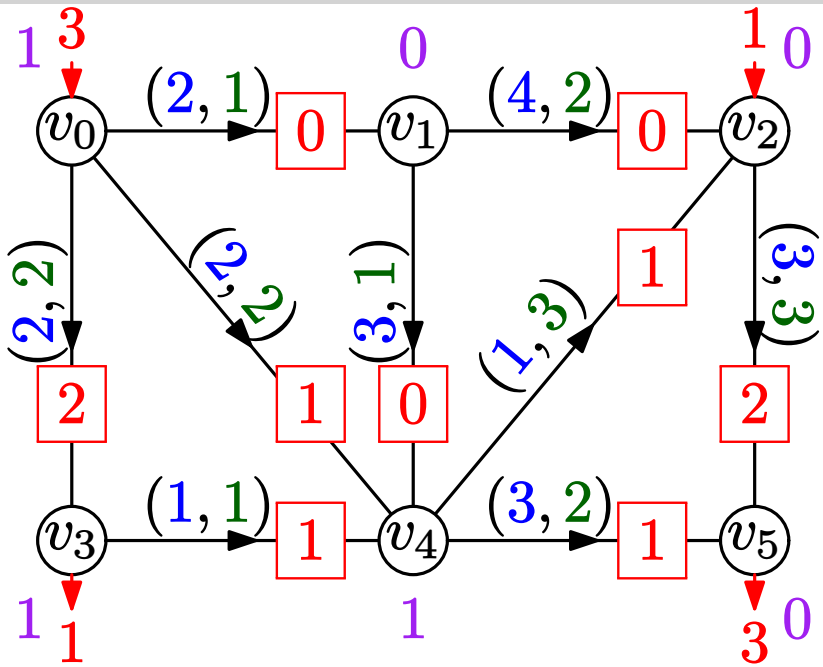


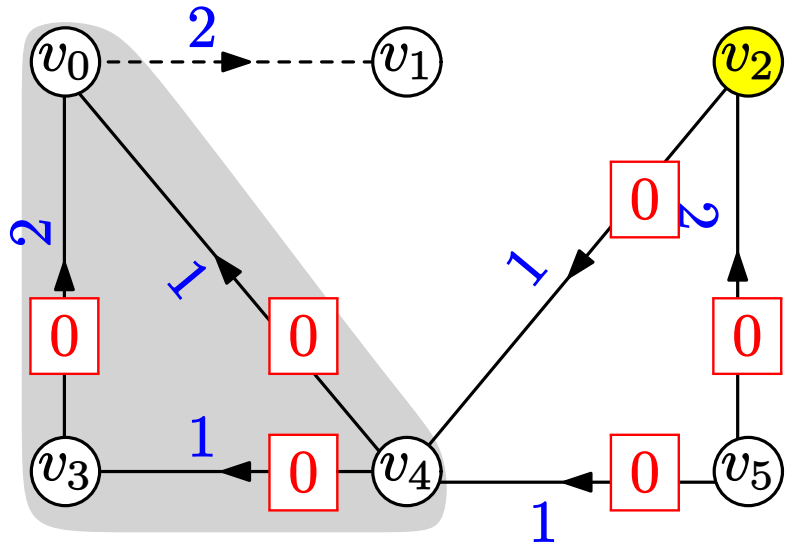
v_2-v_2 流で，
v_2 への流入量を最大に
するものを見つける

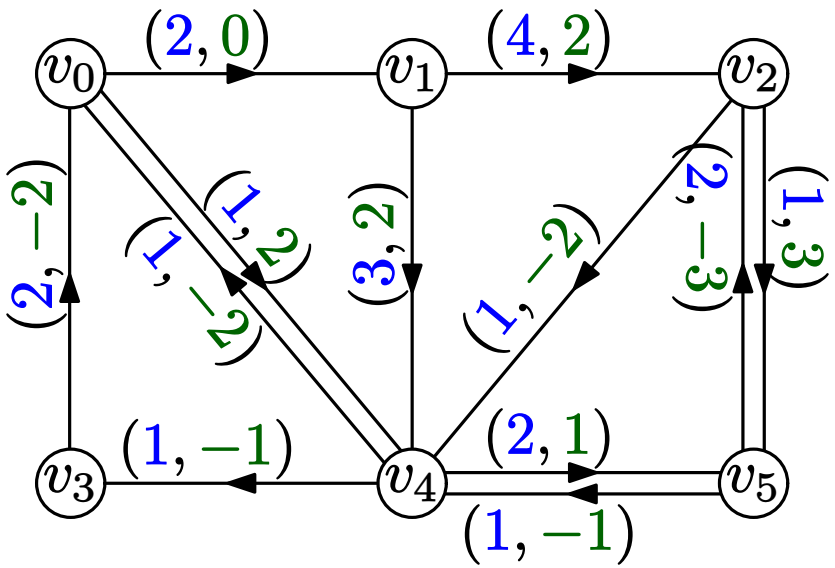
v_2 から到達できる (v_2 以外の)
頂点の集合 W を定める

W の頂点のポテンシャルを
上げる

主双対法：例 (2/8)







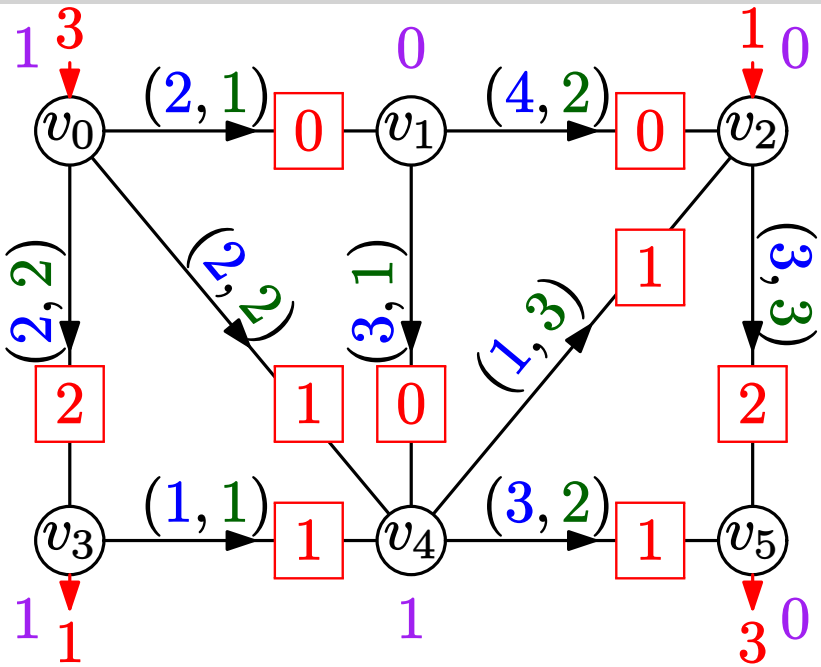
v_2-v_2 流で，
v_2 への流入量を最大に

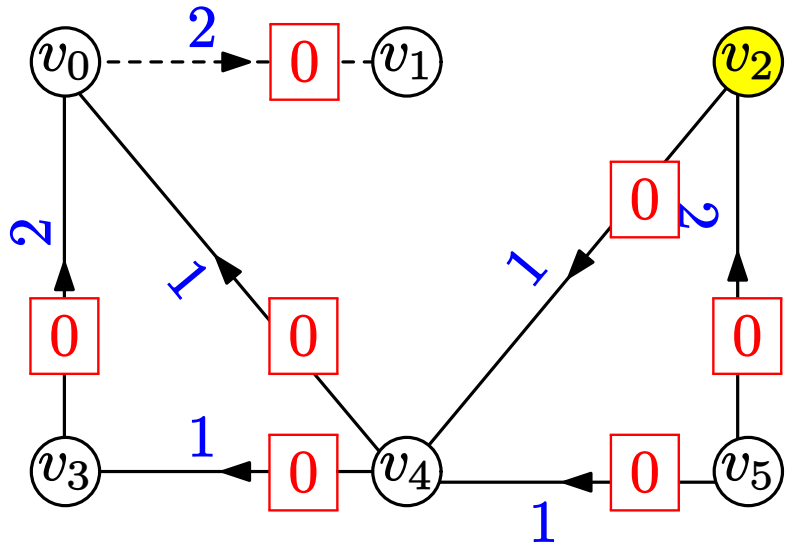
簡約費用が負の弧が減る か
簡約費用が 0 の弧が増える まで

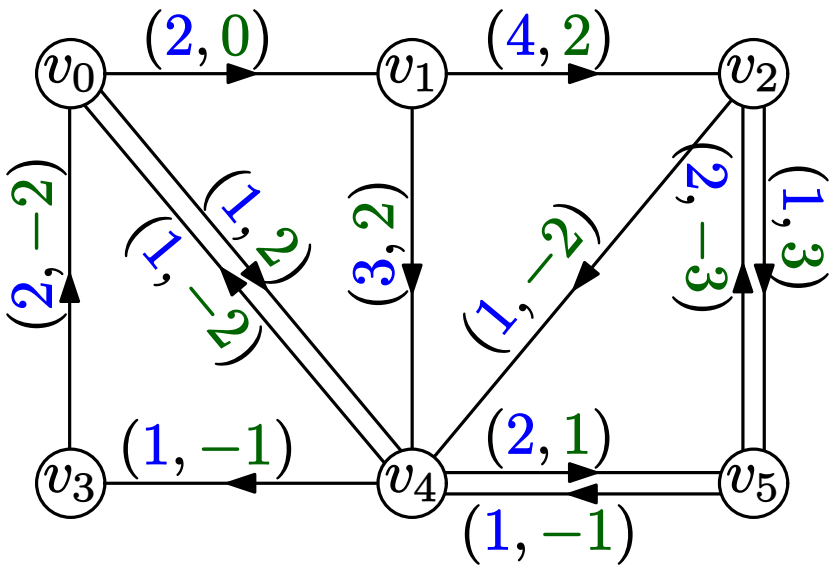
頂点の集合 W を定める

W の頂点のポテンシャルを上げる

主双対法：例 (3/8)

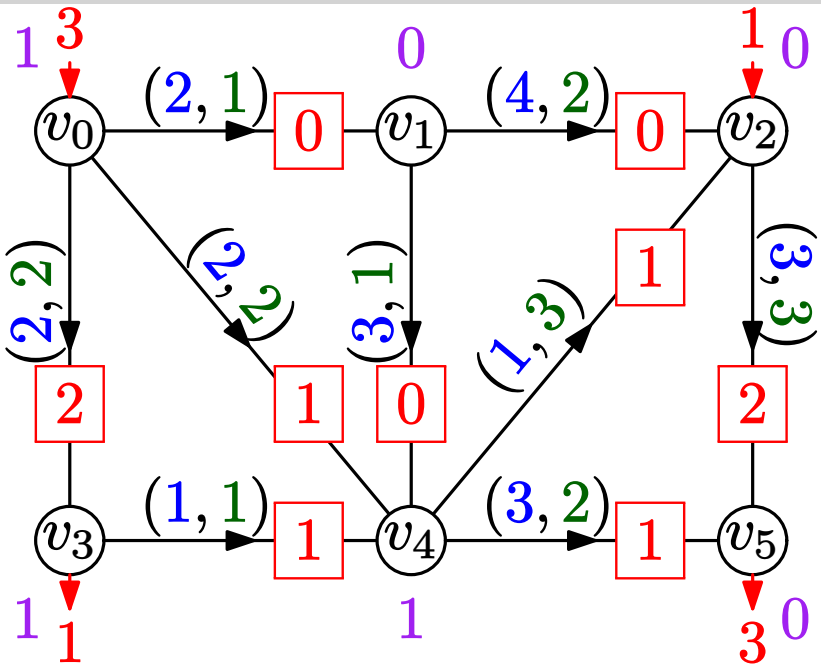


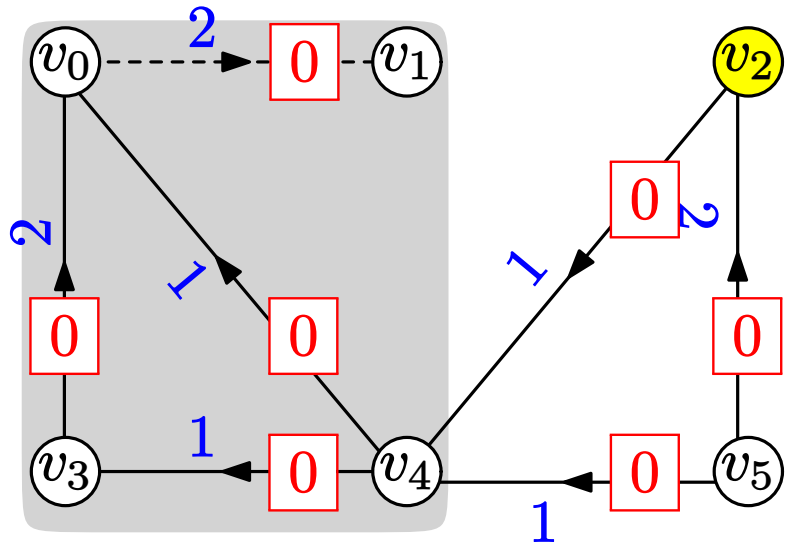


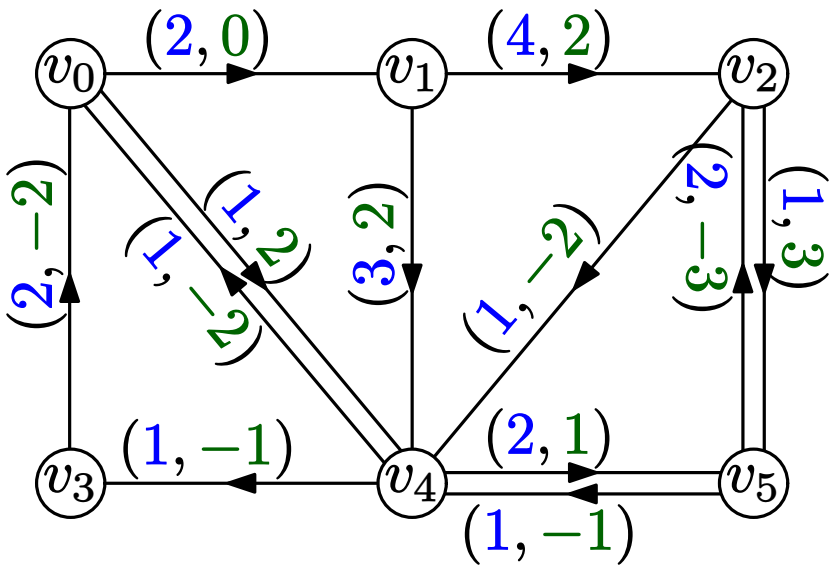


v_2-v_2 流で，
v_2 への流入量を最大に
するものを見つける

主双対法：例 (3/8)



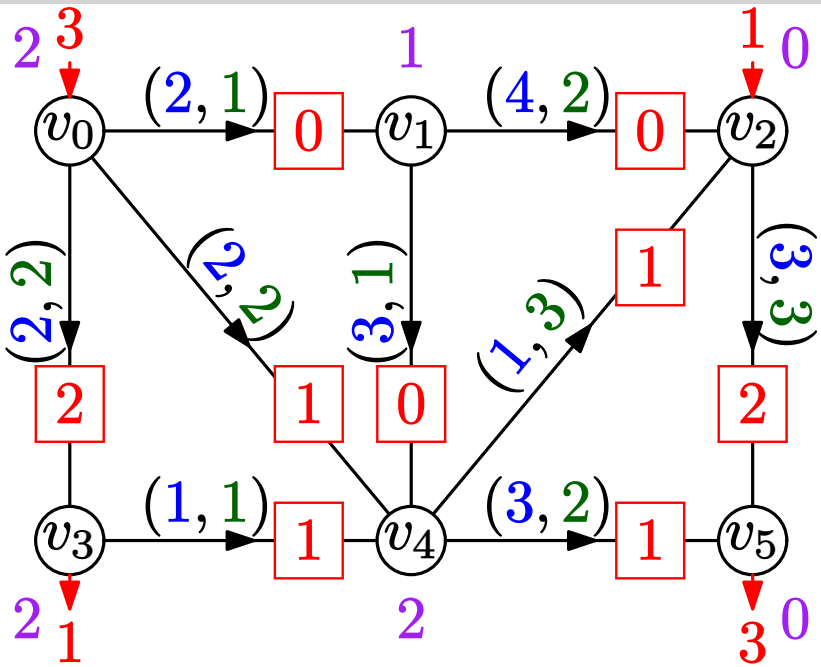


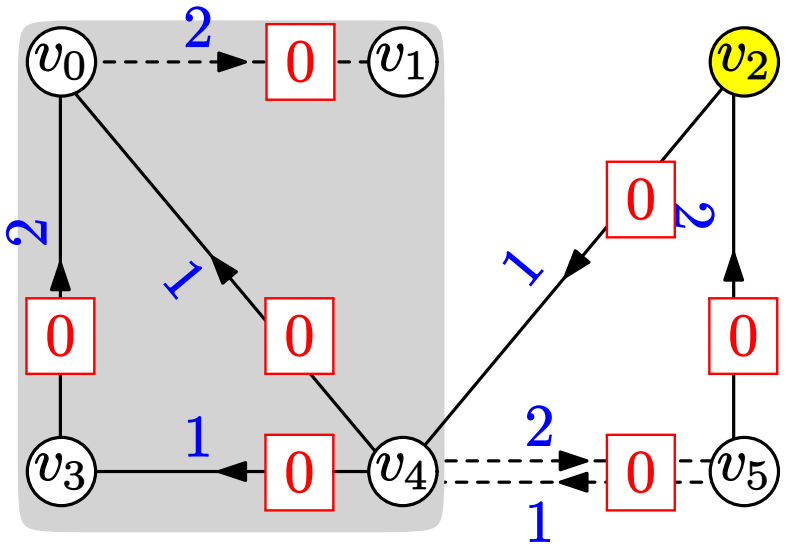


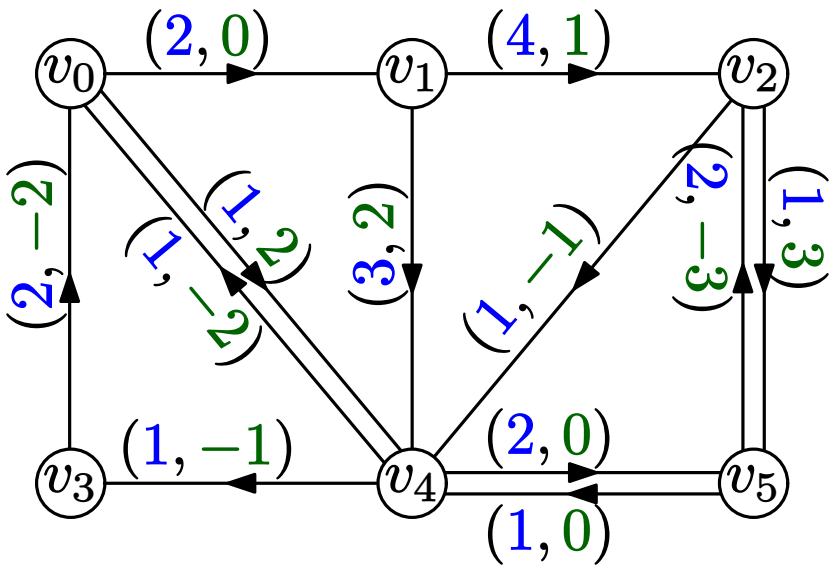
v_2-v_2 流で，
v_2 への流入量を最大に
するものを見つける

v_2 から到達できる (v_2 以外の)
頂点の集合 W を定める

主双対法：例 (3/8)





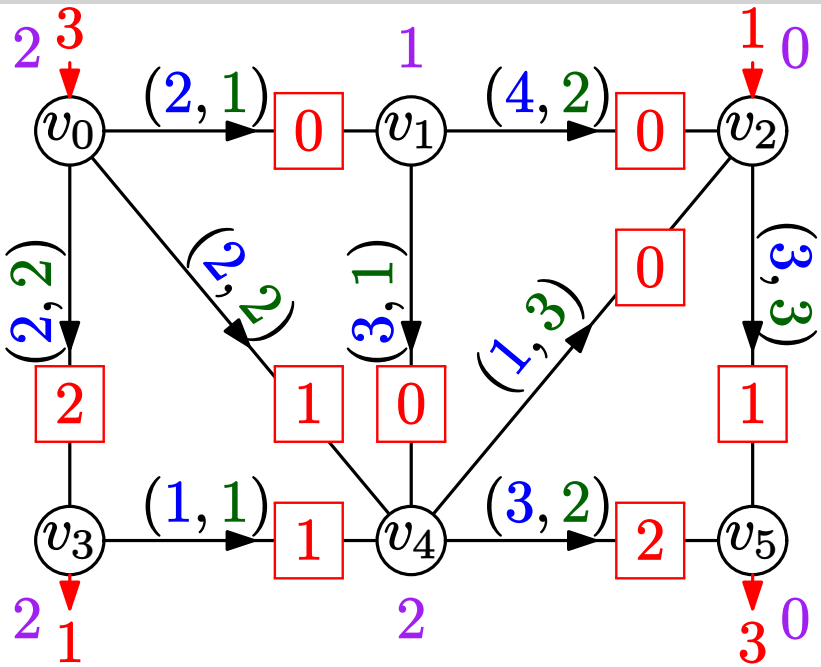


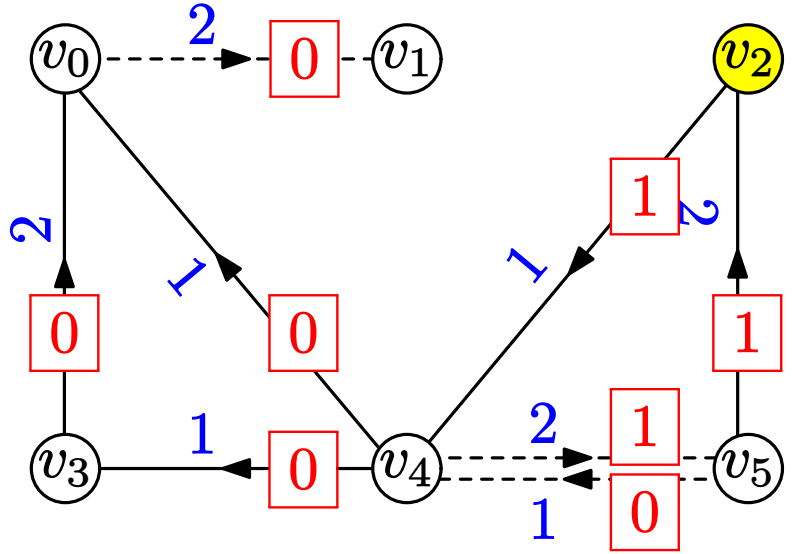
v_2-v_2 流で，
v_2 への流入量を最大に
するものを見つける

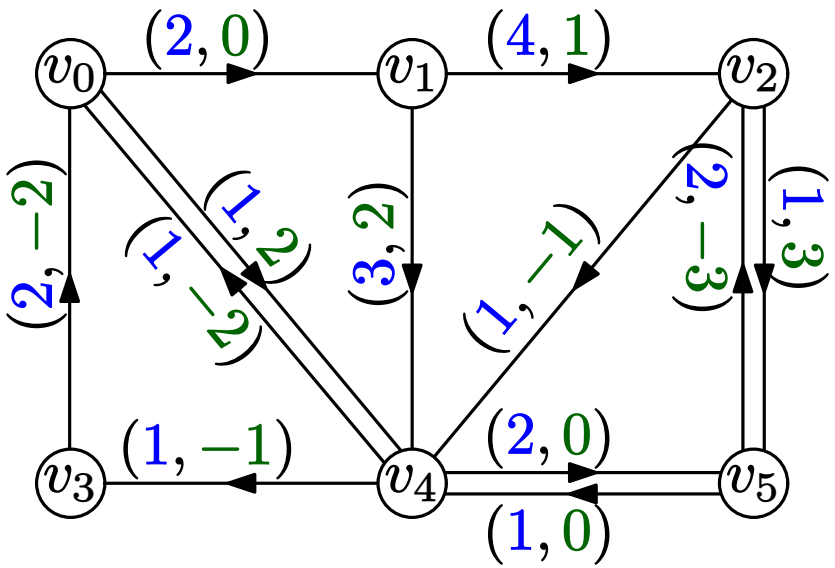
v_2 から到達できる (v_2 以外の)
頂点の集合 W を定める

W の頂点のポテンシャルを
上げる

主双対法：例 (4/8)

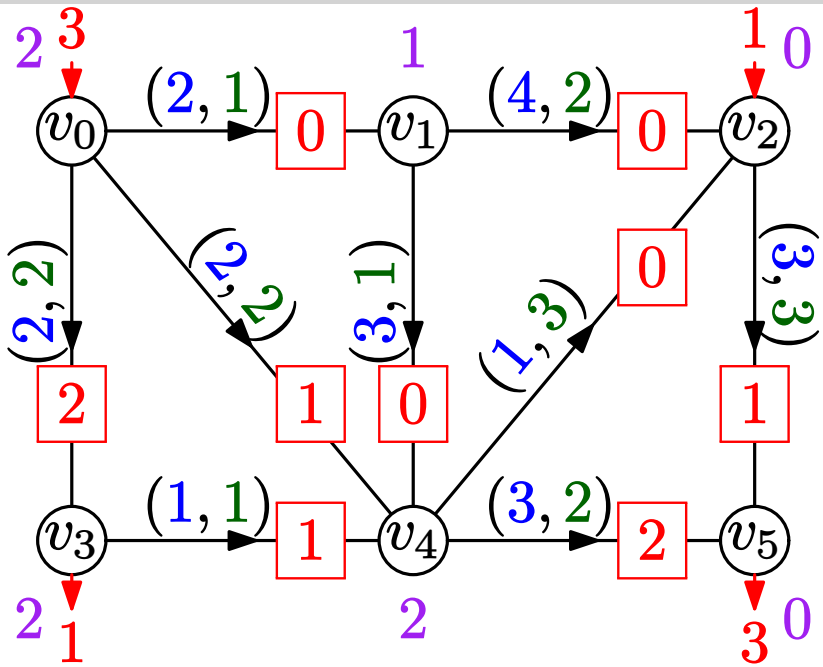


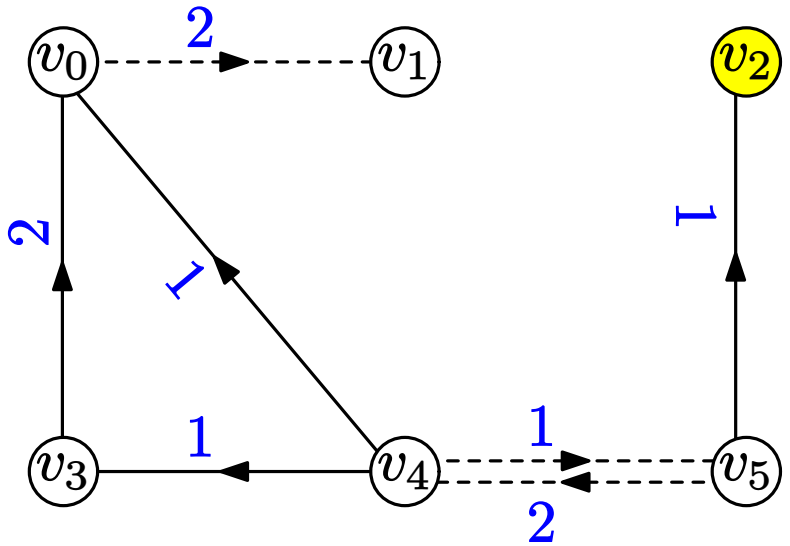


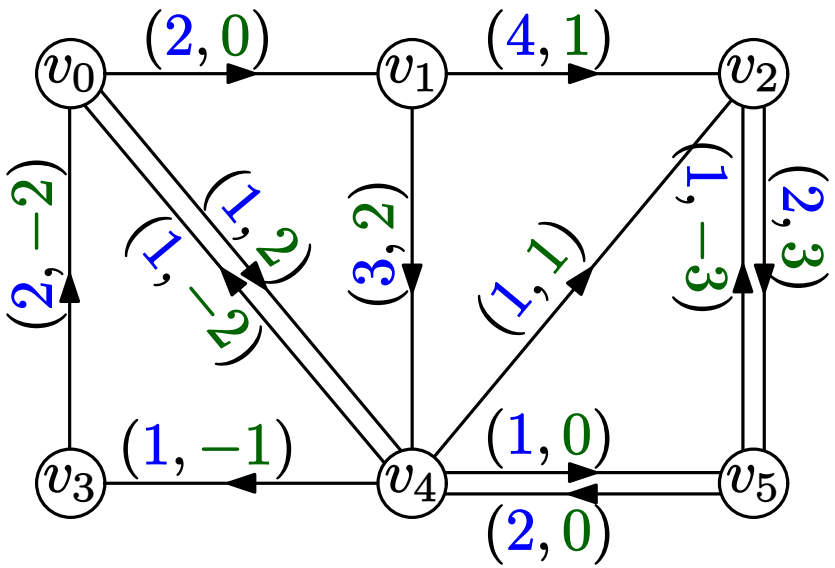


v_2-v_2 流で，
v_2 への流入量を最大に
するものを見つける

主双対法：例 (4/8)

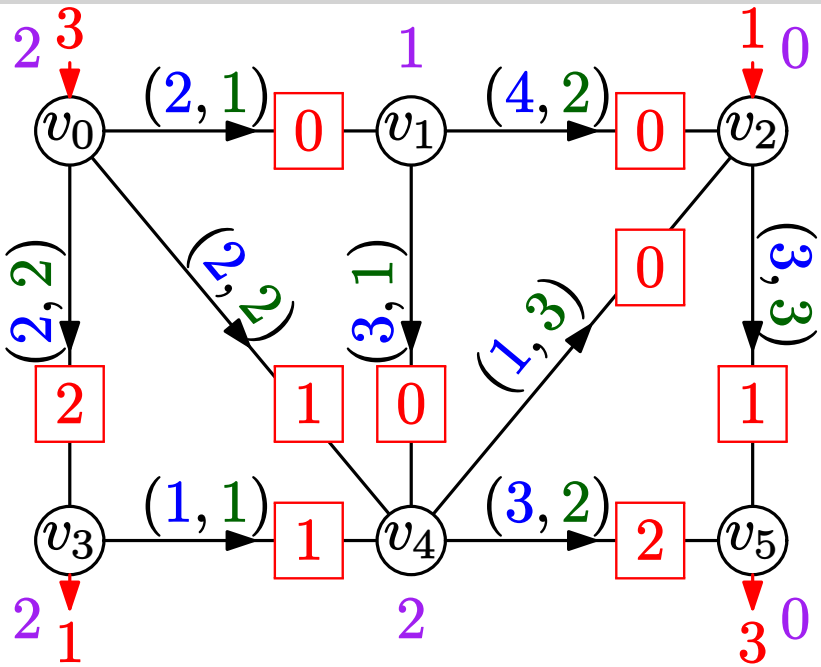


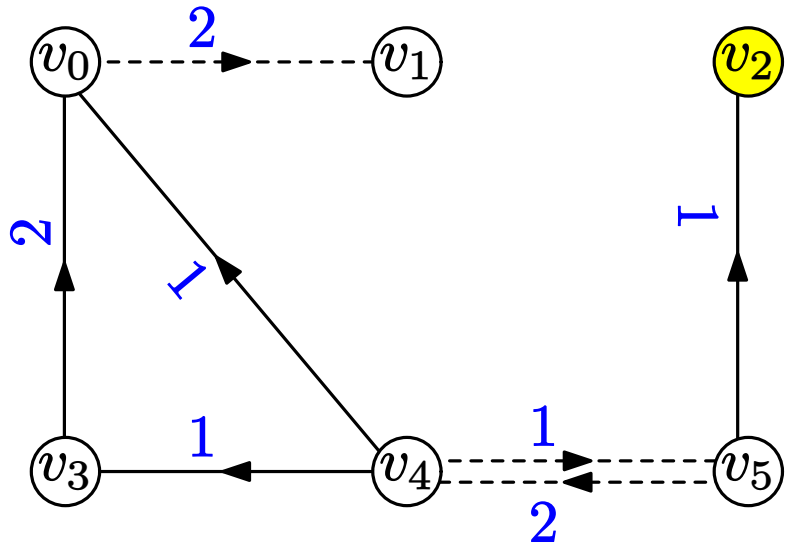


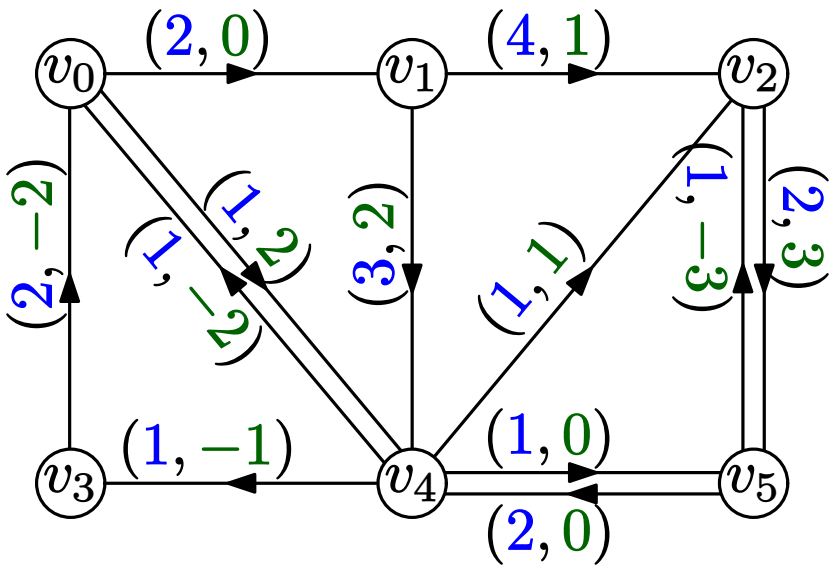


v_2-v_2 流で，
v_2 への流入量を最大に
するものを見つける

主双対法：例 (4/8)



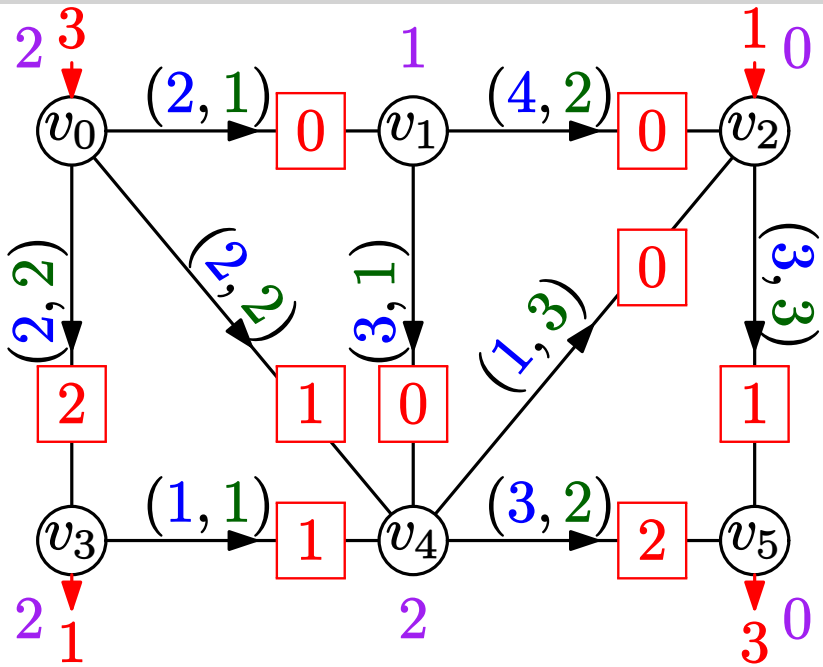


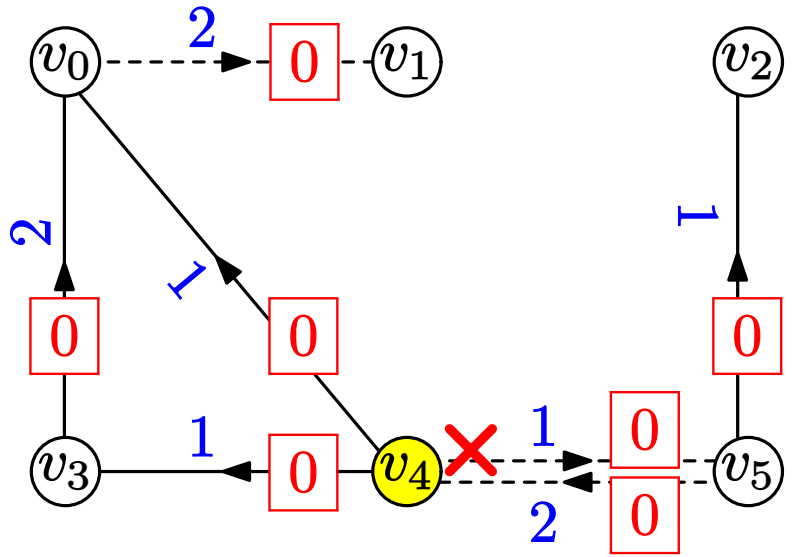


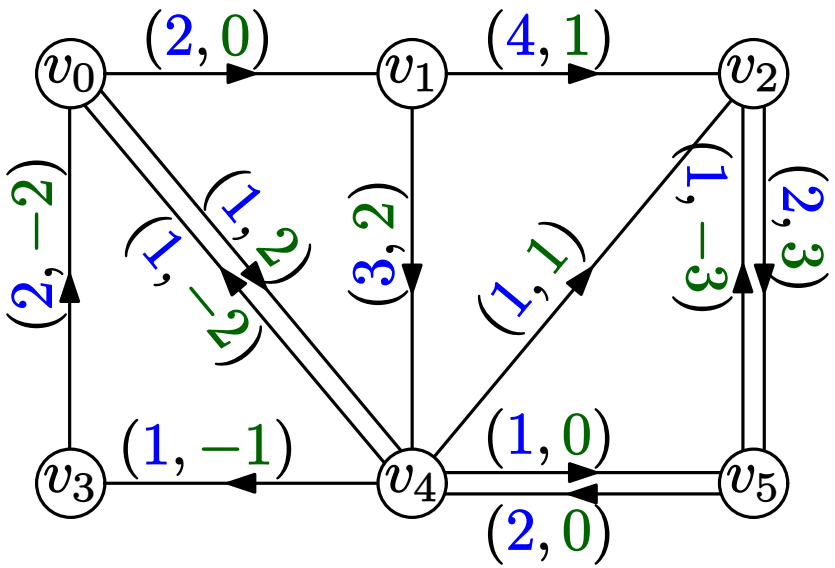
v_2-v_2 流で，
v_2 への流入量を最大に
するものを見つける

v_2 から到達できる (v_2 以外の)
頂点の集合 W を定める

主双対法：例 (5/8)

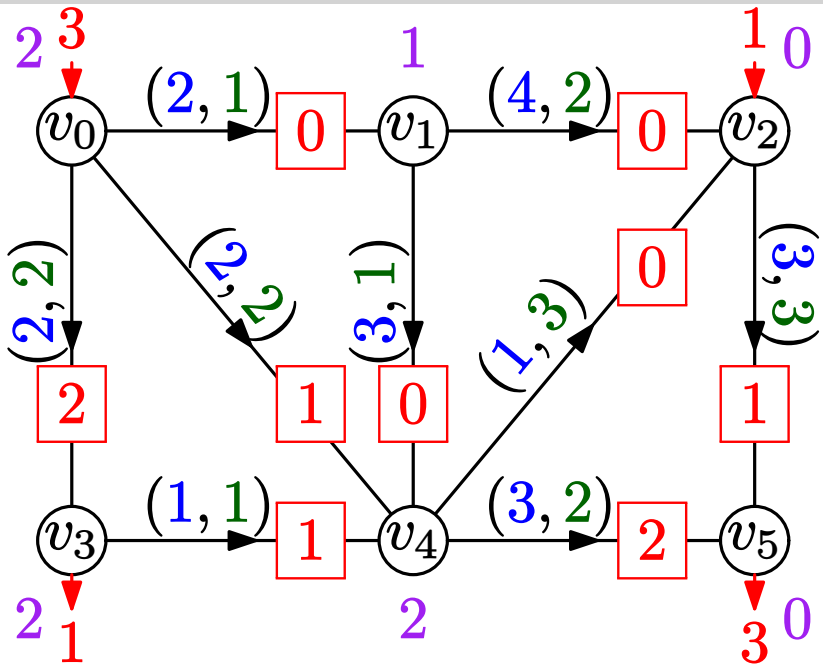


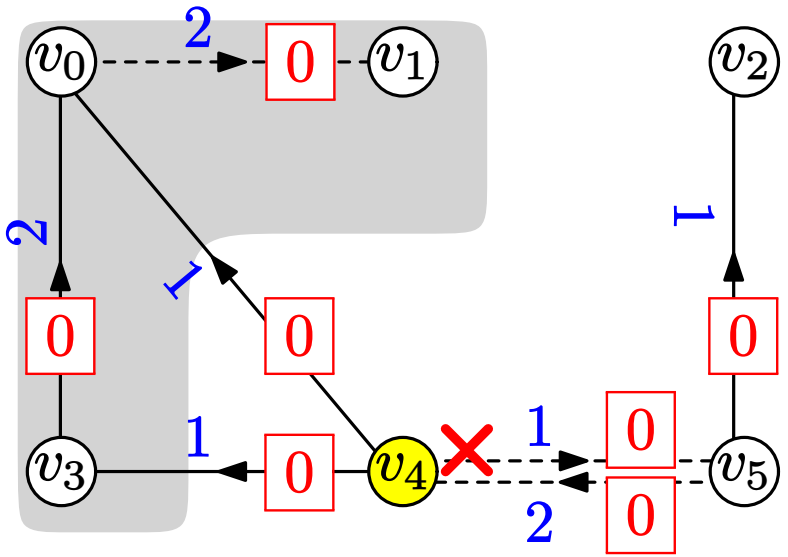


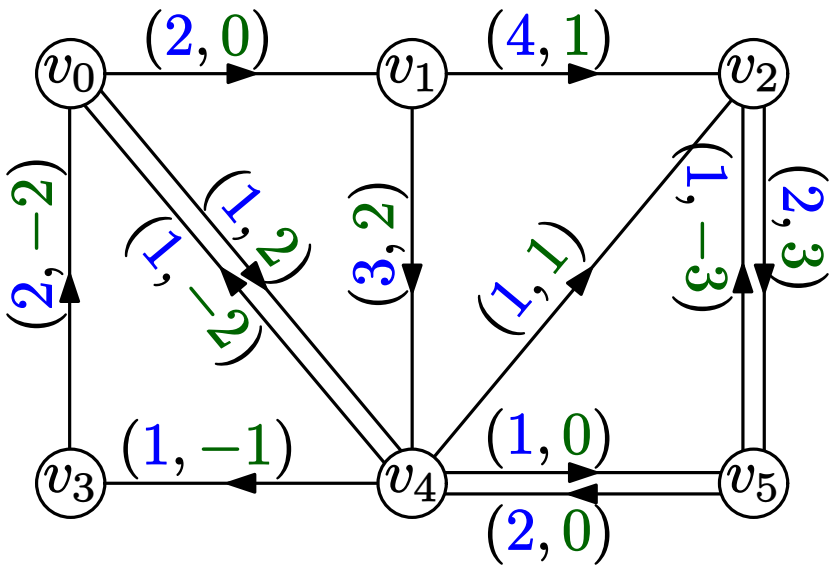


v_4-v_4 流で，
v_4 への流入量を最大に
するものを見つける

主双対法：例 (5/8)



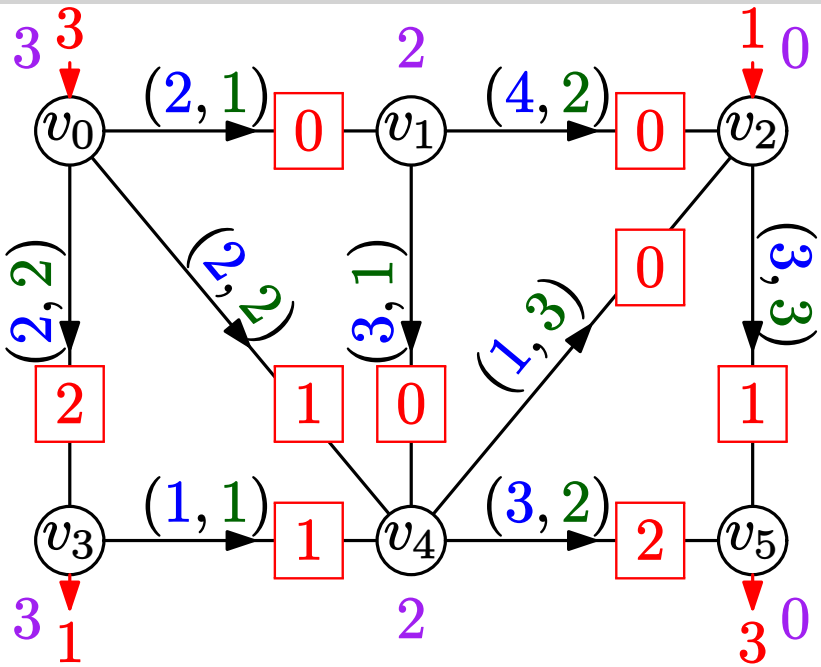


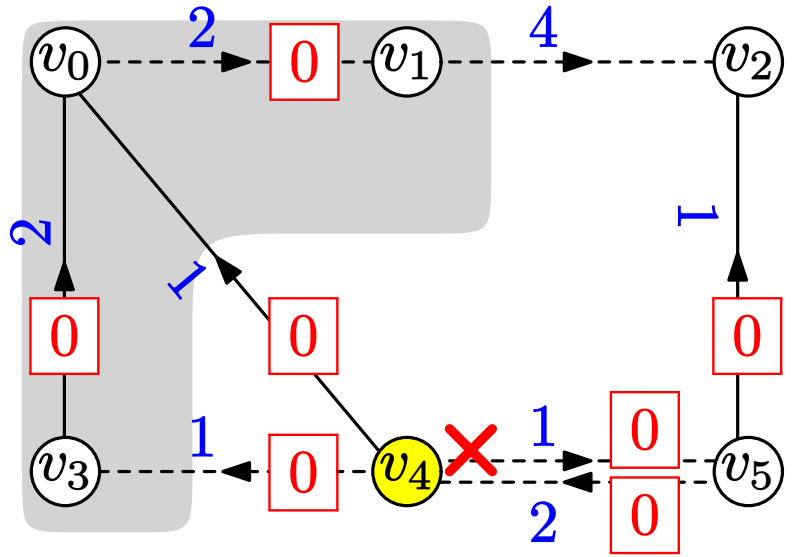


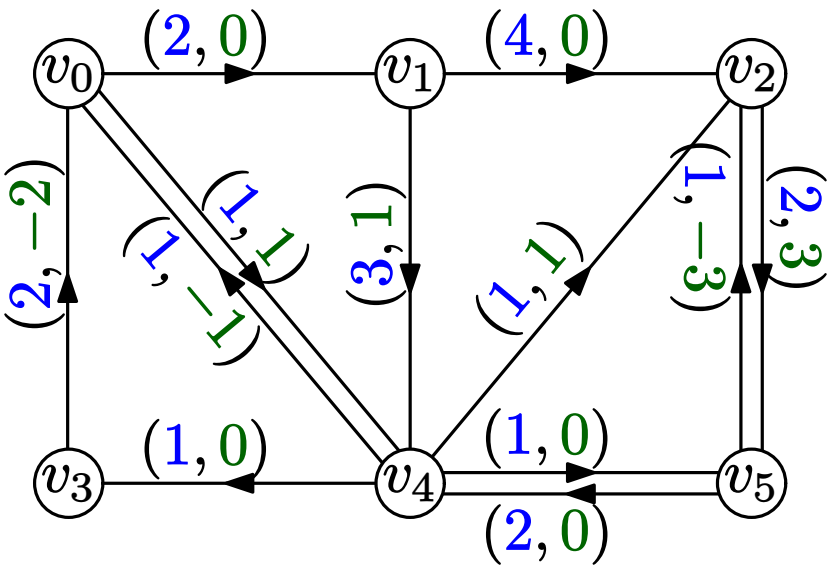
v_4-v_4 流で，
v_4 への流入量を最大に
するものを見つける

v_4 から到達できる (v_4 以外の)
頂点の集合 W を定める

主双対法：例 (5/8)





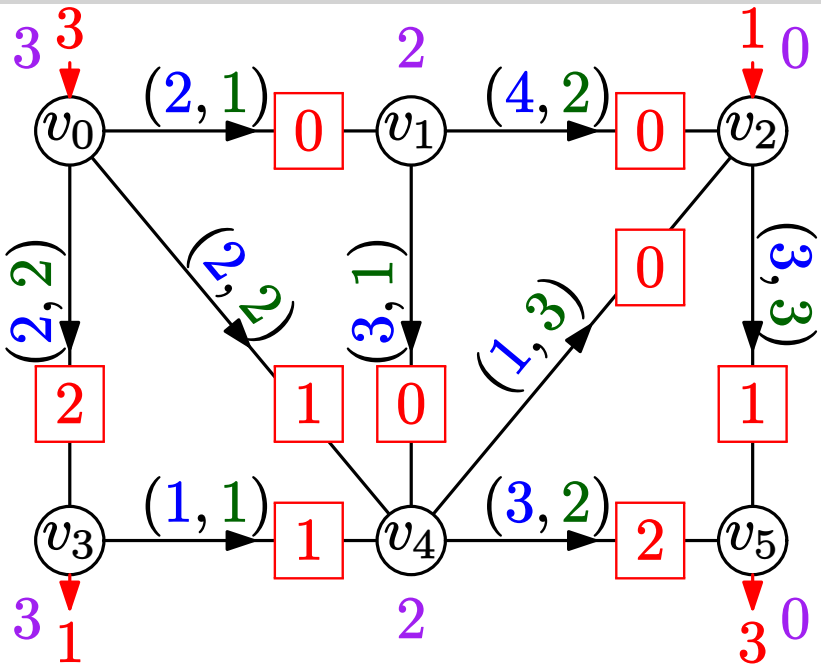


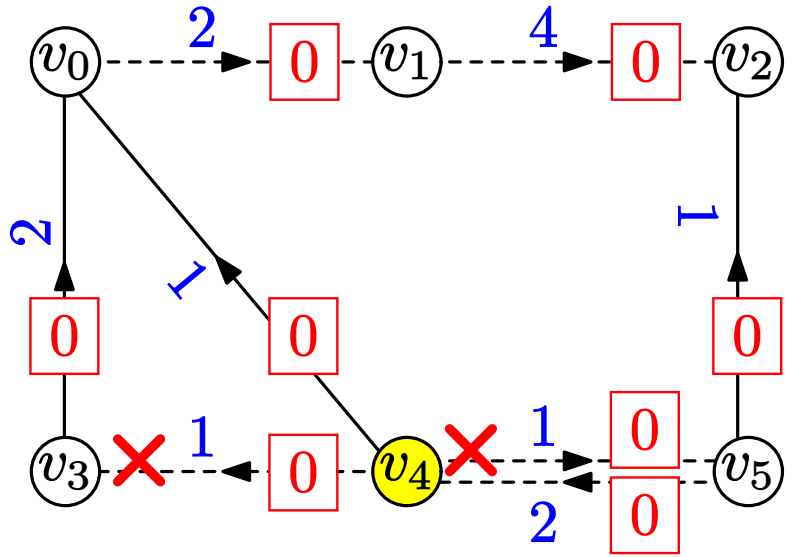
v_4-v_4 流で，
v_4 への流入量を最大に
するものを見つける

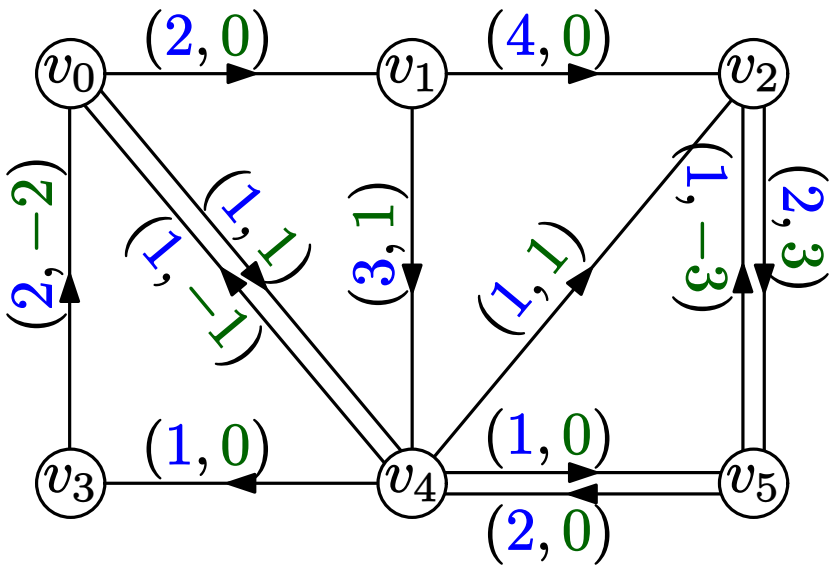
v_4 から到達できる (v_4 以外の)
頂点の集合 W を定める

W の頂点のポテンシャルを
上げる

主双対法：例 (6/8)

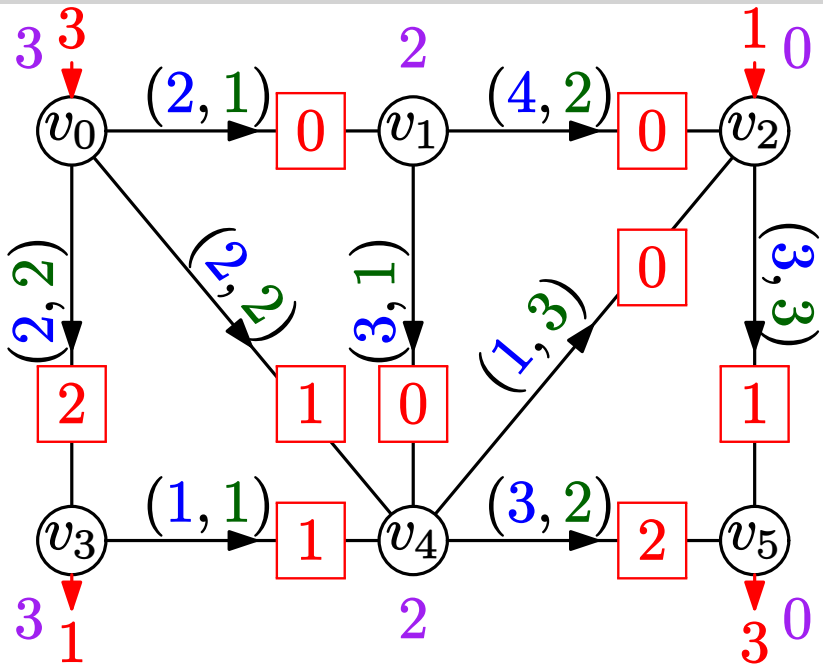


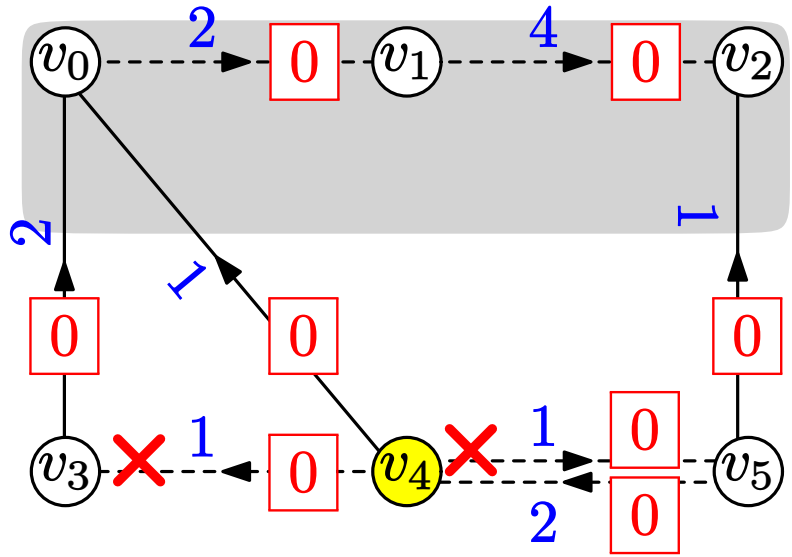


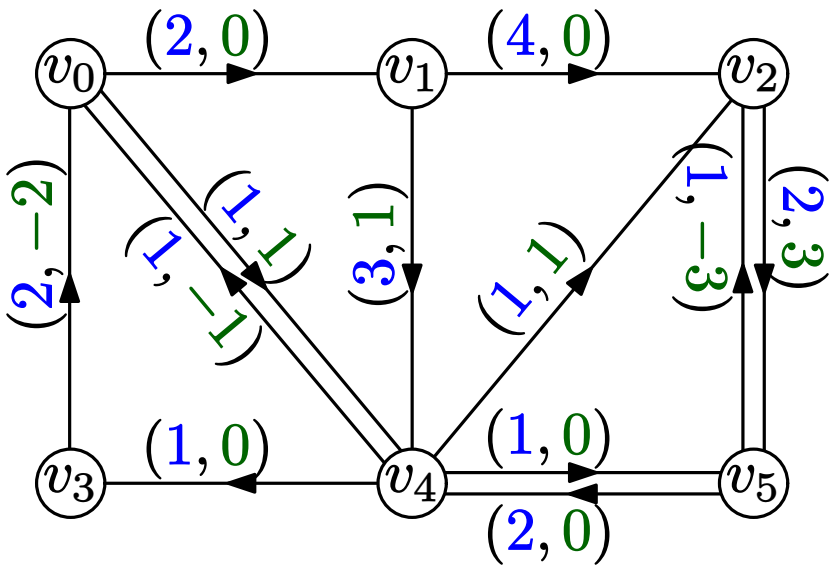


v_4-v_4 流で，
v_4 への流入量を最大に
するものを見つける

主双対法：例 (6/8)



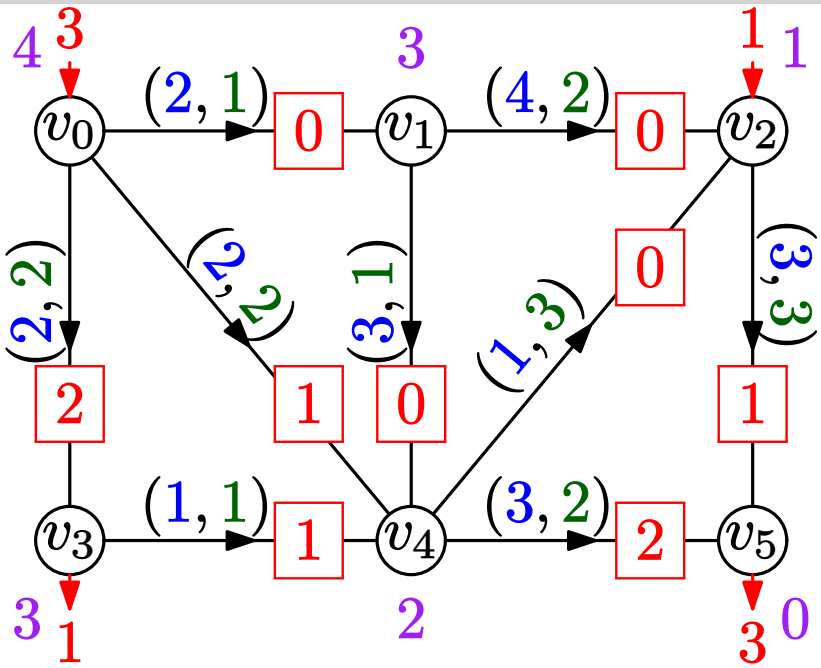


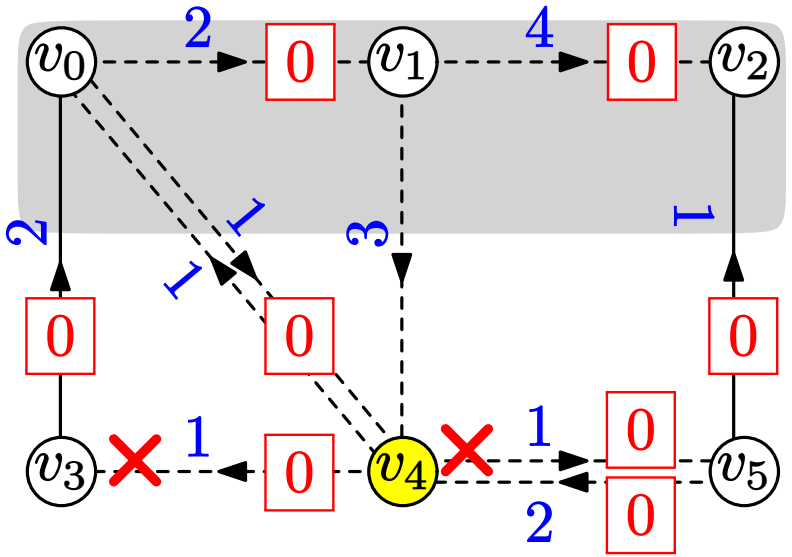


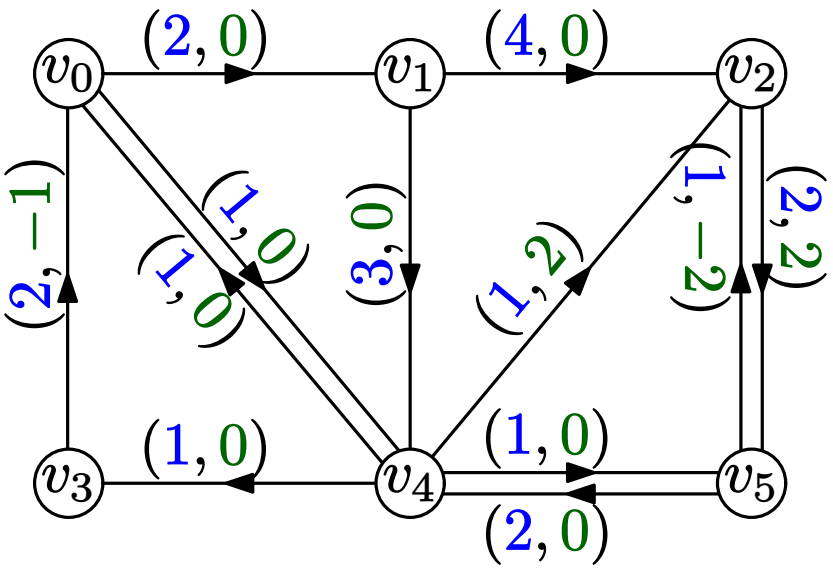
v_4-v_4 流で，
v_4 への流入量を最大に
するものを見つける

v_4 から到達できる (v_4 以外の)
頂点の集合 W を定める

主双対法：例 (6/8)





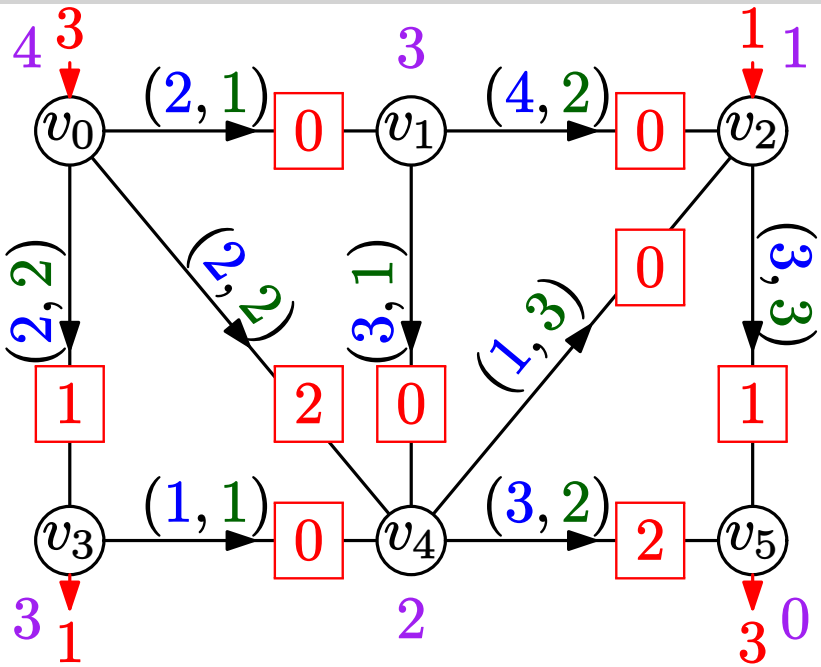


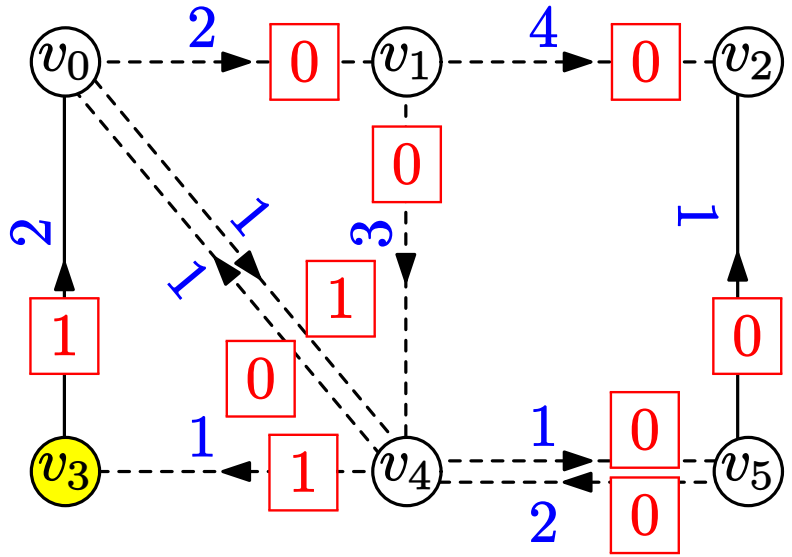
v_4-v_4 流で，
v_4 への流入量を最大に
するものを見つける

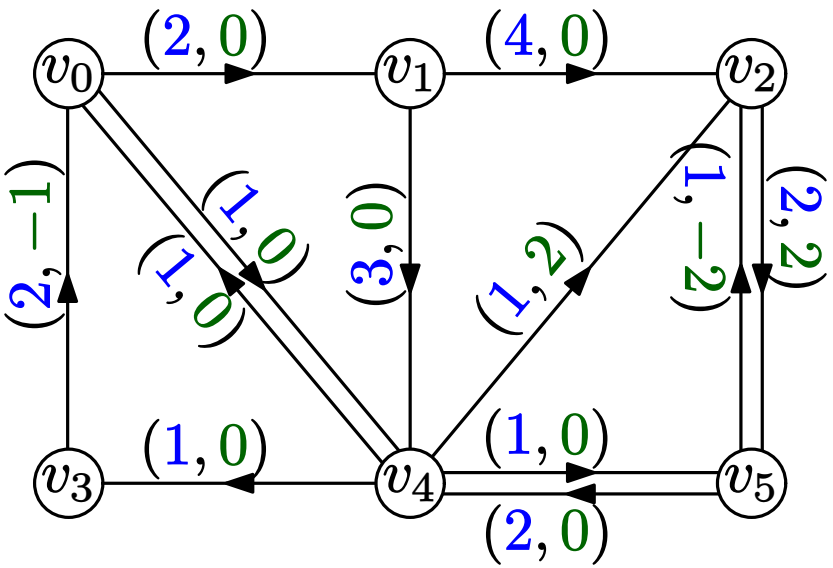
v_4 から到達できる (v_4 以外の)
頂点の集合 W を定める

W の頂点のポテンシャルを
上げる

主双対法：例 (7/8)

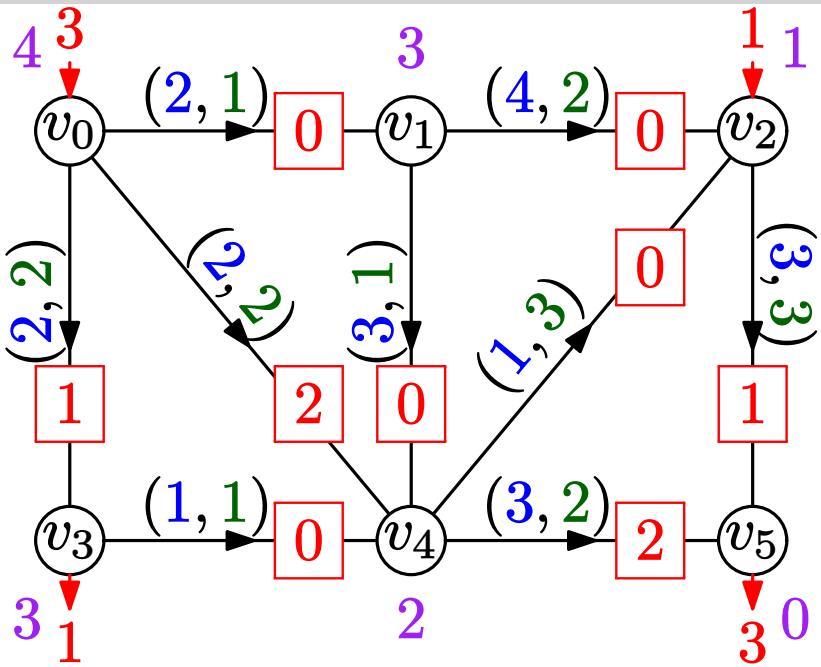


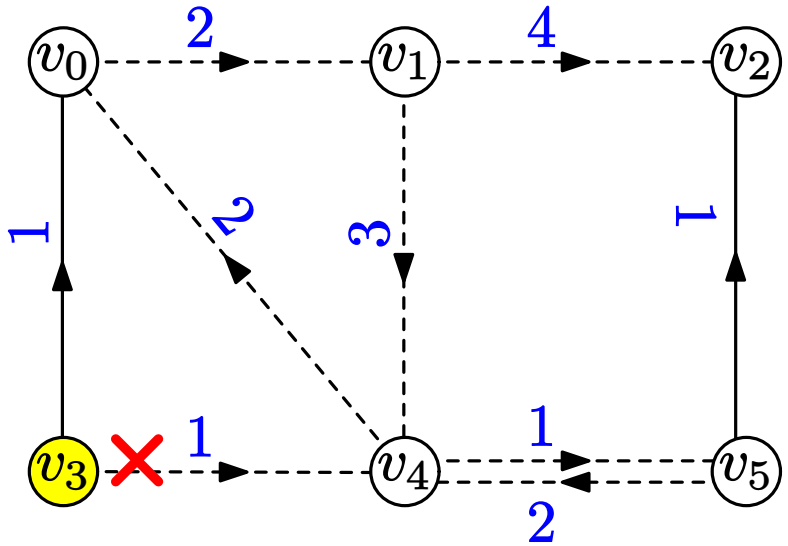




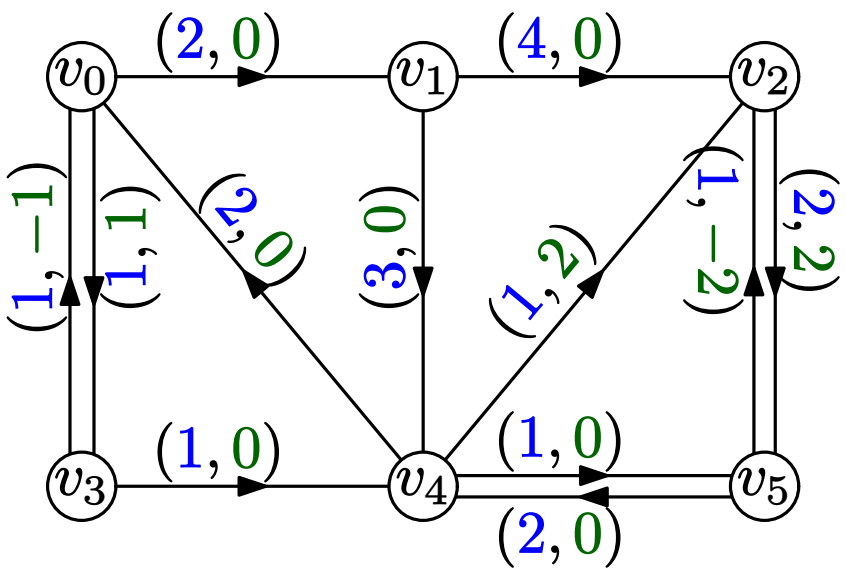
v_3-v_3 流で，
v_3 への流入量を最大に
するものを見つける

主双対法：例 (7/8)

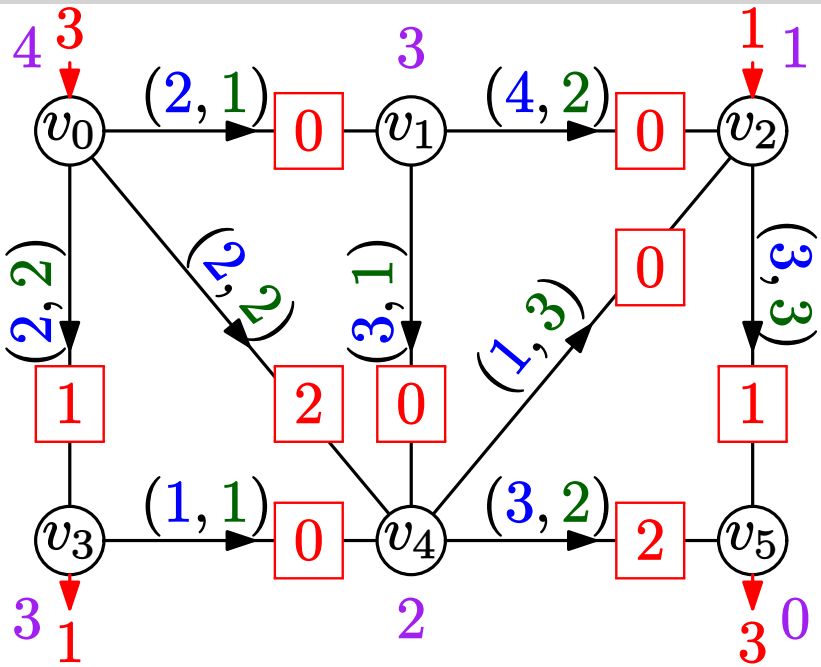


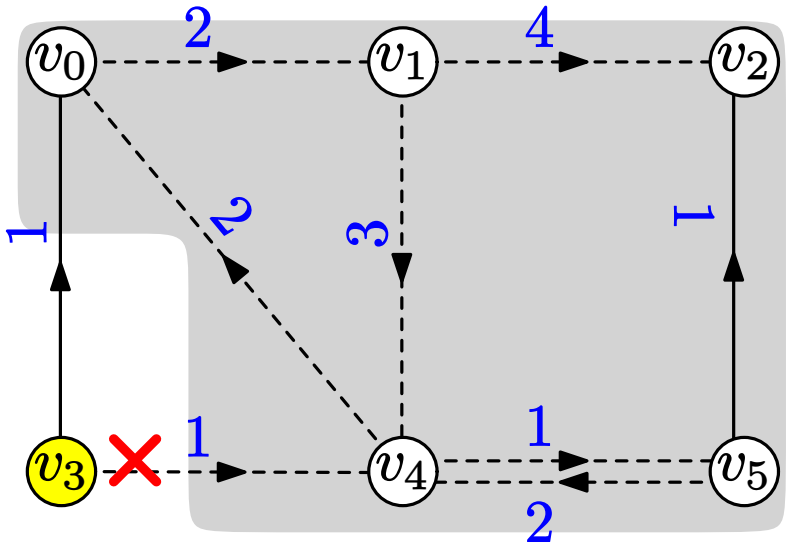


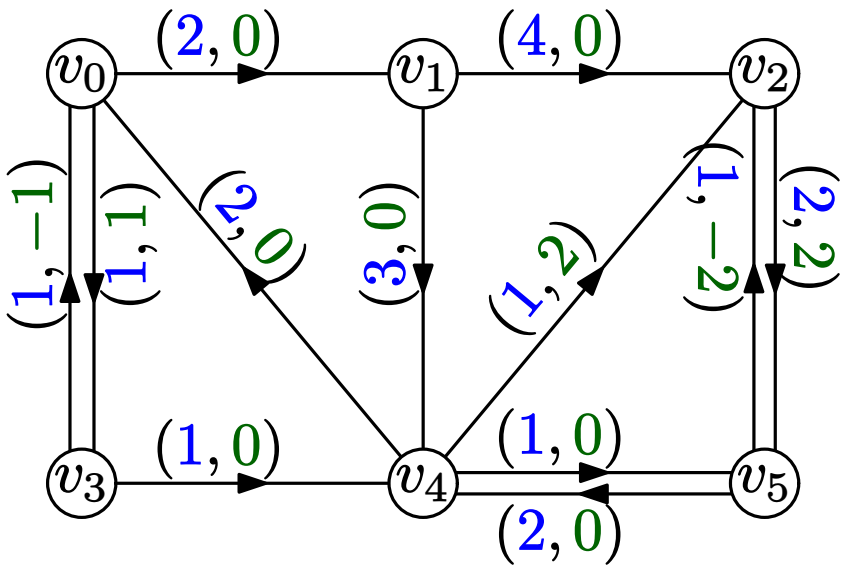
v_3-v_3 流で，
v_3 への流入量を最大に
するものを見つける



主双対法：例 (7/8)



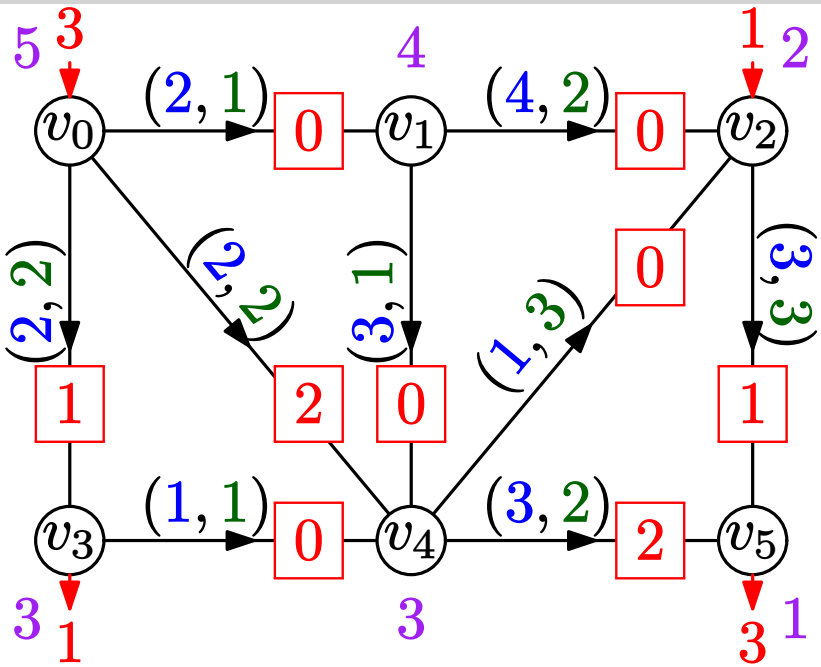


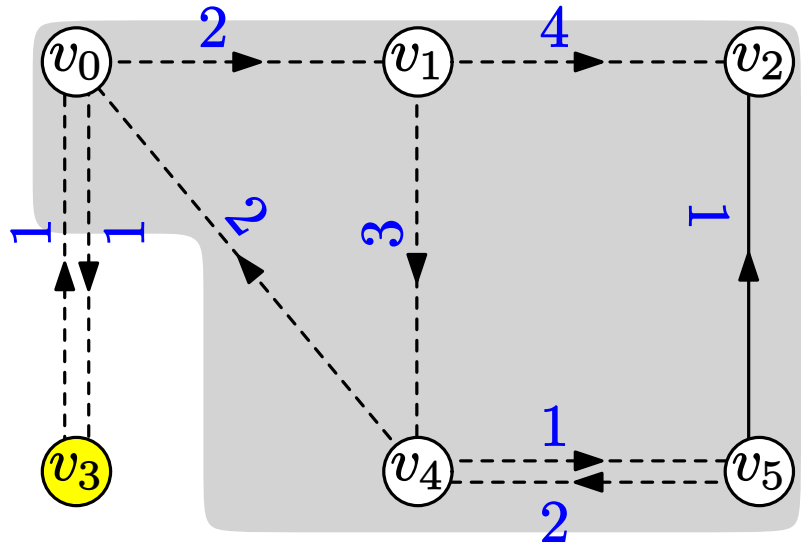


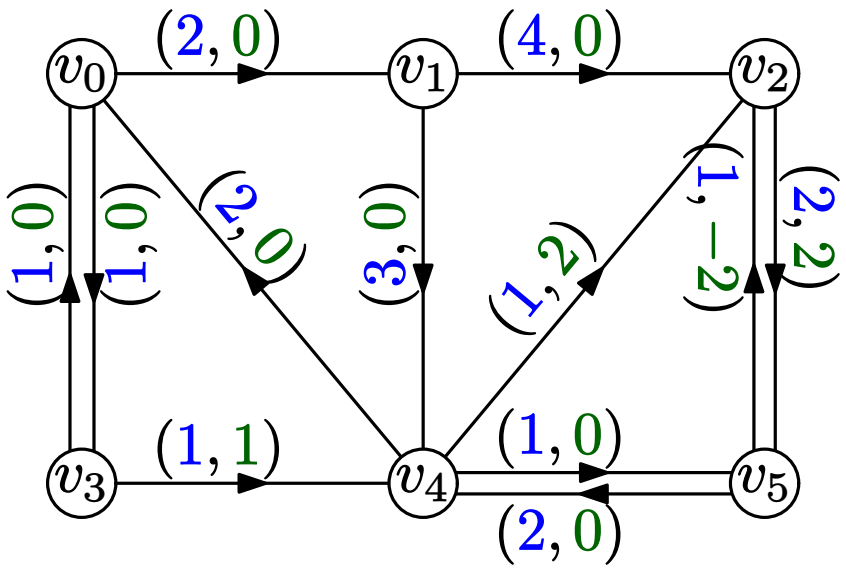
v_3-v_3 流で，
v_3 への流入量を最大に
するものを見つける

v_3 から到達できる (v_3 以外の)
頂点の集合 W を定める

主双対法：例 (7/8)





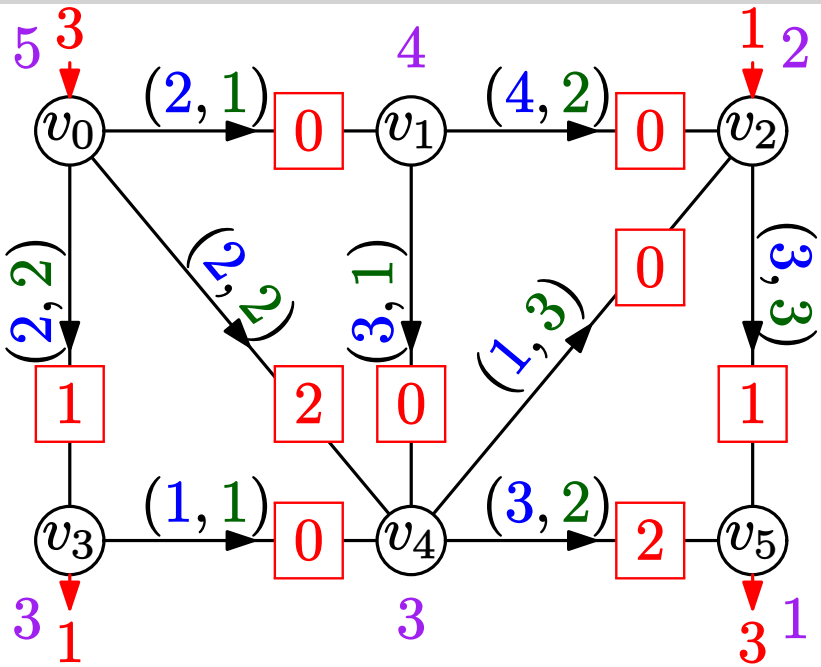


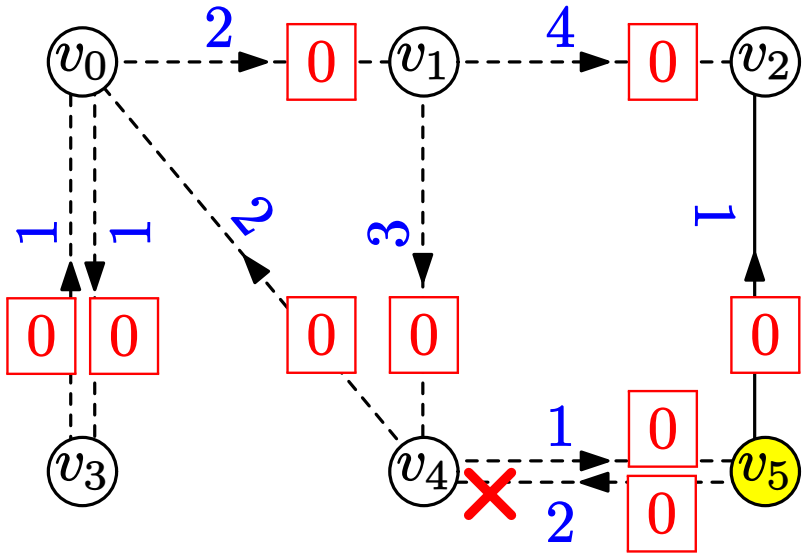
v_3-v_3 流で，
v_3 への流入量を最大に
するものを見つける

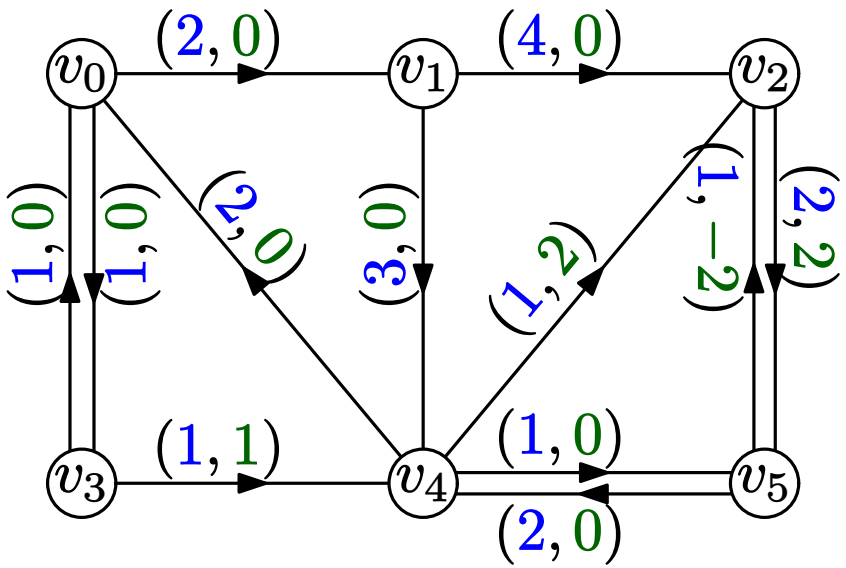
v_3 から到達できる (v_3 以外の)
頂点の集合 W を定める

W の頂点のポテンシャルを
上げる

主双対法：例 (8/8)

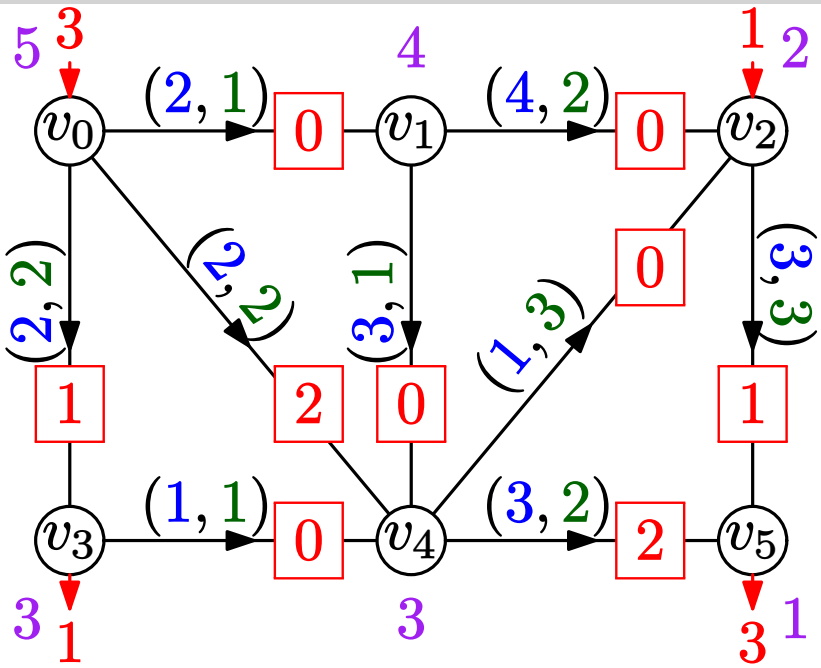


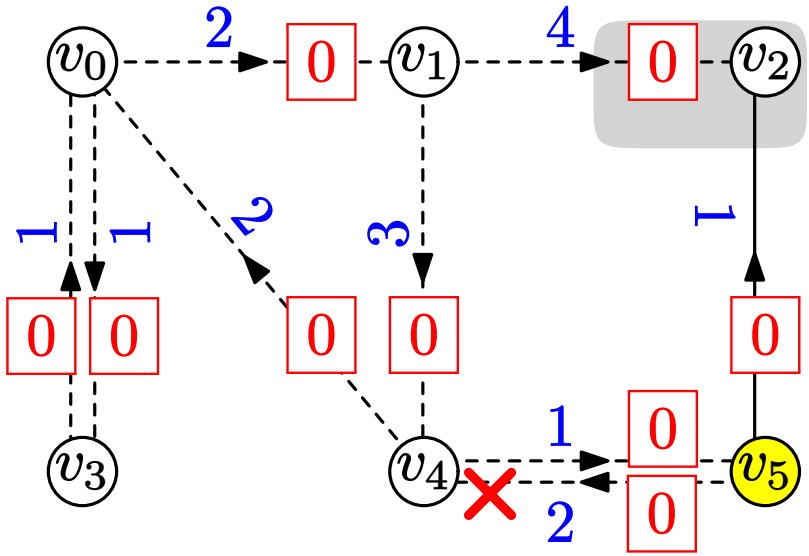


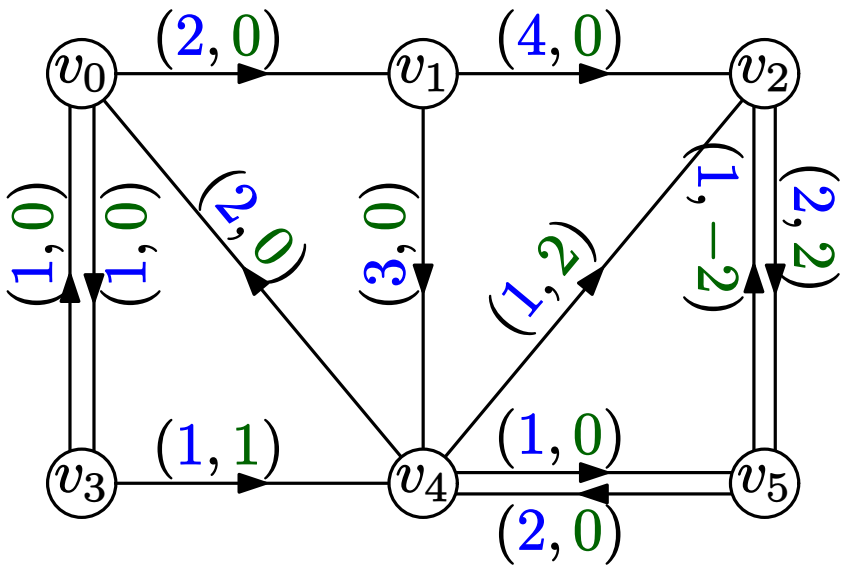


v_5-v_5 流で，
v_5 への流入量を最大に
するものを見つける

主双対法：例 (8/8)



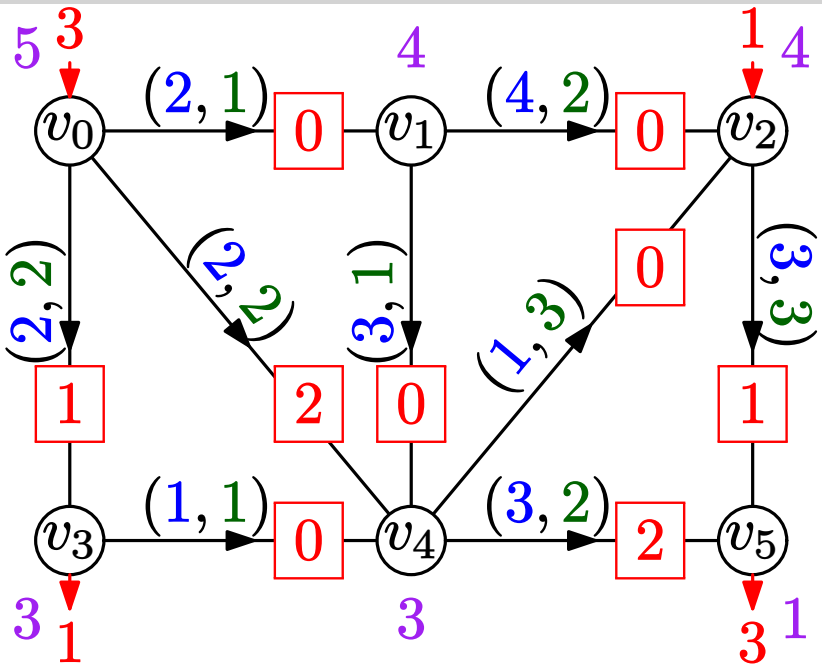


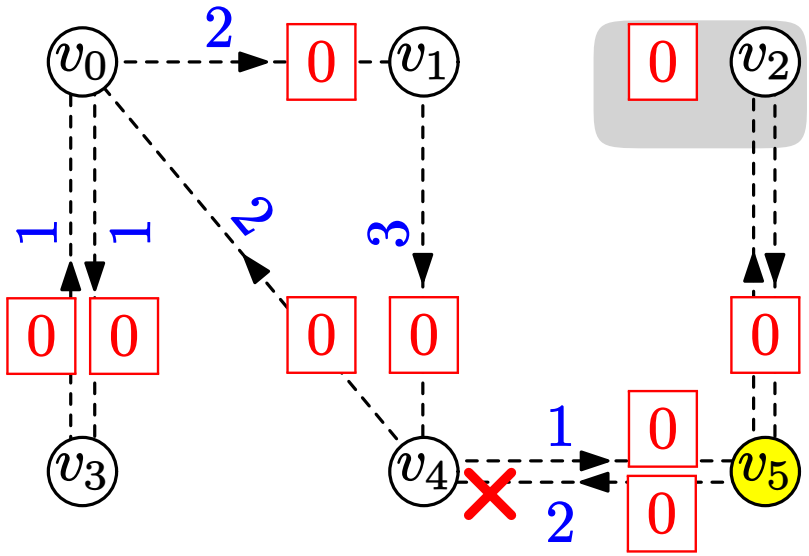


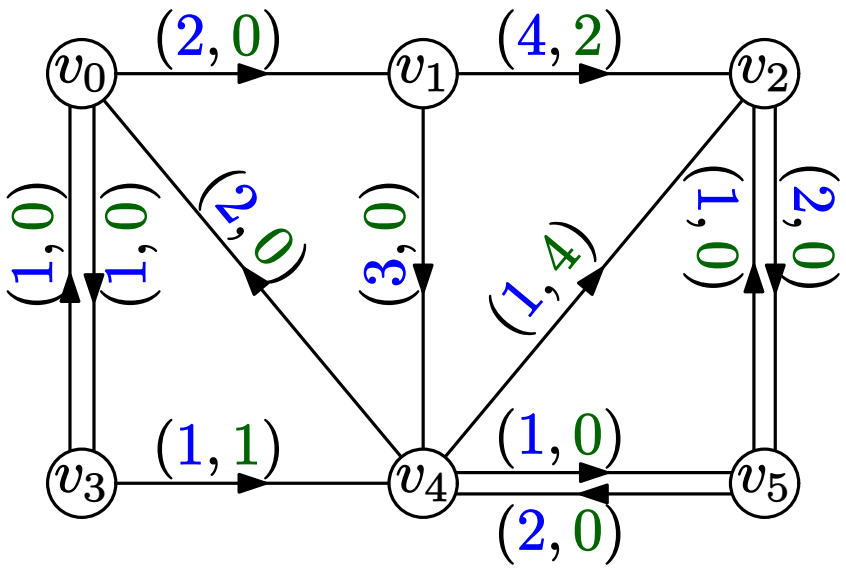
v_5-v_5 流で，
v_5 への流入量を最大に
するものを見つける

v_5 から到達できる (v_5 以外の)
頂点の集合 W を定める

主双対法：例 (8/8)







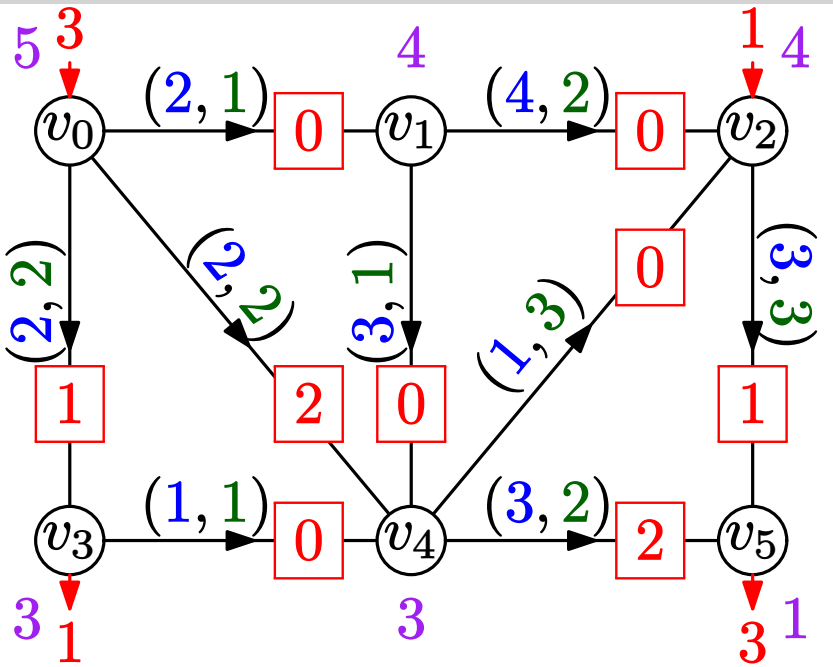
v_5-v_5 流で，
v_5 への流入量を最大に
するものを見つける

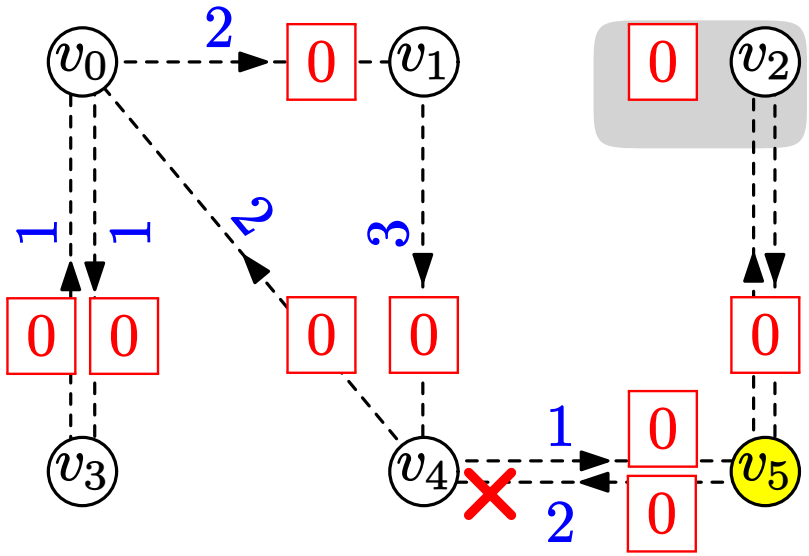
v_5 から到達できる (v_5 以外の)
頂点の集合 W を定める

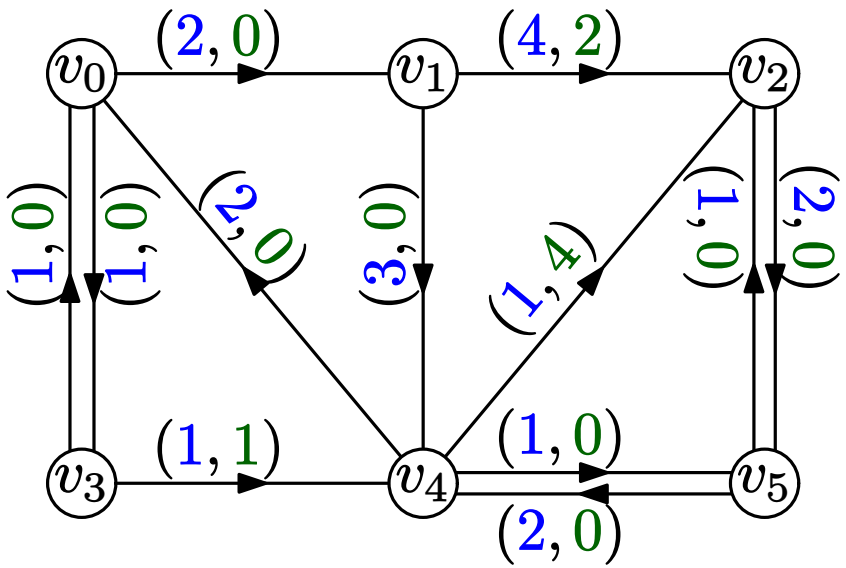
W の頂点のポテンシャルを
上げる

主双対法：例 (8/8)

最小費用 b-流！







v_5-v_5 流で，
v_5 への流入量を最大に
するものを見つける

v_5 から到達できる (v_5 以外の)
頂点の集合 W を定める

W の頂点のポテンシャルを
上げる

主双対法：流れ

アルゴリズム：主双対法 (primal-dual method)

- 初期化：b-流 f，ポテンシャル $p = 0$
- 反復：簡約費用が負の弧がある限り，以下を実行
  1. $v \leftarrow$ 簡約費用が負の弧の始点
  2. 簡約費用が非正の弧のみを用いる v-v 流で，v への流入が最大のものを発見
  3. それを用いて，f を更新
  4. $W \leftarrow v$ から簡約費用が非正の弧のみを用いて到達可能な頂点の集合
  5. W のポテンシャルを一律に増加
- f を出力

ステップ 2, 4 では，v を始点とする弧で簡約費用が 0 のものは使わない

主双対法：流れ

アルゴリズム：主双対法 (primal-dual method)

- 初期化：b-流 f，ポテンシャル $p = 0$
- 反復：簡約費用が負の弧がある限り，以下を実行
  1. $v \leftarrow$ 簡約費用が負の弧の始点
  2. 簡約費用が非正の弧のみを用いる v-v 流で，v への流入が最大のものを発見
  3. それを用いて，f を更新
  4. $W \leftarrow v$ から簡約費用が非正の弧のみを用いて到達可能な頂点の集合
  5. W のポテンシャルを一律に増加
- f を出力

簡約費用が負の弧が減る か
簡約費用が 0 の弧が増える まで

ステップ 2, 4 では，v を始点とす

まとめと予告

最小費用流問題

最適性条件 ⟶ アルゴリズム

- 簡約費用最適性条件
- 負閉路最適性条件
- 正カット最適性条件

- 逐次最短路法，主双対法
- 負閉路消去法
- 正カット消去法

次回の内容

- 主双対法の計算量
  ⇝ 擬多項式時間
- 主双対法の高速化 (費用スケーリング)
  ⇝ 弱多項式時間