離散最適化基礎論

第 12 回

最小費用流問題：逐次最短路法

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2024 年 1 月 9 日

最終更新：2024 年 1 月 10 日　13:26

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)

[復習] b-流：定義

設定：有向グラフ $G = (V, A)$，弧容量関数 $u\colon A \to \mathbb{R}_+$

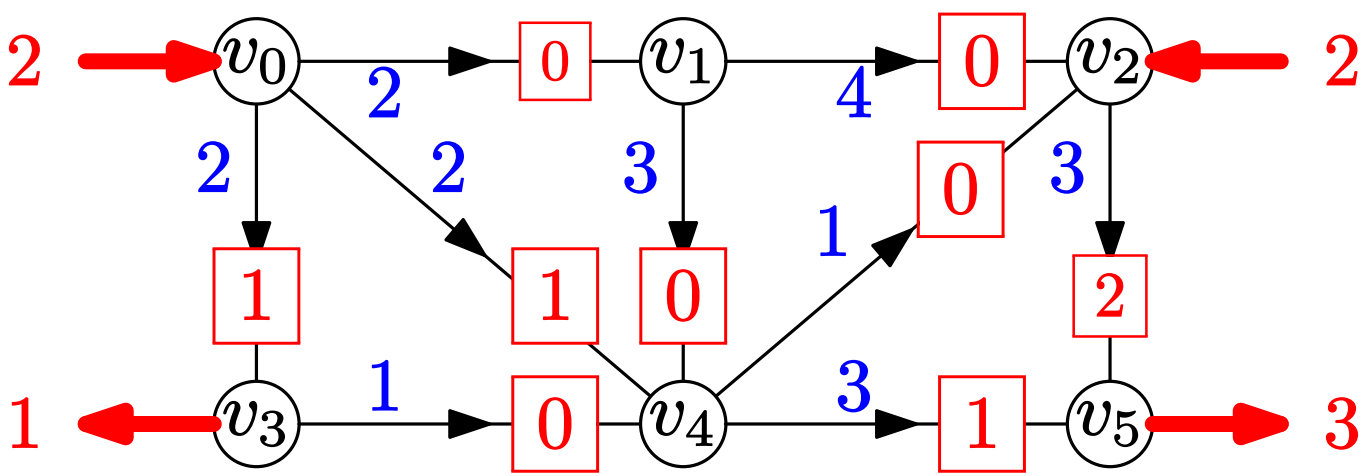
$b\colon V \to \mathbb{R}$ (負の値をとってもよい)

定義：b-流 (b-flow)

b-流 とは 次を満たす関数 $f\colon A \to \mathbb{R}_+$ のこと

1. 任意の弧 $a \in A$ に対して，$0 \le f(a) \le u(a)$
2. 任意の頂点 $v \in V$ に対して，

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



$$b(v_0) = 2,$$
$$b(v_1) = 0,$$
$$b(v_2) = 2,$$
$$b(v_3) = -1,$$
$$b(v_4) = 0,$$
$$b(v_5) = -3$$

[復習] b-流の費用：定義

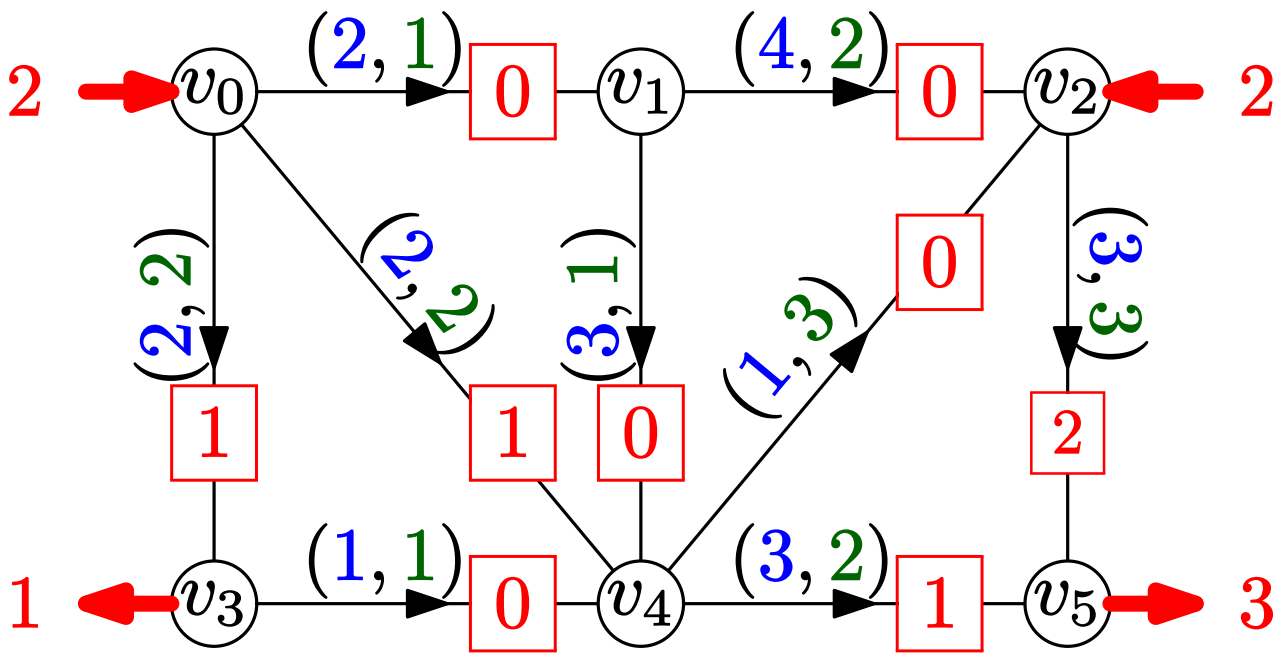
設定：有向グラフ $G = (V, A)$，弧容量関数 $u\colon A \to \mathbb{R}_+$

弧費用関数 $c\colon A \to \mathbb{R}_+$，$b\colon V \to \mathbb{R}$，$b$-流 $f\colon A \to \mathbb{R}_+$

定義：b-流の費用

b-流 f の**費用** を次の式で定義する

$$\mathsf{cost}(f) = \sum_{a \in A} c(a) f(a)$$



$$\begin{aligned}\mathsf{cost}(f) &= 0\cdot 1 + 1\cdot 2 + 1\cdot 2 \\ &\quad + 0\cdot 2 + 0\cdot 1 + 2\cdot 3 \\ &\quad + 0\cdot 1 + 0\cdot 3 + 1\cdot 2 \\ &= 12\end{aligned}$$

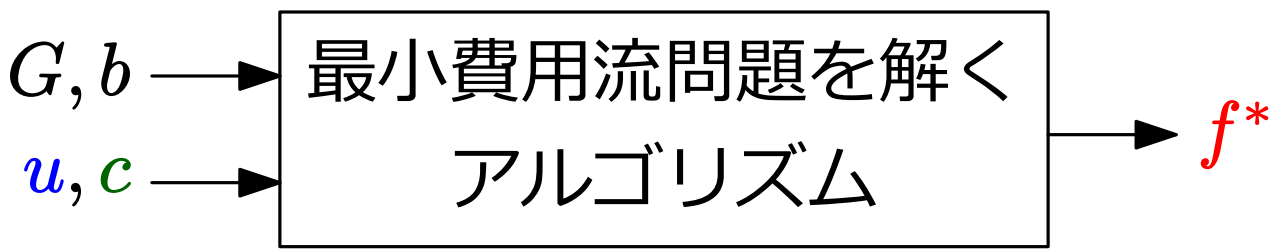
[復習] 最小費用流問題：定義

定義：最小費用流問題 (minimum-cost flow problem)

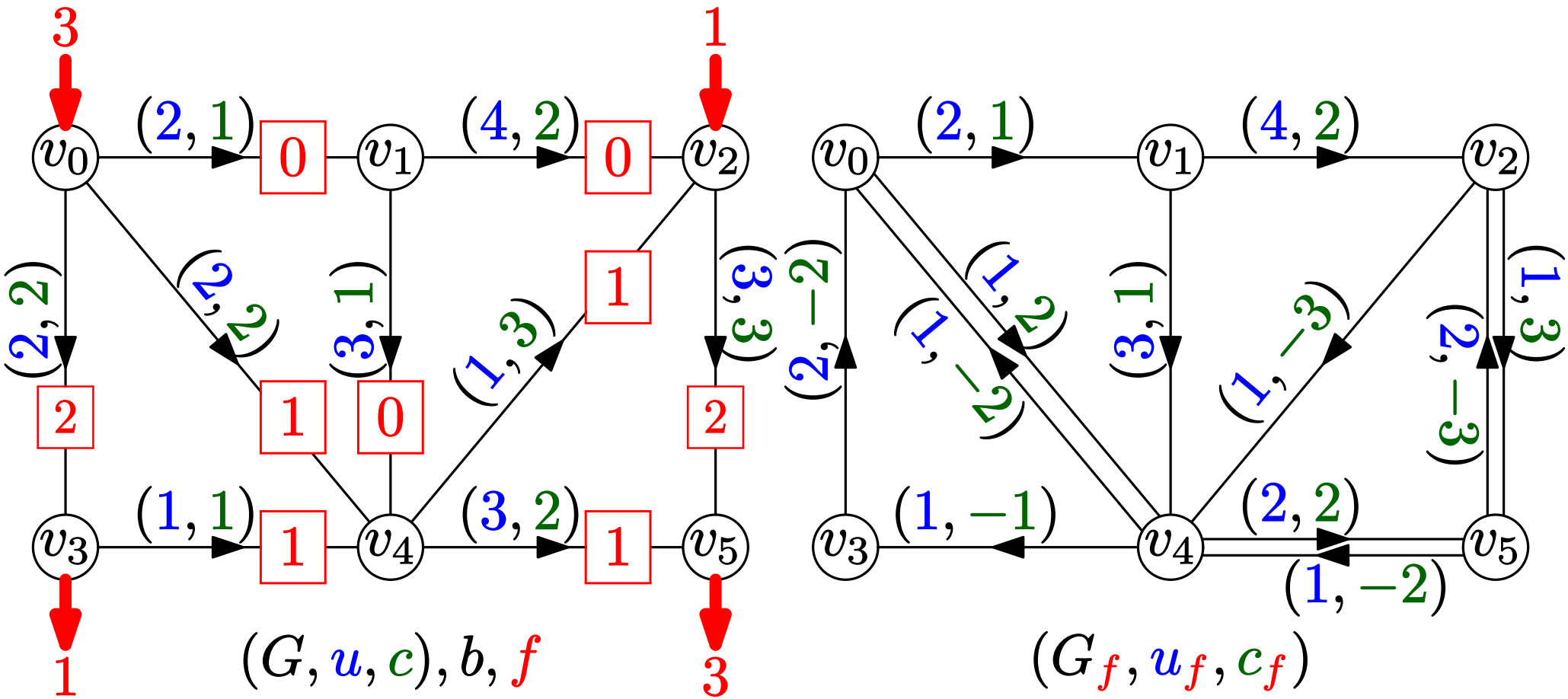
入力
- 有向グラフ $G = (V, A)$，関数 $b\colon V \to \mathbb{R}$
- 弧容量関数 $u\colon A \to \mathbb{R}_+$，弧費用関数 $c\colon A \to \mathbb{R}_+$

出力
- ネットワーク (G, u, c) に対する最小費用 b-流 f^*

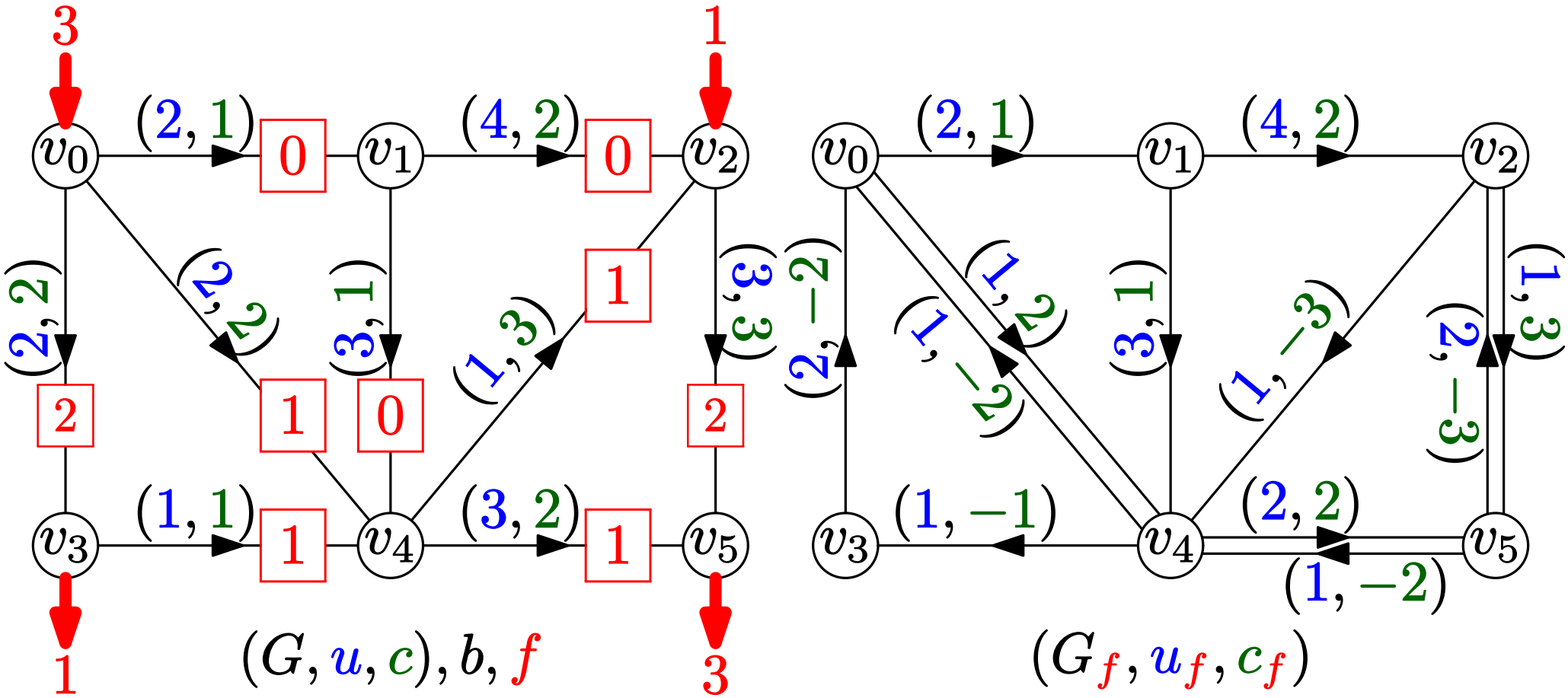
任意の b-流 f に対して，$\mathrm{cost}(f^*) \leq \mathrm{cost}(f)$



[復習] 補助ネットワーク：例



[復習] 補助ネットワーク：例



流 $f(a)$

(容量，費用) $(u(a), c(a))$

逆向き（容量，費用） $(f(a), -c(a))$

順向き（容量，費用） $(u(a)-f(a), c(a))$

[復習] 簡約費用最適性条件

設定：有向グラフ $G = (V, A)$，弧容量関数 $u\colon A \to \mathbb{R}_+$，
弧費用関数 $c\colon A \to \mathbb{R}_+$，$b\colon V \to \mathbb{R}$，$b$-流 $f\colon A \to \mathbb{R}_+$

性質：簡約費用最適性条件 (Ford, Fulkerson '62)

f が (G, u, c) における最小費用 b-流 $\Leftrightarrow$
次を満たすポテンシャル $p\colon V \to \mathbb{R}$ が存在

- 任意の弧 $uv \in A_f$ に対して，$c_f(uv) - p(u) + p(v) \geq 0$

uv の簡約費用

前回のまとめ 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

- 今回
- 負閉路消去法
- 正カット消去法

今回の内容

- アルゴリズム：逐次最短路法 (最短路繰り返し法)

逐次最短路法と相補性定理

相補性定理：f, y, z が最適解

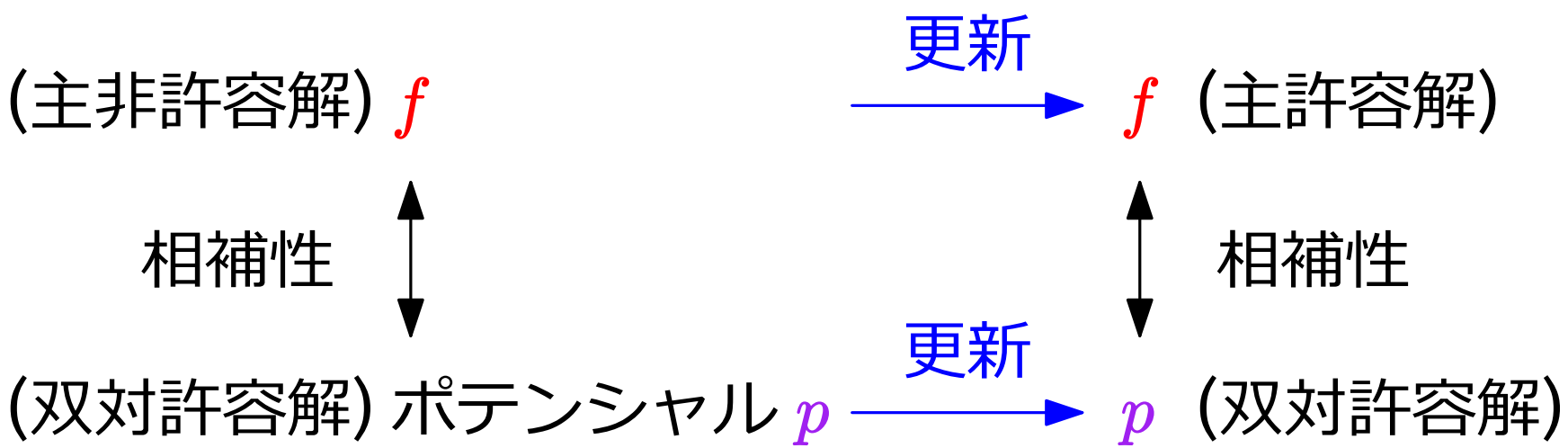
$\Leftrightarrow$ 主許容性 + 双対許容性 + 相補性条件

逐次最短路法に対する視点

双対許容性

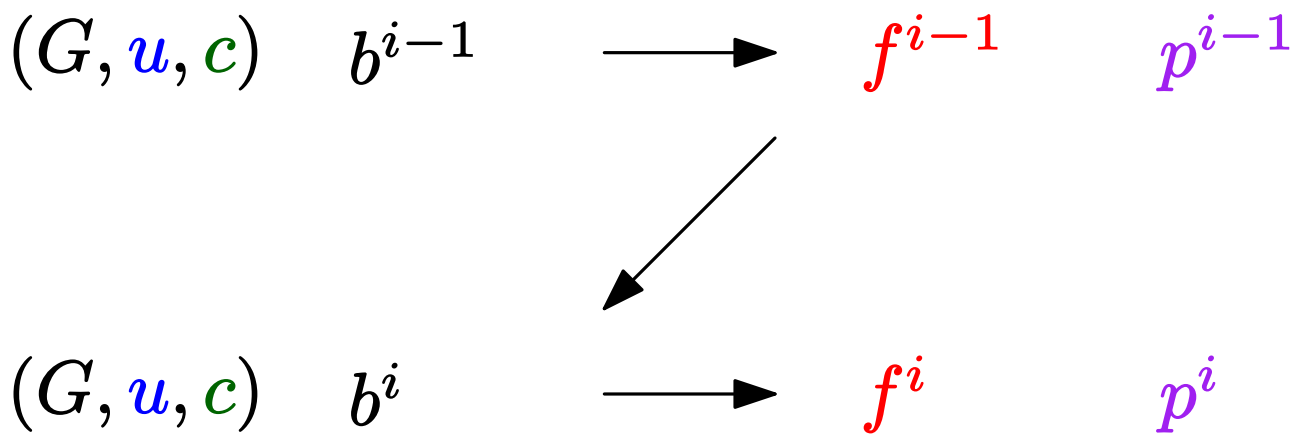
を満たしたまま **主許容性** を満たしに行く

相補性



本日の内容

1. **逐次最短路法**
2. 逐次最短路法：計算量



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

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

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

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

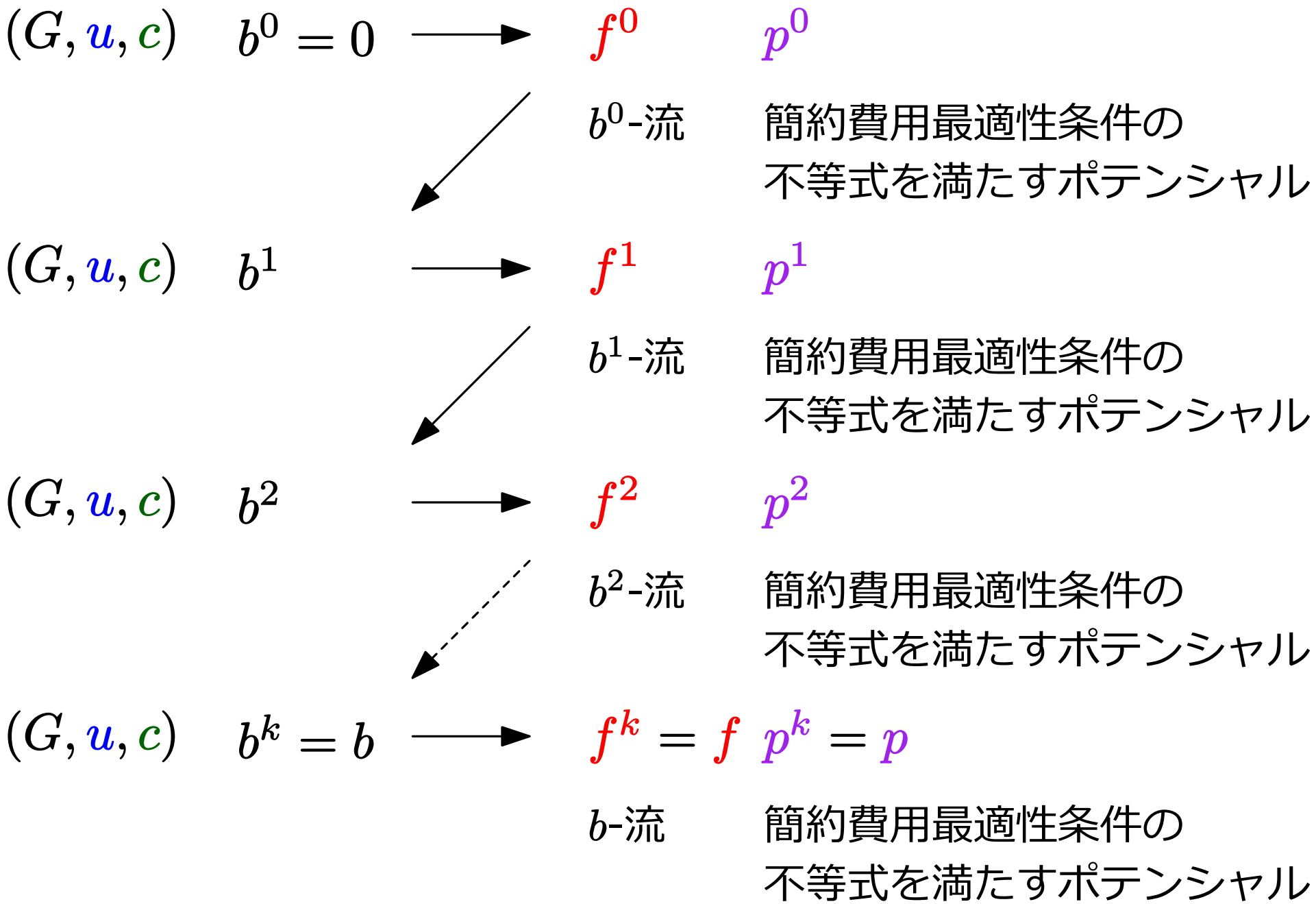
(G, u, c) b^2 → f^2 p^2

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

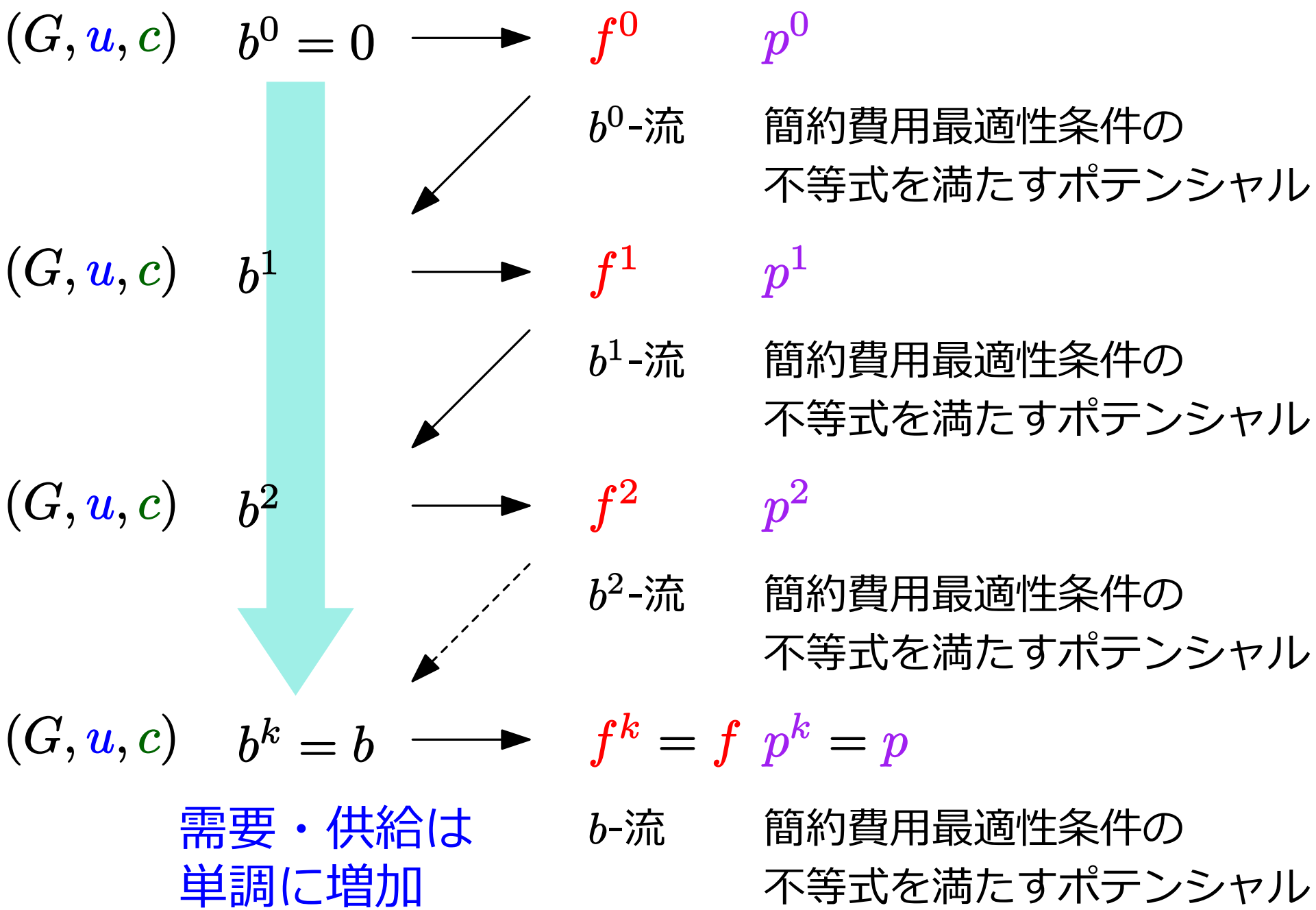
(G, u, c) b → f p

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

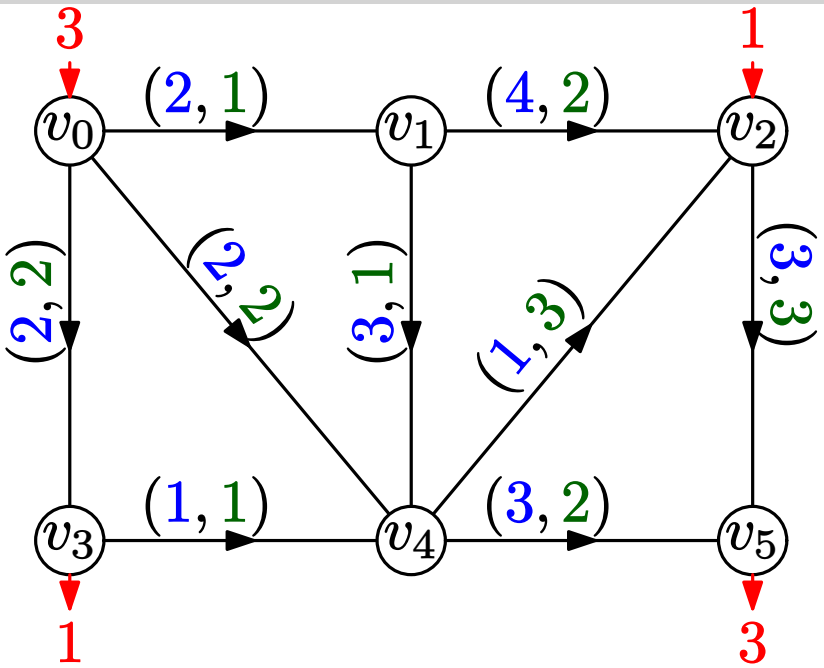
逐次最短路法：基本的な考え方



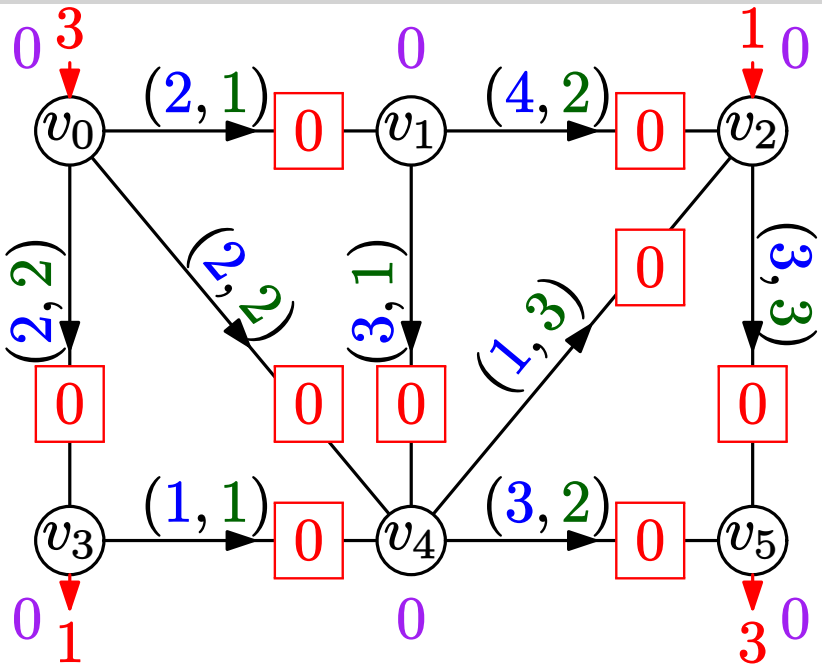
逐次最短路法：基本的な考え方



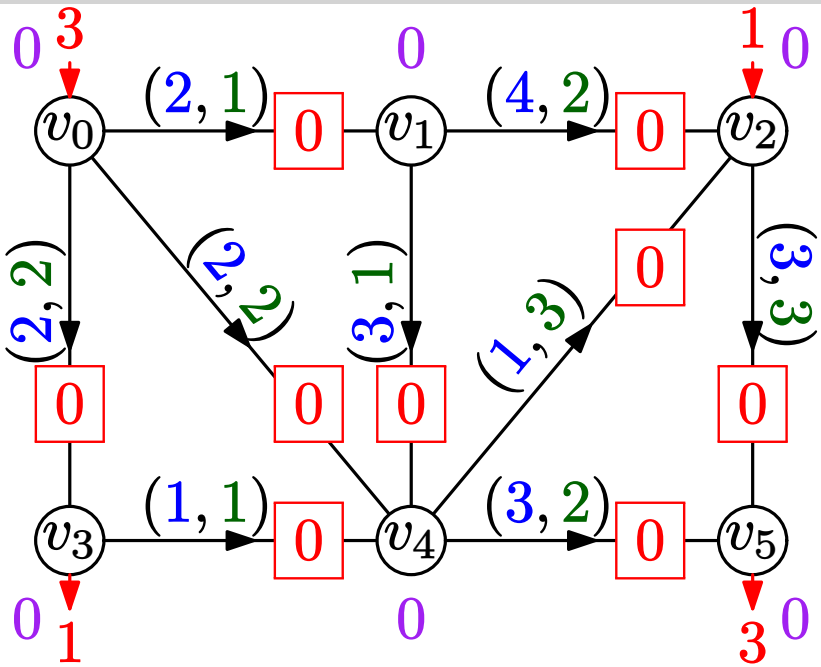
逐次最短路法：例 (1/4)

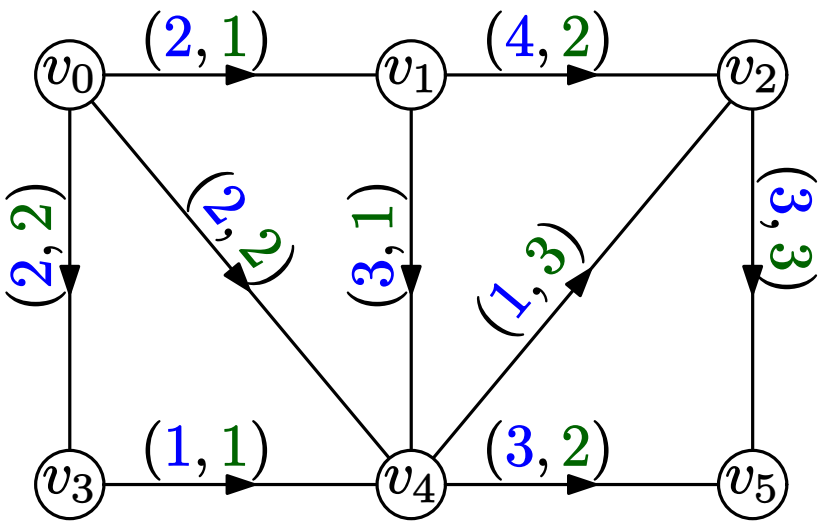


逐次最短路法：例 (1/4)

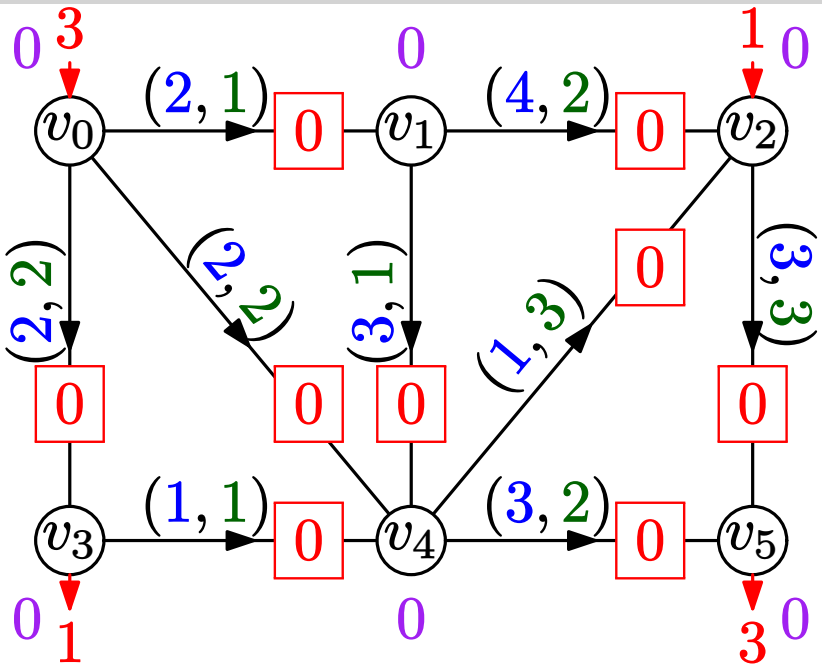

0
3
0
1
0
(2, 1)
(4, 2)
v_0
v_1
v_2
0
0
0
(2, 2)
(2, 2)
(3, 1)
(1, 3)
(3, 3)
0
0
0
0
(1, 1)
(3, 2)
v_3
v_4
v_5
0
0
0
0
1
0
3
0

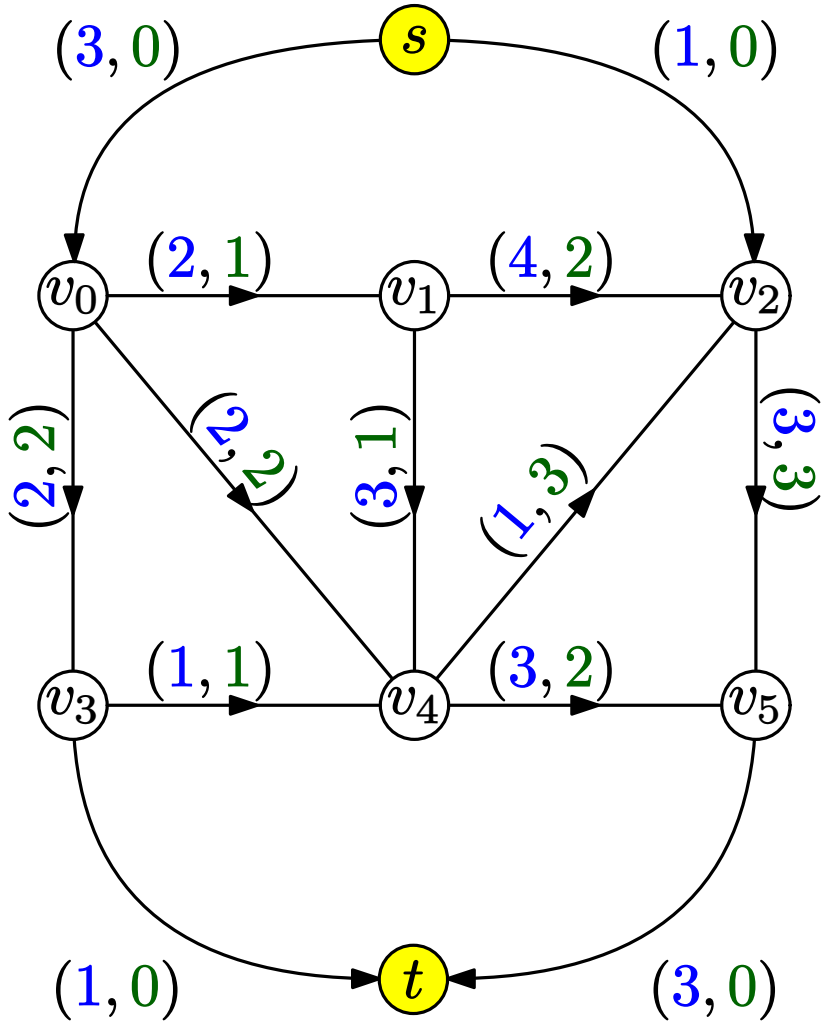
逐次最短路法：例 (1/4)

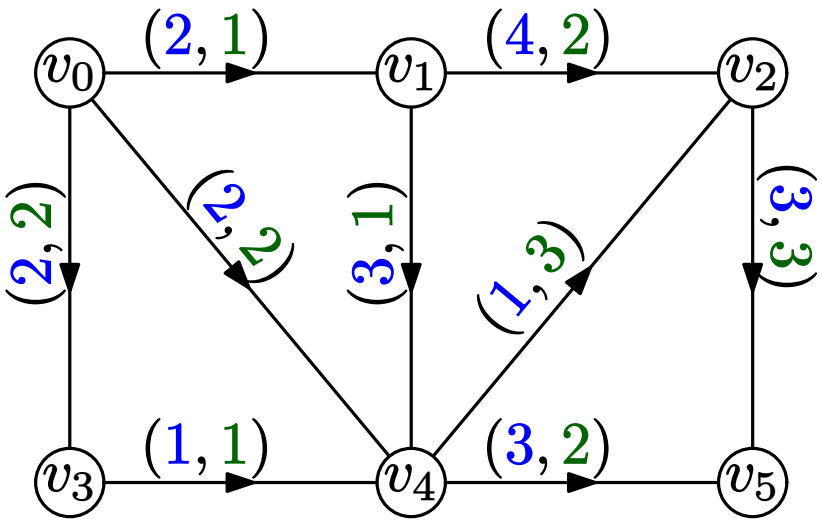

0 3
0
1 0
(2, 1)
(4, 2)
v_0
v_1
v_2
0
0
0
(2, 2)
(2, 2)
(3, 1)
(1, 3)
(3, 3)
0
0
0
0
(1, 1)
(3, 2)
v_3
v_4
v_5
0
0
0
0 1
0
3 0


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

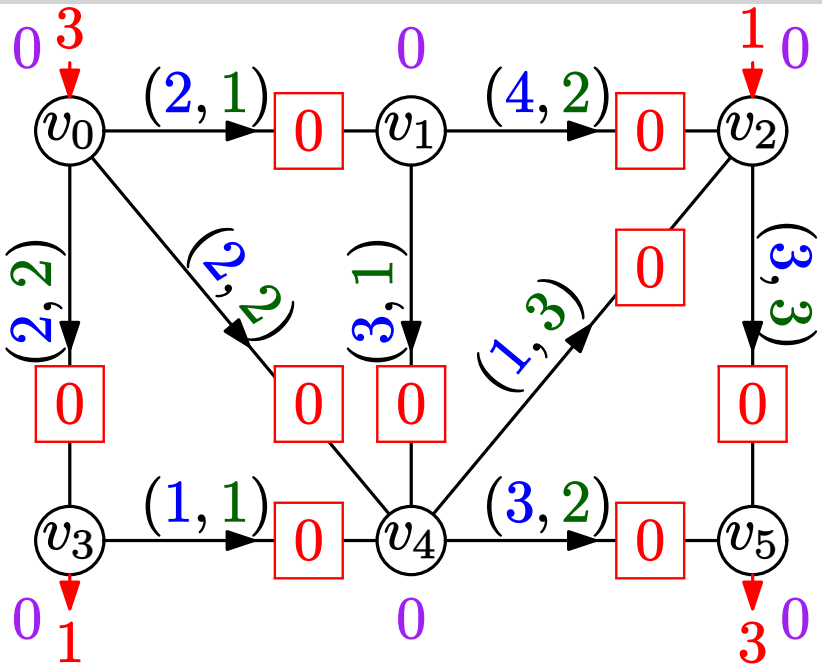
逐次最短路法：例 (1/4)

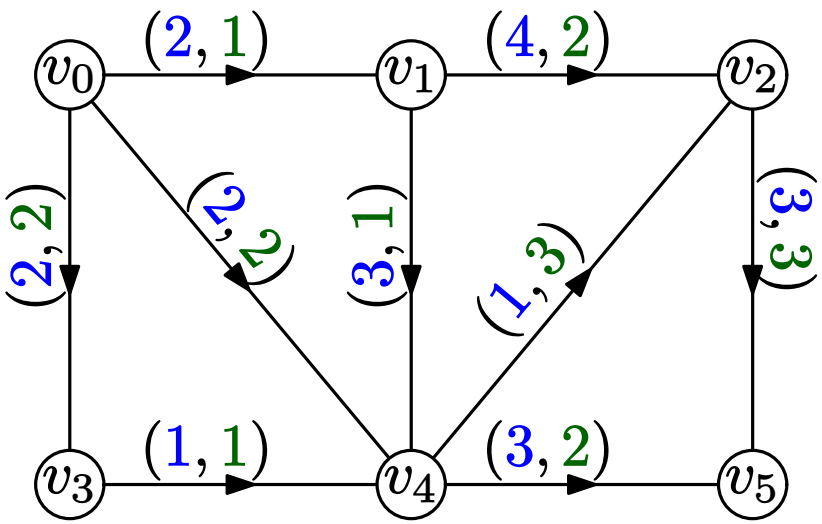

0 3
0
1 0
(2, 1)
(4, 2)
v_0
v_1
v_2
0
0
0
(2, 2)
(2, 2)
(3, 1)
(1, 3)
(3, 3)
0
0
0
0
(1, 1)
(3, 2)
v_3
v_4
v_5
0
0
0
0 1
0
3 0

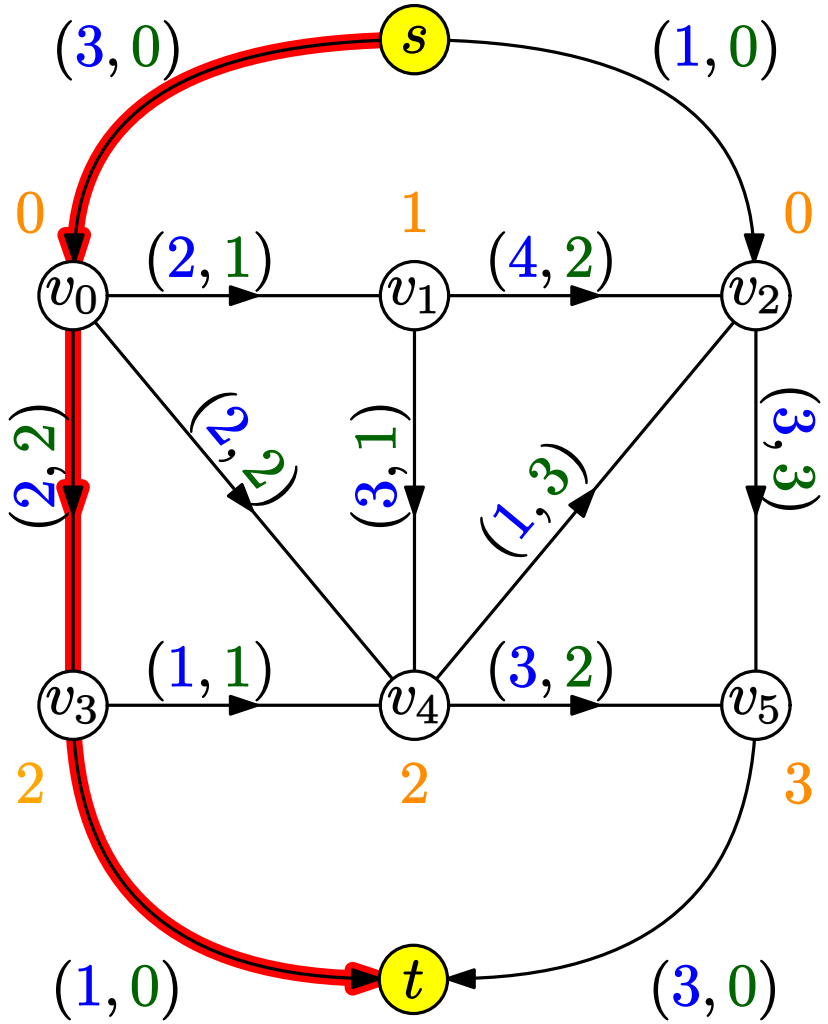

(3, 0)
s
(1, 0)
v_0
(2, 1)
v_1
(4, 2)
v_2
(2, 2)
(2, 2)
(3, 1)
(1, 3)
(3, 3)
(1, 1)
(3, 2)
v_3
v_4
v_5
(1, 0)
t
(3, 0)


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

逐次最短路法：例 (1/4)

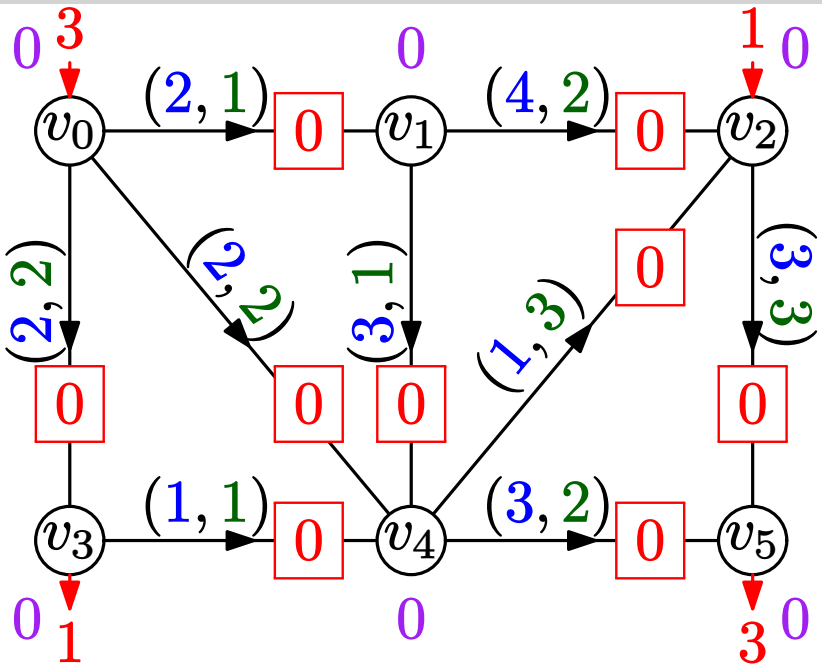


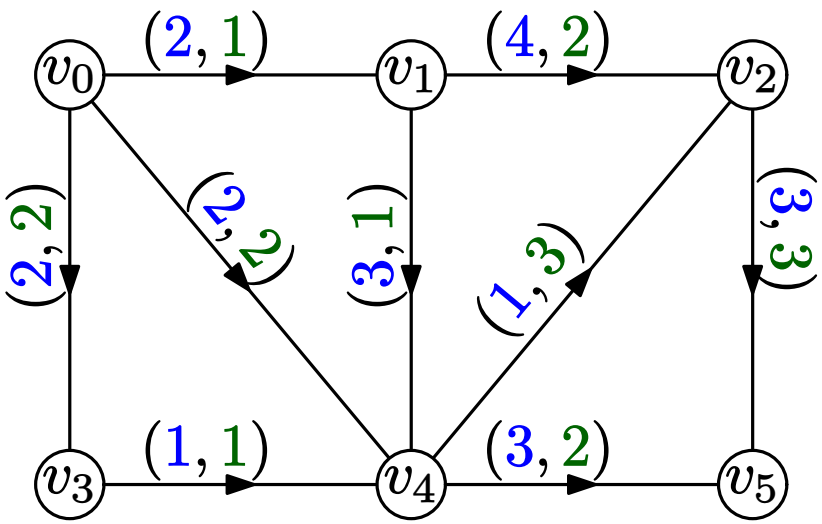


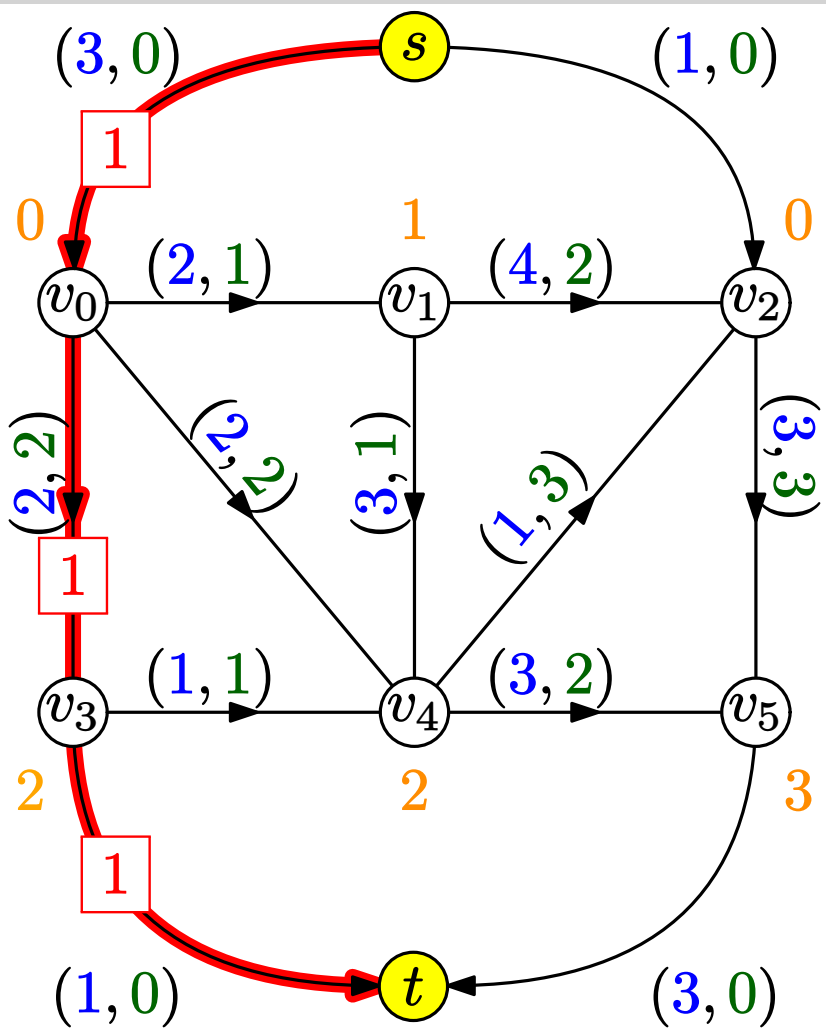


簡約費用を距離とする最短 s-t 路

逐次最短路法：例 (1/4)



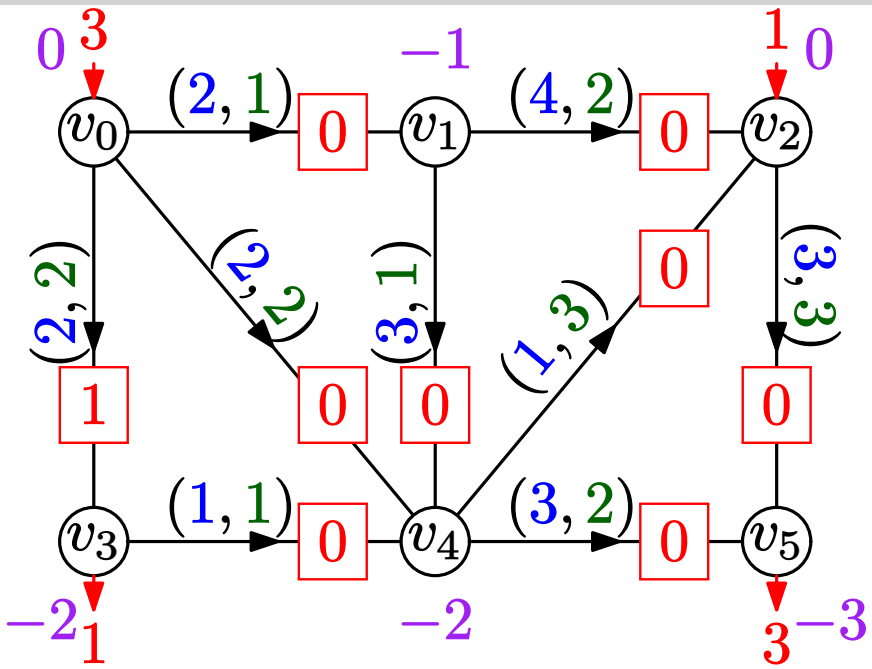




簡約費用を距離とする最短 s-t 路

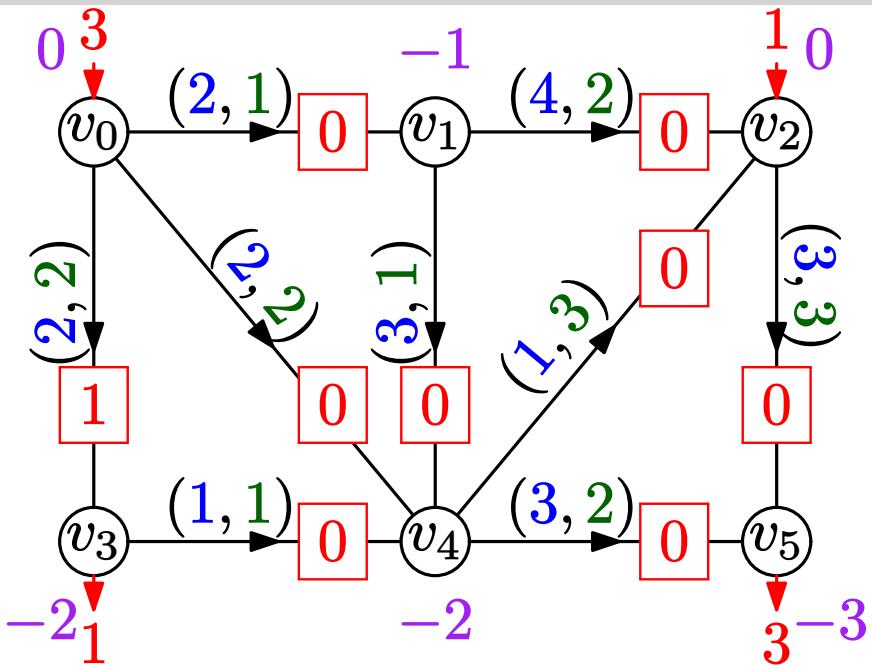
流せるだけ流す

逐次最短路法：例 (2/4)

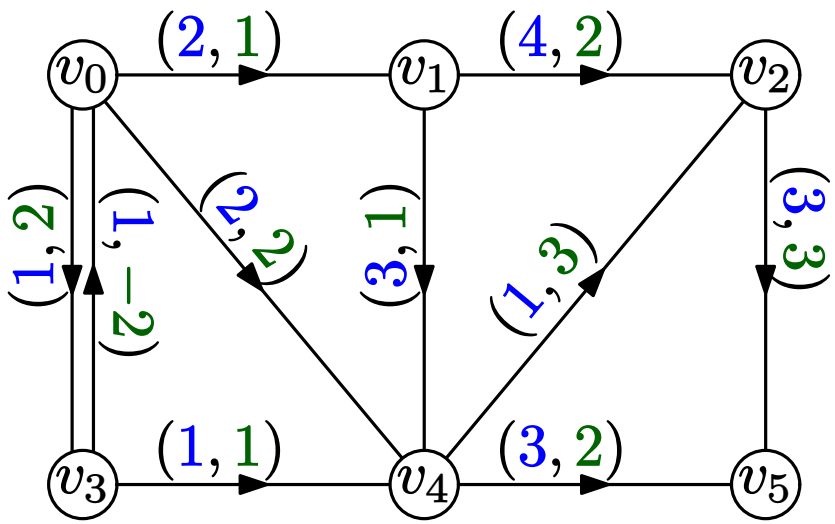


流とポテンシャルの更新

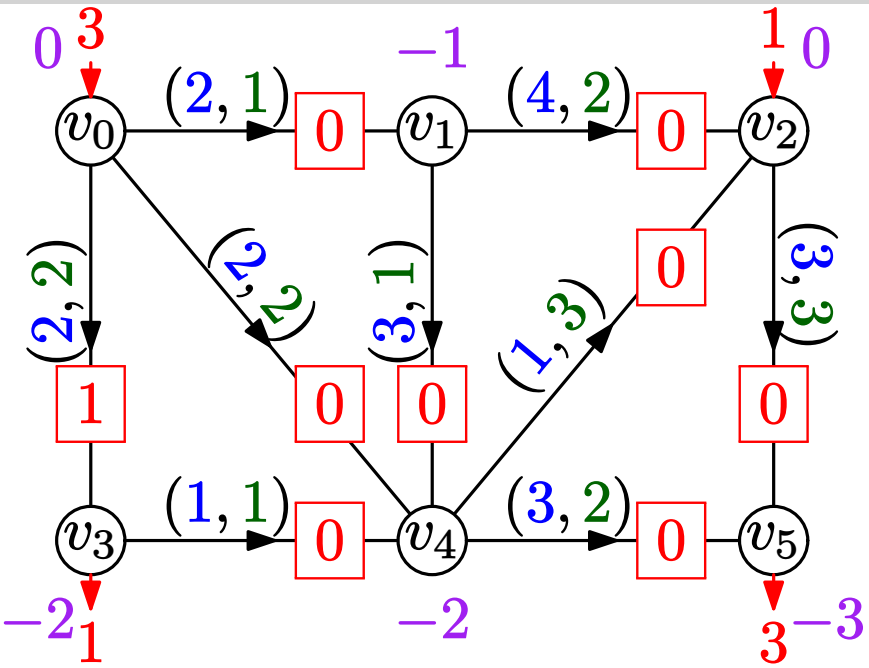
逐次最短路法：例 (2/4)



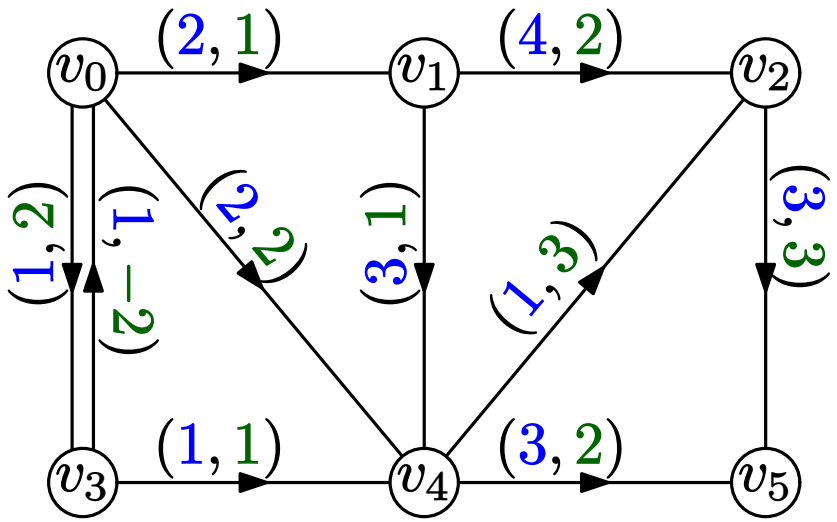
流とポテンシャルの更新

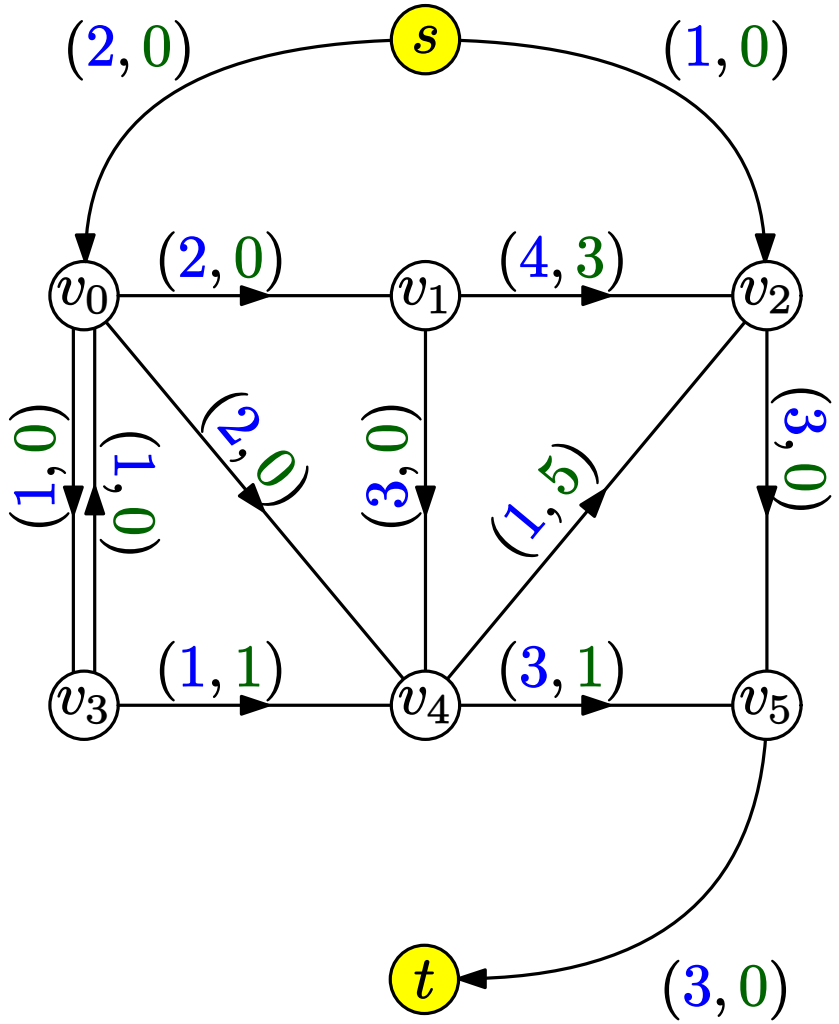


逐次最短路法：例 (2/4)

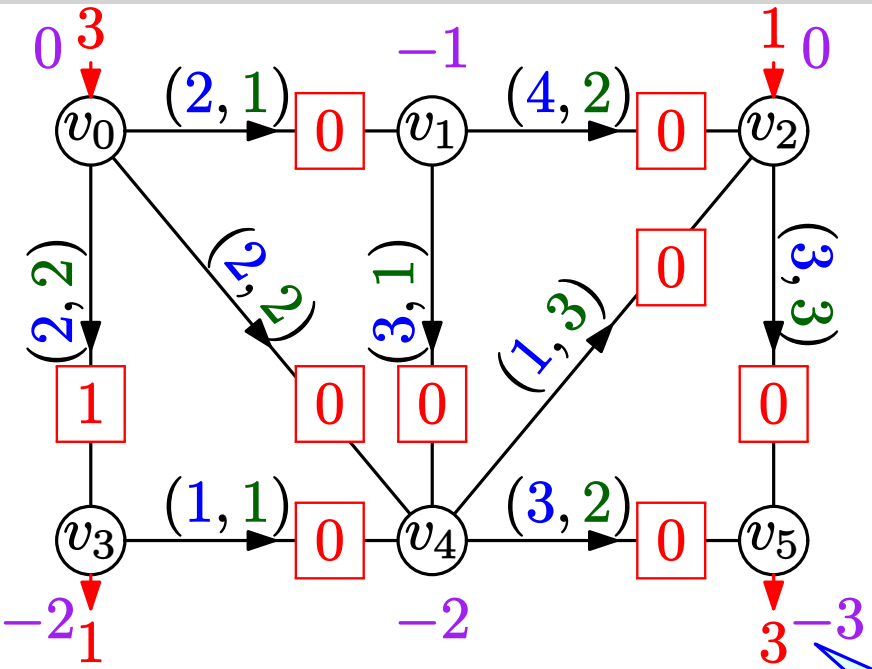


流とポテンシャルの更新

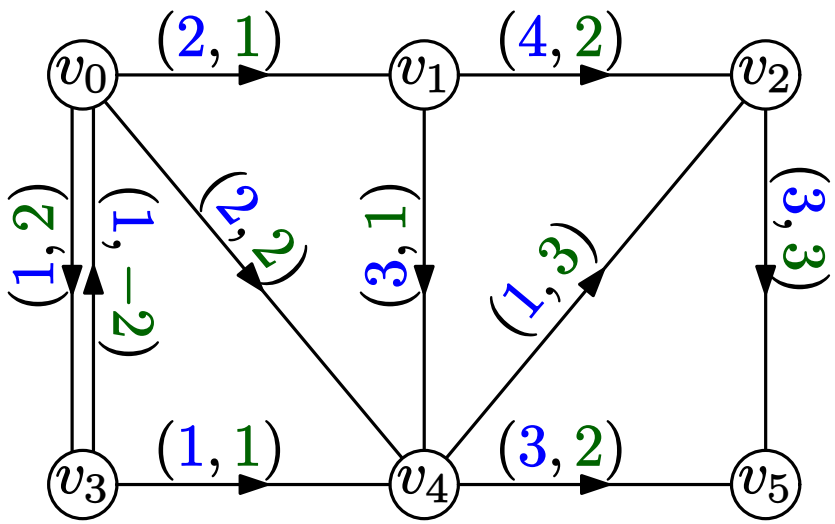


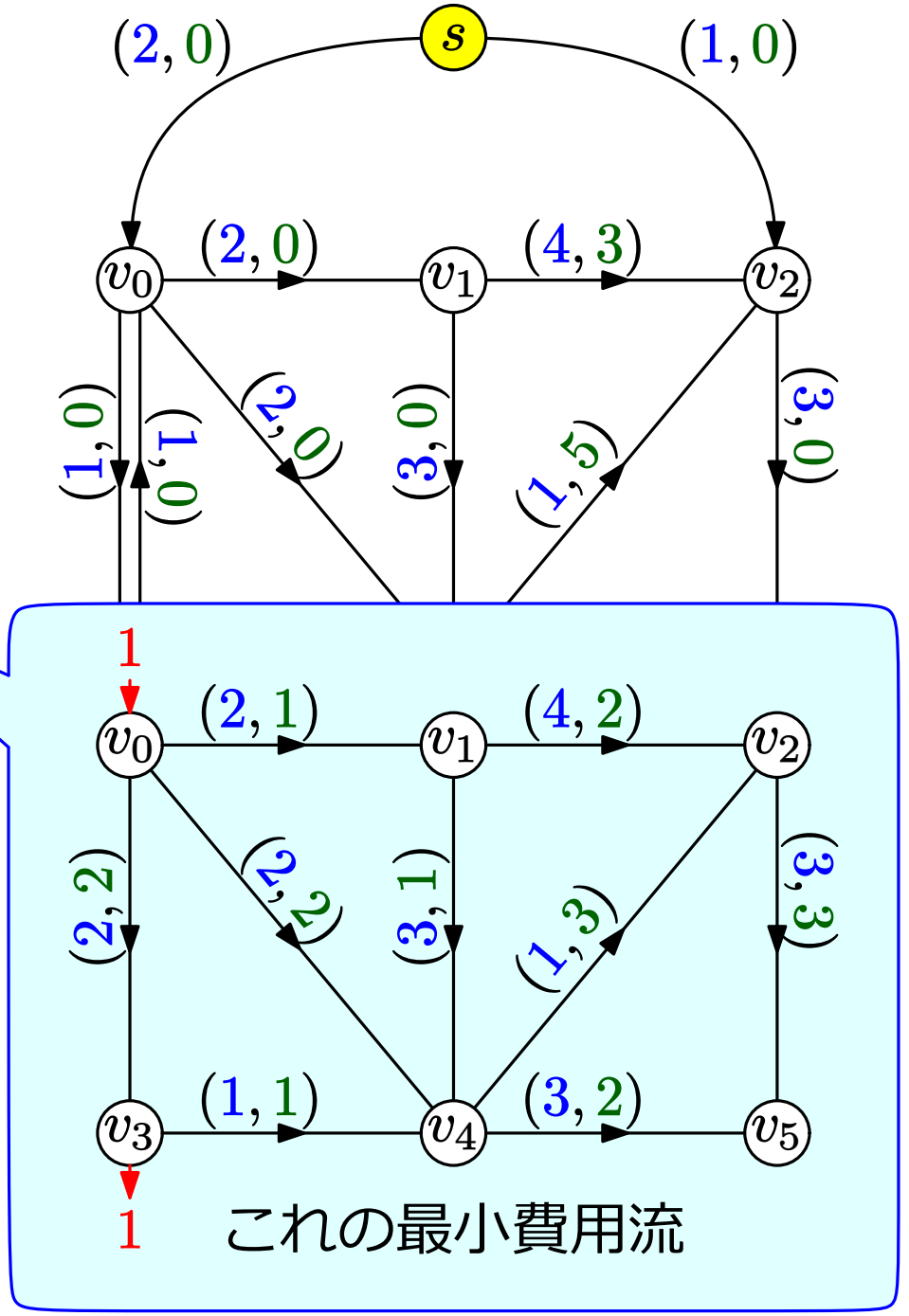


逐次最短路法：例 (2/4)

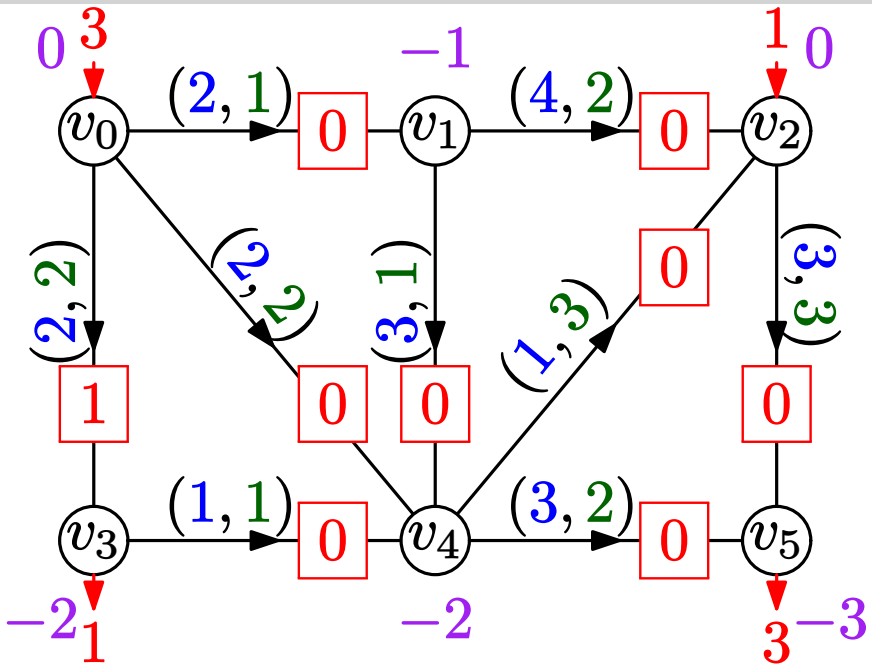


流とポテンシャルの更新

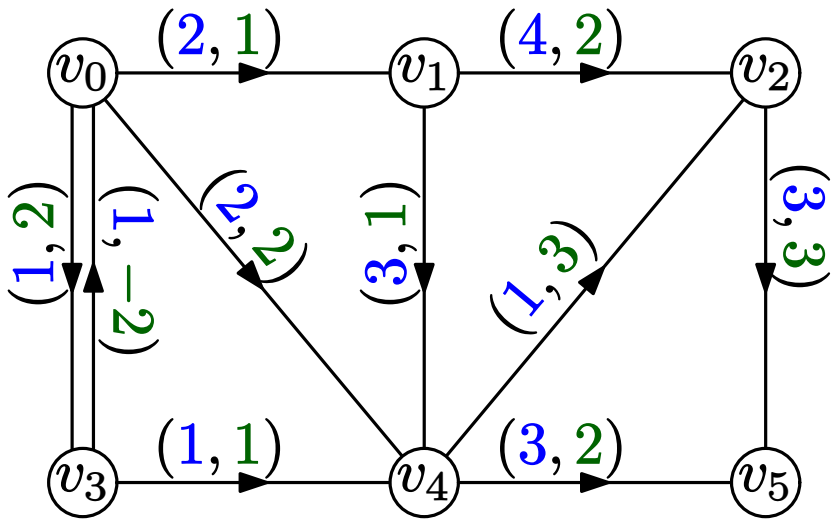


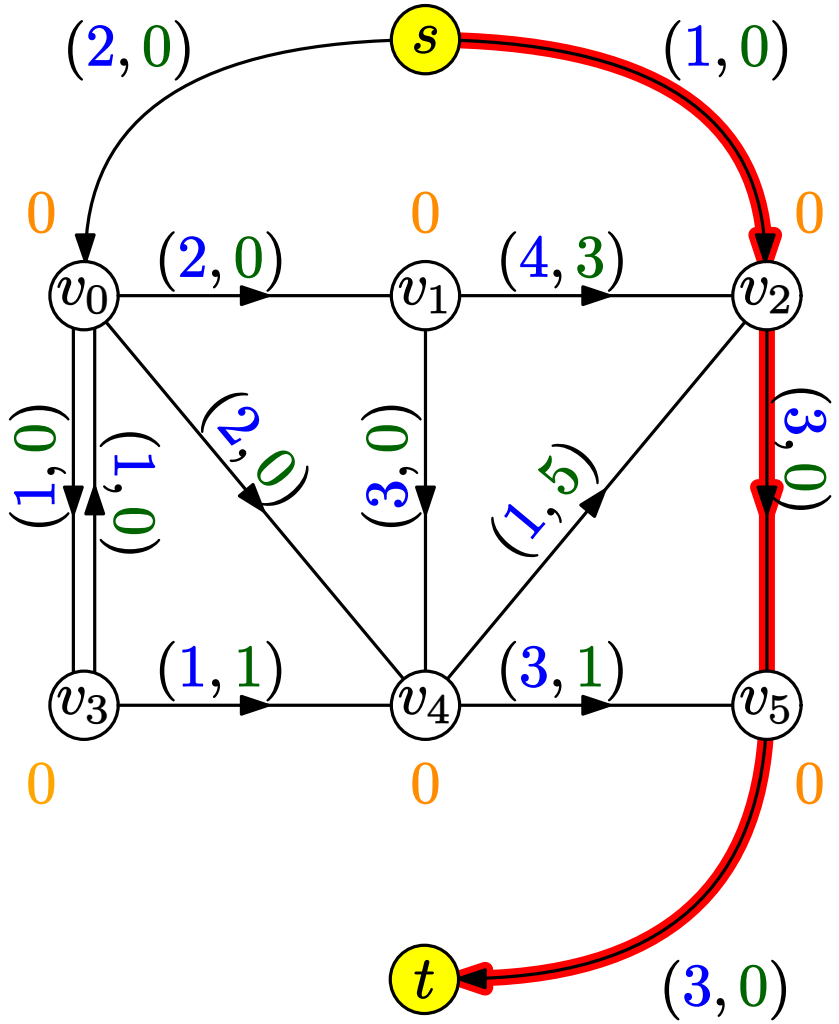


逐次最短路法：例 (2/4)



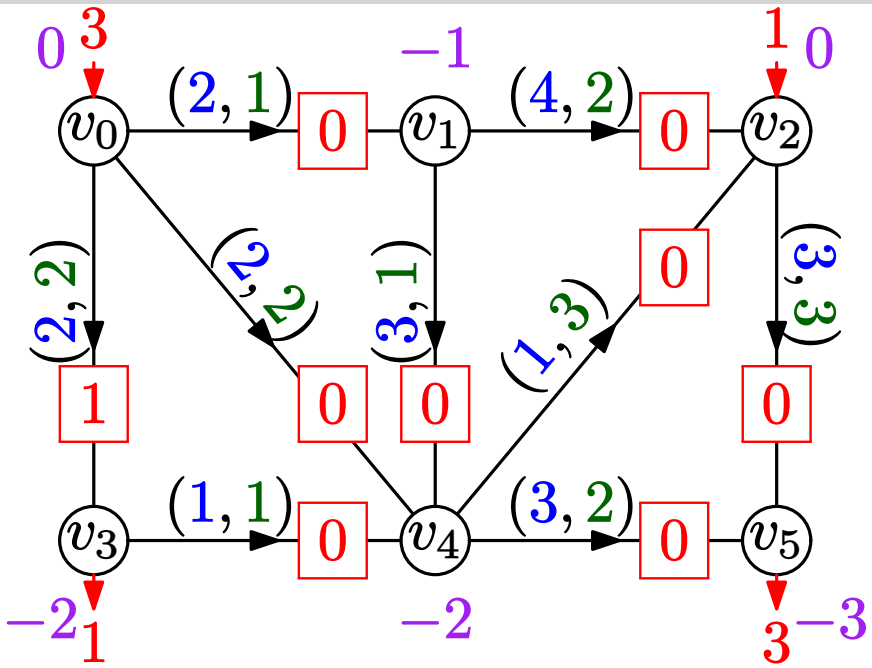
流とポテンシャルの更新



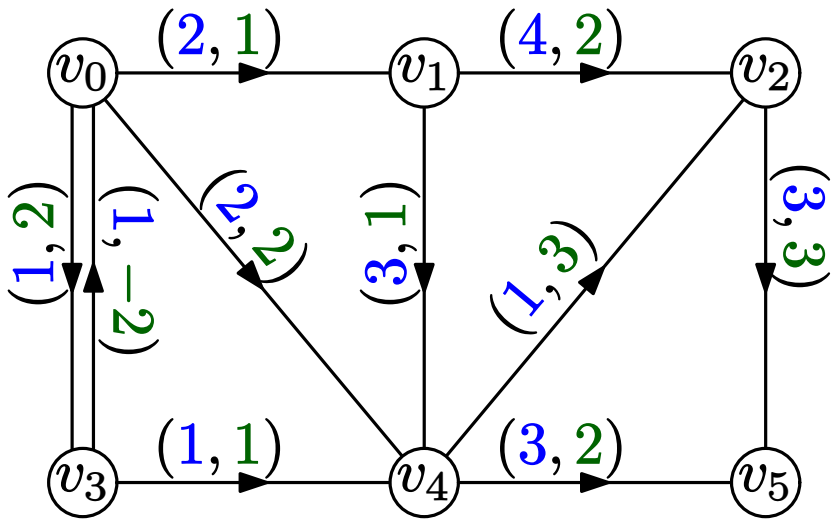


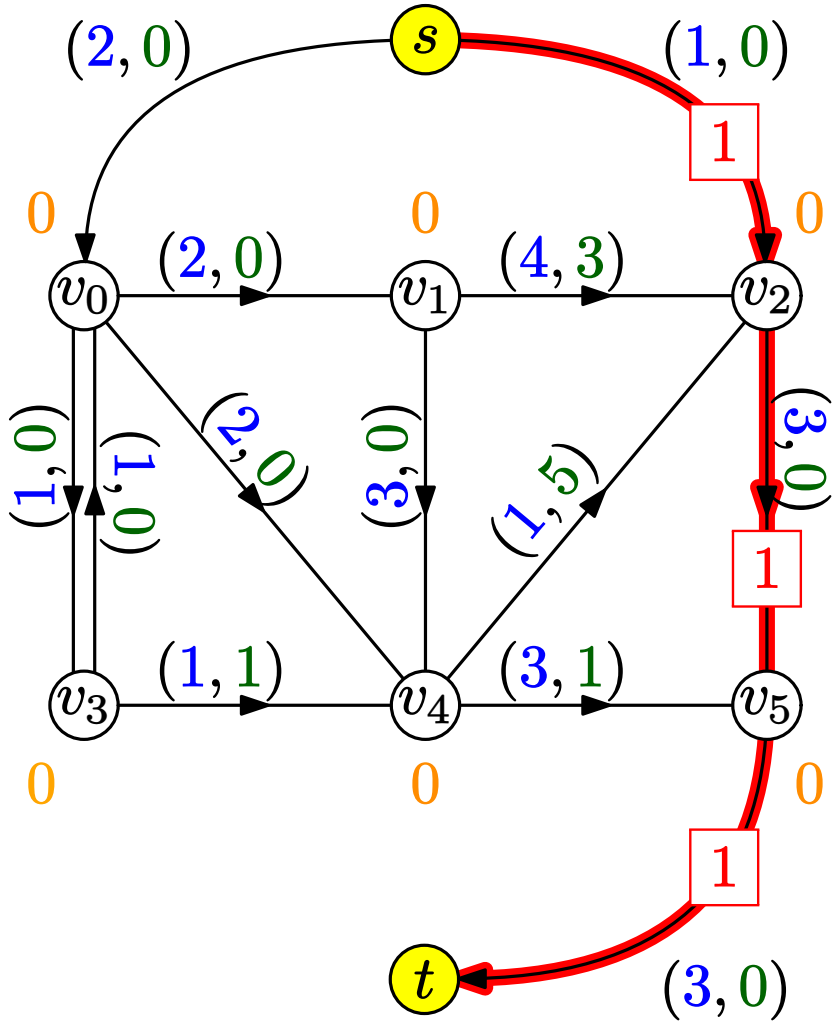
簡約費用を距離とする最短 s-t 路

逐次最短路法：例 (2/4)



流とポテンシャルの更新

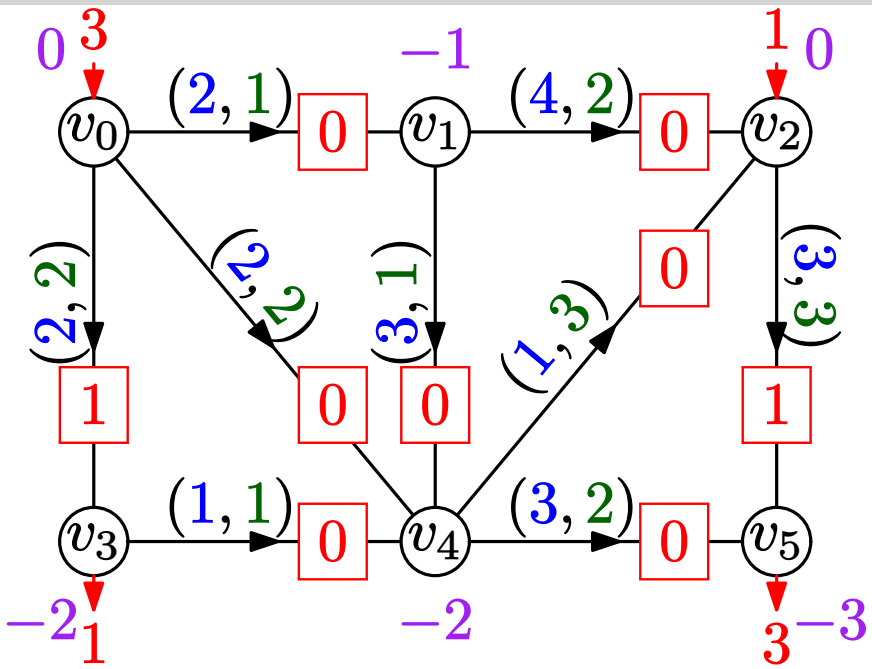




簡約費用を距離とする最短 s-t 路

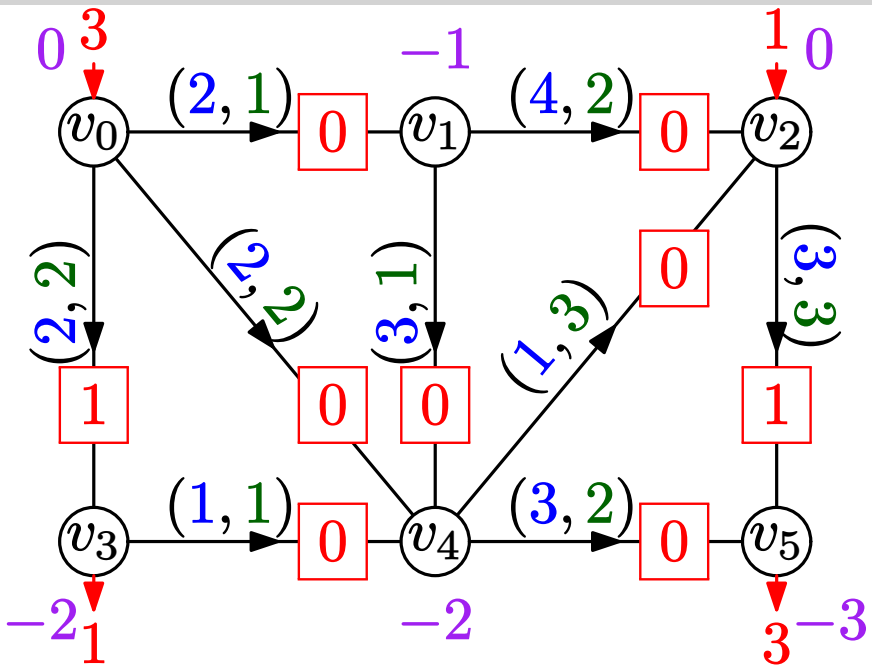
流せるだけ流す

逐次最短路法：例 (3/4)

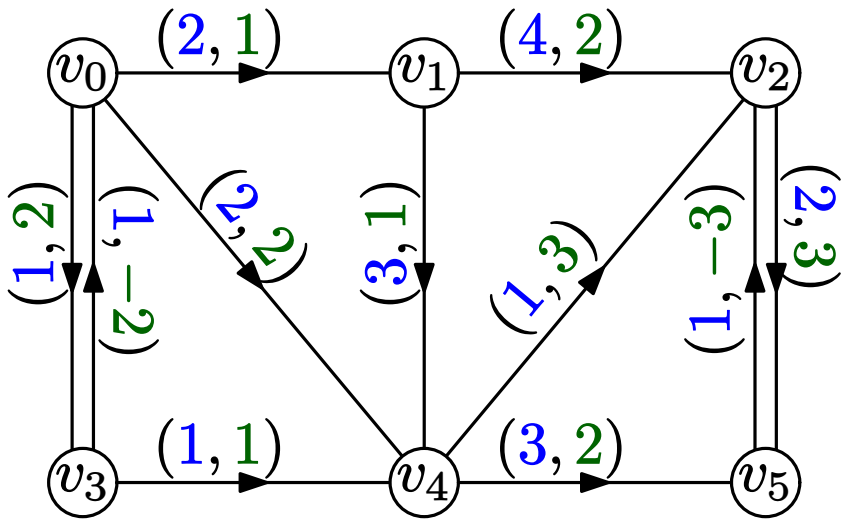


流とポテンシャルの更新

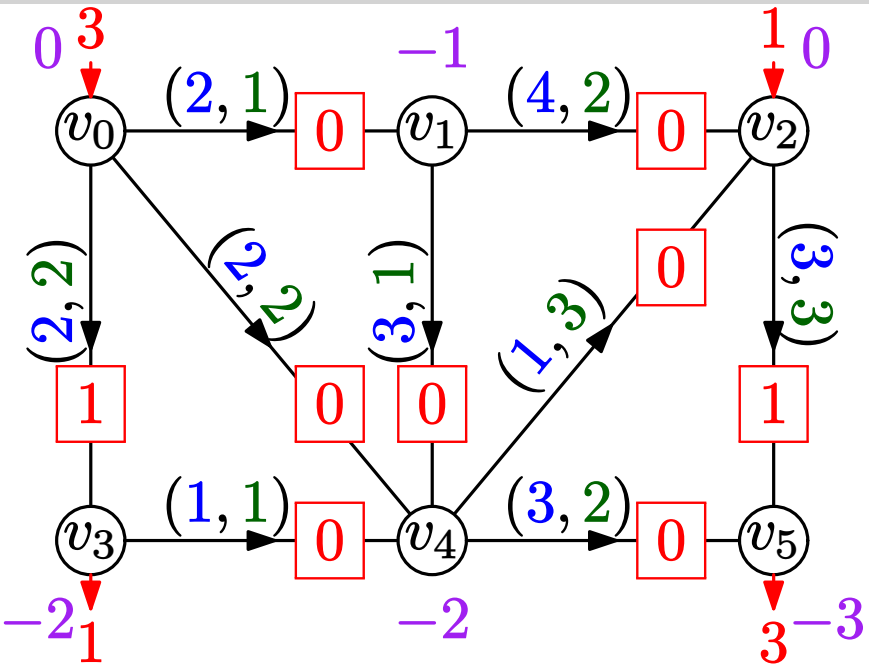
逐次最短路法：例 (3/4)



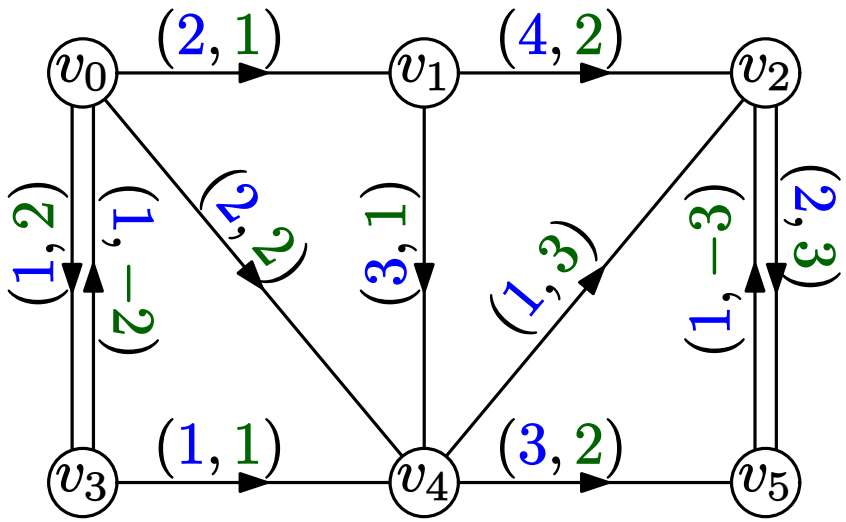
流とポテンシャルの更新

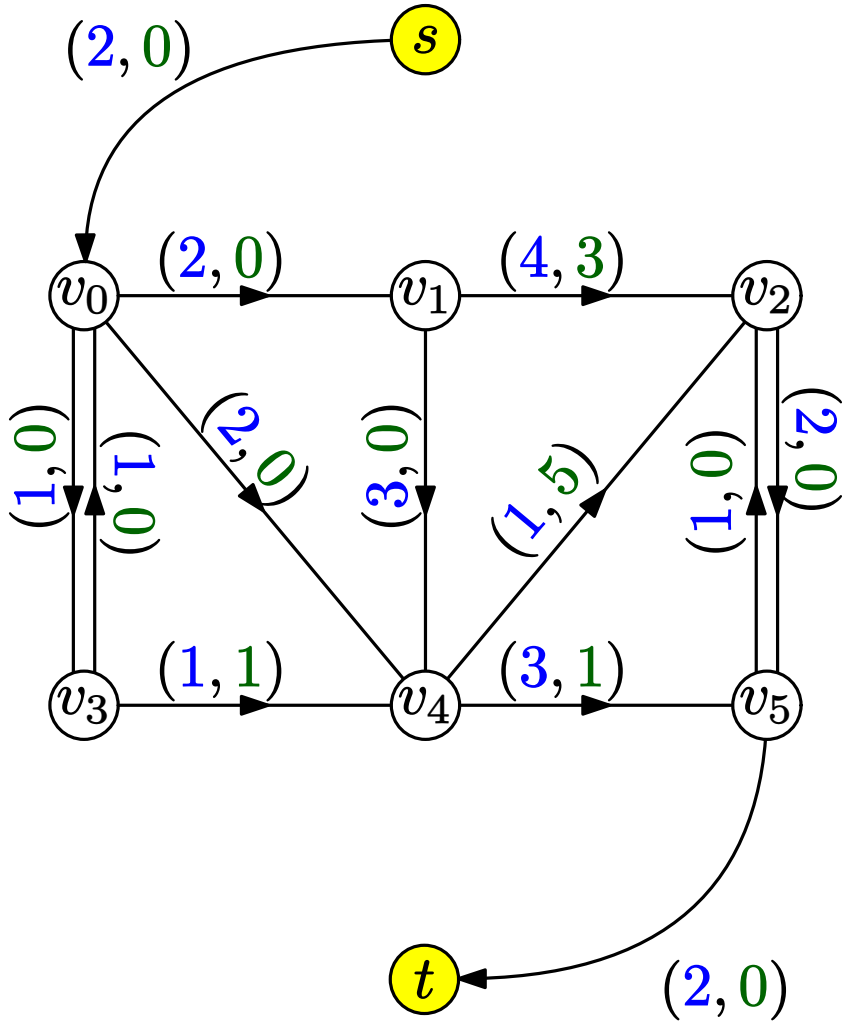


逐次最短路法：例 (3/4)

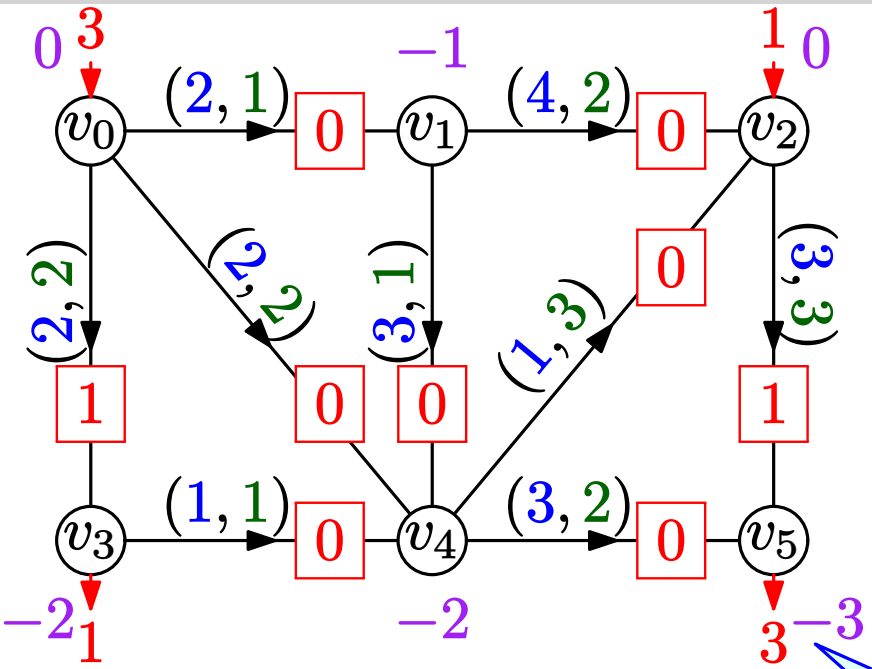


流とポテンシャルの更新

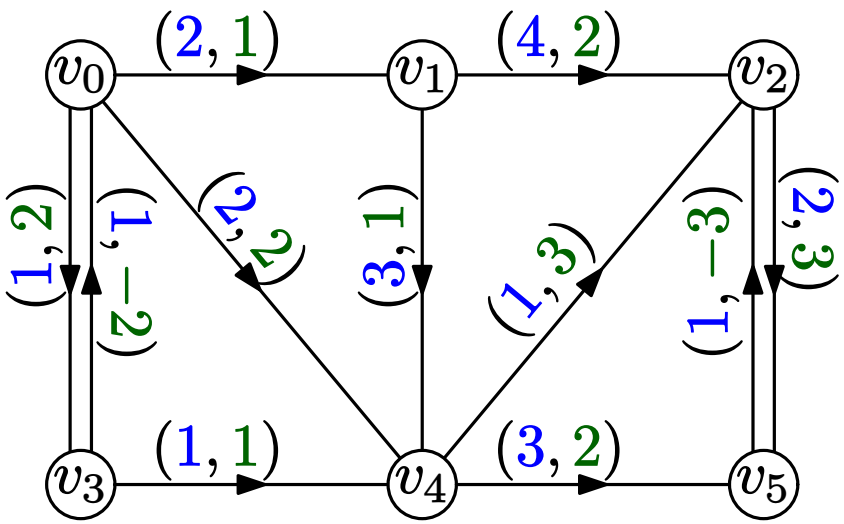


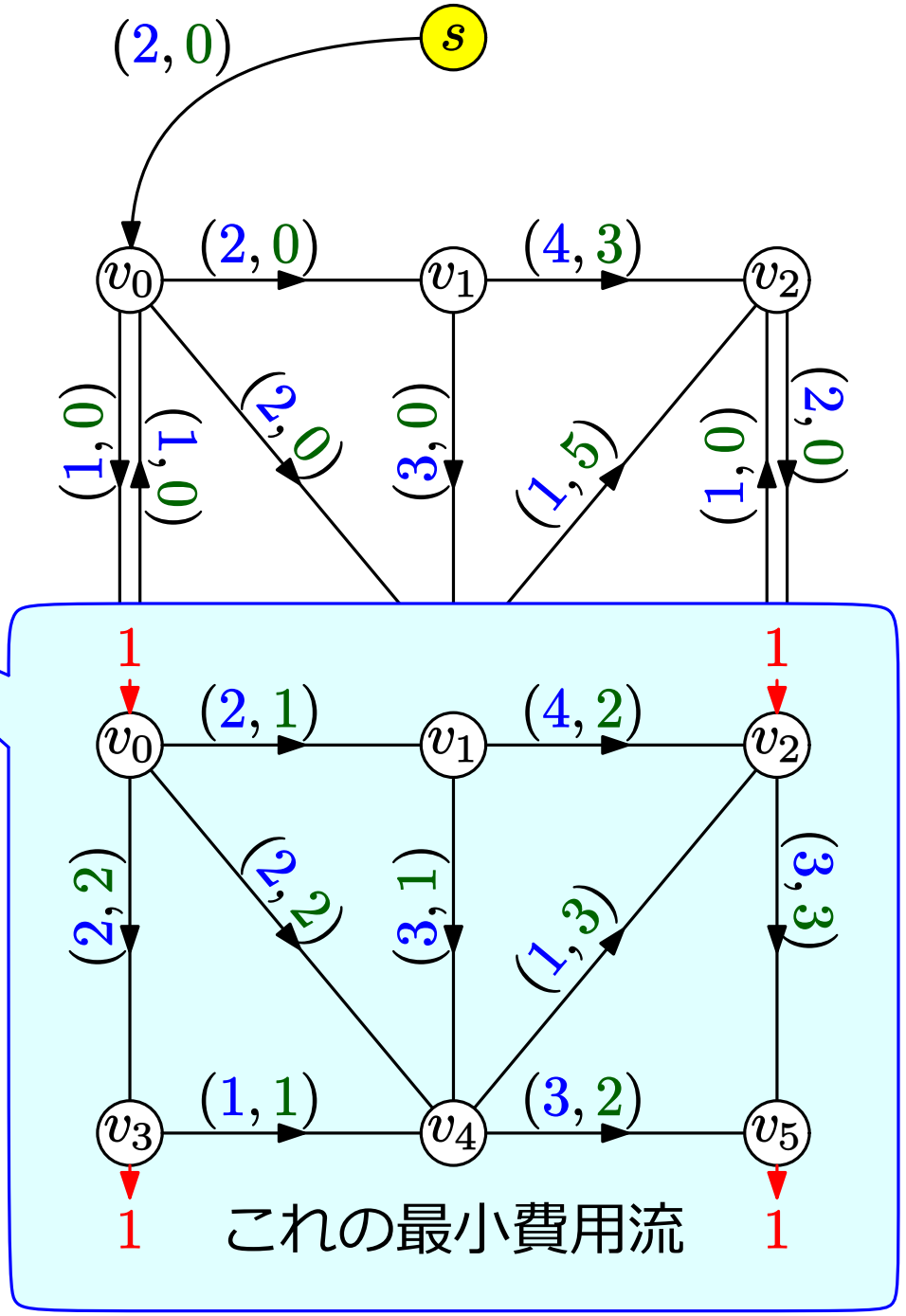


逐次最短路法：例 (3/4)

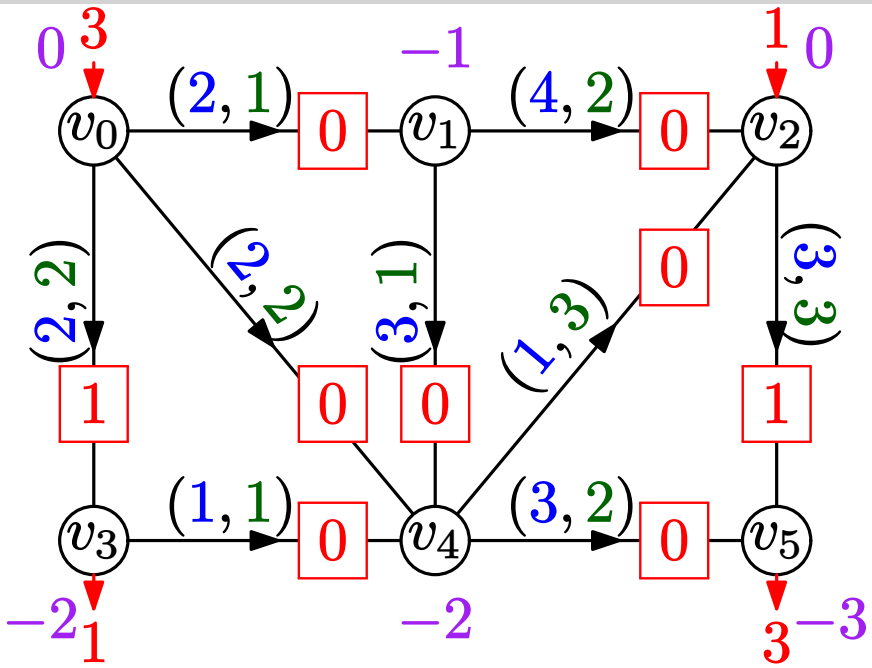


流とポテンシャルの更新

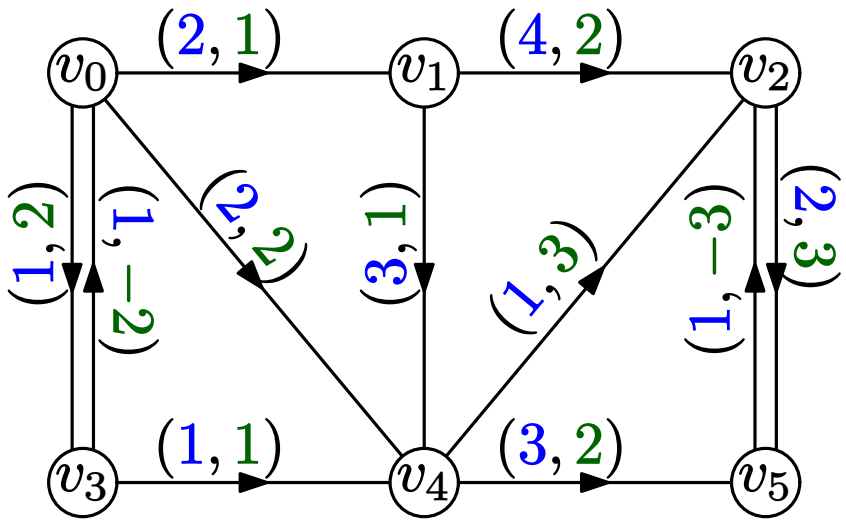


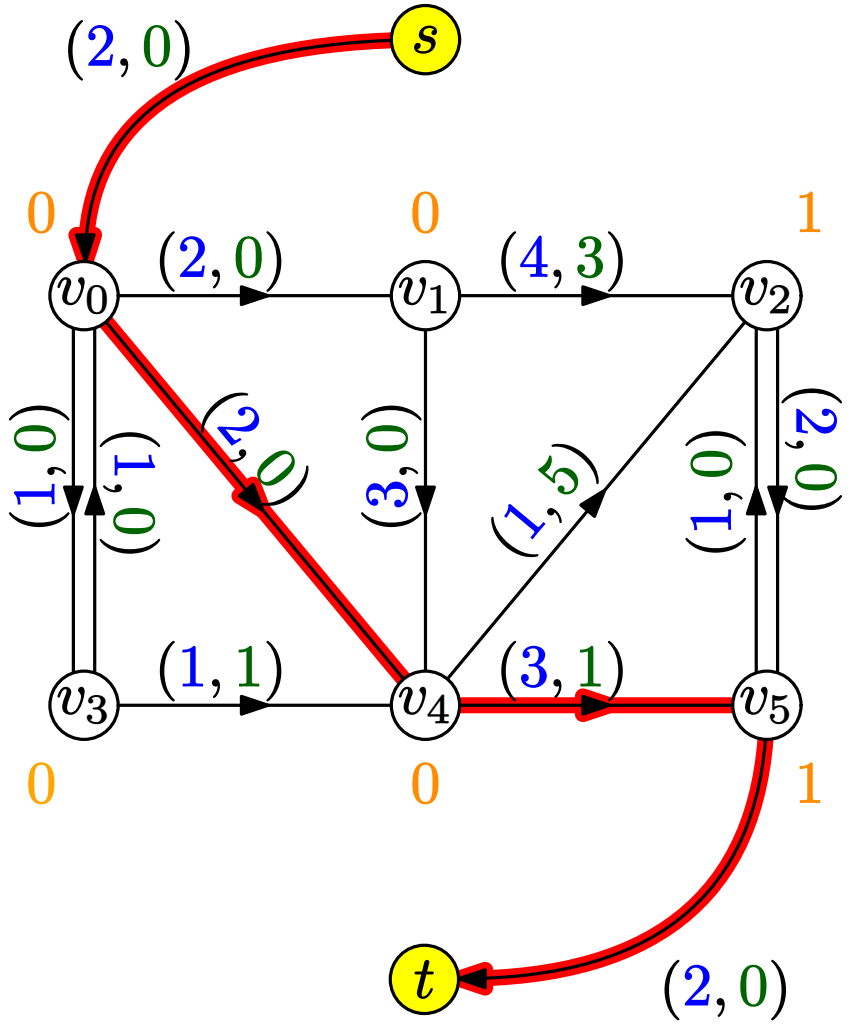


逐次最短路法：例 (3/4)



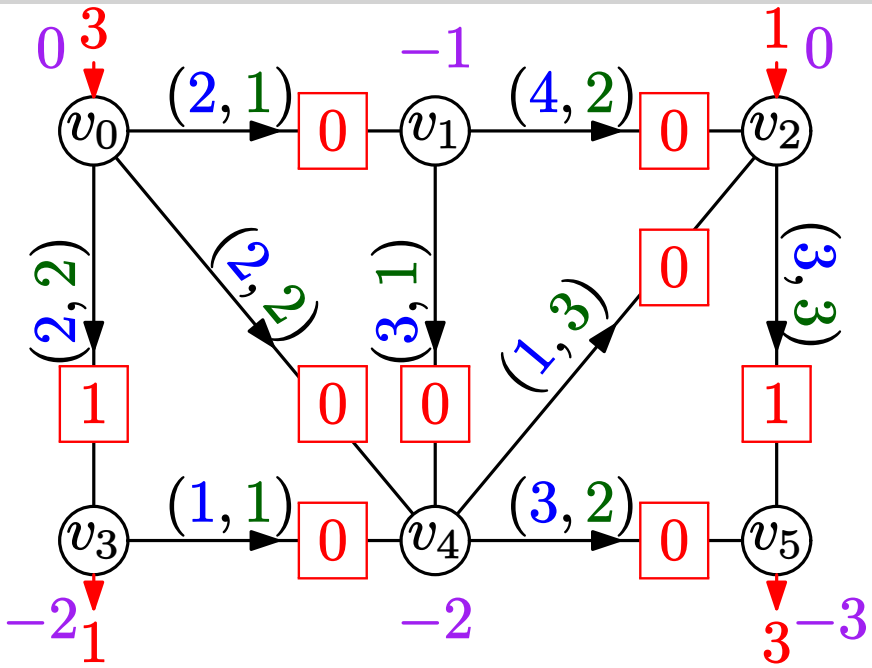
流とポテンシャルの更新



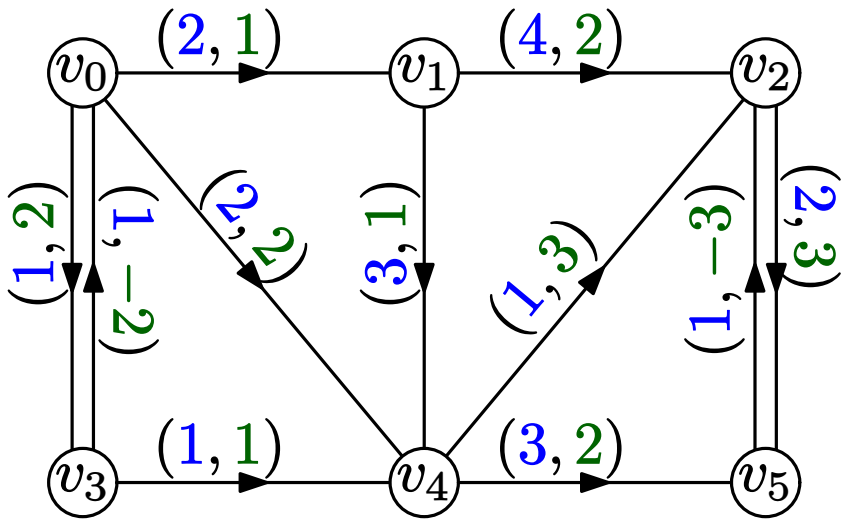


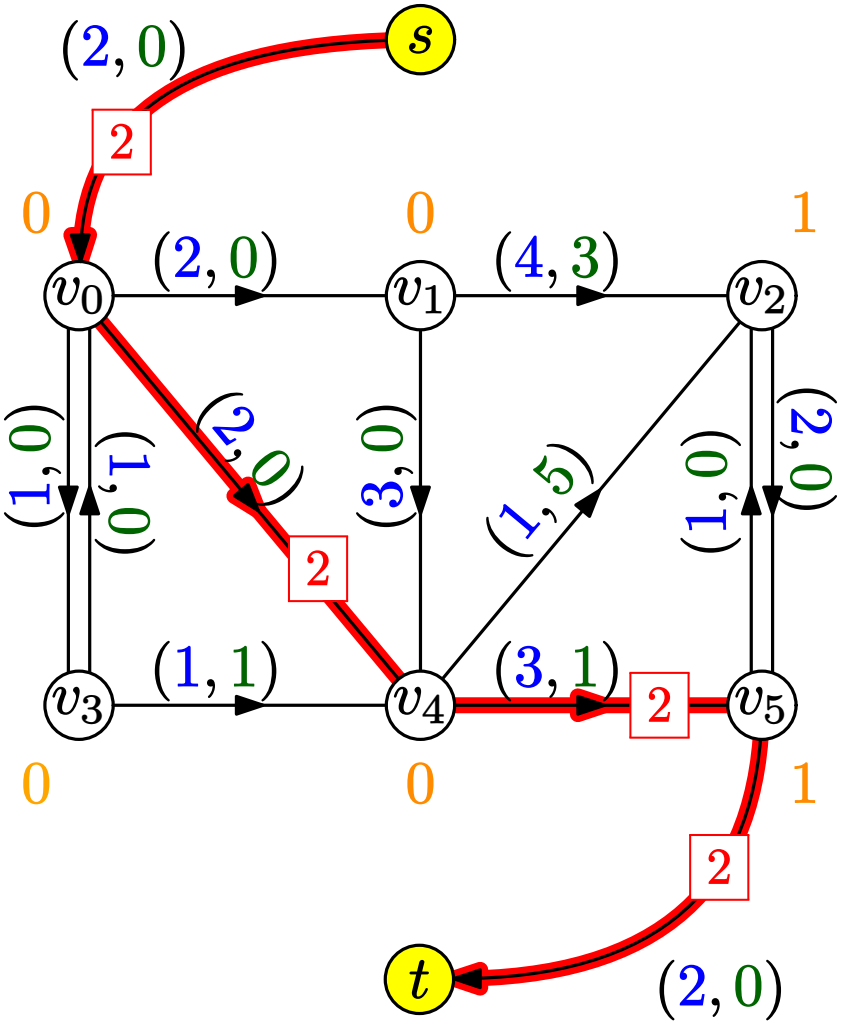
簡約費用を距離とする最短 s-t 路

逐次最短路法：例 (3/4)



流とポテンシャルの更新

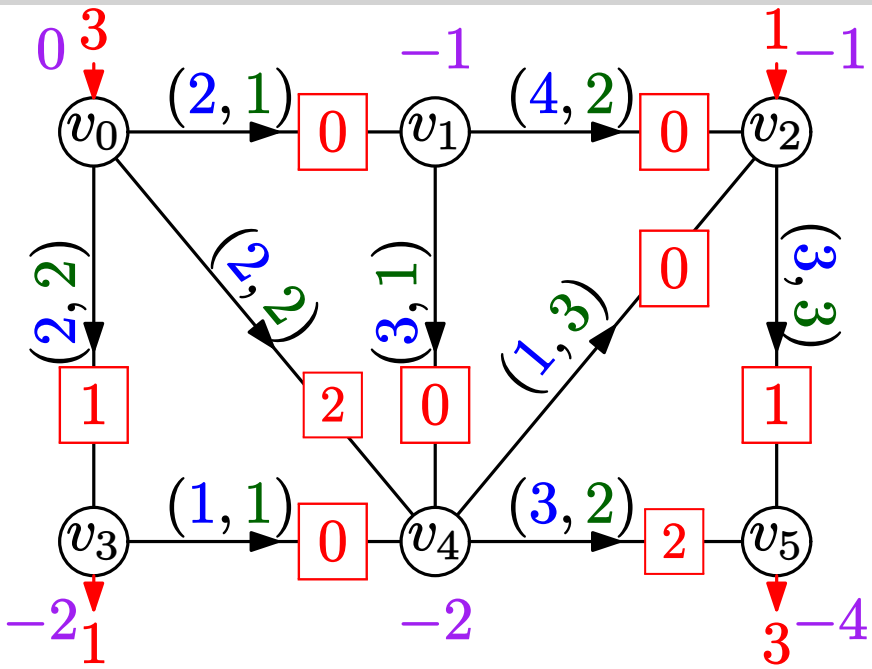




簡約費用を距離とする最短 s-t 路

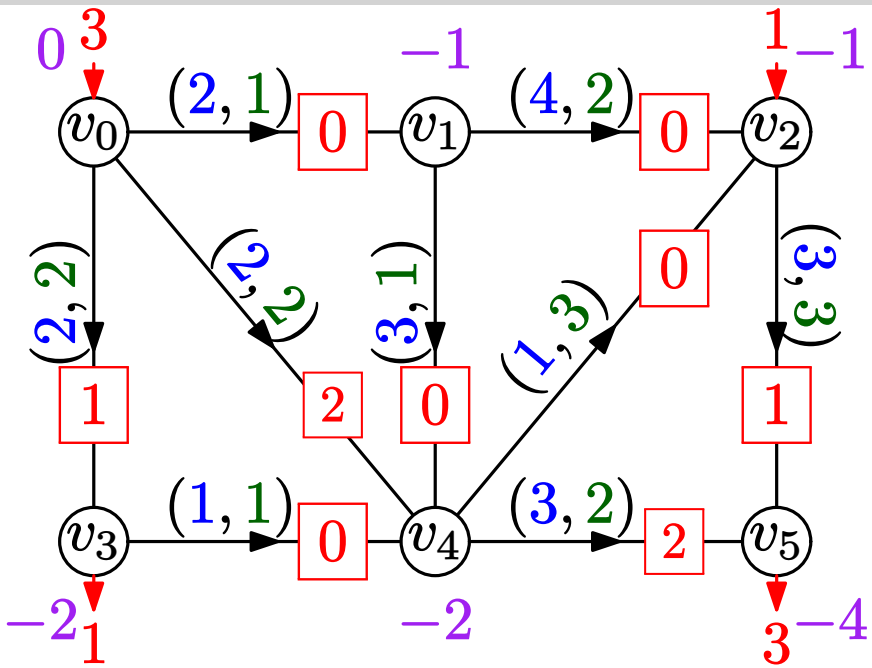
流せるだけ流す

逐次最短路法：例 (4/4)

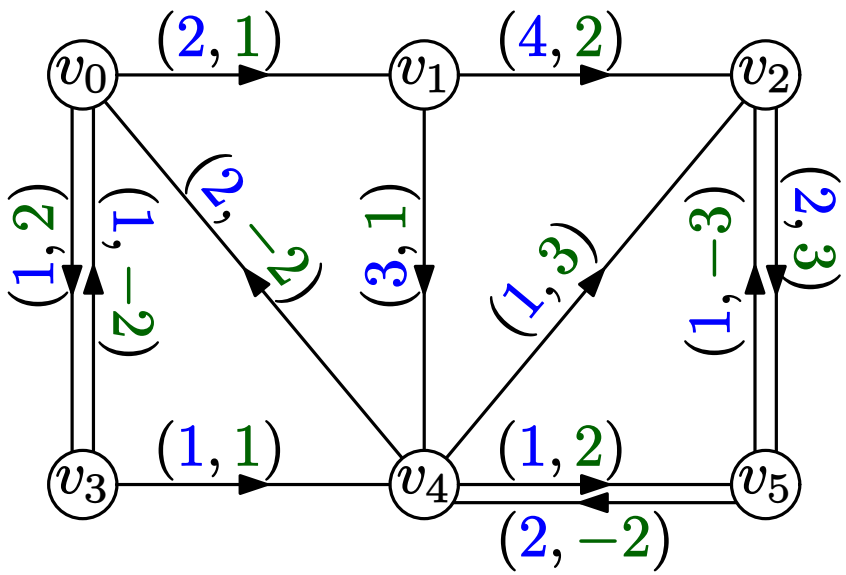


流とポテンシャルの更新

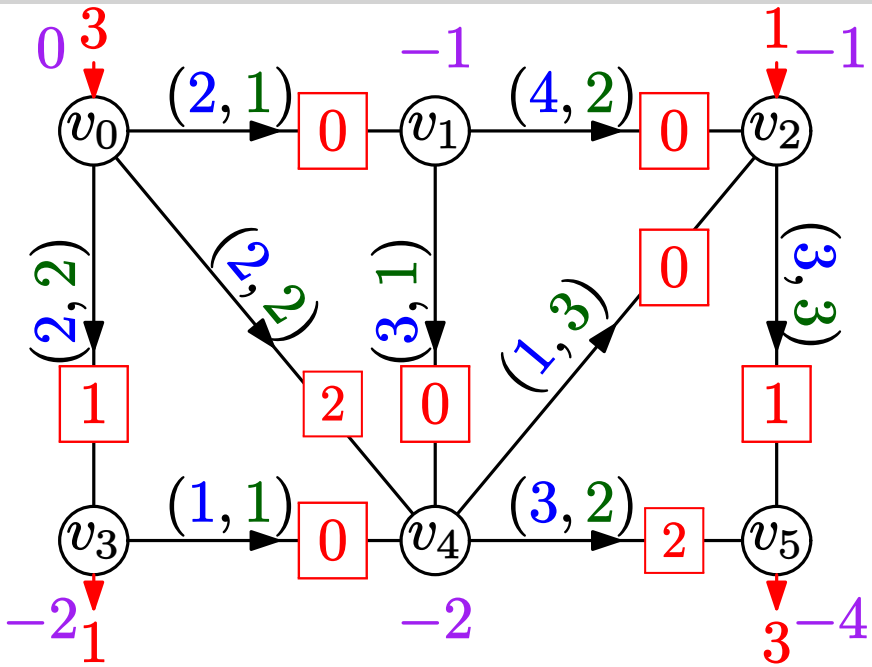
逐次最短路法：例 (4/4)



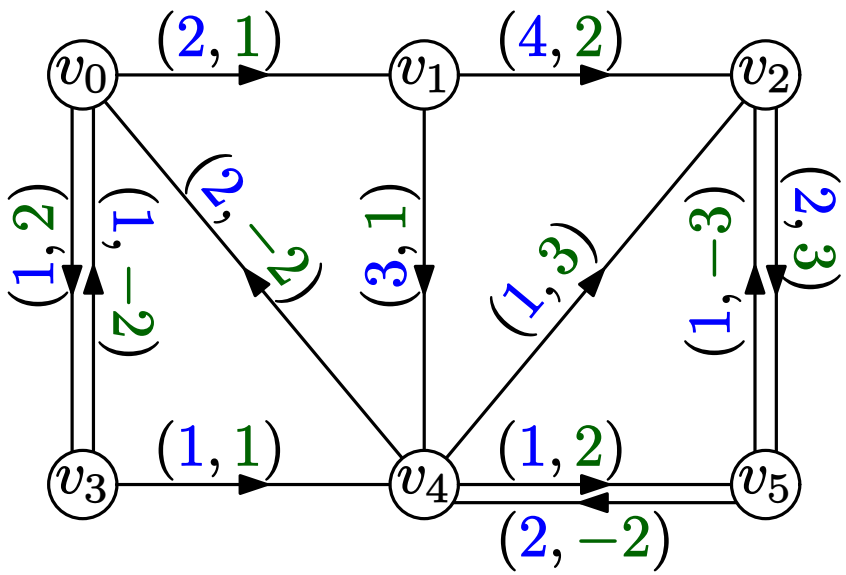
流とポテンシャルの更新

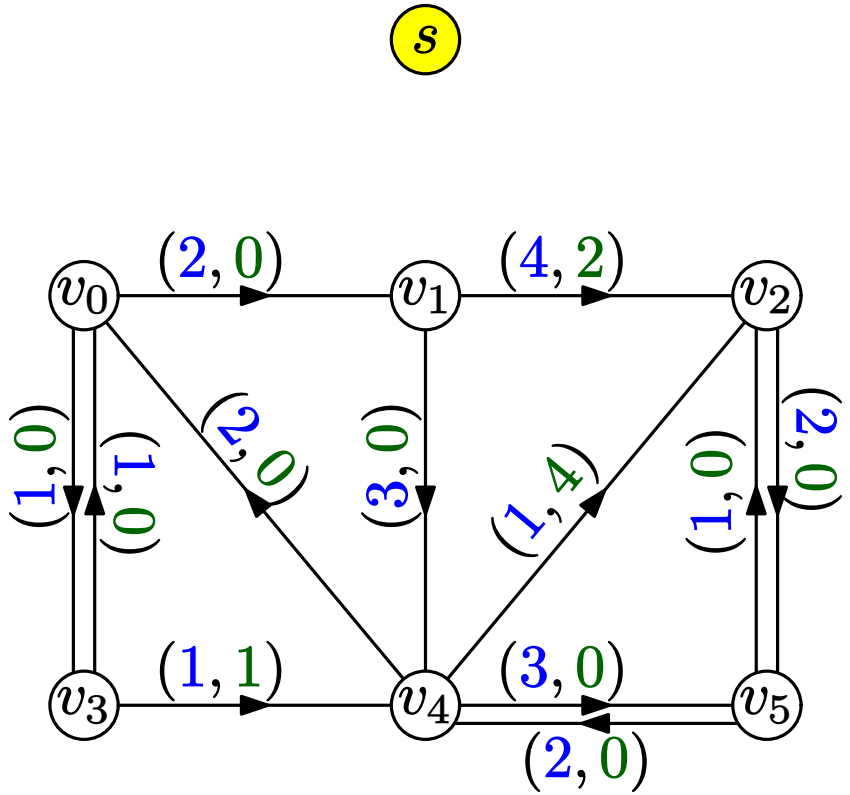


逐次最短路法：例 (4/4)

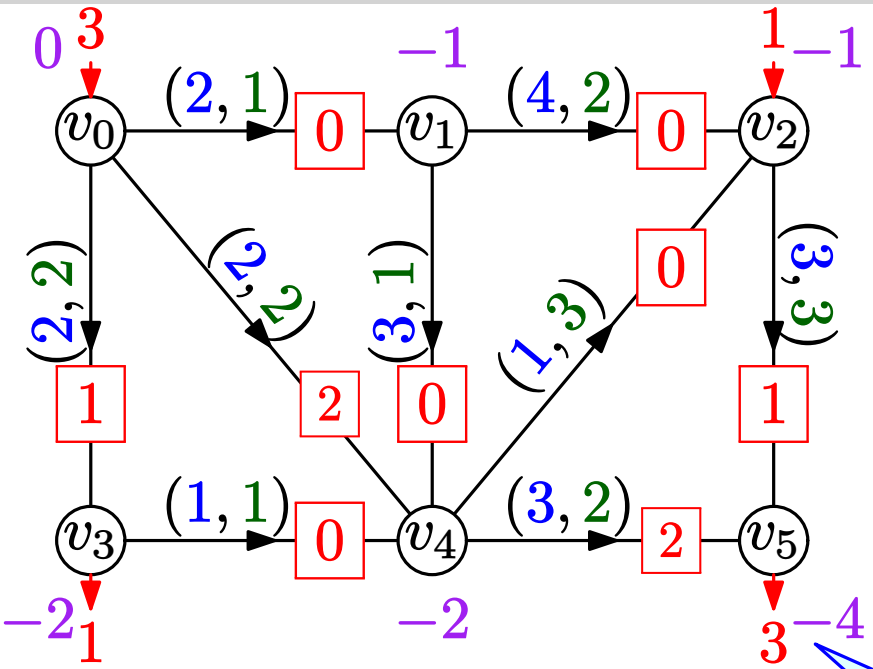


流とポテンシャルの更新

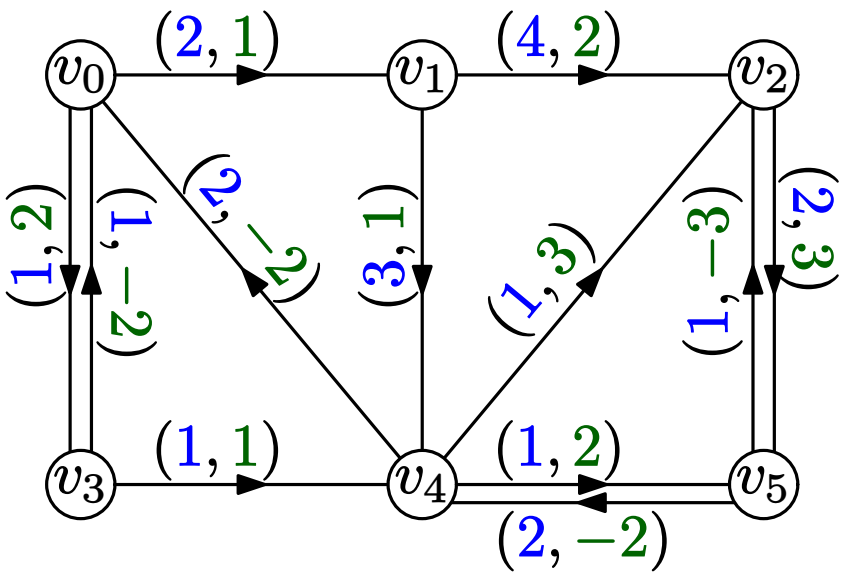


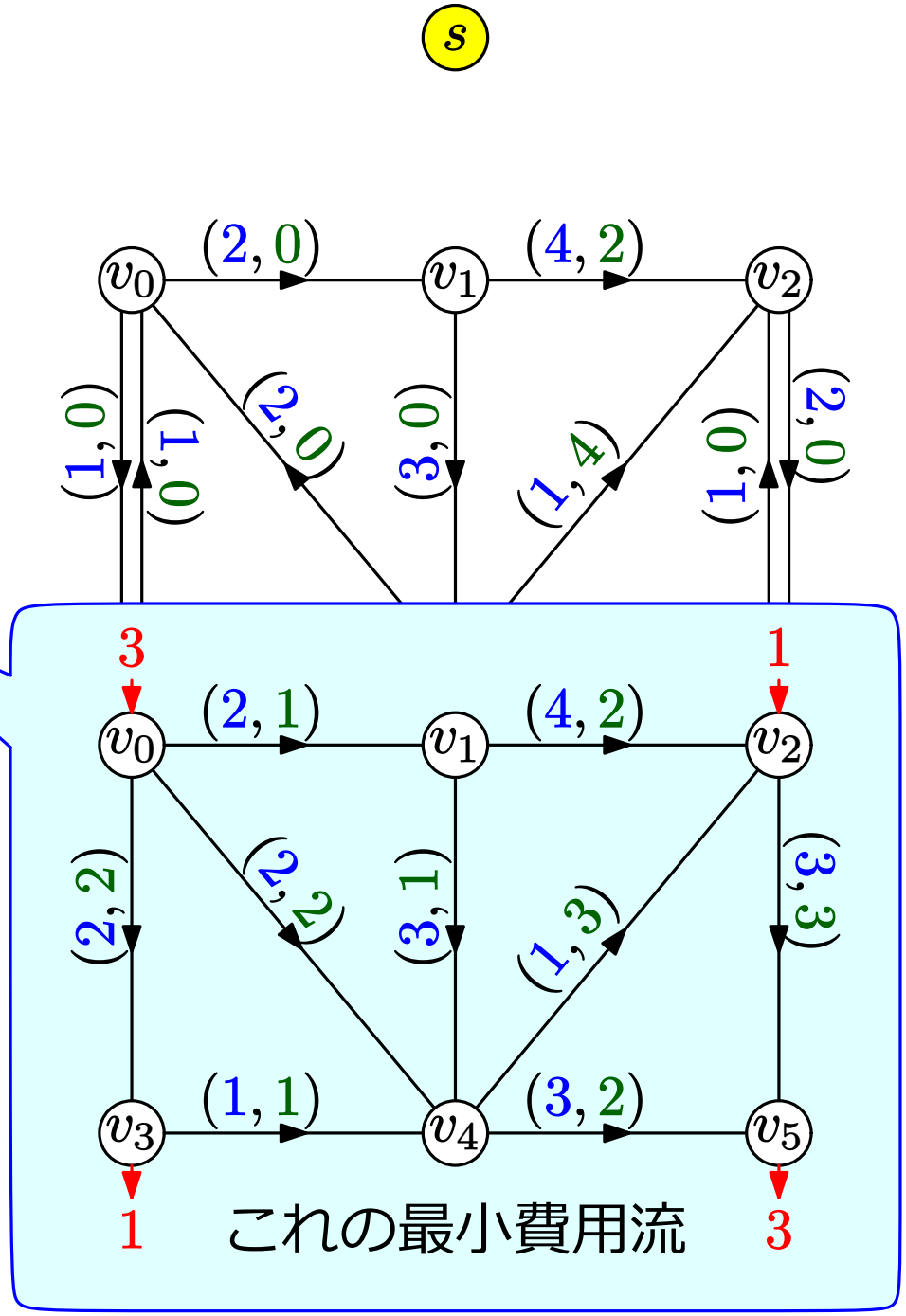


逐次最短路法：例 (4/4)

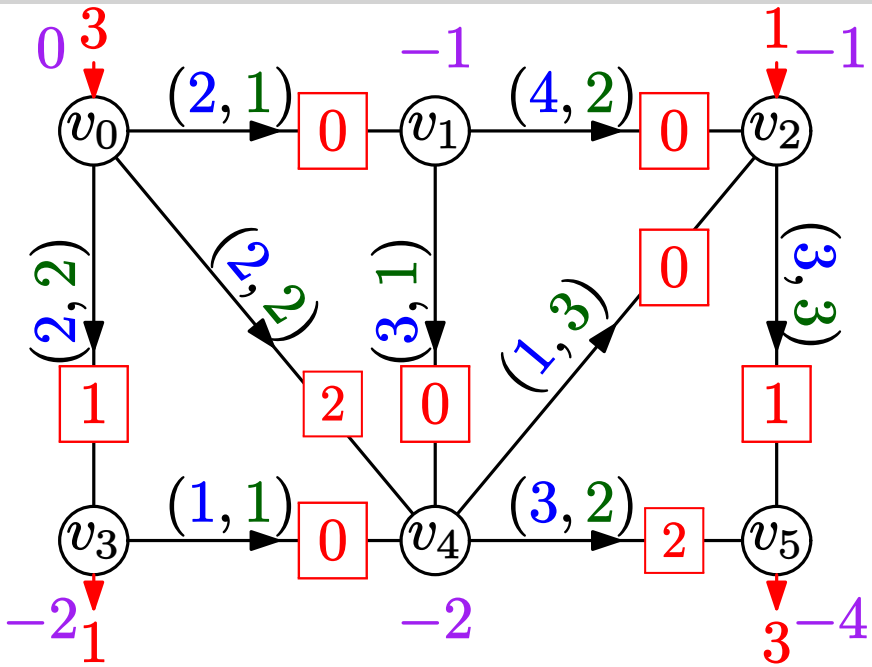


流とポテンシャルの更新

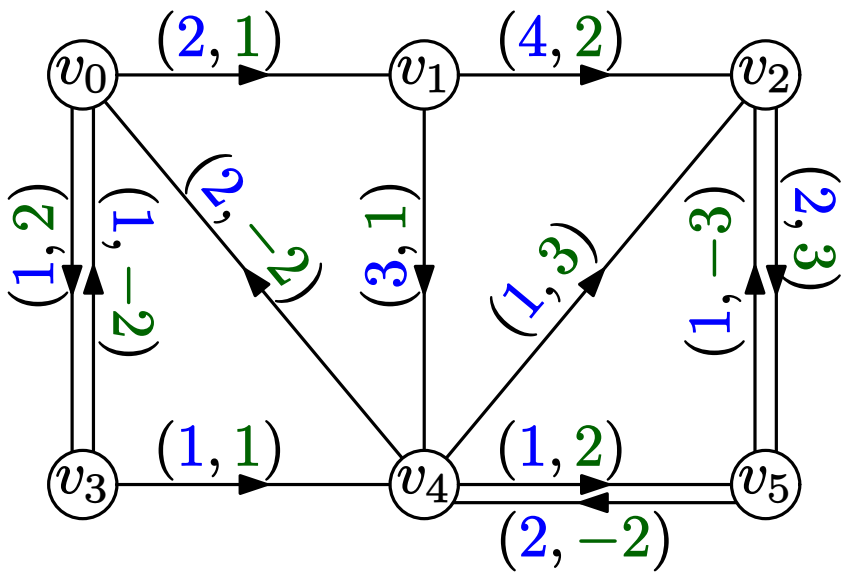


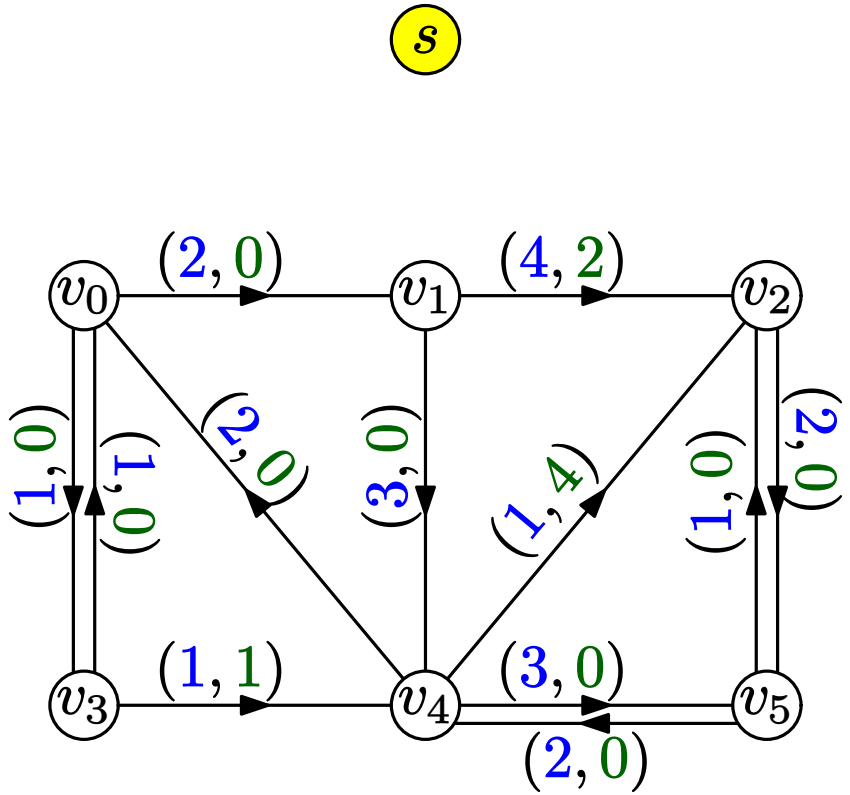


逐次最短路法：例 (4/4)



流とポテンシャルの更新



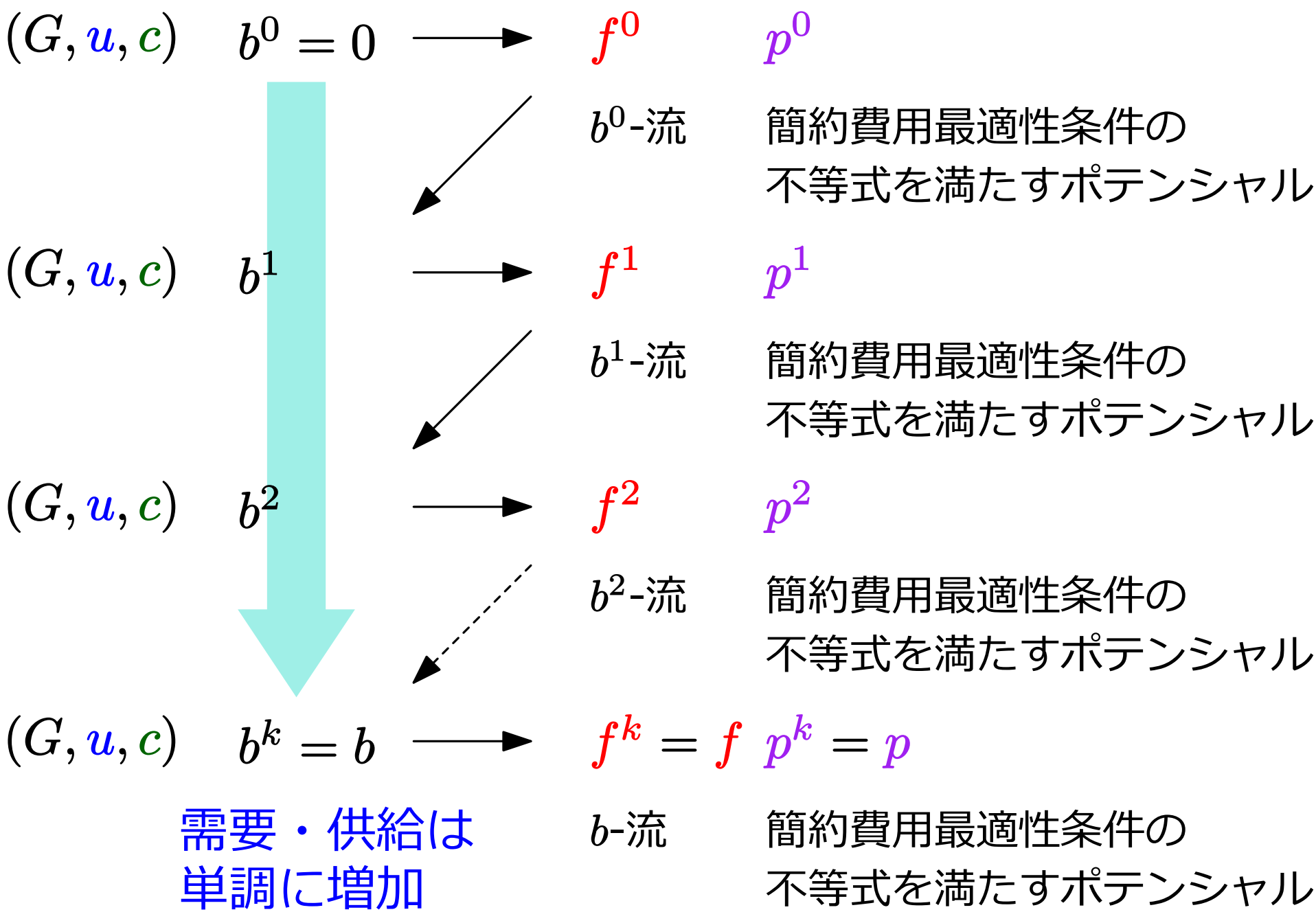


最小費用 b-流が得られた

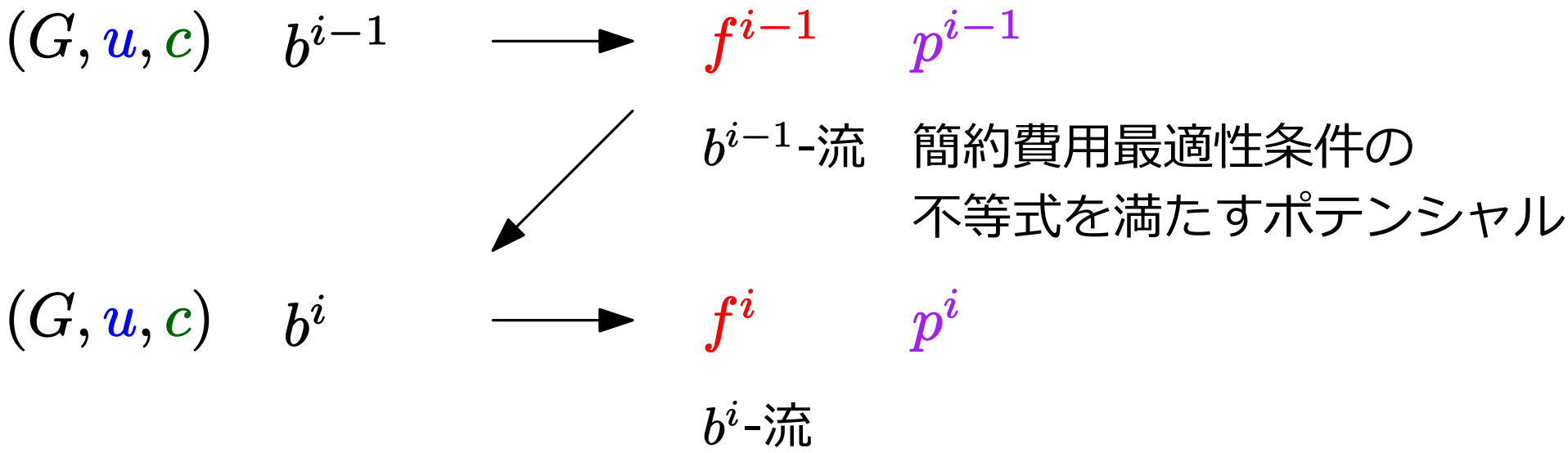
アルゴリズム：逐次最短路法

- 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

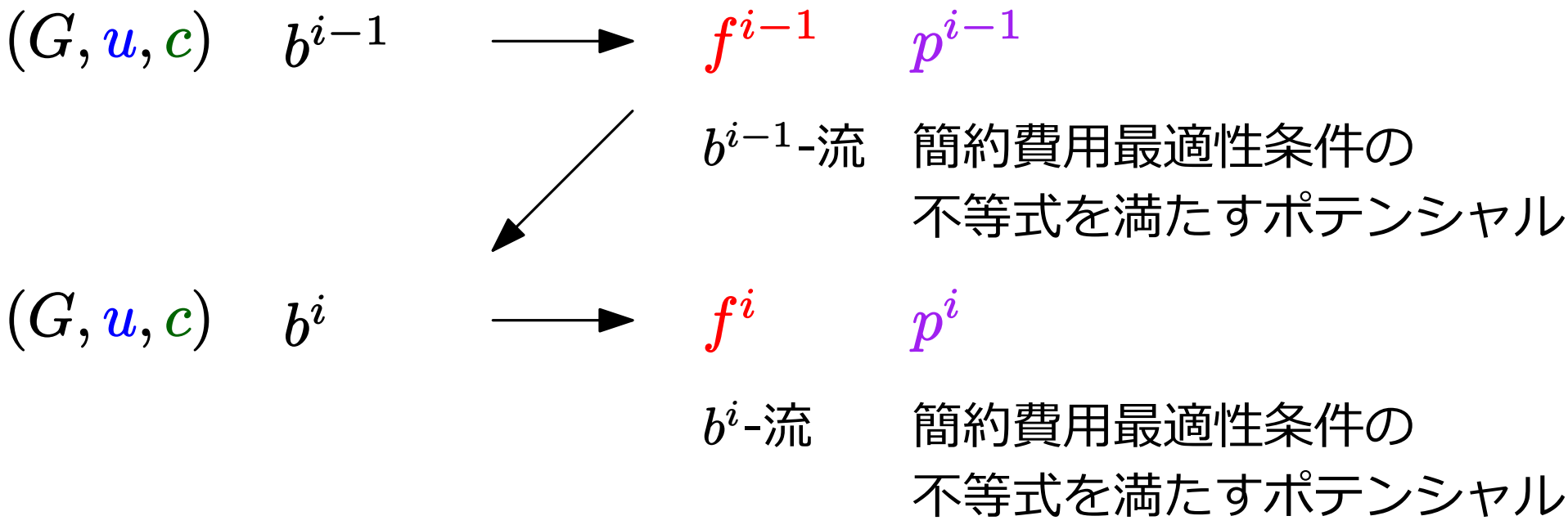
逐次最短路法：基本的な考え方 (再掲)



逐次最短路法：証明すべき事項



逐次最短路法：証明すべき事項



証明すべき事項

p^i が簡約費用最適性条件の不等式を満たす

不変条件

(G, u, c)　b^{i-1} ⟶ f^{i-1}　p^{i-1}

b^{i-1}-流

仮定：$(c_{f^{i-1}})_{uv} - p^{i-1}_u + p^{i-1}_v \geq 0 \quad \forall\, uv \in A_{f^{i-1}}$

(G, u, c)　b^{i} ⟶ f^{i}　$p^{i} = p^{i-1} - d$

b^{i}-流

主張：$(c_{f^{i}})_{uv} - p^{i}_u + p^{i}_v \geq 0 \quad \forall\, uv \in A_{f^{i}}$

不変条件

(G, u, c)　b^{i-1} ⟶ f^{i-1}　p^{i-1}

b^{i-1}-流

仮定：$(c_{f^{i-1}})_{uv} - p^{i-1}_u + p^{i-1}_v \geq 0 \quad \forall\, uv \in A_{f^{i-1}}$

(G, u, c)　b^{i} ⟶ f^{i}　$p^{i} = p^{i-1} - d$

b^{i}-流

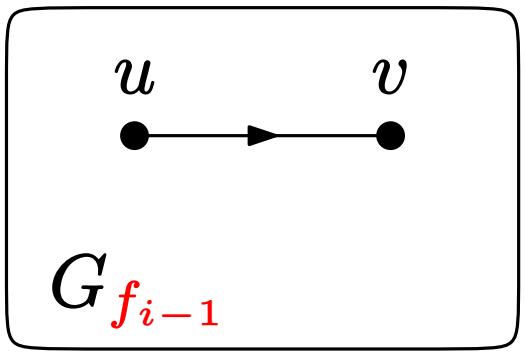
主張：$(c_{f^{i}})_{uv} - p^{i}_u + p^{i}_v \geq 0 \quad \forall\, uv \in A_{f^{i}}$

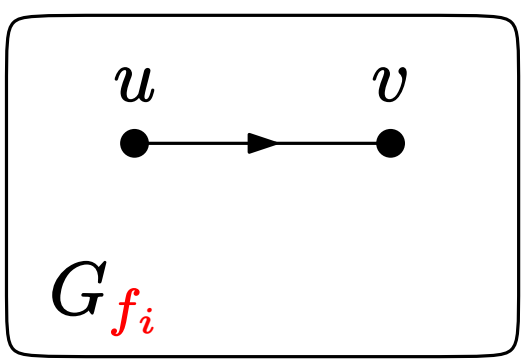
証明：場合分け (1) $uv \in A_{f^i} \cap A_{f^{i-1}}$ (2) $uv \in A_{f^i} - A_{f^{i-1}}$

不変条件：証明 (1/2)

(1) $uv \in A_{f^i} \cap A_{f^{i-1}}$ のとき

$$
\begin{aligned}
&(c_{f_i})_{uv} - p_u^i + p_v^i \\
&= (c_{f_{i-1}})_{uv} - (p_u^{i-1} - d_u) + (p_v^{i-1} - d_v) \\
&= (c_{f_{i-1}})_{uv} - p_u^{i-1} + p_v^{i-1} + d_u - d_v
\end{aligned}
$$



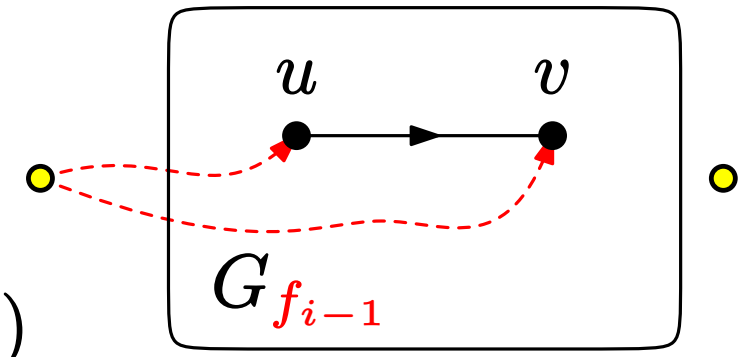


仮定：$(c_{f^{i-1}})_{uv} - p_u^{i-1} + p_v^{i-1} \geq 0 \quad \forall\, uv \in A_{f^{i-1}}$

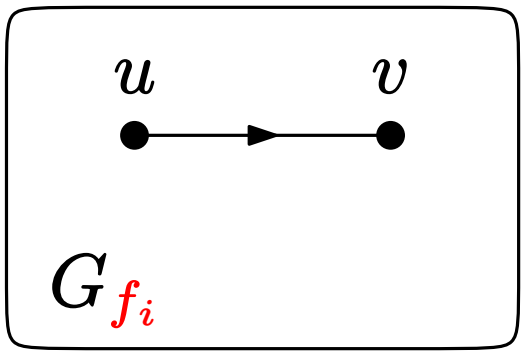
主張：$(c_{f^i})_{uv} - p_u^i + p_v^i \geq 0 \quad \forall\, uv \in A_{f^i}$ $\qquad p^i = p^{i-1} - d$

不変条件：証明 (1/2)

(1) $uv \in A_{f^i} \cap A_{f^{i-1}}$ のとき



$$
\begin{aligned}
&(c_{f_i})_{uv} - p_u^i + p_v^i \\
&= (c_{f_{i-1}})_{uv} - (p_u^{i-1} - d_u) + (p_v^{i-1} - d_v) \\
&= (c_{f_{i-1}})_{uv} - p_u^{i-1} + p_v^{i-1} + \underline{d_u - d_v} \\
&\qquad \geq -((c_{f_{i-1}})_{uv} - p_u^{i-1} + p_v^{i-1})
\end{aligned}
$$

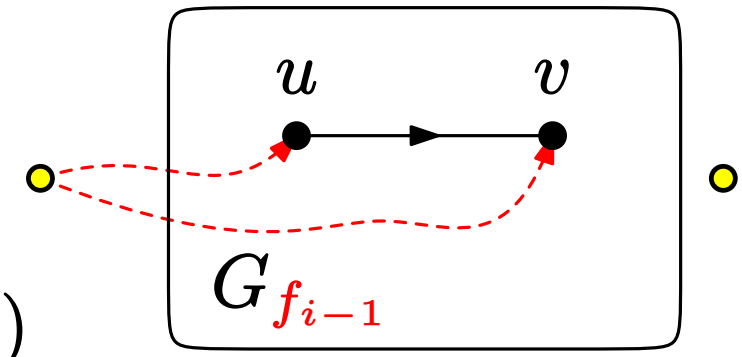


仮定：$(c_{f^{i-1}})_{uv} - p_u^{i-1} + p_v^{i-1} \geq 0 \quad \forall\, uv \in A_{f^{i-1}}$

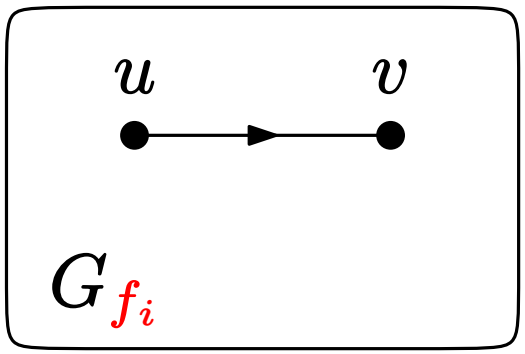
主張：$(c_{f^i})_{uv} - p_u^i + p_v^i \geq 0 \quad \forall\, uv \in A_{f^i}$ $\qquad p^i = p^{i-1} - d$

不変条件：証明 (1/2)

(1) $uv \in A_{f^i} \cap A_{f^{i-1}}$ のとき



$$
\begin{aligned}
&(c_{f_i})_{uv} - p_u^i + p_v^i \\
&= (c_{f_{i-1}})_{uv} - (p_u^{i-1} - d_u) + (p_v^{i-1} - d_v) \\
&= (c_{f_{i-1}})_{uv} - p_u^{i-1} + p_v^{i-1} + \underline{d_u - d_v} \\
&\qquad\qquad \geq -((c_{f_{i-1}})_{uv} - p_u^{i-1} + p_v^{i-1}) \\
&\geq 0
\end{aligned}
$$



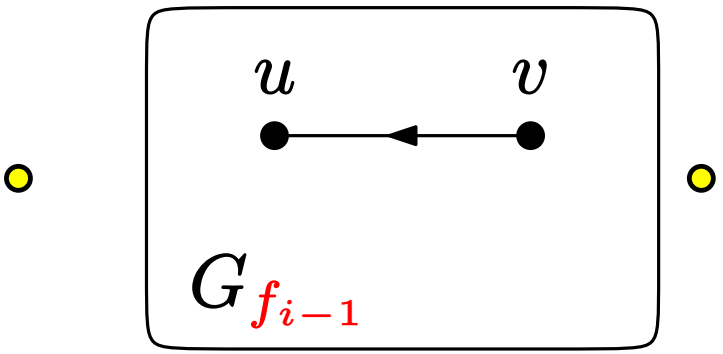
仮定：$(c_{f^{i-1}})_{uv} - p_u^{i-1} + p_v^{i-1} \geq 0 \quad \forall\, uv \in A_{f^{i-1}}$

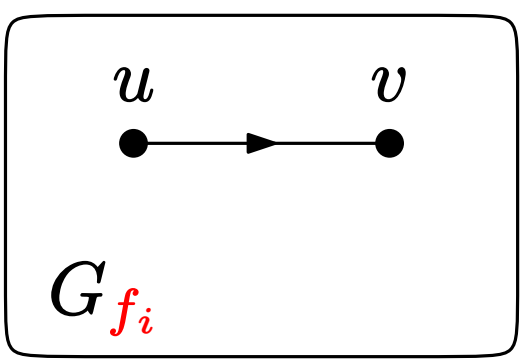
主張：$(c_{f^i})_{uv} - p_u^i + p_v^i \geq 0 \quad \forall\, uv \in A_{f^i}$ $\qquad p^i = p^{i-1} - d$

不変条件：証明 (2/2)

(2) $uv \in A_{f^i} - A_{f^{i-1}}$ のとき

$(c_{f_i})_{uv} - p^i_u + p^i_v$





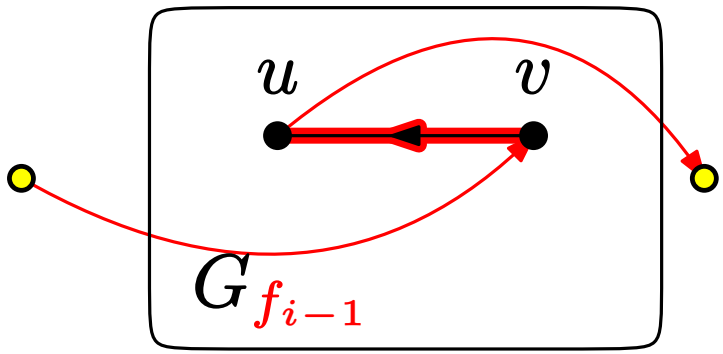
仮定：$(c_{f^{i-1}})_{uv} - p^{i-1}_u + p^{i-1}_v \geq 0 \quad \forall\, uv \in A_{f^{i-1}}$

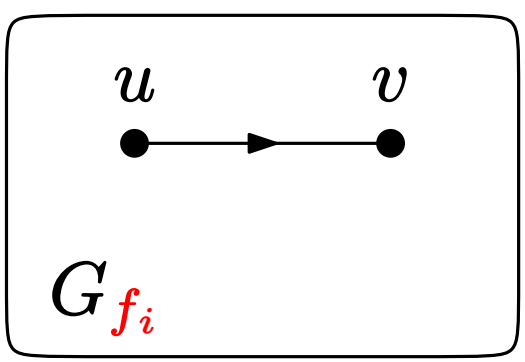
主張：$(c_{f^i})_{uv} - p^i_u + p^i_v \geq 0 \quad \forall\, uv \in A_{f^i}$ $\qquad p^i = p^{i-1} - d$

不変条件：証明 (2/2)

(2) $uv \in A_{f^i} - A_{f^{i-1}}$ のとき

$$(c_{f_i})_{uv} - p^i_u + p^i_v$$



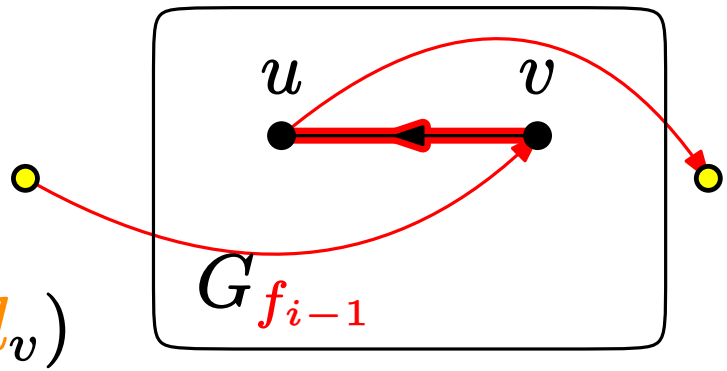


仮定：$(c_{f^{i-1}})_{uv} - p^{i-1}_u + p^{i-1}_v \geq 0 \quad \forall\, uv \in A_{f^{i-1}}$

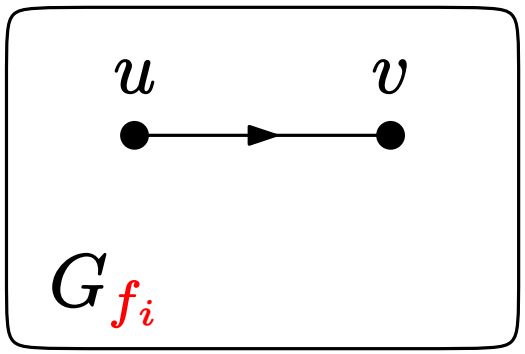
主張：$(c_{f^i})_{uv} - p^i_u + p^i_v \geq 0 \quad \forall\, uv \in A_{f^i}$ $\qquad p^i = p^{i-1} - d$

不変条件：証明 (2/2)

(2) $uv \in A_{f^i} - A_{f^{i-1}}$ のとき



$$
\begin{aligned}
&(c_{f_i})_{uv} - p_u^i + p_v^i \\
&= -(c_{f_{i-1}})_{vu} - (p_u^{i-1} - d_u) + (p_v^{i-1} - d_v) \\
&= -(c_{f_{i-1}})_{vu} - p_u^{i-1} + p_v^{i-1} + d_u - d_v
\end{aligned}
$$

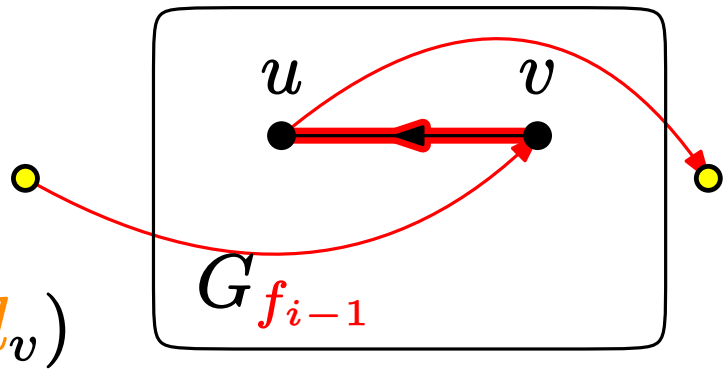


仮定：$(c_{f^{i-1}})_{uv} - p_u^{i-1} + p_v^{i-1} \geq 0 \quad \forall\, uv \in A_{f^{i-1}}$

主張：$(c_{f^i})_{uv} - p_u^i + p_v^i \geq 0 \quad \forall\, uv \in A_{f^i}$ $\qquad p^i = p^{i-1} - d$

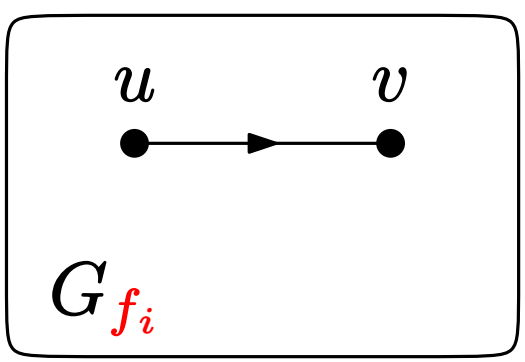
不変条件：証明 (2/2)

(2) $uv \in A_{f^i} - A_{f^{i-1}}$ のとき



$$
\begin{aligned}
&(c_{f_i})_{uv} - p_u^i + p_v^i \\
&= -(c_{f_{i-1}})_{vu} - (p_u^{i-1} - d_u) + (p_v^{i-1} - d_v) \\
&= -(c_{f_{i-1}})_{vu} - p_u^{i-1} + p_v^{i-1} + \underline{d_u - d_v} \\
&\qquad\qquad = (c_{f_{i-1}})_{vu} - p_v^{i-1} + p_u^{i-1} \\
&= 0
\end{aligned}
$$

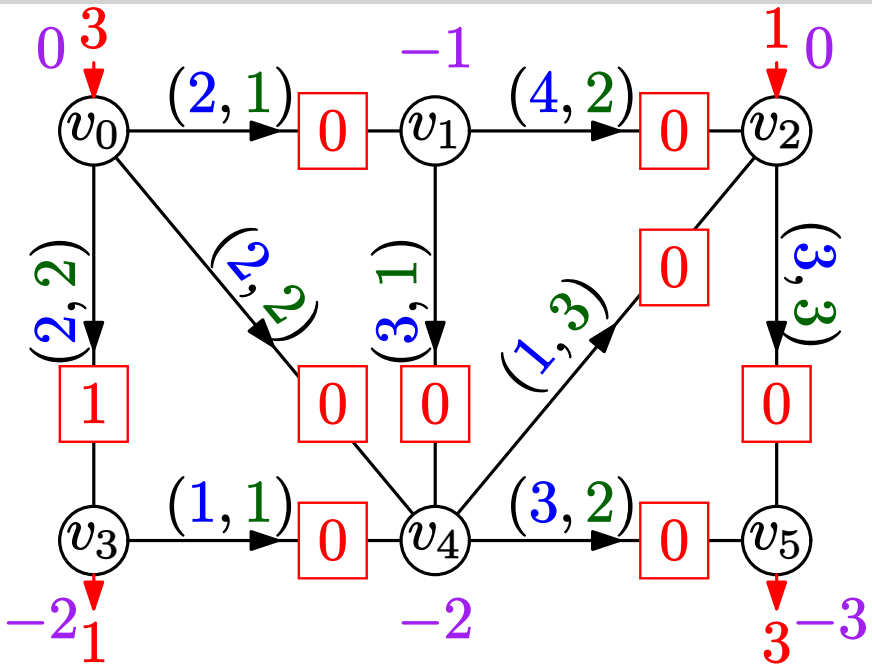
□

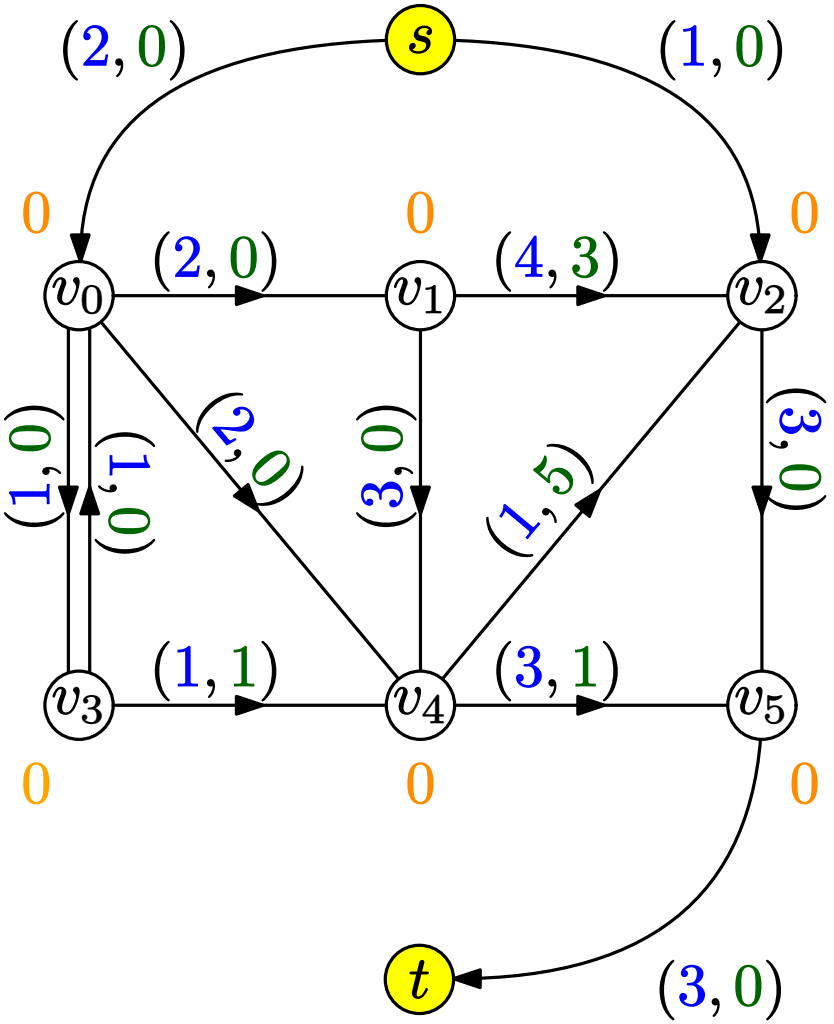


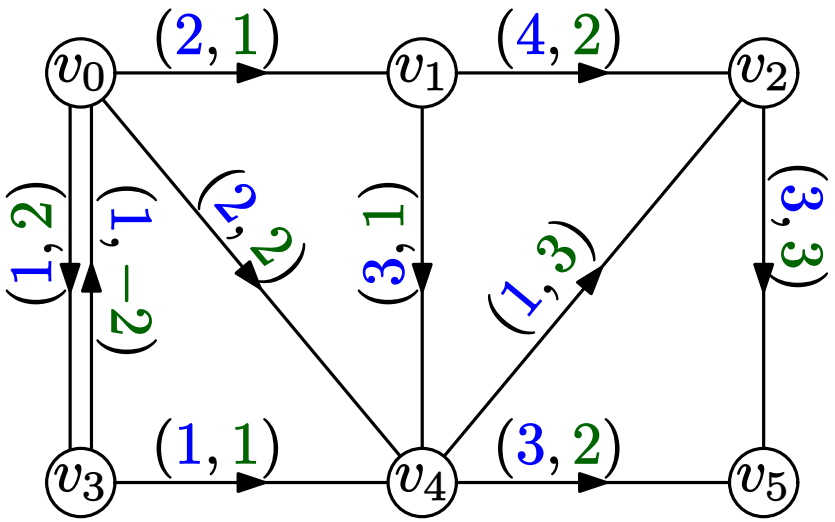
仮定：$(c_{f^{i-1}})_{uv} - p_u^{i-1} + p_v^{i-1} \geq 0 \quad \forall\, uv \in A_{f^{i-1}}$

主張：$(c_{f^i})_{uv} - p_u^i + p_v^i \geq 0 \quad \forall\, uv \in A_{f^i}$ $\qquad p^i = p^{i-1} - d$

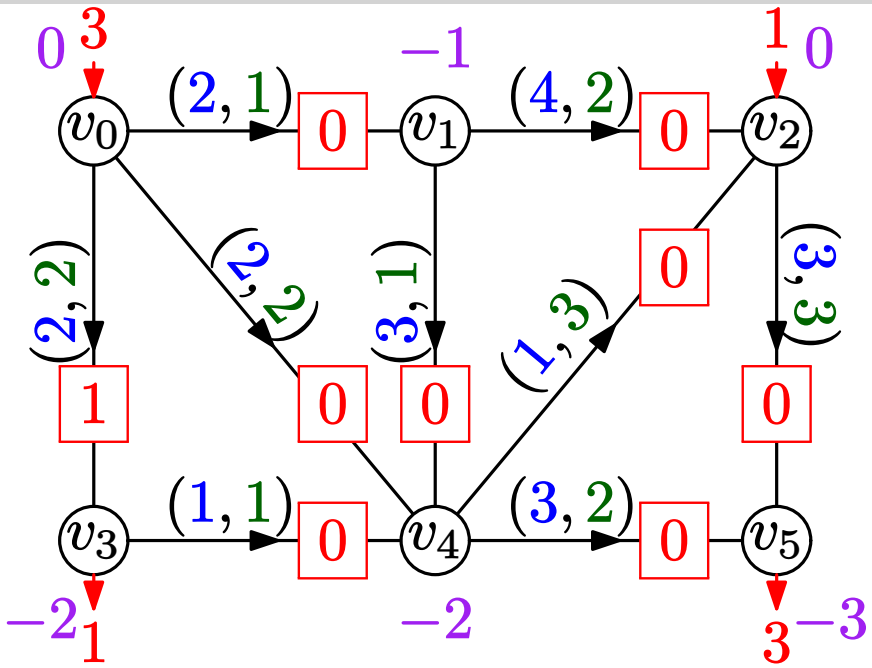
最短路を使わないと？(1/2)

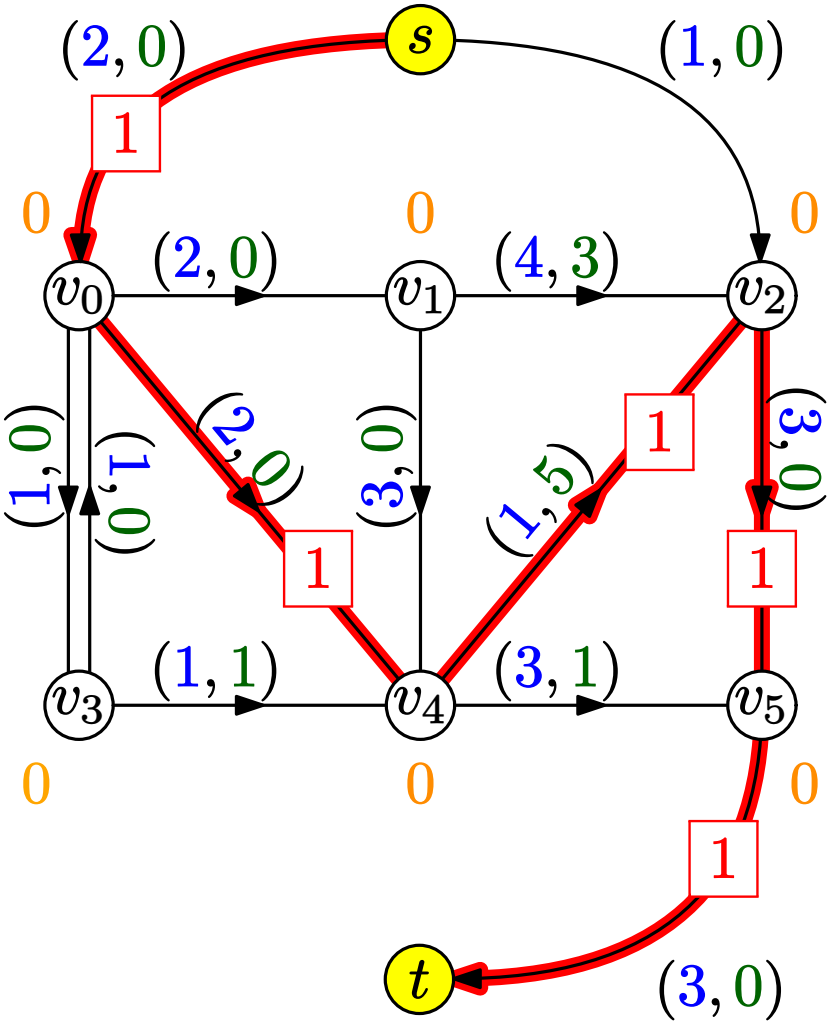


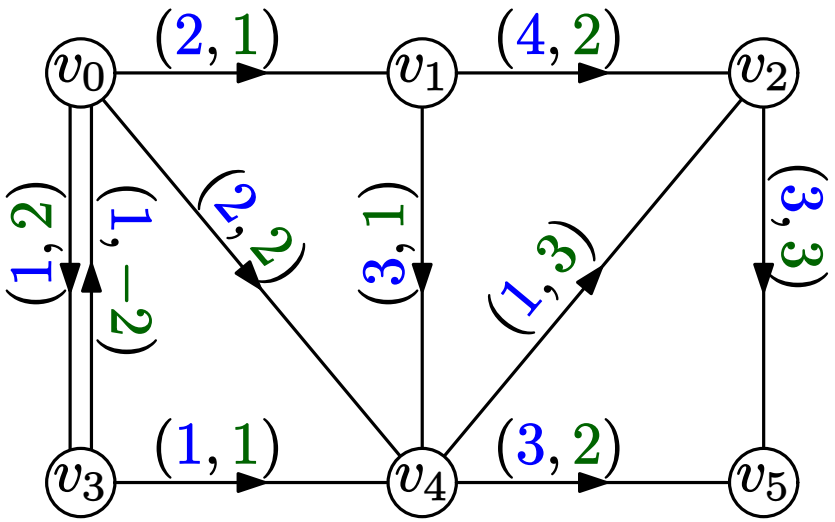




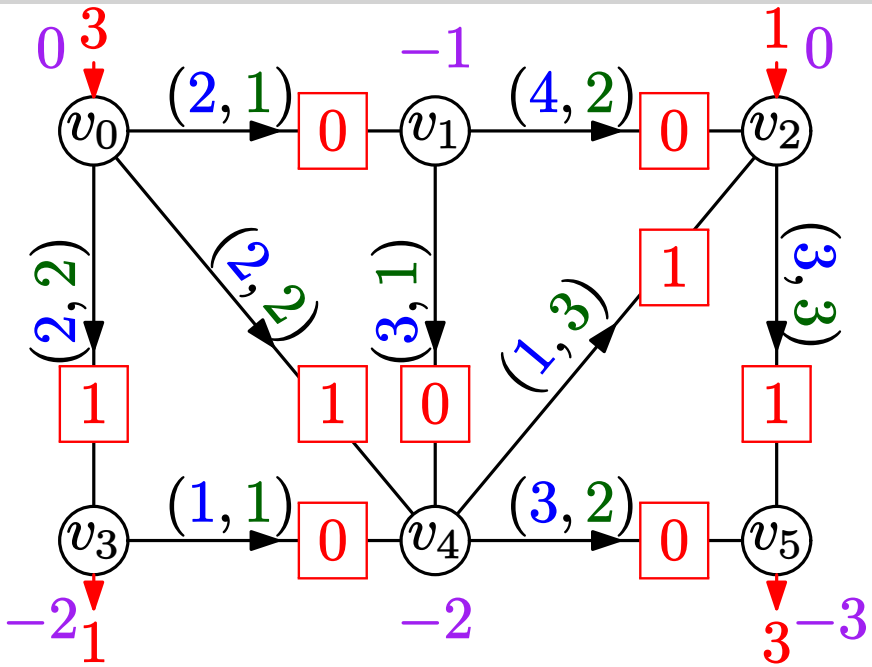
最短路を使わないと？(1/2)

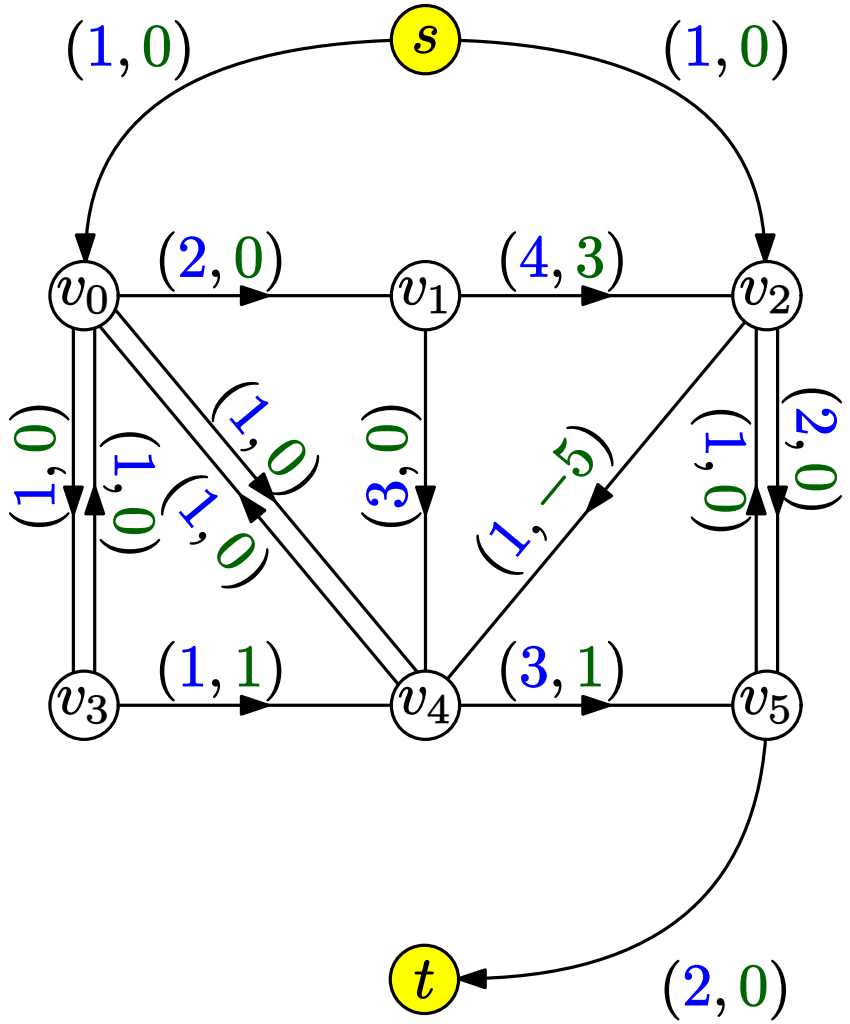


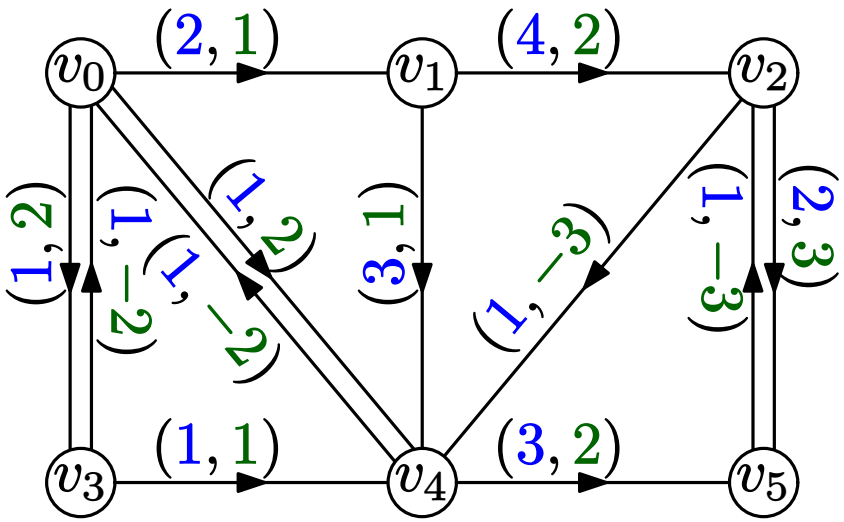




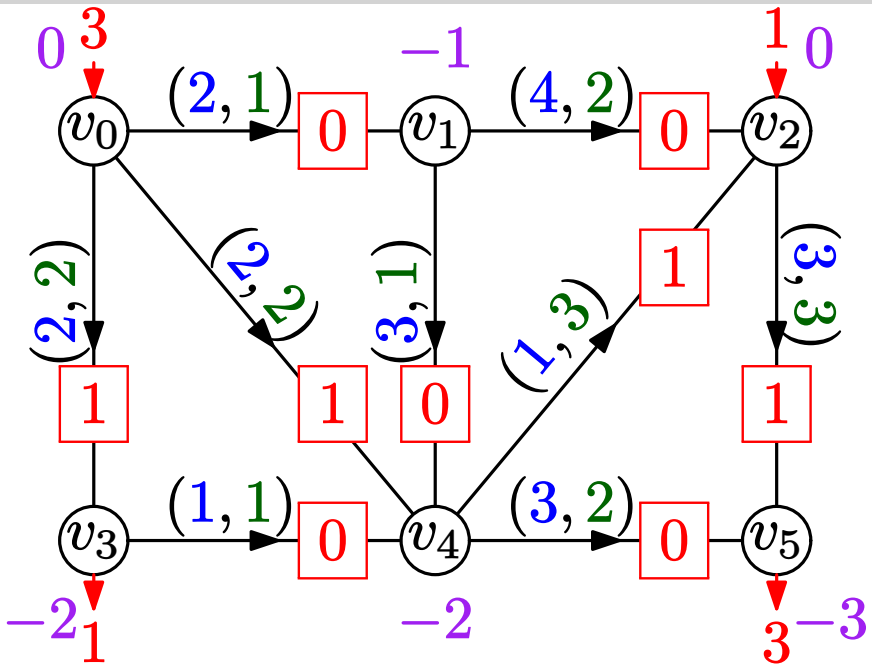
最短路を使わないと？(2/2)

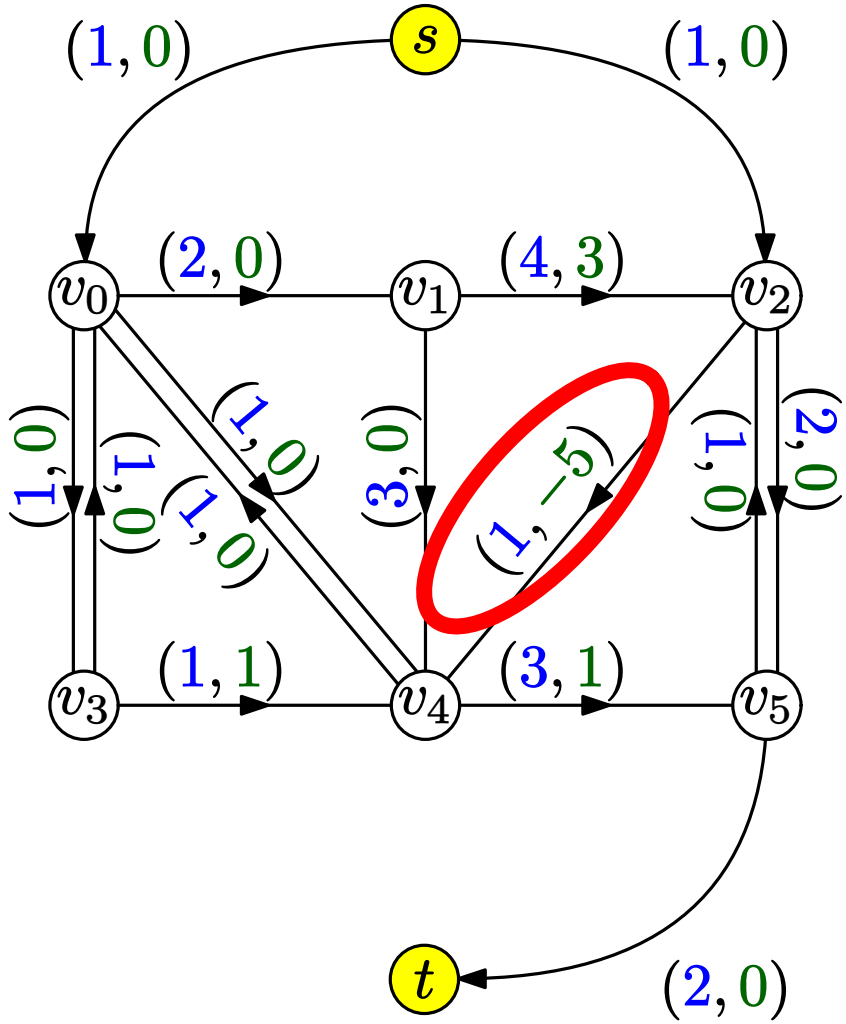


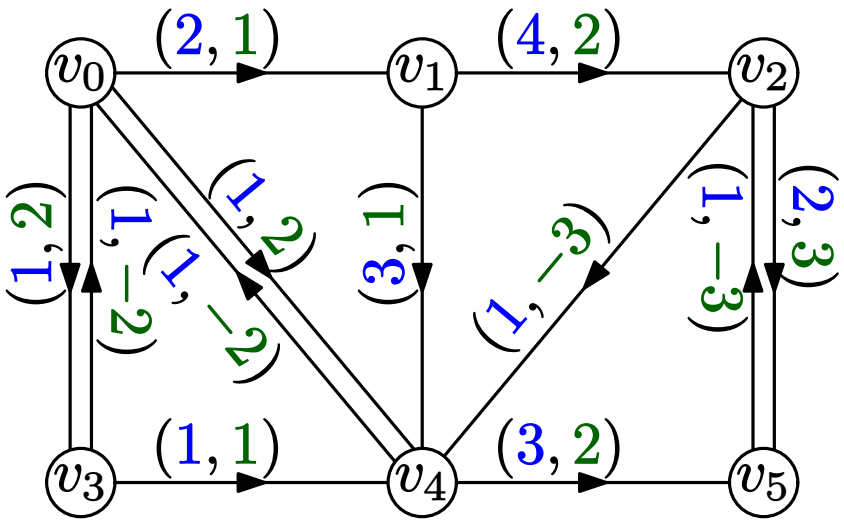




最短路を使わないと？ (2/2)







逐次最短路法：正当性

性質：逐次最短路法の正当性

逐次最短路法が停止する ⇒
出力は最小費用 b-流である

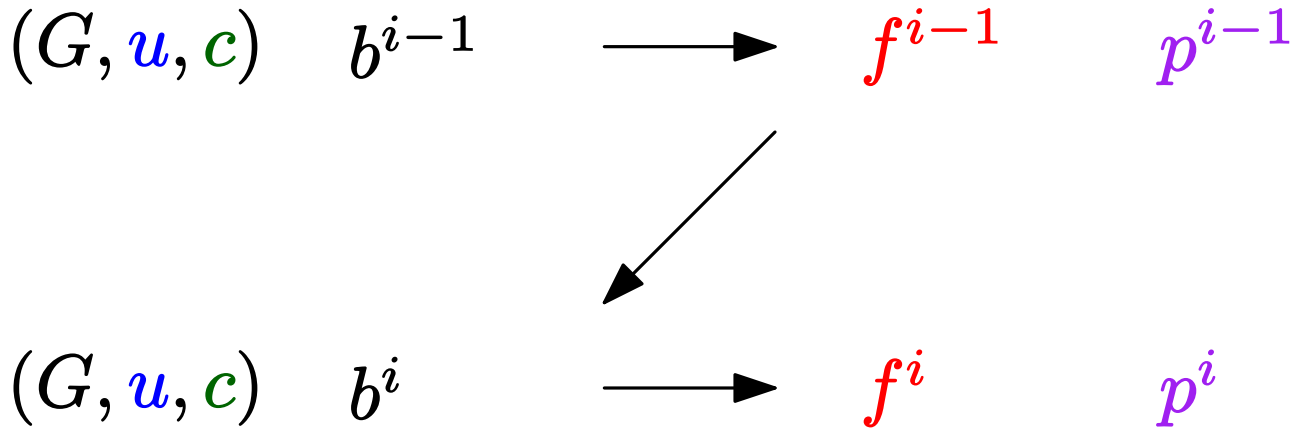
補足：逐次最短路法

- 元の英名：successive shortest path algorithm
- 他の和名：最短路繰返し法
- 源流：
  - Jewell (’58), Iri (’60), Busacker, Gowen (’61)
  - Edmonds, Karp (’72), Tomizawa (’71)

Iri = 伊理 正夫 (1933–2018)
Tomizawa = 富沢 信明 (1942–)

本日の内容

1. 逐次最短路法
2. **逐次最短路法：計算量**

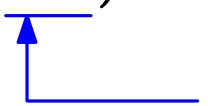


逐次最短路法の計算量

容量・供給/需要が実数の場合

工夫がない $\rightsquigarrow$ 停止しない場合がある

工夫がある $\rightsquigarrow$ $O(m \log n \cdot \mathrm{SP})$ (強多項式時間)



最短路問題の計算量
$O(m + n \log n)$

Orlin ('93)

容量・供給/需要が整数の場合

工夫がない $\rightsquigarrow$ 擬多項式時間 (今から)

工夫がある $\rightsquigarrow$ 弱多項式時間 (次回)

最小費用流問題：入力のサイズ

最小費用流問題の入力	サイズ (ビット)
• 有向グラフ G の頂点集合 V	• $O(\lvert V\rvert \log \lvert V\rvert)$
• 有向グラフ G の弧集合 A	• $O(\lvert A\rvert \log \lvert V\rvert)$
• 弧容量関数 $u\colon A \to \mathbb{Z}_+$	• $O\left(\sum_{a\in A} \log u(a)\right)$
• 弧費用関数 $c\colon A \to \mathbb{Z}_+$	• $O\left(\sum_{a\in A} \log c(a)\right)$
• 供給/需要 $b\colon V \to \mathbb{Z}$	• $O\left(\sum_{v\in V} \log \lvert b(v)\rvert\right)$

最小費用流問題：入力のサイズ

最小費用流問題の入力	サイズ (ビット)
• 有向グラフ G の頂点集合 V	• $O(\lvert V\rvert \log \lvert V\rvert)$
• 有向グラフ G の弧集合 A	• $O(\lvert A\rvert \log \lvert V\rvert)$
• 弧容量関数 $u\colon A \to \mathbb{Z}_+$	• $O\left(\sum_{a\in A} \log u(a)\right)$
• 弧費用関数 $c\colon A \to \mathbb{Z}_+$	• $O\left(\sum_{a\in A} \log c(a)\right)$
• 供給/需要 $b\colon V \to \mathbb{Z}$	• $O\left(\sum_{v\in V} \log \lvert b(v)\rvert\right)$

$U = \max_{a\in A} u(a)$, $C = \max_{a\in A} c(a)$, $B = \max_{v\in V} |b(v)|$ として,

計算量を $n = |V|, m = |A|, U, C, B$ に対する依存性で測る

3 種類の計算時間

最小費用流問題に対して

定義：強/弱/擬多項式時間アルゴリズム

{ **強多項式時間** / **弱多項式時間** / **擬多項式時間** } アルゴリズム とは 計算時間が

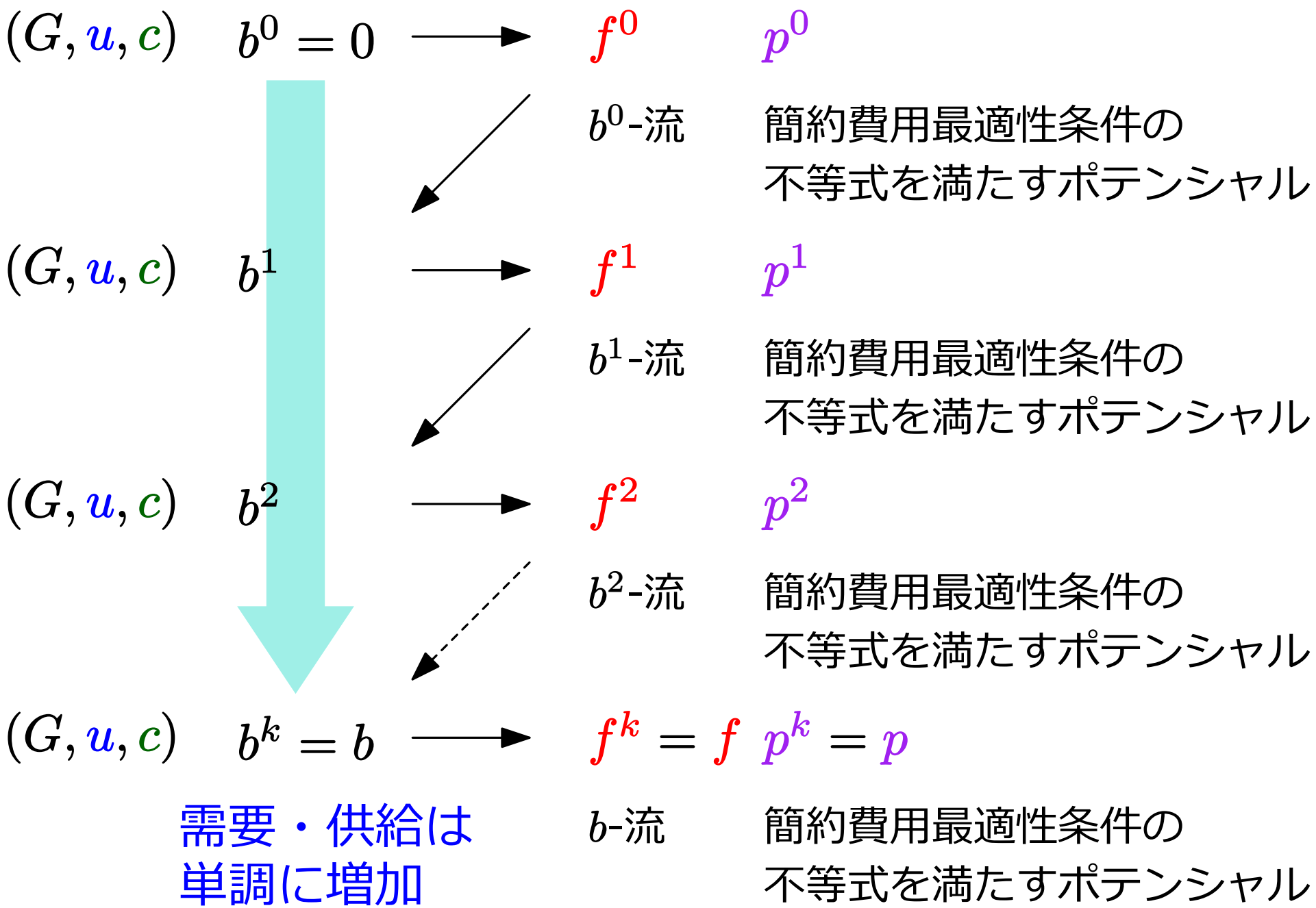
{ n, m / $n, m, \log U, \log C, \log B$ / n, m, U, C, B } の多項式で抑えられる アルゴリズム

注：強多項式時間 $\Rightarrow$ 弱多項式時間 $\Rightarrow$ 擬多項式時間

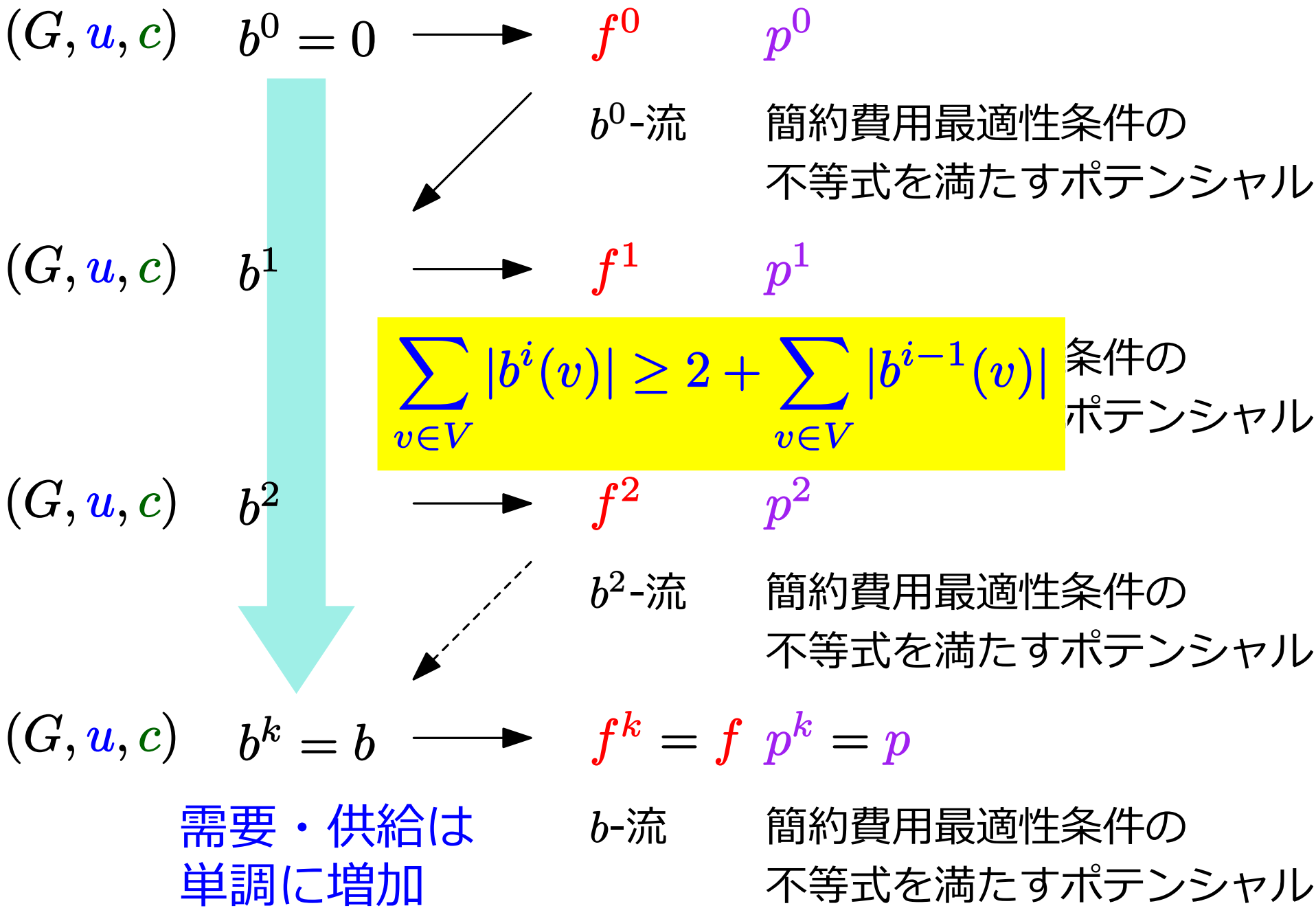
（強多項式時間 $\Rightarrow$ 弱多項式時間：多項式時間アルゴリズムである）

（弱多項式時間 $\Rightarrow$ 擬多項式時間：多項式時間アルゴリズムではない）

逐次最短路法：計算量 (1/2)



逐次最短路法：計算量 (1/2)



逐次最短路法：計算量 (1/2)

(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) $b^k = b$ → $f^\kappa = f$ $p^\kappa = p$

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

逐次最短路法：計算量 (2/2)

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

逐次最短路法：計算量 (2/2)

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

反復回数 $= O(nB)$

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

逐次最短路法：計算量 (2/2)

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

反復回数 $= O(nB)$

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

ここ以外は $O(m)$

逐次最短路法：計算量 (2/2)

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

反復回数 $= O(nB)$

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

ここ以外は $O(m)$

最短路計算 (Dijkstra 法で $O(m + n \log n)$)

逐次最短路法の計算量：擬多項式時間

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)

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**		• **逐次最短路法**
• 負閉路最適性条件		• 負閉路消去法
• 正カット最適性条件		• 正カット消去法

次回の内容

- 逐次最短路法の高速化 (容量スケーリング)
  ⇝ 弱多項式時間
- 主双対法