フローとカット
Flow and Cut

ネットワーク中のフロー (流量、FLOW)

- 交通網中の流れ
  - 都市間を流れる車両の数、及びその上限
  - 都市間を結ぶ航空路線が輸送する人数とその上限
- 物流
  - 倉庫間を移動している商品の数とその上限
- 作業
  - 各工程間のコストとその上限

容量と流れ

- 各弧に容量(上限)が存在する
  - 交通機関の輸送能力
  - 通信速度
- 各弧に実際に流れる流量
  - 容量以下
- 二点間に最大流量を実現する方法？

2端子フロー

- グラフ $G=(V,A)$
- 入口 $s^+ \in V$ と出口 $s^- \in V$ が定義され、s^+ から s^- への有向道がある
- $\forall a \in A$ に、流量上限 $c(a) \geq 0$ が定義されている
  - 弧の重みを上限として解釈する
- $\forall a \in A$ に、流量(flow) $\varphi(a) \geq 0$ が設定される

$$N=\left(G(V,A),s^+,s^-,c\right)$$

流量に対する制約

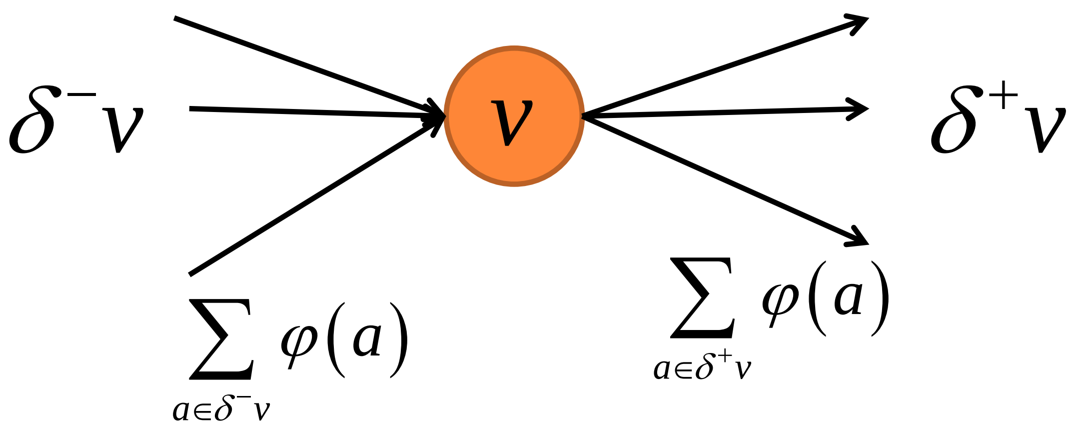
- 容量制約

$$0 \le \varphi(a) \le c(a)$$

- 流量保存則
  - 頂点vで「湧きだし」、「吸い込み」がない

$$\partial\varphi(v) \equiv \sum_{a \in \delta^+ v} \varphi(a) - \sum_{a \in \delta^- v} \varphi(a) = 0$$

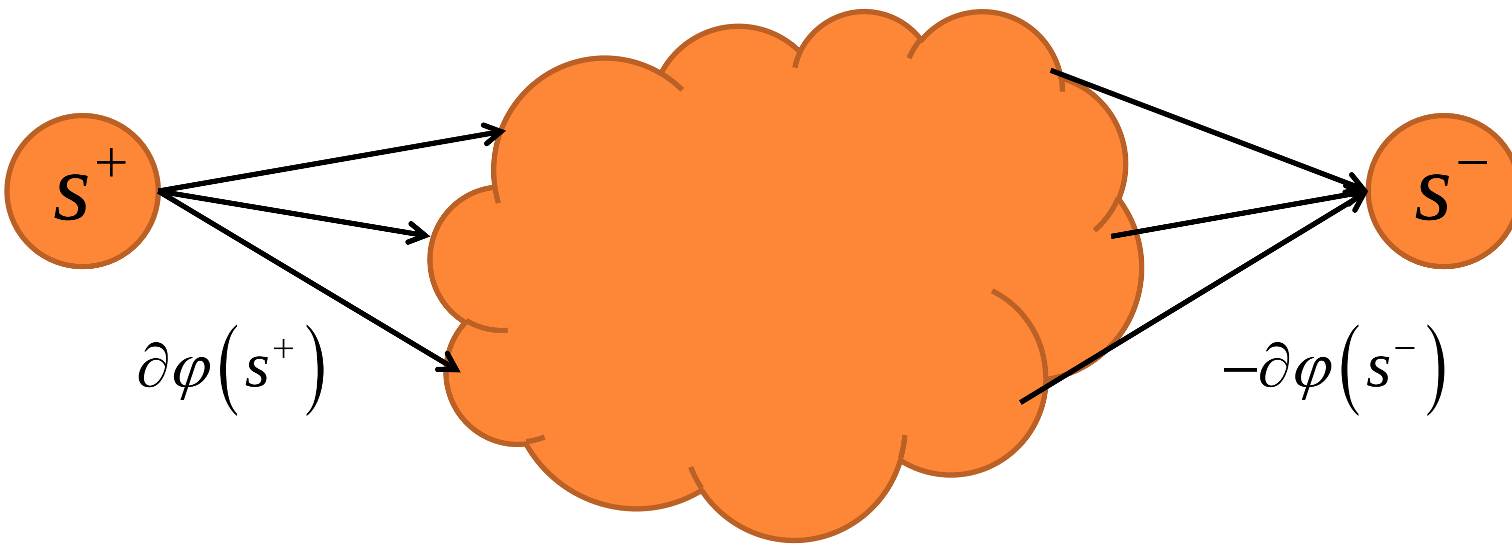
$$\forall v \in V \setminus \{s^+, s^-\}$$



ネットワークフローのイメージ

- ネットワークの流量

$$Q(\varphi) = \partial\varphi(s^+) = -\partial\varphi(s^-)$$

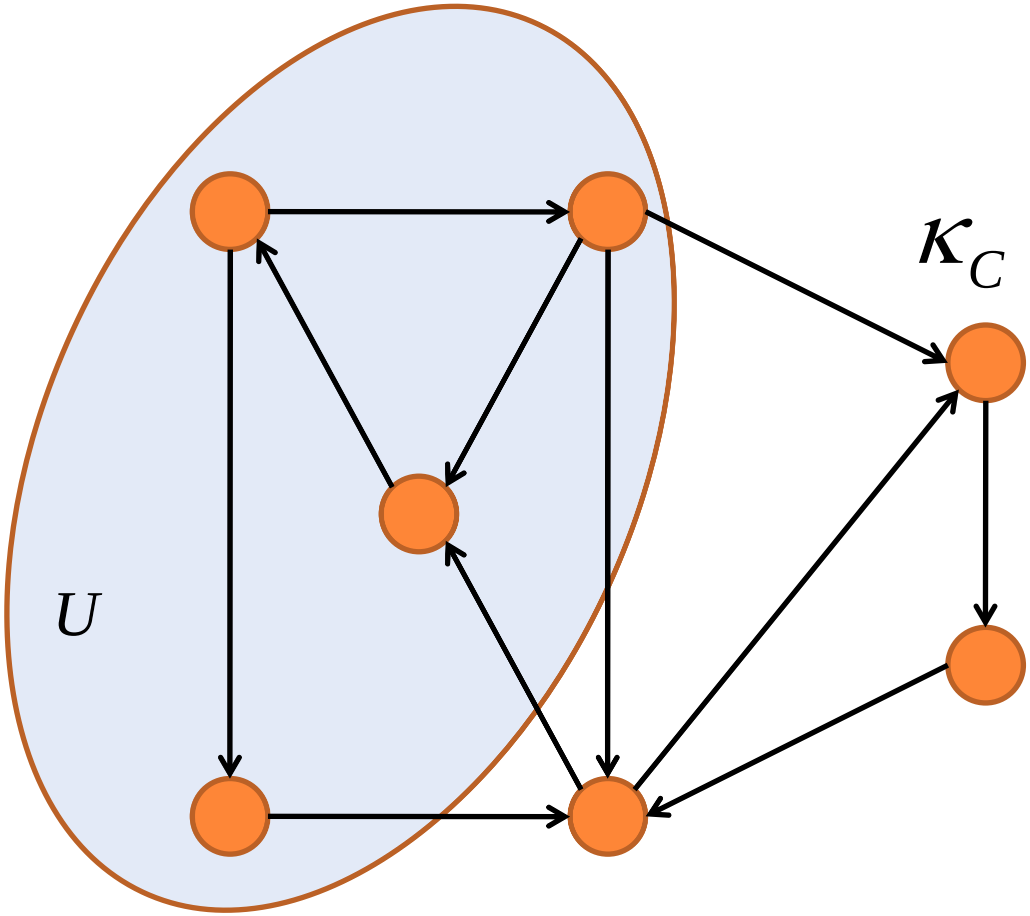


カット(CUT)

- ネットワークのカット$U \subseteq V$
  - $\left(s^{+} \in U\right) \wedge \left(s^{-} \notin U\right)$となる頂点の集合
- カットの容量
  - $\Delta^{+}U$ ： U から出て、$U \setminus V$ へ入る弧の全体

$$\kappa_C\left(U\right) = \sum_{a \in \Delta^{+}U} c\left(a\right)$$

カットとその境界



$\Delta^{+}U$

$$\kappa_C(U) = \sum_{a \in \Delta^{+}U} c(a)$$

$\Delta^{-}U$

最大流量は最小カットに対応

- N 中の任意のフロー φ と任意のカット U に対して次が成り立つ。

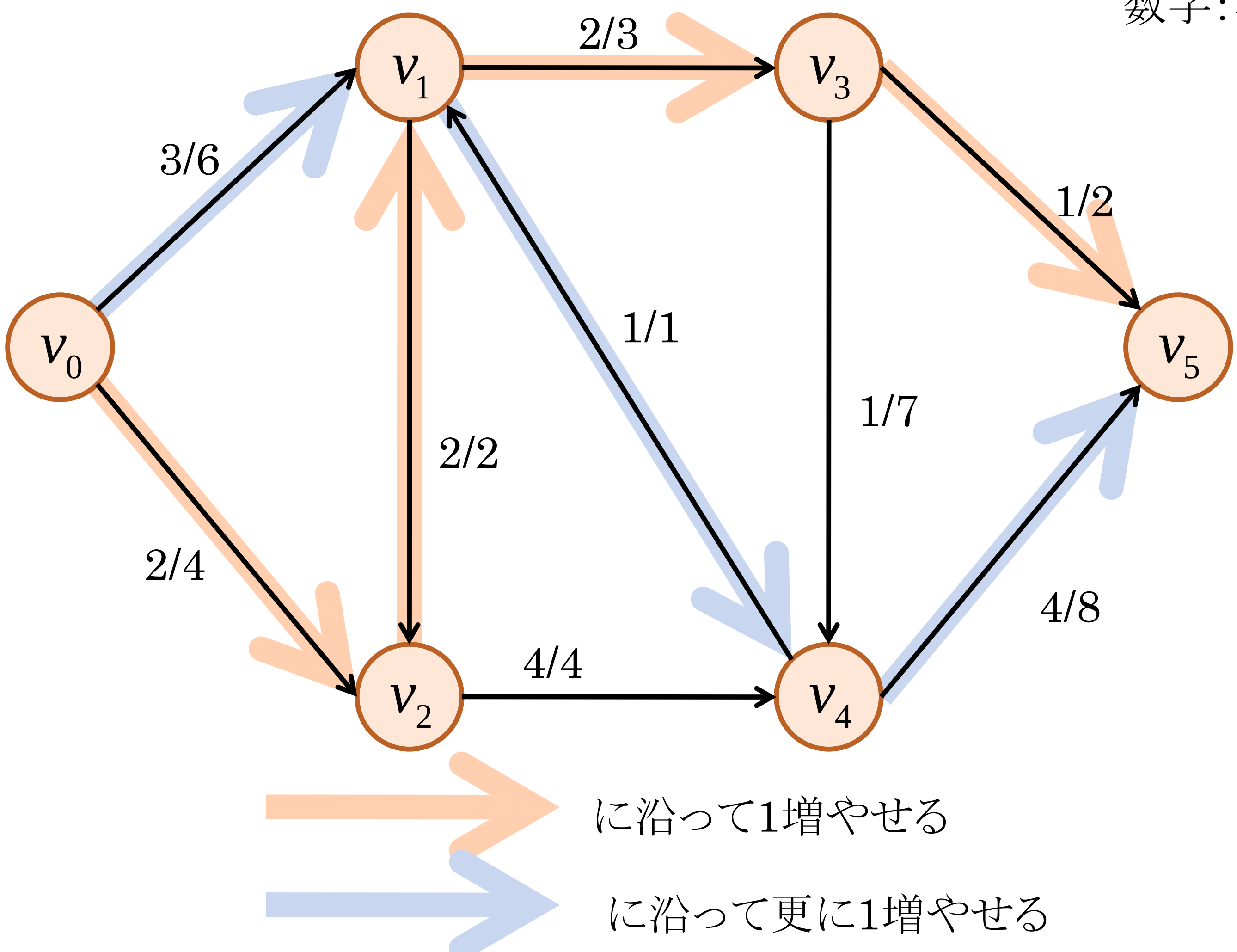
$$Q(\varphi) \leq \kappa_C(C)$$

- 証明

$$\begin{aligned} Q(\varphi) &\leq \sum_{a \in \Delta^+ U} \varphi(a) - \sum_{a \in \Delta^- U} \varphi(a) \\ &\leq \sum_{a \in \Delta^+ U} c(a) - 0 \\ &= \kappa_C(U) \end{aligned}$$

最大フローを見付ける考え方

数字:流量/容量



補助ネットワーク(Auxiliary Network)

$$N = \left(G_\varphi\left(V, A_\varphi\right), s^+, s^-, c_\varphi\right)$$

$$A_\varphi = A_\varphi{}^+ \cup A_\varphi{}^-$$

A_φ^+ ：順方向に容量に余裕のある孤の集合

A_φ^- ：元のネットワークと逆方向の弧の集合。ただし流量 0 の弧を除く

元のネットワークの$\forall a \in A$に対して

- $\varphi(a) < c(a)$ならば
  - $A_\varphi^+ \leftarrow A_\varphi^+ \cup \{a\}$ ：弧の追加
  - $c_\varphi(a) = c(a) - \varphi(a)$ ：容量の設定
  - $\varphi(a) = c(a)$である弧はA_φ^+に追加されないことに注意

- $0 < \varphi(a)$ ならば

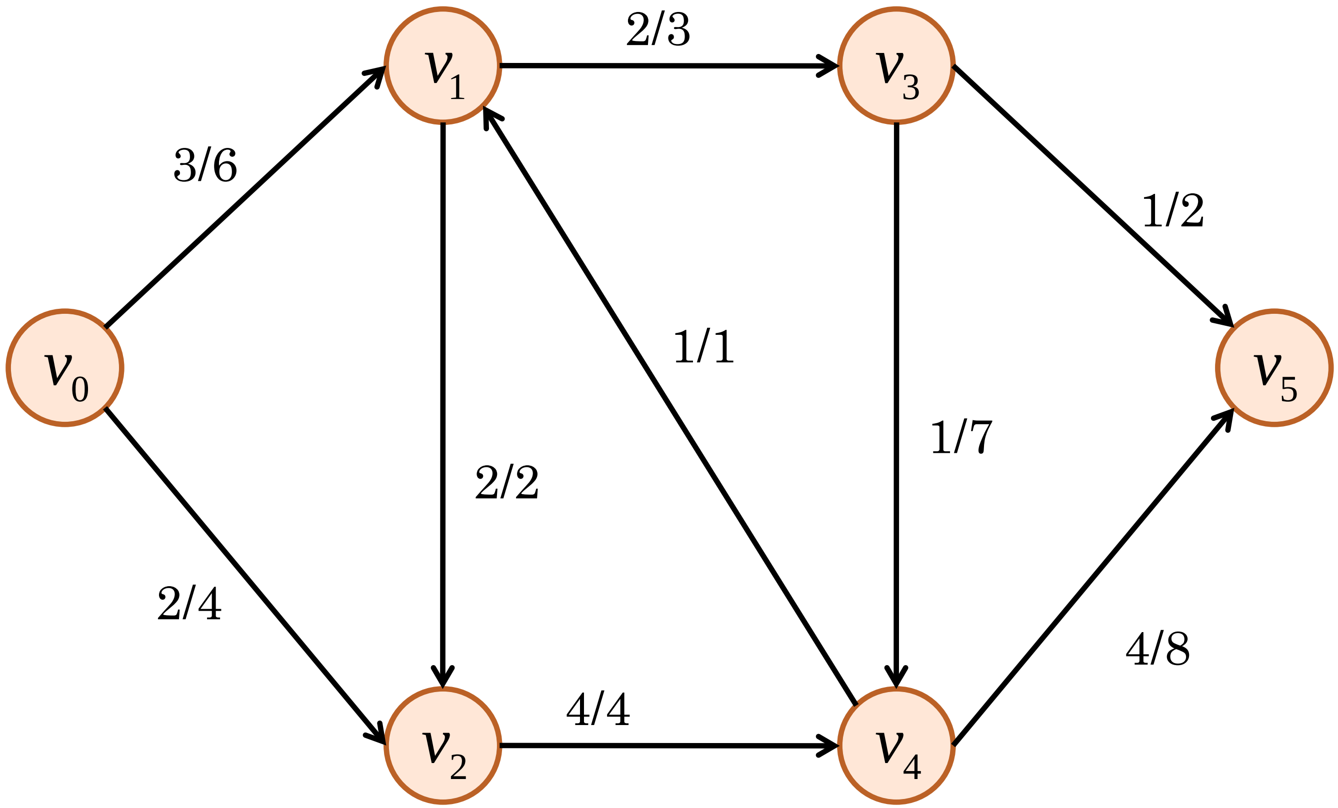
  - $A_{\varphi}^{-} \leftarrow A_{\varphi}^{-} \cup \{b\}$ ：a と逆向きの弧 b を追加

  - $c_{\varphi}(b) = \varphi(a)$ ：容量設定

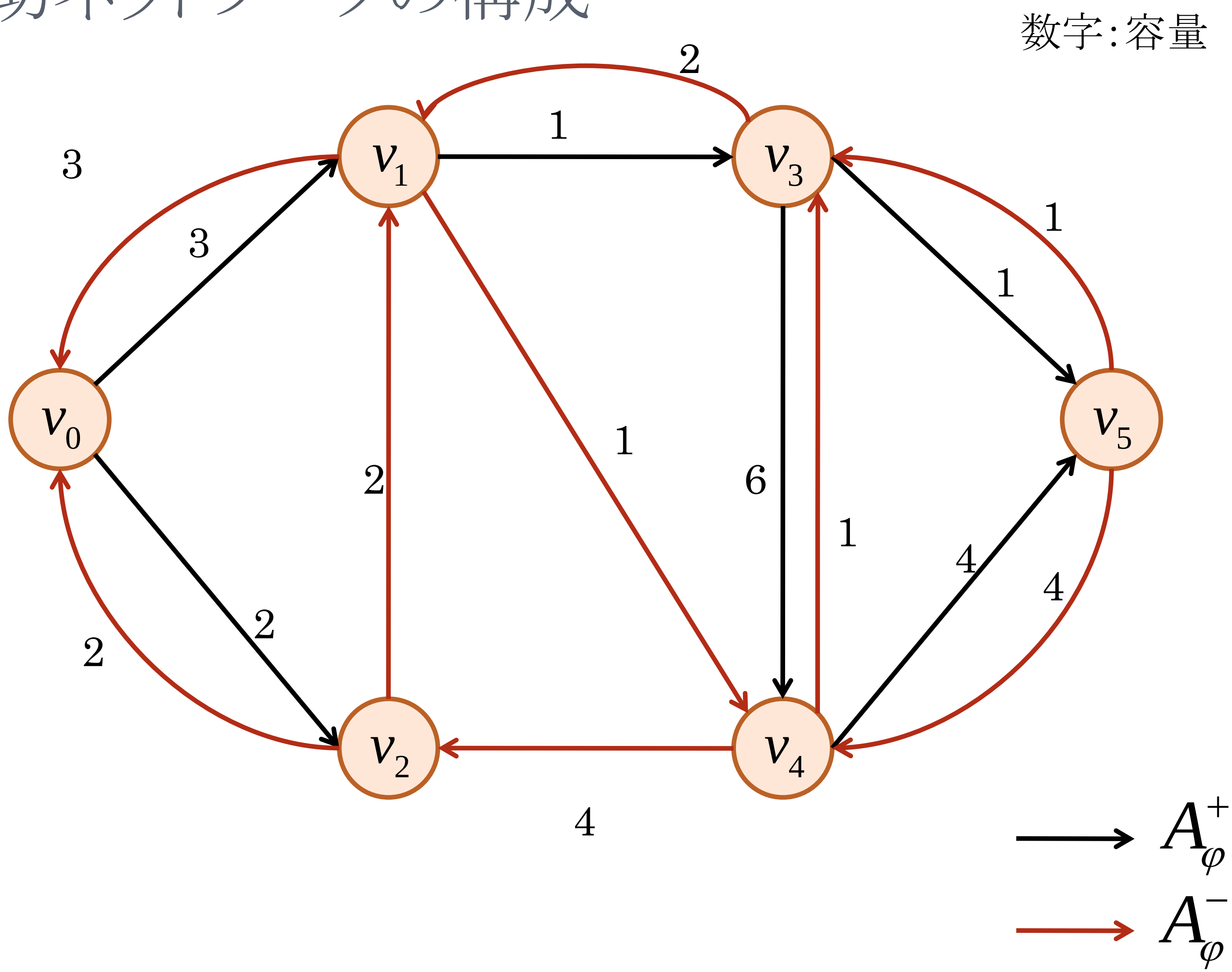
  - 流量が 0 の弧以外はすべて含まれることに注意
  - 元のグラフと逆向きの弧であることに注意

補助ネットワークの構成

数字:流量/容量



補助ネットワークの構成



補助ネットワーク中にs^+からs^-への有向道Pがあれば以下のdだけ流量を増やすことができる。

$$d = \min\left(c_\varphi(a) \middle| a \in P\right) > 0$$

Pを増加道と呼ぶことにする。

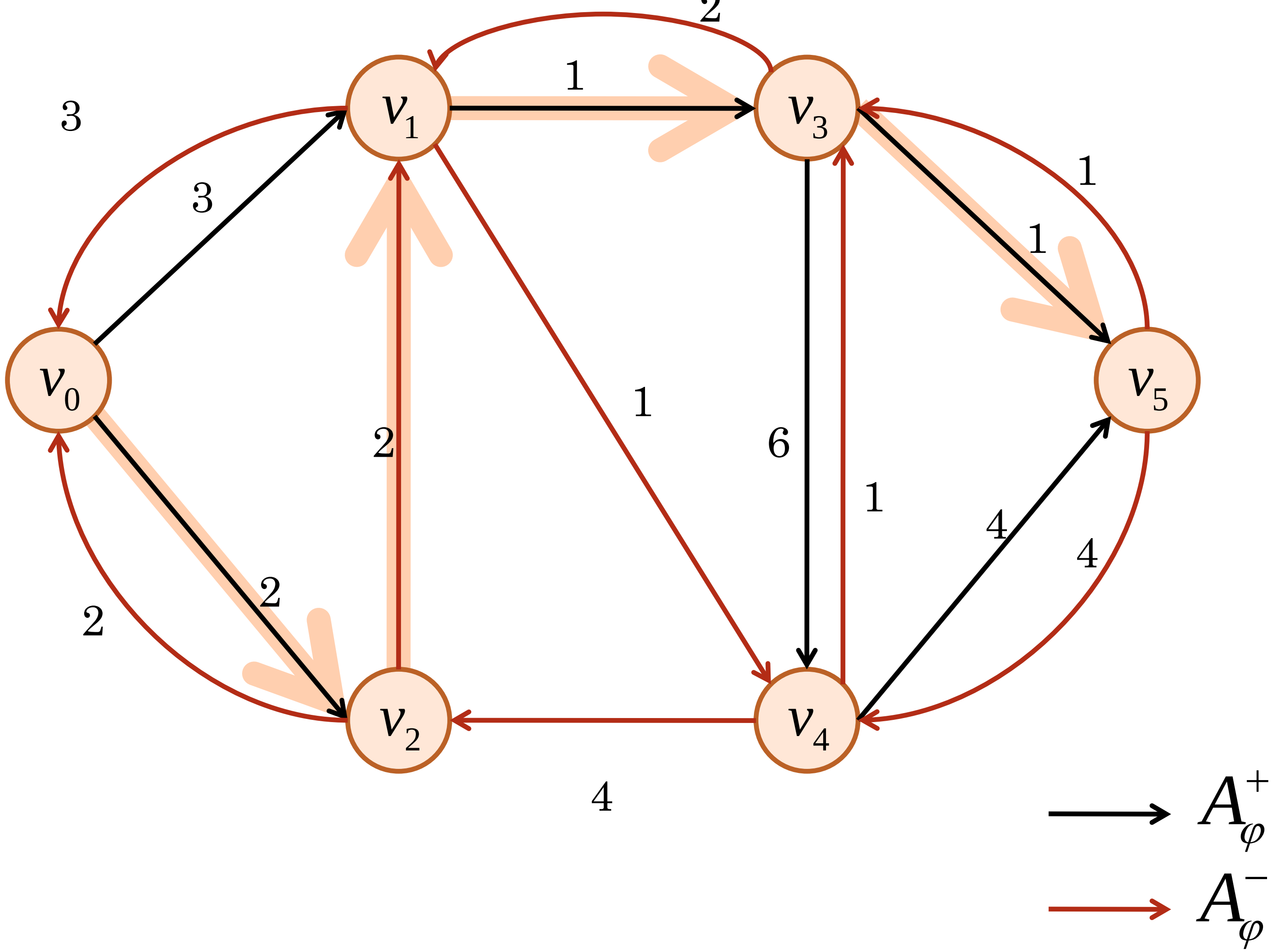
- このときの補助ネットワーク中の新しい流量

$$\varphi'(a) = \begin{cases} \varphi(a) + d & \text{for } a \in A_\varphi^+ \wedge a \in P \\ \varphi(a) - d & \text{for } a \in A_\varphi^- \wedge a \in P \\ \varphi(a) & \text{otherwise} \end{cases}$$

- 容量cが整数関数ならば、流量は整数

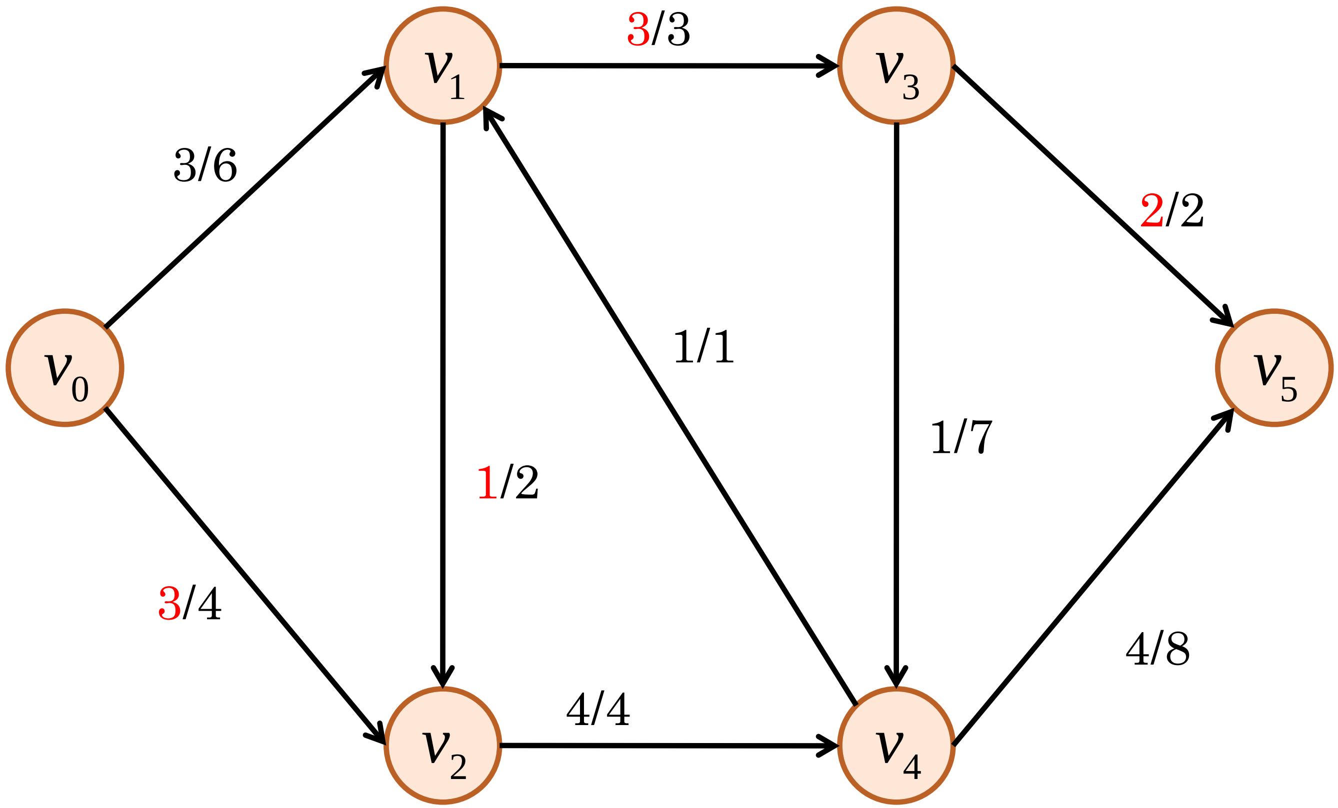
補助ネットワーク内の増加道

数字:容量



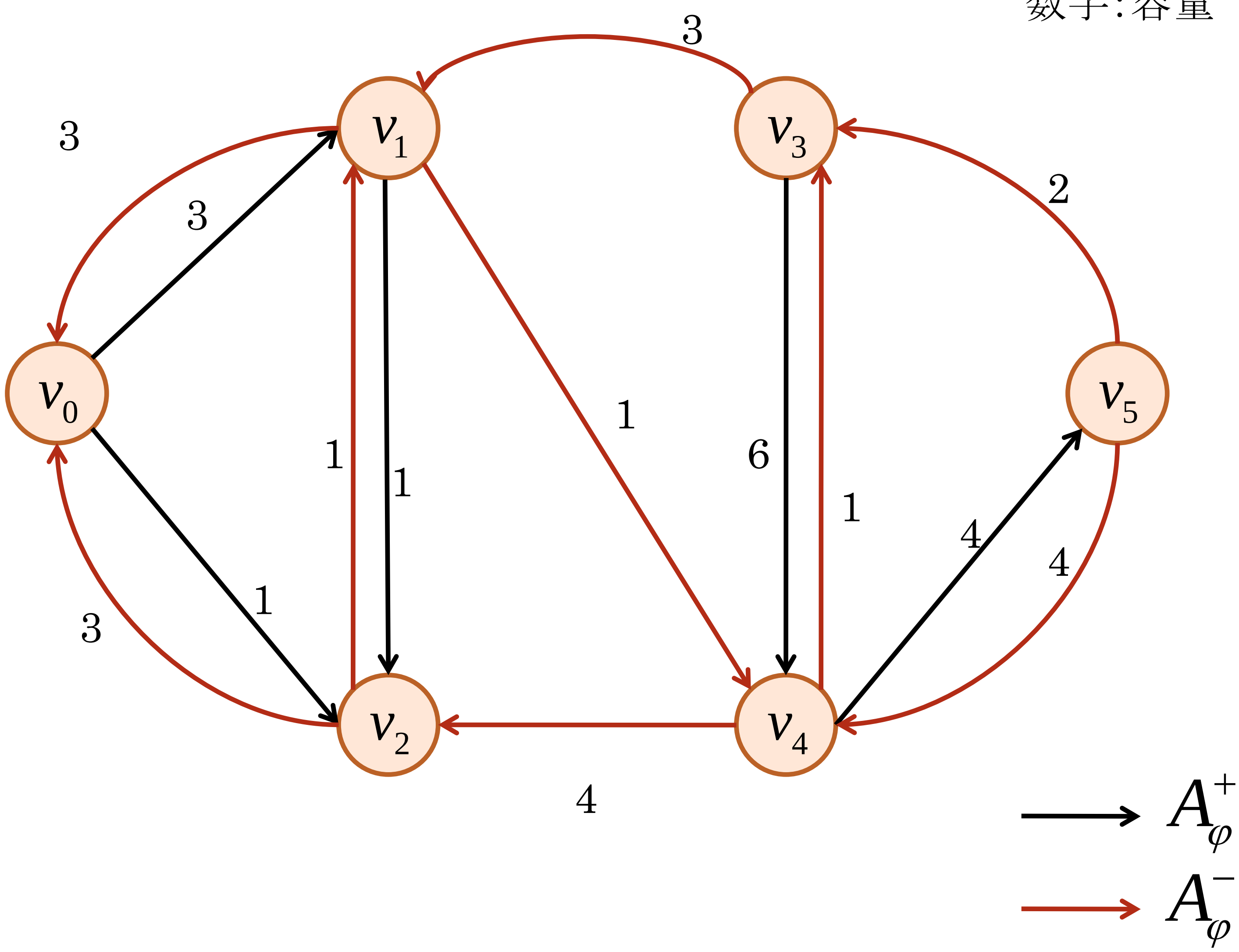
流量增加

数字:流量/容量



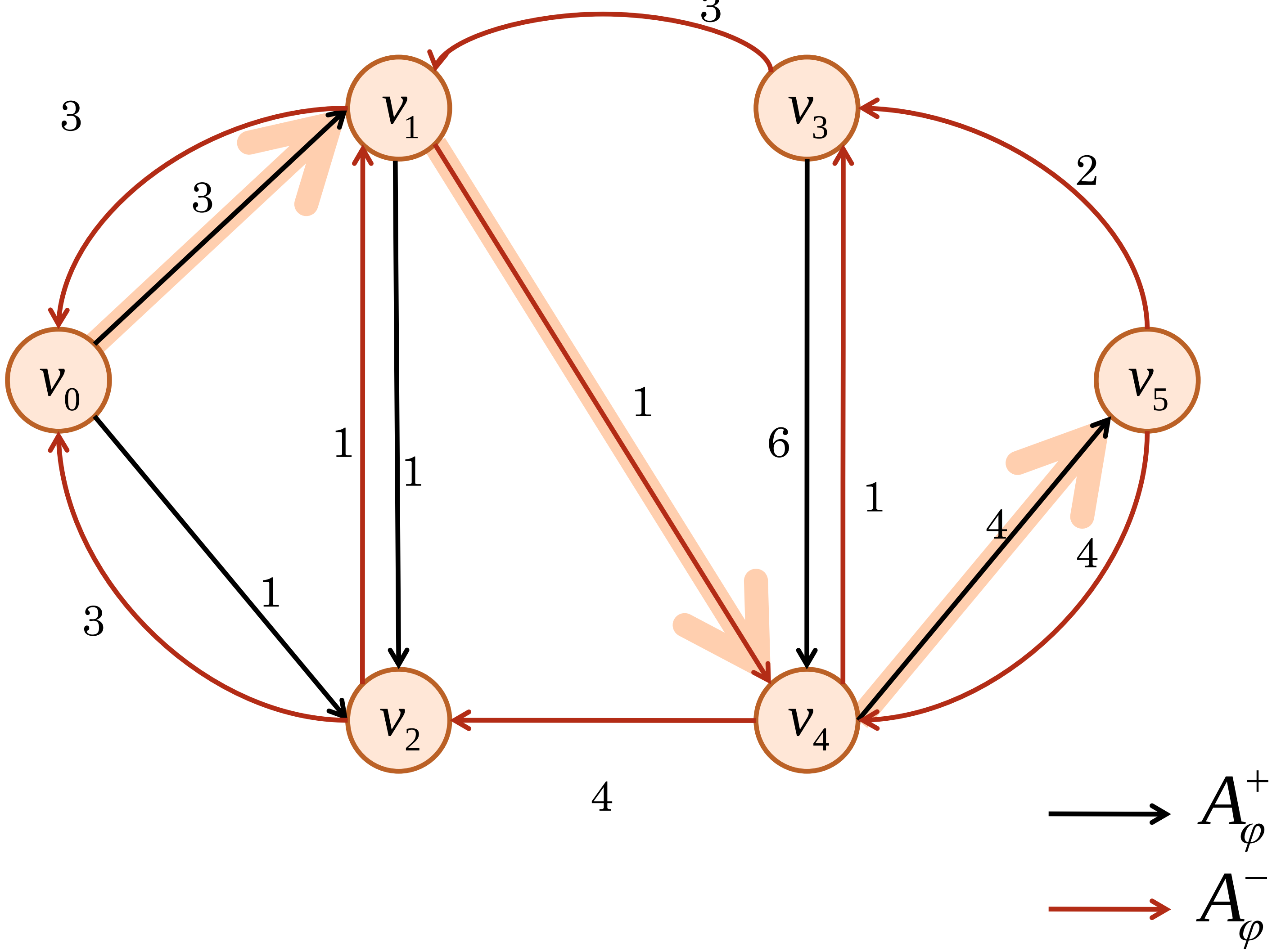
補助ネットワークの更新

数字:容量



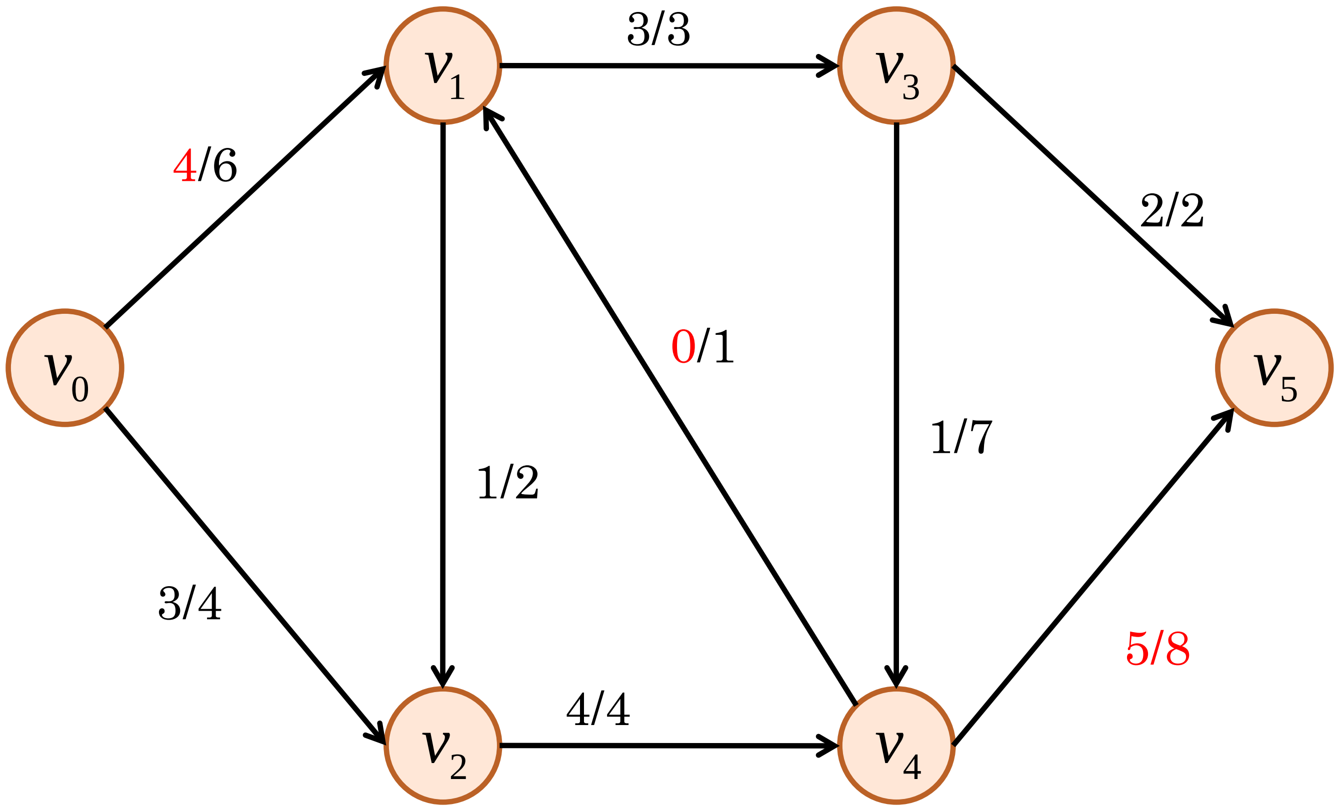
補助ネットワーク内の増加道(再)

数字:容量



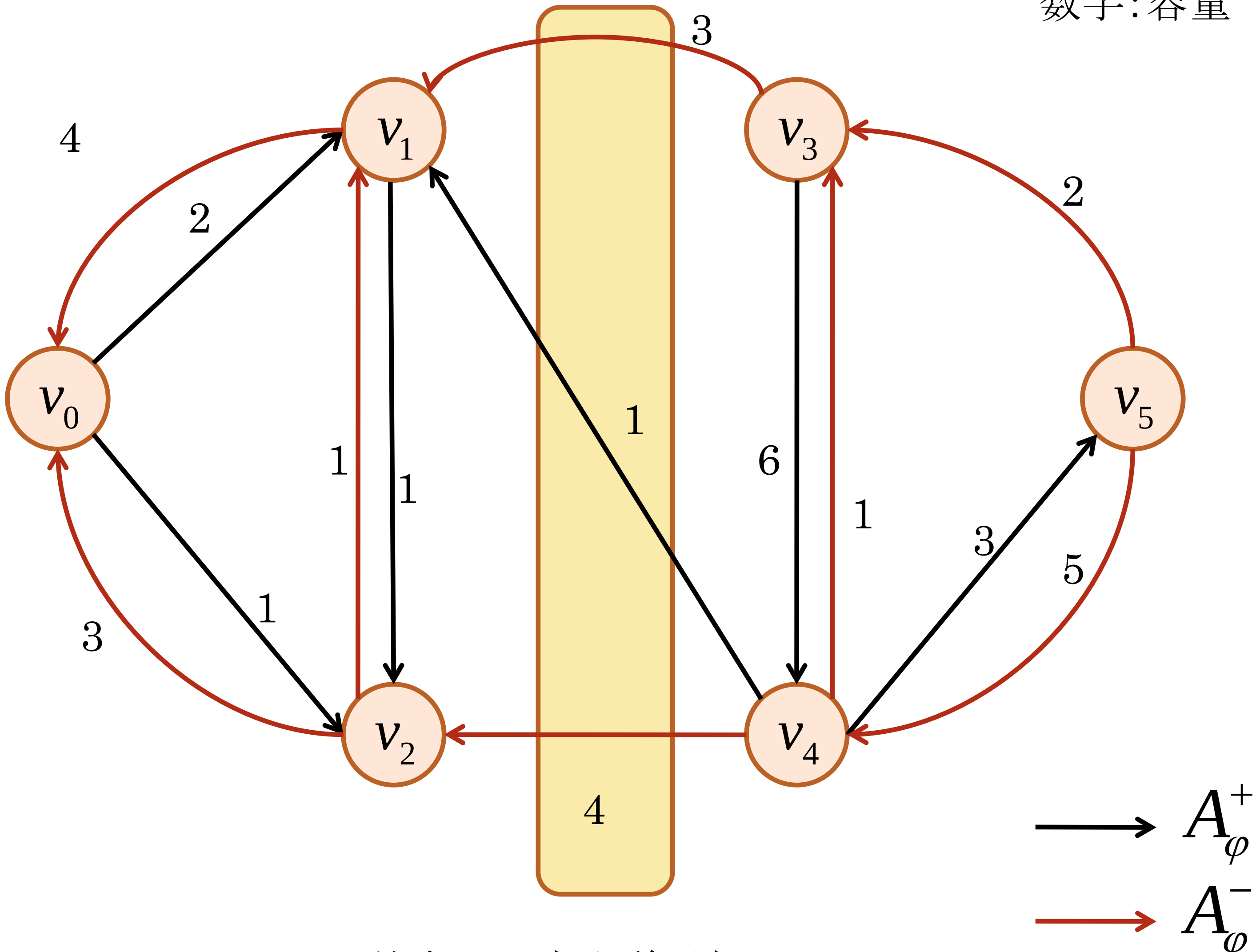
流量增加

数字:流量/容量



補助ネットワークの更新(再)

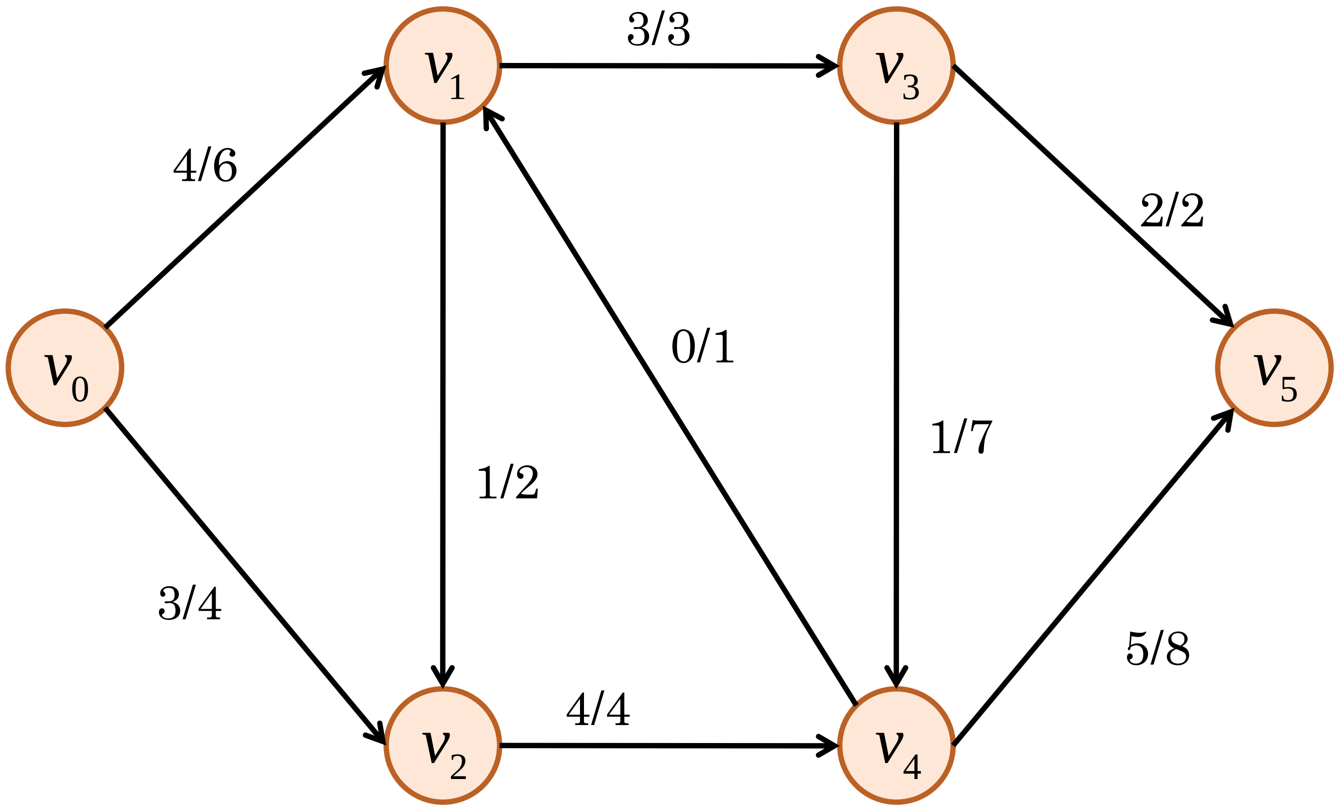
数字:容量



終点への有向道が無い

最大流量の実現

数字:流量/容量



アルゴリズムとしての整理

補助ネットワーク N を作る

幅優先探索で始点から終点への道 P を得る

while($P \neq$ null){

$$d = \min\left(c_\phi(a) \middle| a \in P\right)$$

update(N, P, d) : 補助ネットワーク更新

幅優先探索で始点から終点への道 P を得る

}

deploy(N) : 元のネットワークへ反映

補助ネットワーク更新アルゴリズム

update(N, P, d){

foreach($a \in P$){

$c_\phi(a) \to c_\phi(a) - d$

if($c_\phi(a) = 0$){ a を削除}

b ： $a \in A_\phi^\pm$ ならば対応する弧 $b \in A^\mp{}_\phi$

if(b が存在しない） b を作る

$c_\phi(b) \to c_\phi(b) + d$

}

}

元のネットワークへの反映アルゴリズム

N_A ：補助ネットワーク

N ：元のネットワーク

deploy(N_A){

foreach($a \in A$){ A は N の弧

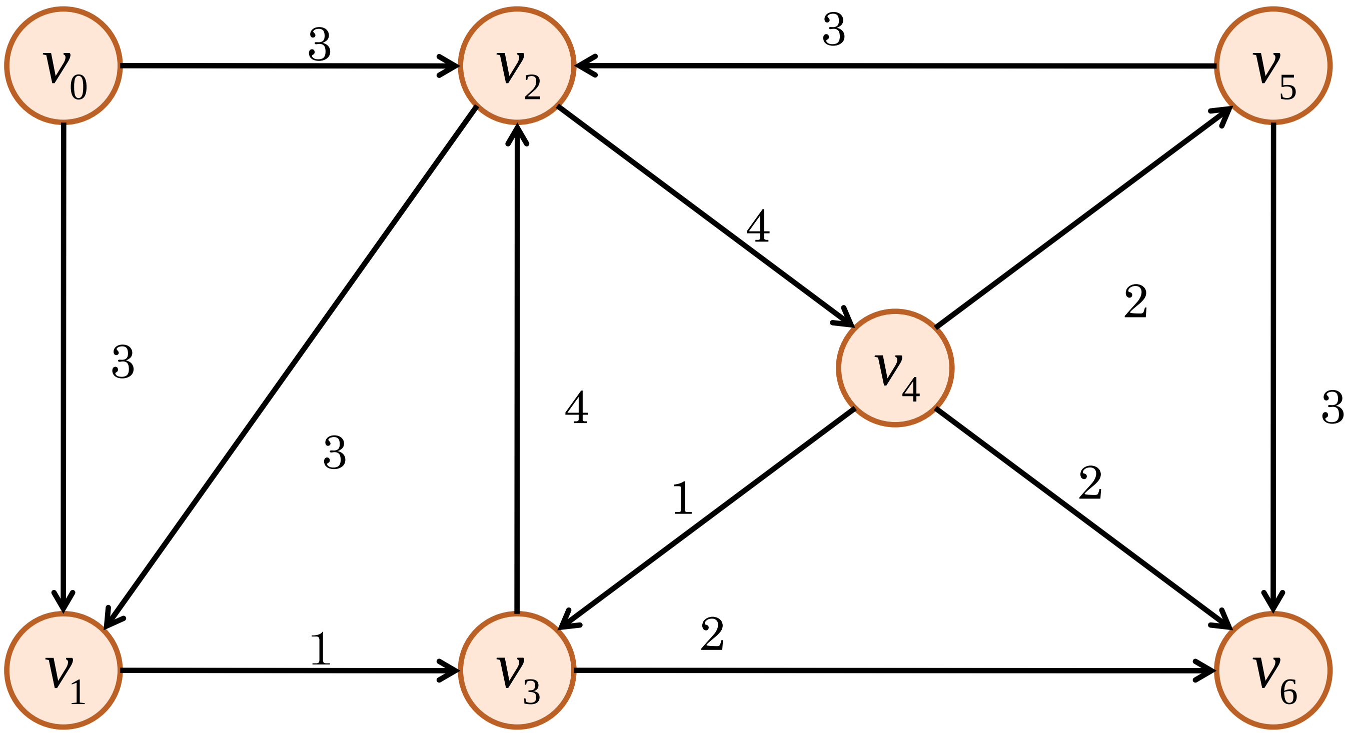
$b \in A_\phi^-$ は対応する弧

$c(a) = c(b)$

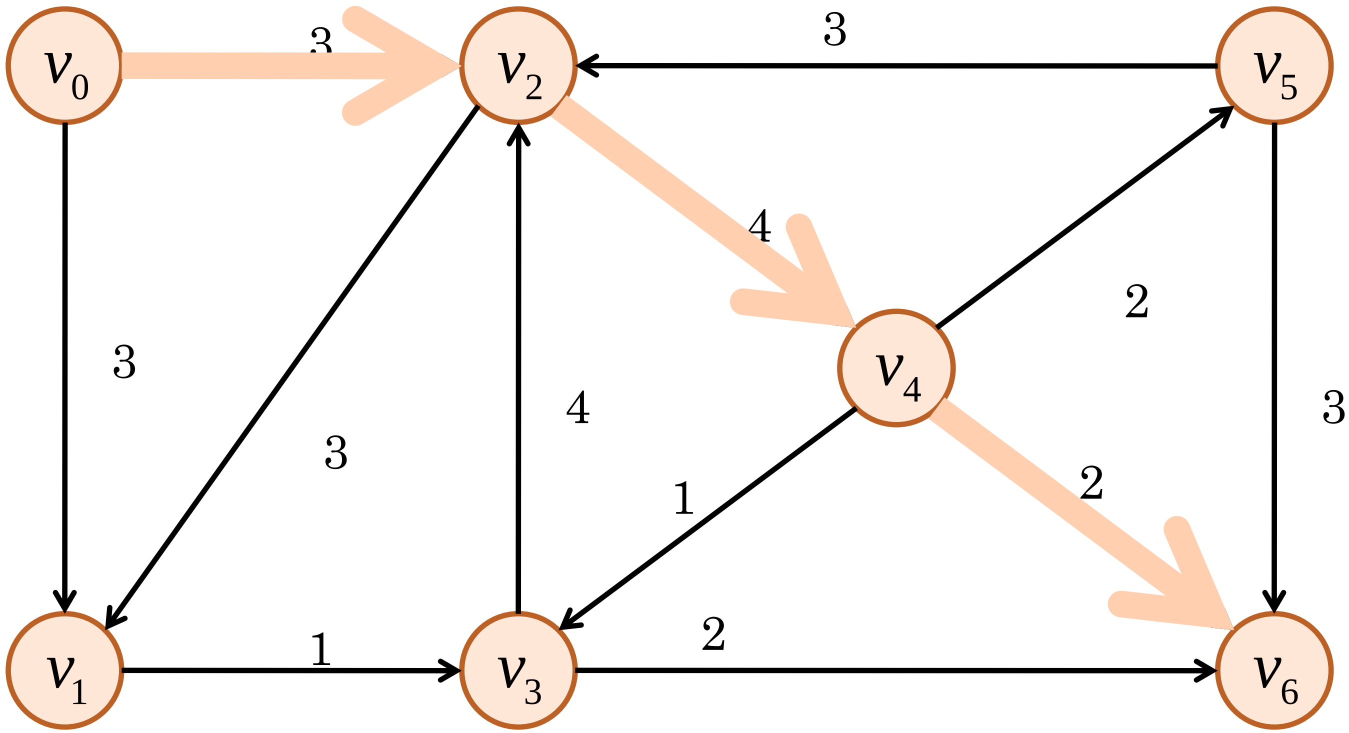
}

}

例2 $v_0 \rightarrow v_6$

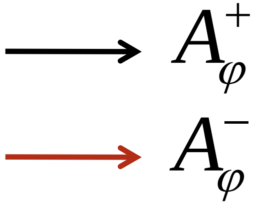


例2：補助ネットワーク構築と増加道探索

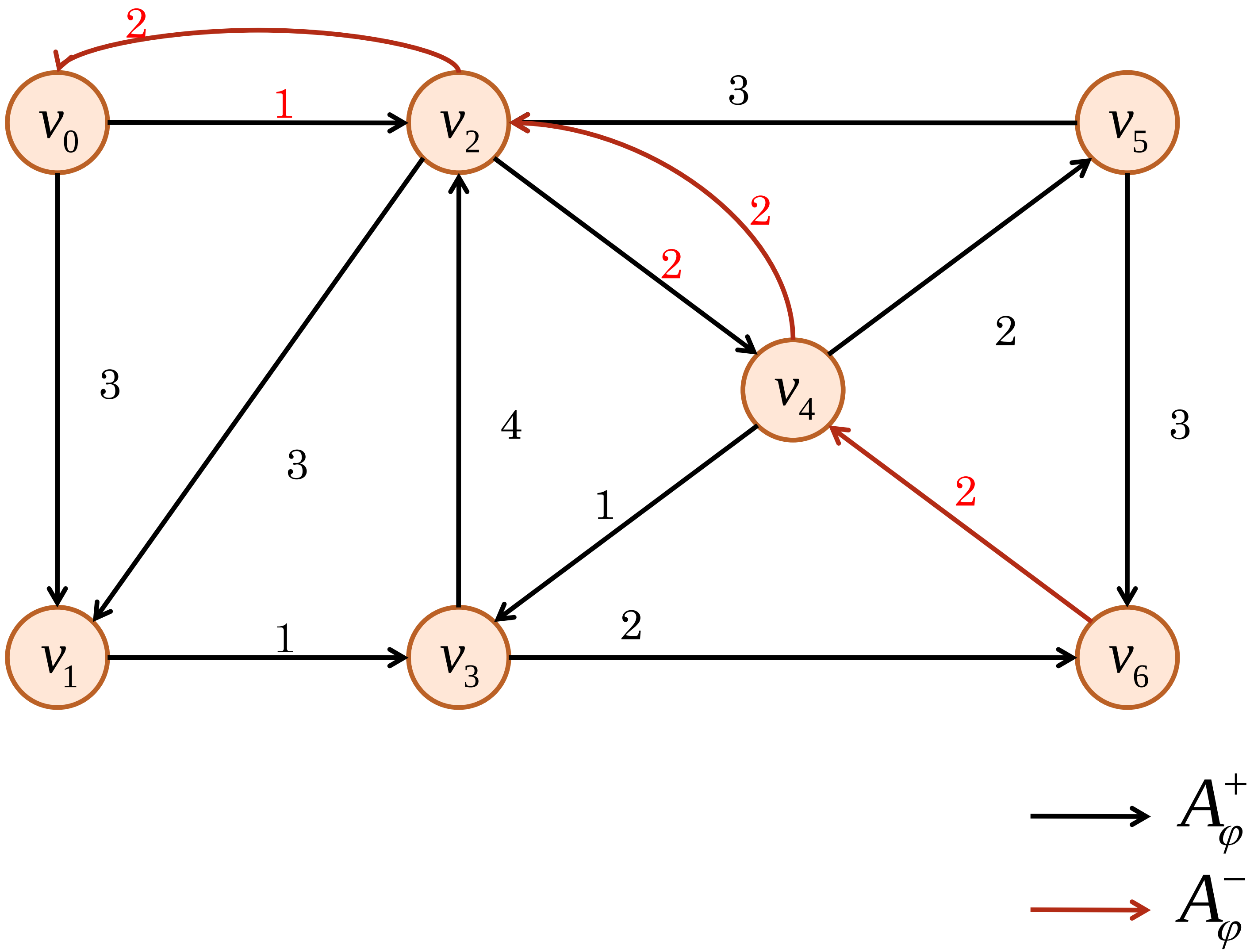


$d = 2$

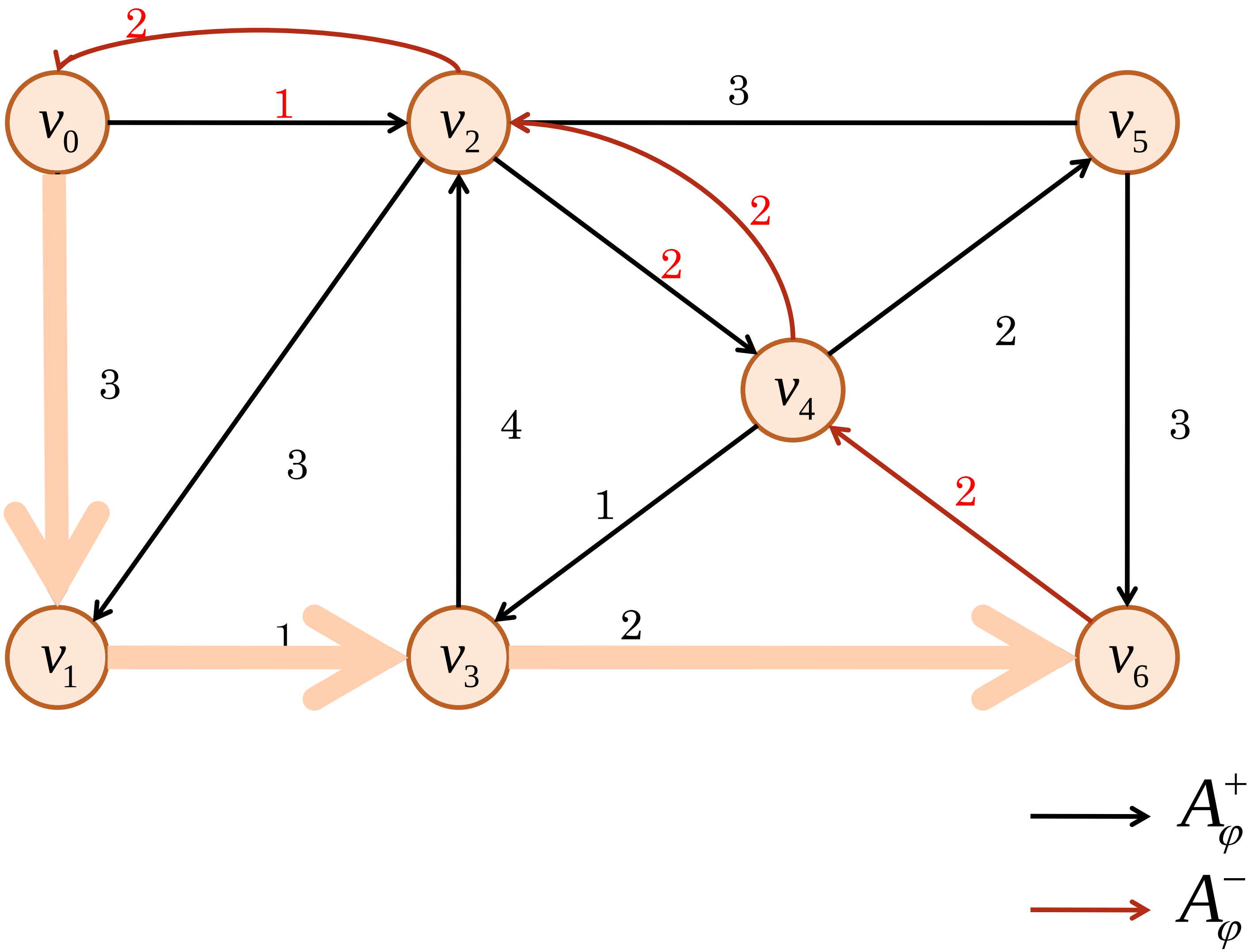
流量0なので、逆向きは存在しない



例2：補助ネットワーク更新

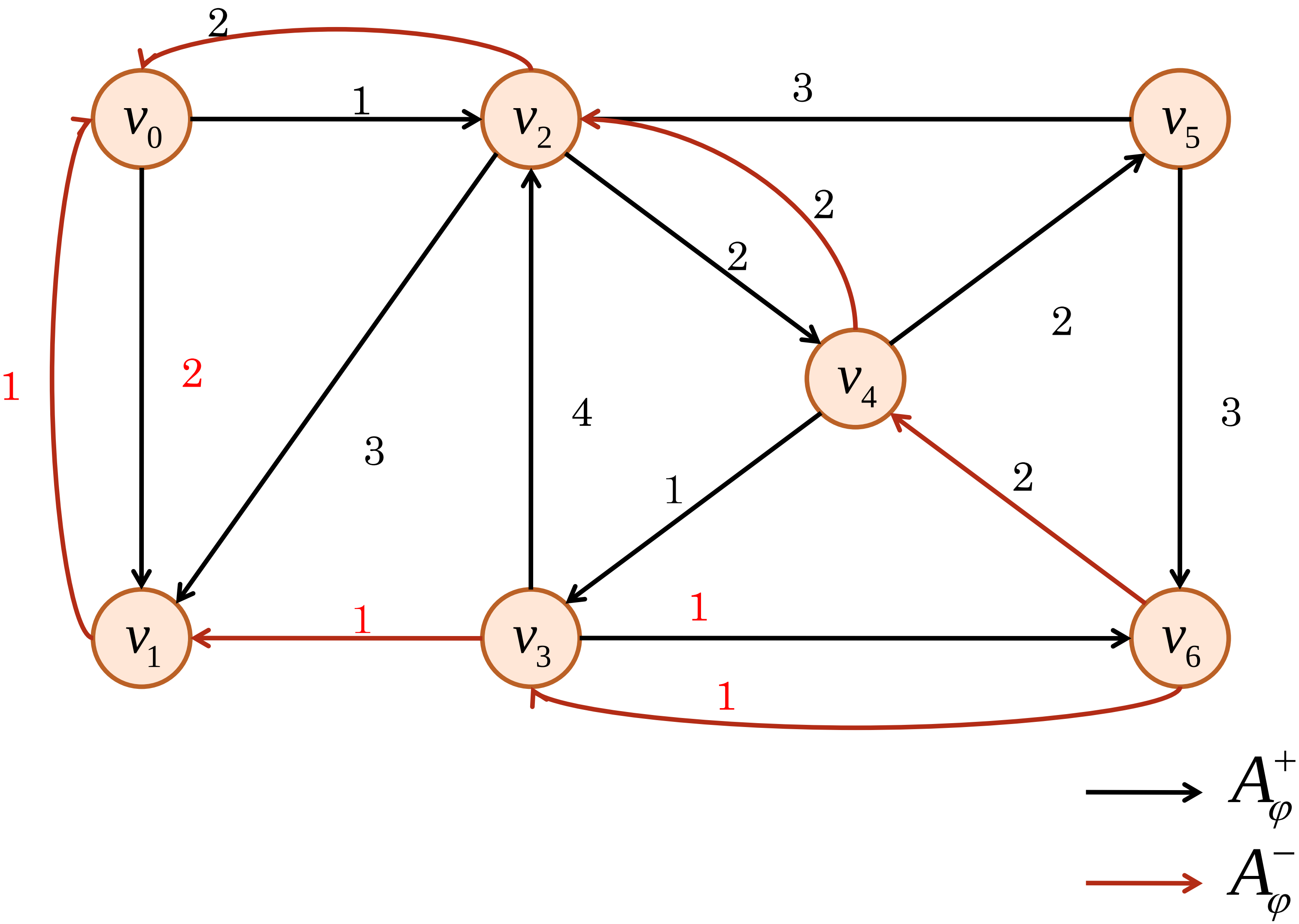


例2：補助ネットワーク中の増加道探索

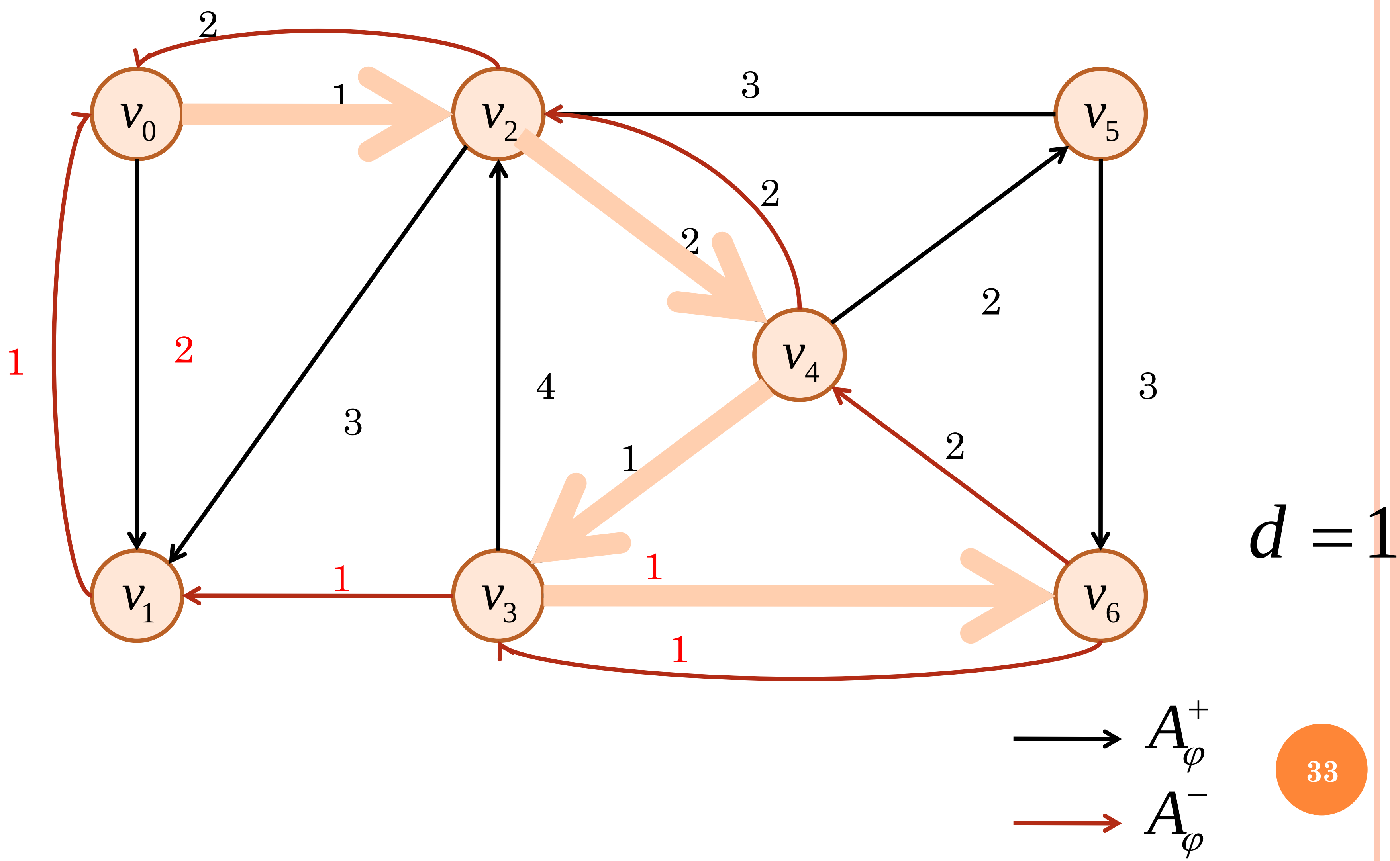


$d = 1$

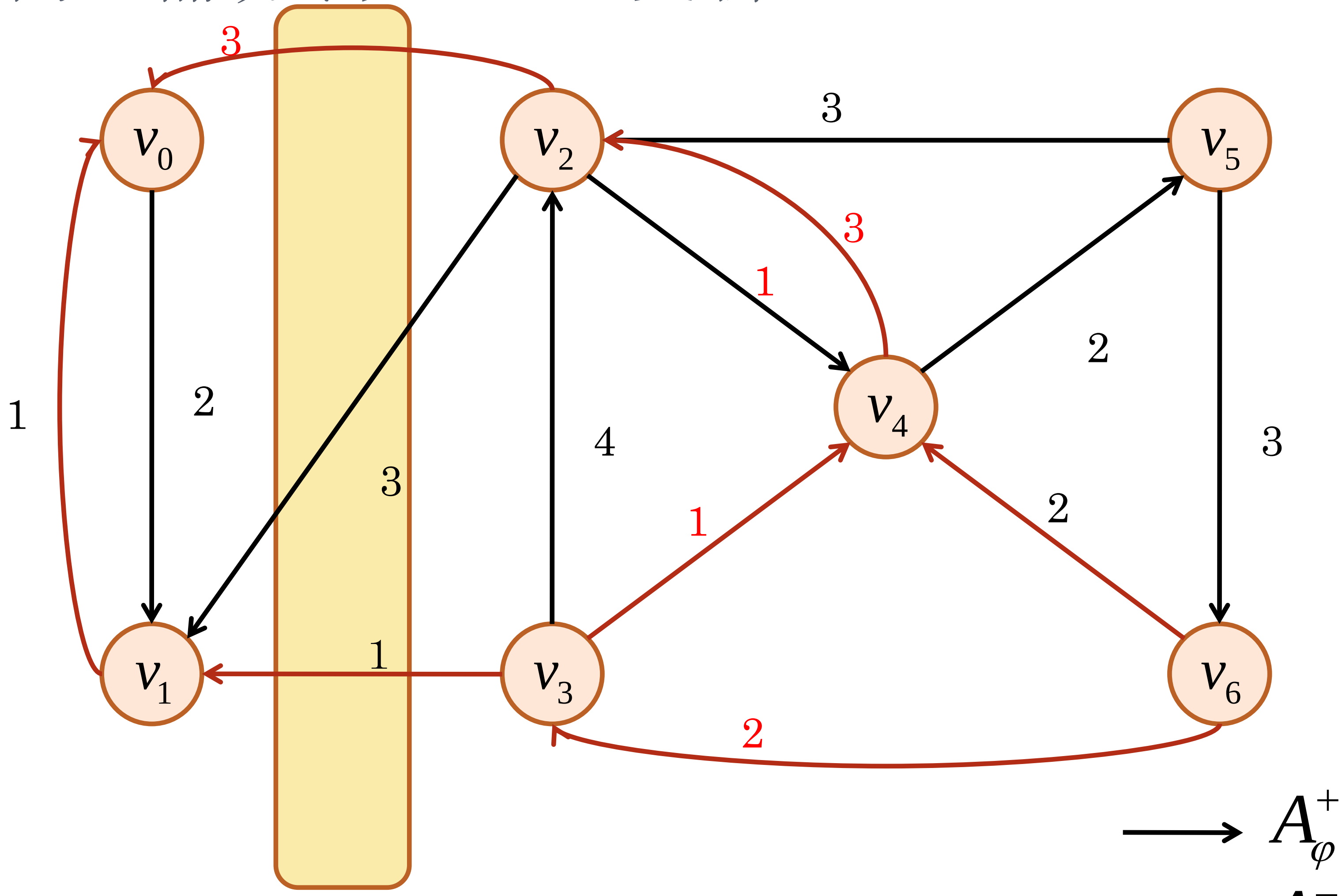
例2：補助ネットワーク更新



例2：補助ネットワーク更新



例2：補助ネットワーク更新



終点への有向道が無い

例2:最大流量の実現

数字:流量/容量