カットとフロー

離散数学・オートマトン
2024年後期
佐賀大学理工学部　只木進一

1. ネットワークとフロー
2. 補助ネットワークの導入
3. 最大フローのアルゴリズム
4. カット: Cut

ネットワークとフロー: Networks and Flows

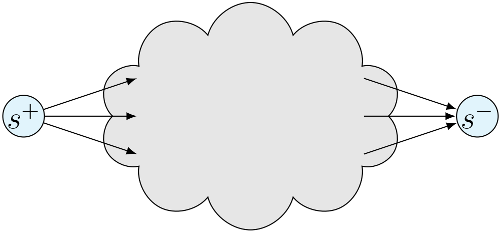
- 交通網中の流れ
  - 都市間を流れる車両の数、及びその上限
  - 都市間を結ぶ航空路線が輸送する人数、及びその上限
- 物流
  - 倉庫間を移動している商品数とその上限
- 作業
  - 各工程における処理数とその上限

容量と流れ: Capacities and Flows

- 各辺に容量 (上限) がある
    - 交通機関の輸送能力
    - 通信速度
- 各辺に実際に流れる流量
    - 容量以下
- ネットワークの二点に最大流量を実現する方法

2端子フロー

- 有向ネットワーク
- 入口 s^+ と出口 s^-
- 入口から出口までの有向道が存在する
- 各辺に容量が定義され、それ以下の流量を割り当てる。



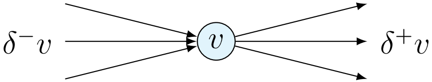
2 端子フローの定義

- グラフ $G = (V, E)$
- 入口 s^+ と出口 s^- の間に有向道がある
- $\forall e \in E$ に、流量上限 (capacity) $c(e) \geq 0$ を定義する
- $\forall e \in E$ に、流量 (flow) $c(e) \geq \phi(e) \geq 0$ を設定する
- 入口 s^+ から出口 s^- への最大流量を求める

流量に対する制約

- $\forall v \in V \setminus \{s^+, s^-\}$
- **容量による制約**: $0 \le \phi(e) \le c(e)$
- **流量保存則**: **頂点** v **で「湧き出し** (source)**」と「吸い込み** (sink)**」がない**

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



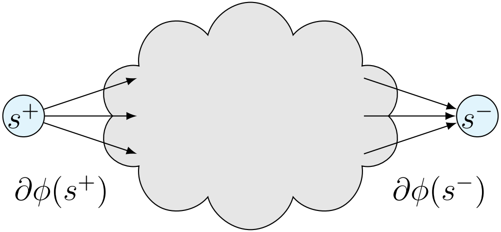
$$\sum_{e \in \delta^- v} \phi(e) \qquad\qquad \sum_{e \in \delta^+ v} \phi(e)$$

ネットワークフローのイメージ

- ネットワークの流量

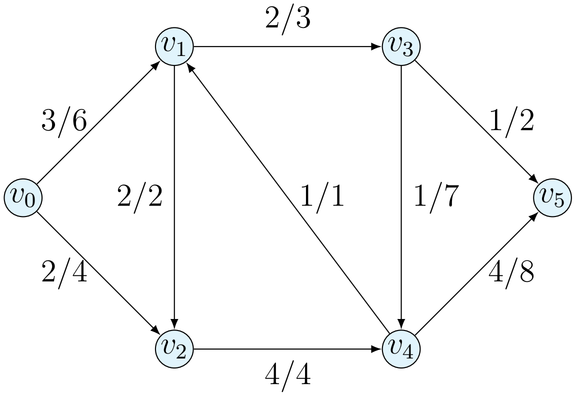
$$Q(\phi) = \partial\phi(s^+) = -\partial\phi(s^-) \tag{1.1}$$

- s^+ から入った流れは全て s^- へ至る

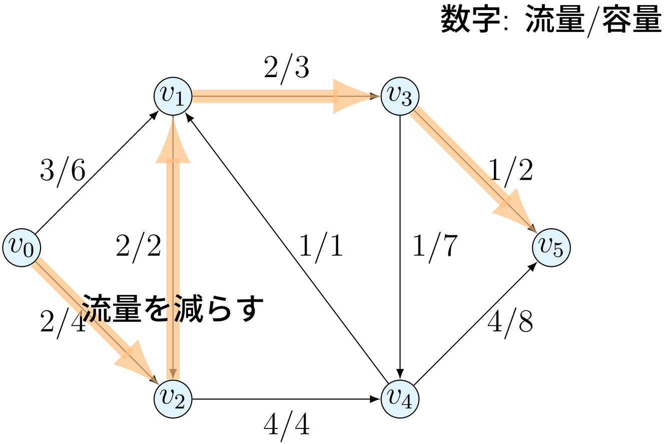


最大フローを見つける考え方

数字: 流量/容量

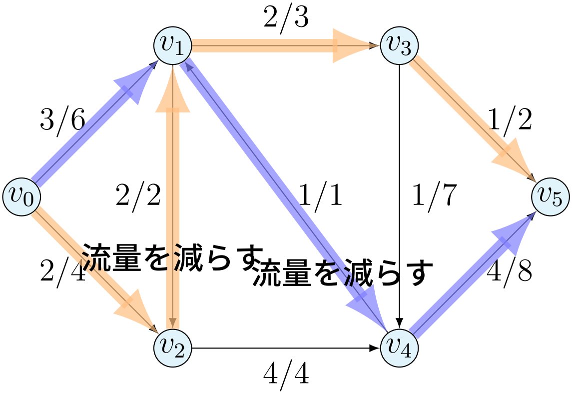


最大フローを見つける考え方: 流量を増やせる道を見つける



最大フローを見つける考え方: 流量を増やせる道を見つける2

数字: 流量/容量



わかりやすいアルゴリズムへ

- わかりにくい点
  - 辺の向きとは逆方向に「流す」
  - 逆向きの辺の流量を減らす
  - 容量と流量の二つの量が出てくる
- 補助ネットワークの導入
  - 各辺に一つの量

補助ネットワーク: Auxiliary Network

$$N_A = \left(G_\phi\left(V, E_\phi\right), s^+, s^-, c_\phi\right) \tag{2.1}$$

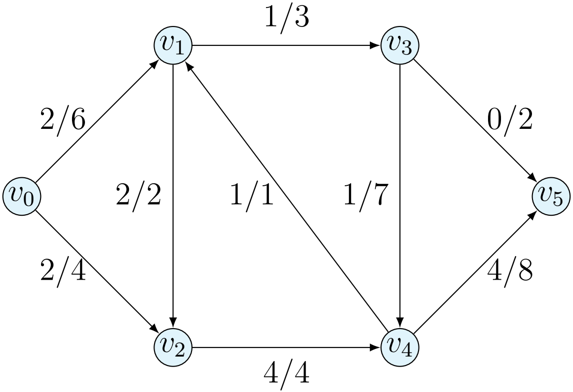
$$E_\phi = E_\phi^+ \cup E_\phi^- \tag{2.2}$$

- E_ϕ^+: 元のネットワークと順方向の辺。容量として、容量の残り余裕を設定
- E_ϕ^-: 元のネットワークと逆方向の辺。容量として、流量を設定

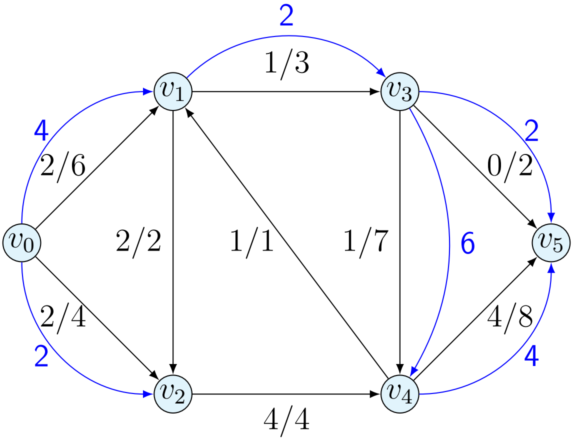
E_ϕ^+ の構成

- 元のネットワーク G の $\forall e \in E$ に対して、
  - 辺の追加: $E_\phi^+ \ \{\forall e \in E\}$
  - 容量の設定: 残りの容量: $c(e) \leftarrow c(e) - \phi(e)$

E_ϕ^+ の構成



E_ϕ^+ の構成

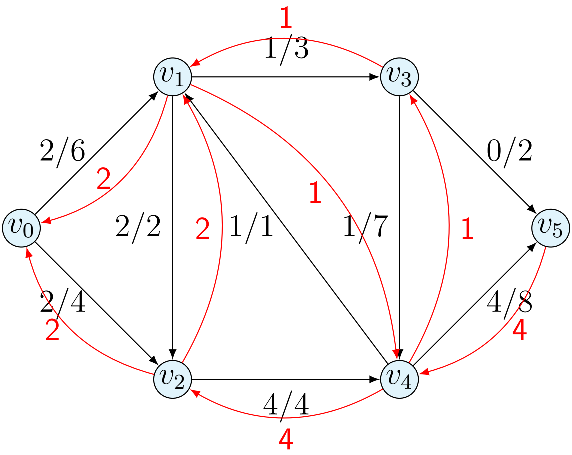


$c(e) = 0$ である $e \in E_\phi^+$ は表示しない

E_ϕ^- の構成

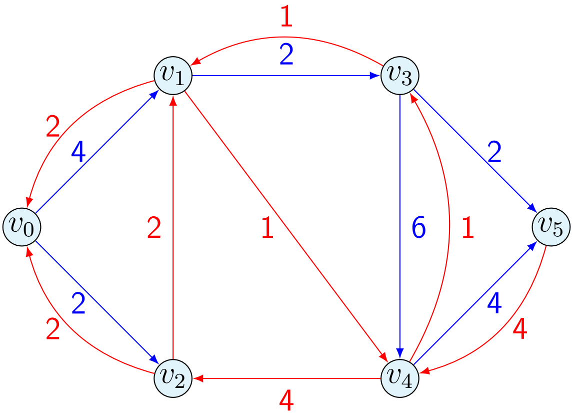
- 元のネットワーク G の $\forall e \in E$ に対して
  - e と逆向きの辺 $e^\dagger$ の追加:$E_\phi^- = \{e^\dagger \mid \forall e \in E\}$
  - 容量の設定: 削減可能流量: $c(e^\dagger) \leftarrow \phi(e)$
  - 元のネットワークと辺の向きが逆である

E_ϕ^- の構成



$c(e) = 0$ である $e \in E_\phi^+$ は表示しない

補助ネットワーク



増加道を見つける

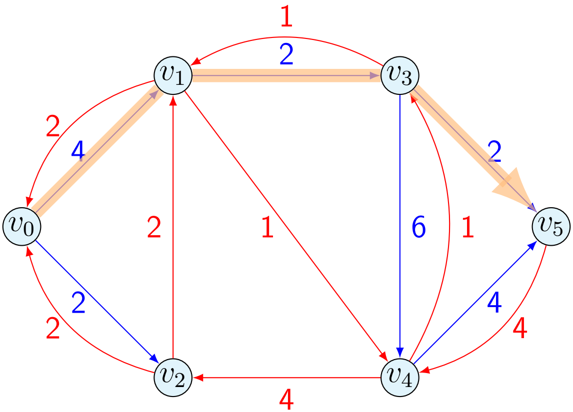
- 補助ネットワーク中に、s^+ から s^- への有向道 P (有向道) があれば、
$$d = \min_{e \in P} c_\phi(e) \tag{2.3}$$
だけ流量を増やすことができる
  - $e \in E_\phi^+$ ならば、容量に余裕がある
  - $e \in E_\phi^-$ ならば、逆方向の辺の流量を減らすことができる

- このときの、ネットワーク中の新しい流量

$$\phi'(e) = \begin{cases} \phi(e) + d & \text{for } e \in E_\phi^+ \wedge e \in P \\ \phi(e) - d & \text{for } e \in E_\phi^- \wedge e \in P \\ \phi(e) & \text{otherwise} \end{cases} \tag{2.4}$$

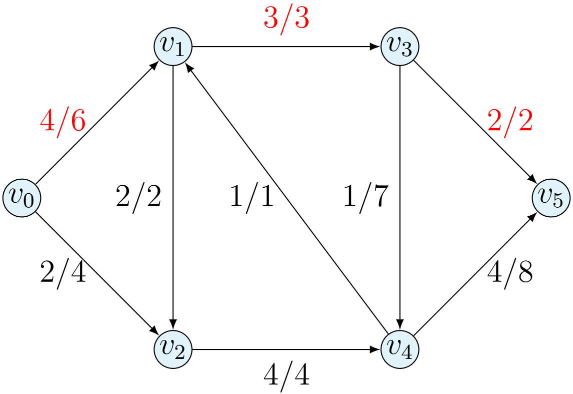
- 容量が整数ならば流量は整数

増加道を見つける

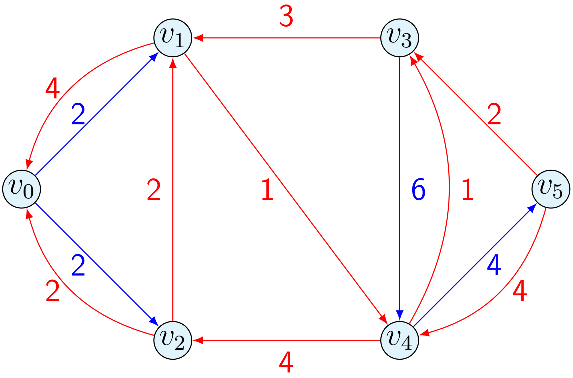


容量 0 の辺は対象外とする

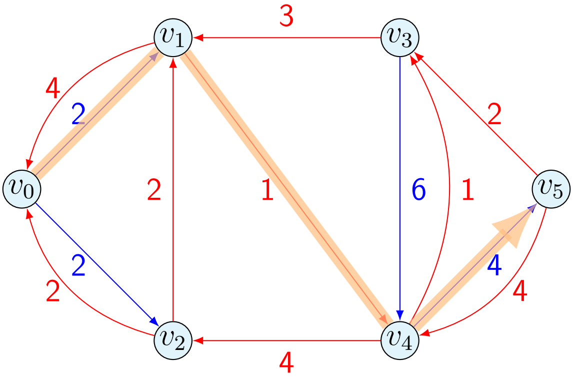
流量増加 $d = 2$



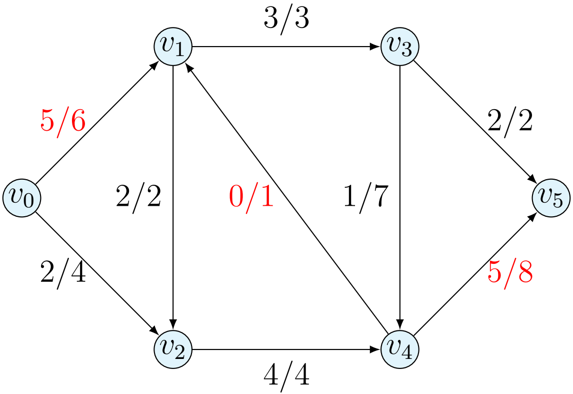
補助ネットワーク構成



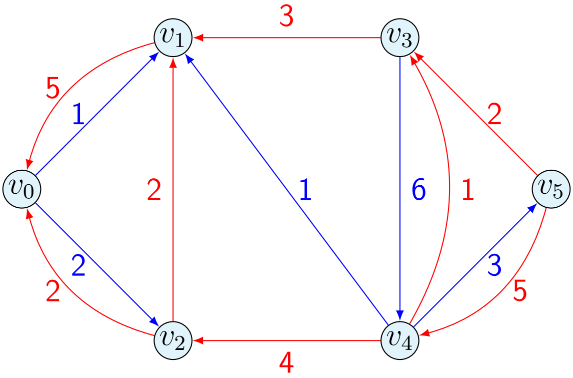
増加道を見つける



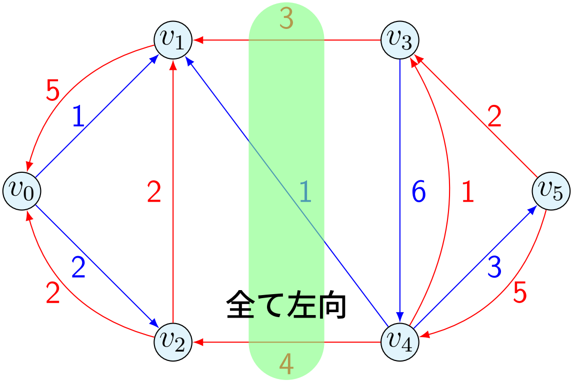
流量増加 $d = 1$



補助ネットワーク構成



補助ネットワークに有向道はない



いちいち元のネットワーク戻す必要があるか

- 補助ネットワークの各辺の容量を更新するアルゴリズム
- $e \in E_phi^{\pm}$ の容量を更新したら、$e \in E_phi^{\mp}$ も更新

アルゴリズムとして整理

Algorithm 1 最大フローのアルゴリズム

補助ネットワーク N_A を構成する
s^+ から s^- への有向道 P を得る
while P が存在 **do**
  $d = \min_{e \in P} c_\phi(e)$
  update(N_A, P, D) ▷ N_A を更新
  s^+ から s^- への有向道 P を得る
end while
deploy(N_A) ▷ 元のネットワークへ反映

Algorithm 2 補助ネットワーク更新

procedure UPDATE(N_A, P, d)
  for $e \in P$ **do**
    $c_\phi(e) = c_\phi(e) - d$
    $e \in E_\phi^{\pm}$ ならば、対応する辺 $e^\dagger \in E_\phi^{\mp}$
    $c_\phi(e^\dagger) = c_\phi(e^\dagger) + d$
  end for
end procedure

Algorithm 3 元のネットワークへの反映

procedure DEPLOY(N_A)
  for $e \in E$ **do**
    $f \in E_\phi^-$ は a に対応する辺
    $\phi(e) = c_\phi(f)$
  end for
end procedure

カット: Cut

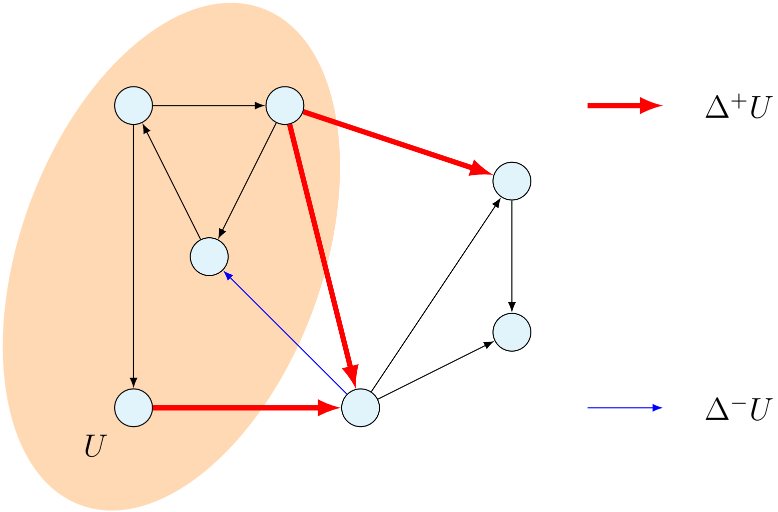
- ネットワークのカット: $U \subset V$
  - s^+ を含み、s^- を含まない頂点集合

$$\left(s^+ \in U\right) \wedge \left(s^- \notin U\right) \tag{4.1}$$

- カットの容量: $\kappa_{\mathrm{C}}(U)$
  - $\Delta^+ U$: U から出て、$U \setminus V$ へ入る辺全体

$$\kappa_{\mathrm{C}}\left(U\right) = \sum_{e \in \Delta^+ U} c(e) \tag{4.2}$$

カットとその境界



流量とカット

- N 中の任意のフロー ϕ と任意のカット U

$$Q(\phi) \leq \kappa_{\mathrm{C}}(U) \tag{4.3}$$

- 直感的には: s^+ と s^- の途中にあるボトルネック部分で流量上限が定まる

$$\begin{aligned} Q(\phi) &= \sum_{e \in \Delta^+ U} \phi(e) - \sum_{e \in \Delta^- U} \phi(e) \\ &\leq \sum_{e \in \Delta^+ U} c(e) - 0 = \kappa_{\mathrm{C}}(U) \end{aligned} \tag{4.4}$$

最大流量と最小カット

$$\max Q(\phi) \leq \min \kappa_{\mathrm{C}}(U) \tag{4.5}$$

- 実際には等号がなりたつ
- つまり、ボトルネック容量で最大流量が定まる

最大流量と最小カット: 等号が成り立つこと

- ネットワーク N の最大流量 ϕ が実現しているならば
  - 補助ネットワーク N_A には、s^+ から s^- への有向道は存在しない
  - 注意: 容量 0 の辺は存在しないものとする
- 補助ネットワーク中の s^+ から到達可能な頂点集合: $W \subset V$
  - N_A には、W から外向きの辺は存在しない。

W への内向きの辺 e は、以下のいずれかである

- $e^\dagger \in E_\phi^-$: N の対応する辺 e の容量を使い切っている

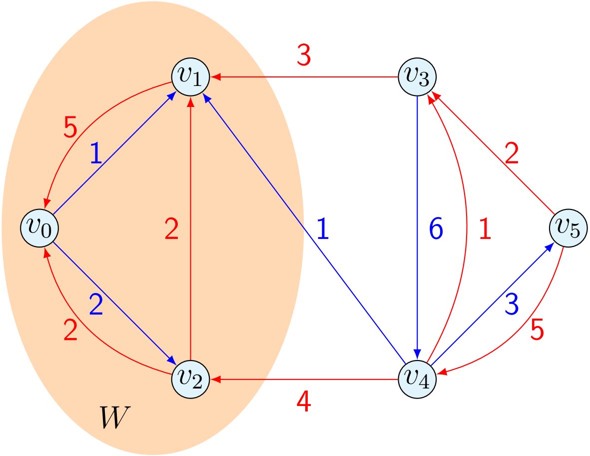
$$\phi(e) = c(e) \tag{4.6}$$

- $e \in E_\phi^+$: N の辺の流量は 0

$$\phi(e) = 0 \tag{4.7}$$

$$\begin{aligned} Q(\phi) &= \sum_{e \in \Delta^+ W} \phi(e) - \sum_{e \in \Delta^- W} \phi(e) \\ &= \sum_{e \in \Delta^+ W} c(e) - 0 = \kappa_{\mathrm{C}}(W) \end{aligned} \tag{4.8}$$

補助ネットワークにおけるカット



元のネットワークにおけるカット