# 最大流最小カット定理

BV21013 堀毛晴輝

2023 年 11 月 5 日

## 目次

# 1 はじめに

競技プログラミングにおいて，グラフ理論の知識を問われることがある．
今回は，グラフ理論の中の最大流最小カット定理を証明し，競技プログラミングの問題に応用する．

# 2 最大流最小カット定理

まず，有向グラフとネットワーク，フローを定義する．

**Def 2.1 (有向グラフ)**

空でない有限集合 $V$ と $V \times V$ の部分集合 $E$ の対 $G = (V, E)$ を有向グラフという．
また，各 $v \in V$ を頂点，各 $(u, v) \in E$ を辺という．

**Def 2.2 (ネットワーク)**

次のような有向グラフ $D = (V, E, s, t, c)$ をネットワークという．

- 始点 (ソース) $s \in V$ が 1 つ存在する．
- 終点 (シンク) $t \in V$ が 1 つ存在する．
- 各辺 $e \in E$ に対し，容量 $c(e) \geq 0$ が定まっている．

**Def 2.3 (フロー)**

次の制約条件下でネットワーク $D = (V, E, s, t, c)$ の各辺 $e \in E$ に対し，関数 $f : E \to \mathbb{R}$ を $s - t$ フロー (または単にフロー) という．

- 容量制限: 任意の $e \in E$ に対し， $0 \leq f(e) \leq c(e)$ を満たす．
- フロー保存則: $s, t$ を除く任意の $v \in V$ に対し， $\displaystyle\sum_{v\text{ に入る辺 }e} f(e) = \sum_{v\text{ から出る辺 }e} f(e)$ を満たす．

また，ネットワーク $D$ のフロー $f$ に対して， $s$ から出る ( $t$ に入る) 合計流量をフロー $f$ の値と呼び，$\mathrm{val}(f)$ と表す．

ネットワーク $D$ に対し， $\mathrm{val}(f)$ を最大化するようなフロー $f$ を最大流といい，これを求める問題を最大流問題という．
たとえば， これはフローの一例である: (各辺について， $f/c$ でフローを表す)

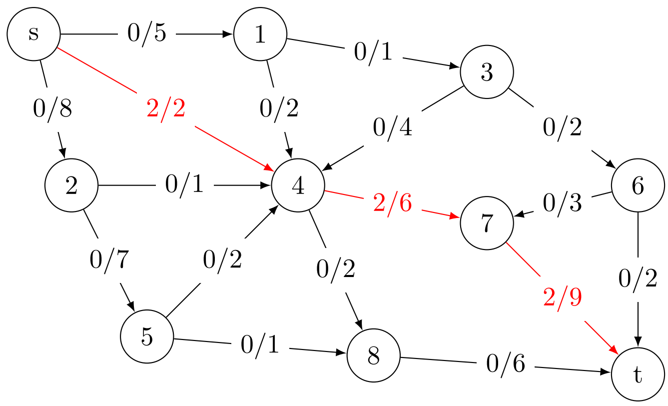


図 1　フローの一例

つぎに，ネットワーク $D$ に対するカットを定義する．

**Def 2.4 (カット)**

ネットワーク $D=(V,E)$ について，ある頂点集合 $S\subset V$ に対し $S$ から出ていく辺の集合をカットといい，$(S,S^c)$ で表す．また，その容量の和をカットの容量といい，$\mathrm{cap}((S,S^c))$ で表す．

とくに，$s\in S, t\in S^c$ となるようなカットを $s-t$ カットという．

カットの容量が最小となるようなカット $(S,S^c)$ をネットワーク $D$ の最小カットという．

最小カットの容量は，言い換えると「ネットワーク $D$ に対して，$s$ から $t$ へのパスが存在しなくなるために除去しなければならない辺の容量の和の最小値」である．

たとえば，これは $s-t$ カットの一例である:（ $S=\{s,1,2,4\}$ ）

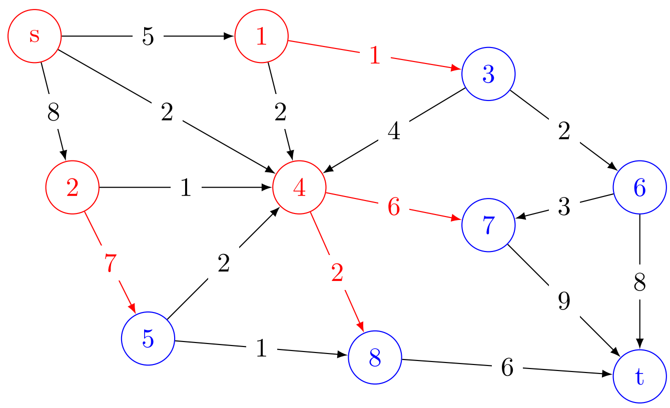


図 2　カットの一例

この場合，カットの容量は $\mathrm{cap}((S,S^c))=1+6+2+7=16$ である．

最大流と最小カットは，次の定理によって関係付けられる．

**Thm 2.5 (最大流最小カット定理)**

任意のネットワーク $D$ において，最大流 $f$ の値 $\mathrm{val}(f)$ と最小カット $(S,S^c)$ の容量 $\mathrm{cap}((S,S^c))$ は等しい．

まず，次の Prop 2.6 を示す．

**Prop 2.6**

ネットワーク $D$ のフロー $f$ とカット $(S, S^c)$ が $\mathrm{val}(f) = \mathrm{cap}((S, S^c))$ を満たすならば，$f$ は $D$ の最大流であり，$(S, S^c)$ は $D$ の最小カットである．

**Proof：** ネットワーク $D$ の任意の $s-t$ フロー $f$ と任意の $s-t$ カット $(S, S^c)$ について，

- $\mathrm{val}(f) =$ ($s$ から出る辺の流量総和)
- 任意の $v \in S \backslash \{s\}$ について，($v$ から出る辺の流量総和) $=$ ($v$ に入る辺の流量総和)

$S$ の収支に着目すると，

$$\mathrm{val}(f) + (S\text{ に入る辺の流量総和}) = (S\text{ から出る辺の流量総和})$$

移項して，

$$\mathrm{val}(f) = (S\text{ から出る辺の流量総和}) - (S\text{ に入る辺の流量総和})$$

($S$ から出る辺の流量総和) $\leq$ ($S$ から出る辺の容量総和) $= \mathrm{cap}((S, S^c))$ であり，($S$ に入る辺の流量総和) $\geq 0$ であるから，$\mathrm{val}(f) \leq \mathrm{cap}((S, S^c))$ が成り立つ．

これより，$\mathrm{val}(f) = \mathrm{cap}((S, S^c))$ を満たすフロー $f$ とカット $(S, S^c)$ が存在すれば，フローの値はそれ以上大きくできないため，$f$ は最大流であり，$(S, S^c)$ は最小カットであることがわかる．よって Prop 2.6 が示された． ■

Prop 2.6 より，任意のネットワーク $D$ について，$\mathrm{val}(f) = \mathrm{cap}((S, S^c))$ を満たすようなフロー $f$ と最小カット $(S, S^c)$ が存在すれば，最大流最小カット定理が示せる．

$\mathrm{val}(f) = \mathrm{cap}((S, S^c))$ を満たすフロー $f$ と最小カット $(S, S^c)$ は，Ford - Fulkerson 法によって求めることができる．

Ford - Fulkerson 法を用いるために，残余容量と残余ネットワークを定義する．

**Def 2.7 (残余容量)**

ネットワーク $D = (V, E)$ とそのフロー $f$ について，次で定義される容量を残余容量という．

$$c_f((u, v)) \coloneqq \begin{cases} c((u,v)) - f((u,v)) & (u,v) \in E \\ f((v,u)) & (v,u) \in E \\ 0 & \text{otherwise} \end{cases}$$

**Def 2.8 (残余ネットワーク)**

ネットワーク $D = (V, E)$ とそのフロー $f$ について，残余容量 $c_f$ によって定義されるネットワーク $D_f = (V, E_f)$ を残余ネットワークという．( $E_f \coloneqq c_f$が 0 でないような辺の集合 )

たとえば，図 3 のようなネットワーク $D$ とフロー $f$ に対して，残余ネットワーク $D_f$ は 図 4 のようになる．

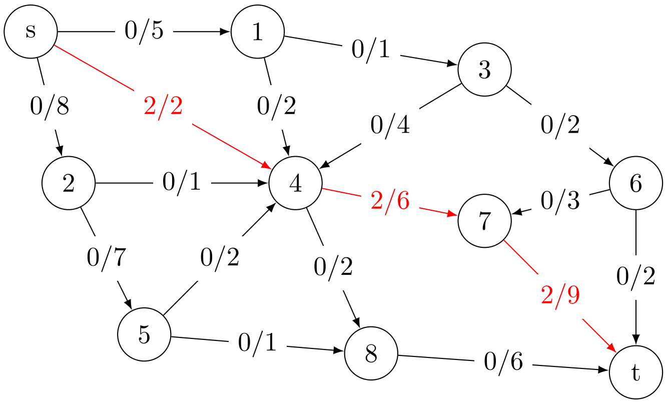


図 3　ネットワーク $D$ とフロー $f$

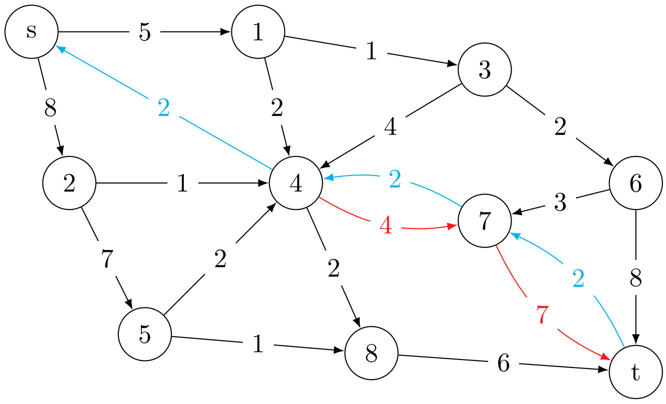


図 4　残余ネットワーク $D_f$

残余ネットワークについて，赤い辺は「あとどれくらいフローを流せるか」，青い辺は「どれくらい戻すことができるか」を表している．

さらに，増加道を定義する．

**Def 2.9 (増加道)**

残余ネットワークで，始点 $s$ から終点 $t$ へのパスで，パス上の任意の辺 $e$ について $0 < c_f(e)$ が成り立つようなパスを増加道という．

残余ネットワークに増加道が存在するとき，その増加道上の辺のうち残余容量が最小の辺の残余容量分を増加道に流すことができる．

Ford - Fulkerson 法とは，次のようなアルゴリズムである．

Ford - Fulkerson 法

フロー $f$ をはじめ 0 として，残余ネットワーク $D_f$ に対して増加道が存在する限り，増加道に沿ってフローを増加させる．

**Prop 2.10**

Ford - Fulkerson 法によって得られるフロー $f$ は最大流である．

**Proof:** Prop 2.6 より，Ford - Fulkerson 法によって得られるフロー $f$ について，$\mathrm{val}(f) = \mathrm{cap}((S, S^c))$ を満たすような最小カット $(S, S^c)$ が存在することを示せばよい．

Ford - Fulkerson 法によって得られたフロー $f$ は，残余ネットワーク $D_f$ について $s-t$ パスが存在しない．

ネットワーク $D$ について，Ford - Fulkerson 法によって得られたフローを $f$ とし，$f$ に対する残余ネットワーク $D_f$ について，$s-v$ パスが存在するような頂点 $v \in V$ の集合を $S$ とする (すなわち，$S$ は $s$ から到達可能な頂点の集合)．

$D_f$ には $s-t$ パスが存在しないので，$(S, S^c)$ は $s-t$ カットである．

辺 $(u, v) \in (S, S^c)$ について:

- $S$ の定義より $c_f((u,v)) = c((u,v)) - f((u,v)) = 0$ であるから，$f((u,v)) = c((u,v))$ ．

辺 $(u, v) \in (S^c, S)$ について:

- $S$ の定義より $c_f((v,u)) = f((u,v)) = 0$ ．

これより，

$$\begin{aligned} \mathrm{val}(f) &= (S \text{ から出る辺の流量総和}) - (S \text{ に入る辺の流量総和}) \\ &= \sum_{e \in (S, S^c)} f(e) - \sum_{e \in (S^c, S)} f(e) \\ &= \sum_{e \in (S, S^c)} c(e) - \sum_{e \in (S^c, S)} 0 \\ &= \mathrm{cap}((S, S^c)) \end{aligned}$$

よって，$S$ を $s$ から到達可能な頂点の集合とすると，$\mathrm{val}(f) = \mathrm{cap}((S, S^c))$ を満たす．

したがって，Ford - Fulkerson 法によって得られたフロー $f$ は最大流であり，$\mathrm{val}(f) = \mathrm{cap}((S, S^c))$ を満たすようなカット $(S, S^c)$ が存在する． ■

これより，Thm 2.5 は次のように示される．

**Proof of Thm 2.5:** 任意のネットワーク $D$ に対して，Ford - Fulkerson 法を用いることで，$\mathrm{val}(f) = \mathrm{cap}((S, S^c))$ を満たすような最大流 $f$ と最小カット $(S, S^c)$ を得ることができる．

これより，任意のネットワーク $D$ において，最大流 $f$ の値 $\mathrm{val}(f)$ と最小カット $(S, S^c)$ の容量 $\mathrm{cap}((S, S^c))$ は等しい． ■

辺の容量を実数としていたが，ここでは整数である，と仮定する．

**Thm 2.11 (整数性定理)**

すべての辺に流れるフローが整数の最大フローが存在し，その値は整数となる．

**Proof:** Ford - Fulkerson 法で増加道を用いてフローを増やしたとき，増える値は必ず整数であることを用いればよい．詳しい証明は略． ■

残余ネットワークについて，$s$ から $t$ への増加道は DFS を用いて $O(|E|)$ で見つけることができる．また，増加道に沿ってフローを増やしたとき，増加量は必ず 1 以上であるから，Ford - Fulkerson 法は $O(|E| \, \mathrm{val}(f))$ で実行できる．

さらに，増加道を BFS で探すと $O(|V||E|^2)$ となり，これは Edmonds - Karp 法と呼ばれる．

# 3 Dinic 法

Dinic 法を導入するために，DAG を定義する．

**Def 3.1** (***DAG*(有向非巡回グラフ)**)

閉路のない有向グラフのことを DAG という．ある頂点 $v$ から出発し，頂点 $v$ に戻ってくるようなパスが存在しないグラフである．

Dinic 法とは，次のようなアルゴリズムである．

Dinic 法

暫定解であるフロー $f$ を持ち，次の 2 つを増加道が存在する限り繰り返す．

1 $G_f$ 上で $s$ から $t$ への最短経路 DAG $G_f^\star$ を求める．
2 $G_f^\star$ に増加道が存在する限り，次を繰り返す:
- $G_f^\star$ 上で増加道を見つけ，そのパスに流せるだけ流す．
- 残余容量が 0 になった辺， $s$ からたどり着けない頂点， $t$ へたどり着けない頂点，それらに隣接する辺を $G_f^\star$ から取り除く．

その後，流したフローを $f$ に反映する．

Dinic 法によって得られるフロー $f$ の残余ネットワーク $G_f$ には，$s$ から $t$ への増加道が存在しないことから，Ford - Fulkerson 法と同様に Dinic 法によって得られるフローは最大流である．

**Prop 3.2**

Dinic 法の計算量は， $O(|V|^2|E|)$ である．

Prop 3.2 を示すために，次の Lem 3.3 を示す．

**Lem 3.3**

Dinic 法の step 1, 2 の繰り返しは，高々 $|V|-1$ 回しか行われない．

**Proof:** あるタイミングについて，step 1 での $s$ から $v$ への最短経路 (存在しないなら $\infty$ ) を $\mathrm{label}(v)$ とする．

このとき， $G_f$ 上 $s-t$ 最短経路 DAG $G_f^\star$ の任意の辺 $(u,v)$ について $\mathrm{label}(u)+1=\mathrm{label}(v)$ が成り立つ．

step 2 で， $G_f^\star$ 上のフローを $f$ に反映させるとき，
- $G_f$ に追加されうる辺 $(u,v)$ は， $G_f^\star$ の逆辺なので，$\mathrm{label}(u)=\mathrm{label}(v)+1$ となる辺のみである．
- $G_f^\star$ 上の任意の $s-t$ パスは，その少なくとも 1 辺が $G_f$ から削除される．

の 2 つが成り立つ．

step 2 終了後の $G_f$ の任意の $s-t$ パスは， $\mathrm{label}(t)$ より長いことを示す．

更新後の $G_f$ の $s-t$ パスについて，このパス上での label からなる列は，初項 0 , 末項 $\mathrm{label}(t)$ で，高々 1 ずつしか増加しない．これが長さ $\mathrm{label}(t)$ 以下になるためには，このパス上のすべての辺 $(u,v)$ で $\mathrm{label}(u)+1=\mathrm{label}(v)$ を満たして，長さ $\mathrm{label}(t)$ になるしかない．

これより， step 1, 2 を行うことで， $G_f$ の $s-t$ 最短経路は，真に長くなることがわかる．$s-t$ パスの長さは高々 $|V|-1$ なので，step 1, 2 の繰り返しは高々 $|V|-1$ 回である． ■

**Proof of Prop 3.2 :** step 1 は，BFS をすることで $O(|E|)$ .

$G_f^\star$ 上で増加道を見つけるには， $s$ から出る辺を辿っていけばよいので， $O(|V|)$ ．更新したときに残余容量が 0 になる辺が必ず 1 つ以上存在するので，更新は $O(|E|)$ ．これより，step 2 は $O(|V||E|)$ で完了する．

従って，step 1, 2 全体の計算量は $O(|V||E|)$ であり，これらは高々 $|V|-1$ 回しか行われないため，Dinic 法の計算量は $O(|V|^2|E|)$ である． ■

Ford - Fulkerson 法の計算量は最大流の値に依存していたが，依存しない形で計算量を抑えることができた．

## 4　例題

**ABC010 D 浮気予防**

高橋くんは SNS で友人関係にある人を辿り，見つけた女の子にメッセージを送ります．なぎさちゃんは，高橋くんのメッセージを女の子が見ることのないように，SNS に工作をします．行える工作は以下の 2 つです:

- 特定の 2 人の友人関係を解消する．
- 高橋くん以外の特定の 1 人のパスワードを変え，ログインできなくする．

なぎさちゃんは，できるだけ工作の回数を少なくし，全員の女の子が高橋くんのメッセージを閲覧できないようにします．なぎさちゃんが工作をする必要のある回数は何回ですか？

頂点を人，辺を友人関係とし，0 を高橋くん，赤い頂点を女の子とする．さらに新たな頂点 $G$ を追加し女の子から $G$ への辺を張ったネットワークを考える (各辺の容量は 1 とする).

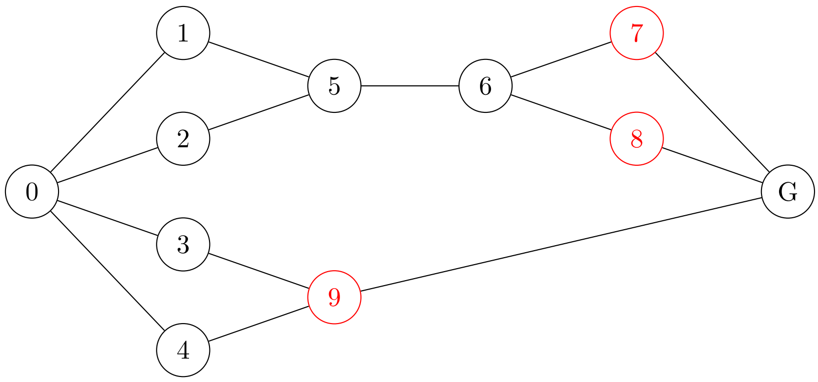


図 5　頂点を人，辺を人間関係としたネットワーク

女の子 $i$ が高橋くんのメッセージを見ることができないとき，少なくとも

- 高橋くんと女の子 $i$ が連結でない．
- 女の子 $i$ のパスワードが変えられていて，ログインできなくなっている．

のどちらか一方は成り立つ．先ほど考えたグラフについて，

- 「特定の 2 人の友人関係を解消する」:
  $\rightarrow$ 友人関係の辺を削除する．
- 「1 人のパスワードを変え，ログインできなくする」:
  $\rightarrow$ 女の子 $i$ から $G$ への辺を削除する．

という操作の置き換えをすると，最小 $0 - G$ カットを求める問題に帰着できる．

最大流最小カット定理より，このネットワークの 0 から $G$ への最大流の値が求める答えである．

# 5　おわりに

最大流最小カット定理を証明し，Ford - Fulkerson 法と Dinic 法を紹介した．

この考え方をさらに発展させることで，競技プログラミングで「燃やす埋める」と呼ばれる問題を解くことができる．

今後の発展として，燃やす埋める問題について理解し，解くことを目標としたい．

# 参考文献


[1] 早稲田大学 早水桃子研究室 “離散数学入門#8: 最大流問題 (1)：フローネットワークの基礎知識”
`https://www.youtube.com/watch?v=QTNCb-d4_80&list=PLCo60G1m_ibpJgfB4WcGwWybC6sfyawoL&index=10`

[2] 早稲田大学 早水桃子研究室 “離散数学入門#9: 最大流問題 (2)：増加道アルゴリズムと最大流最小カット定理”
`https://www.youtube.com/watch?v=TjOA3vKOHCI&list=PLCo60G1m_ibpJgfB4WcGwWybC6sfyawoL&index=11`

[3] Dinic 法とそのとき間計算量
`https://misawa.github.io/others/flow/dinic_time_complexity.html`