

第13章 - 連結度と関連問題

数理情報系3年

甲本健太

本では 節点・枝 という用語を使用していますが，このスライドでは

- 節点 → 頂点
- 枝 → 辺

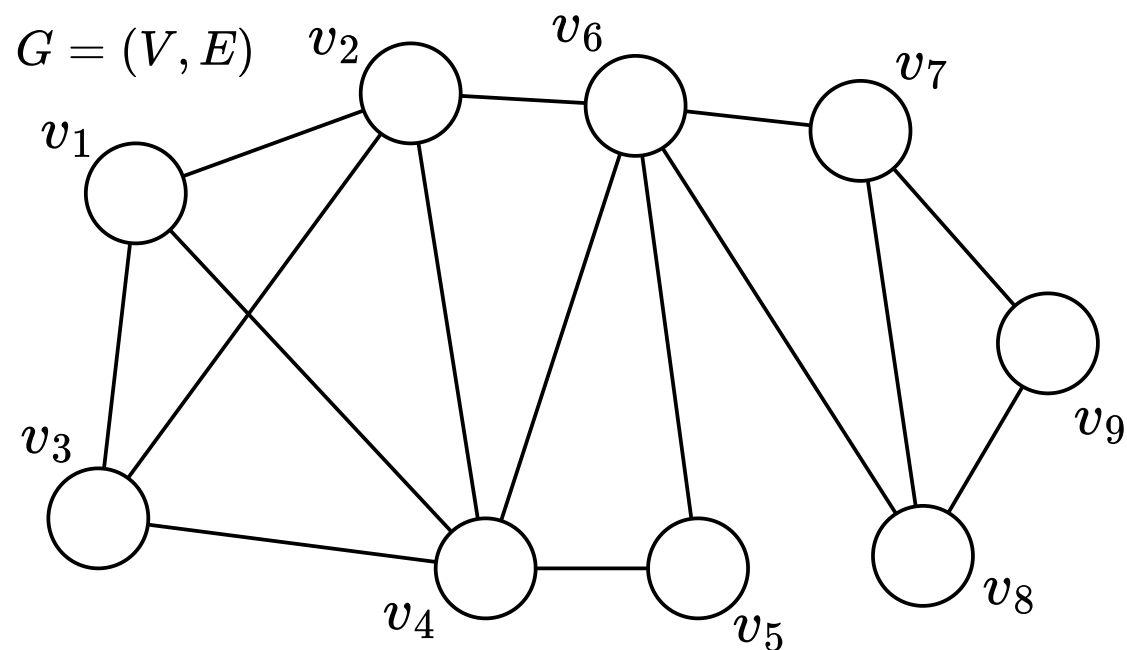
に読み替えています．

2. 連結度とはなにか

下の図で表される通信ネットワークを考える.

グラフの2頂点間に辺が存在するとき, その2頂点間は通信が可能である.

このグラフでは, 頂点 v_6 が故障したとき, v_1 と v_9 間は通信不可能になってしまう.



→ 連結度は, ネットワークの耐故障性の度合いを示す指標の一つ

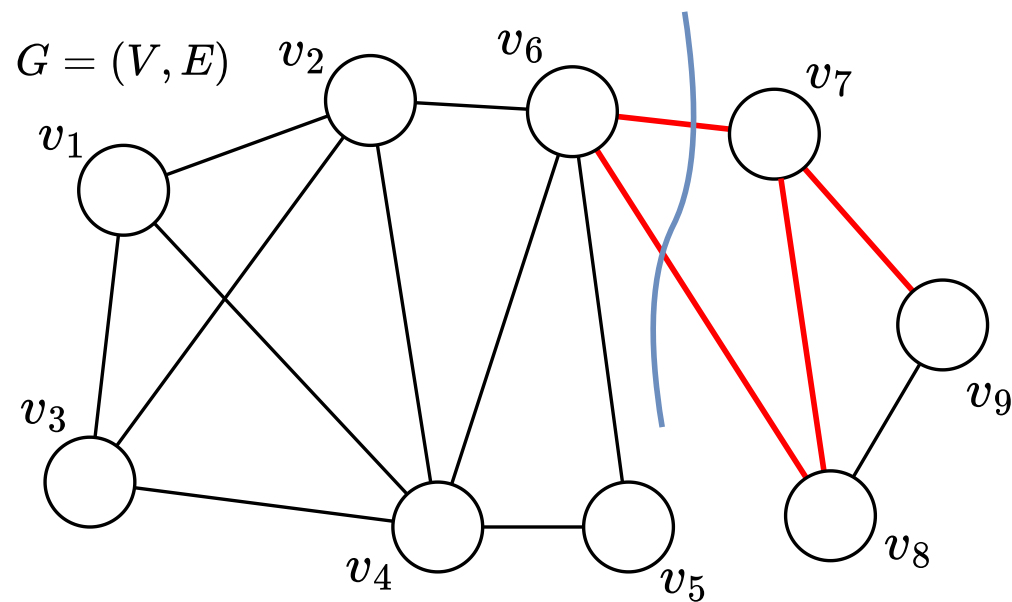
用語の定義 (1/2)

カット

削除するとグラフが非連結になる辺集合，頂点集合をそれぞれ**辺カット**，**点カット**という

ただし， $\{(v_6, v_7), (v_6, v_8), (v_7, v_8), (v_7, v_9)\}$ はカットではあるが，その部分集合 $\{(v_6, v_7), (v_6, v_8)\}$ だけでもカットである．

ここでは**カット**と言うとき，その真部分集合はいずれもカットではないことを仮定する



用語の定義 (2/2)

辺カットの表記

先の仮定より, G における辺カットはある頂点集合 $X \subseteq V$ に対し, X と $V - X$ の間にまたがる辺の集合として定義できる. 文脈によっては $E_G(X, V - X)$ と表記する.

$$\{(v_6, v_7), (v_6, v_8)\} \longleftrightarrow E_G(\{v_1, v_2, v_3, v_4, v_5, v_6\}, \{v_7, v_8, v_9\})$$

辺カット / 点カット

2頂点 $u, v \in V$ に対し, 削除すると u と v が非連結になる辺集合 $E' \subseteq E$ (頂点集合 $V' \subset V$) のことを u と v を分ける辺カット (点カット) という.

最小辺カット / 最小点カット

$|V'|$ ($|E'|$) をカットのサイズとするとき, サイズが最小のカットを最小辺カット (最小点カット) と定義する.

2.2 カットによる連結度の定義

辺連結度

グラフ $G = (V, E)$ において, 2頂点 $u, v \in V$ 間の**局所辺連結度**は, u と v を分ける最小辺カットのサイズで定義され, $\lambda_G(u, v)$ で表す.

点連結度

2頂点の $u, v \in V$ 間の**局所点連結度**は $(u, v) \notin E$ であれば, u と v を分ける最小点カットのサイズ, $(u, v) \in E$ であれば, u, v 間の辺を除いたグラフでの u と v を分ける最小点カットのサイズに1加えた値で定義され, $\kappa_G(u, v)$ で表すこととする.

グラフ全体の連結度

グラフ G 全体の辺連結度 $\lambda(G)$ は、すべての $u, v \in V$ 間の局所辺連結度の最小値で定義される。

同様に、グラフ G 全体の点連結度 $\kappa(G)$ はすべての $u, v \in V$ 間の局所点連結度の最小値で定義される。

すなわち、

$$\lambda(G) = \min\{\lambda_G(u, v) \mid u, v \in V\}$$

$$\kappa(G) = \min\{\kappa_G(u, v) \mid u, v \in V\}$$

である。

実例 (辺連結度)

v_1, v_9 間の局所辺連結度は $\lambda_G(v_1, v_9) = 2$

v_1, v_9 を分ける最小辺カットは

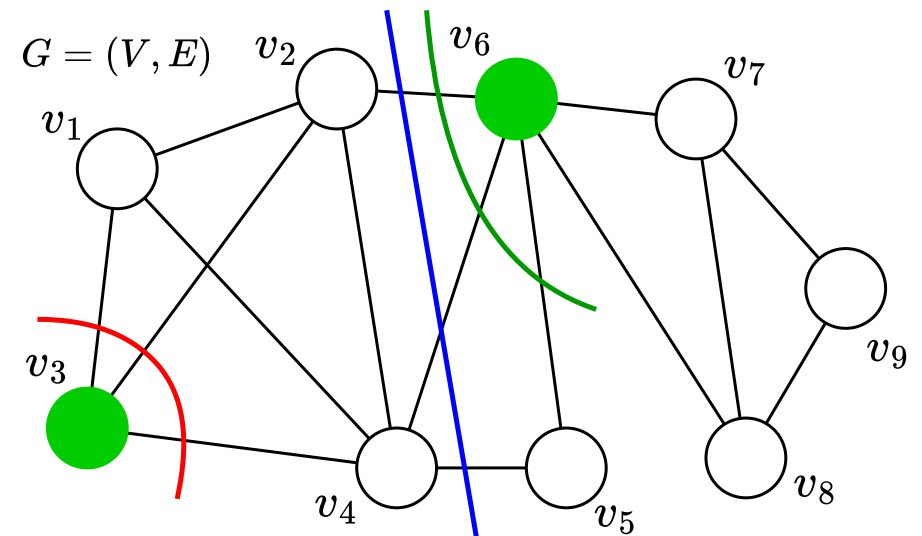
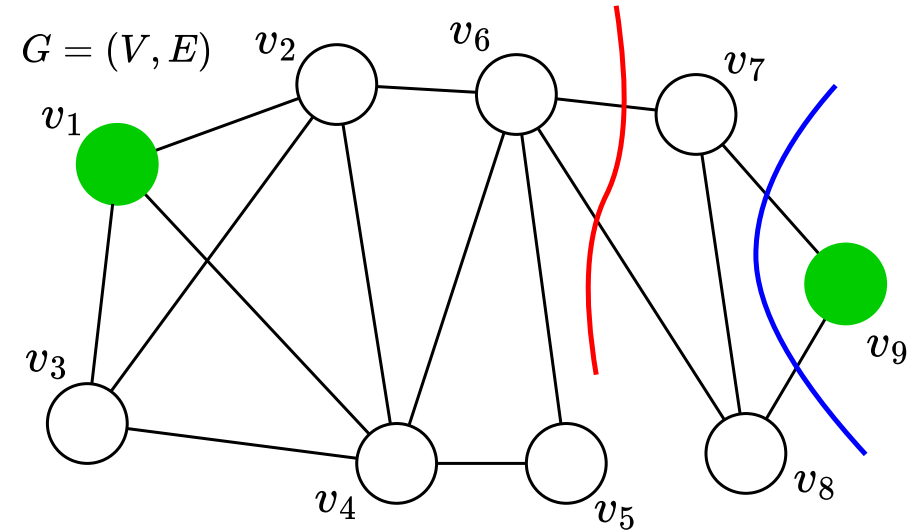
- $\{(v_6, v_7), (v_6, v_8)\} \rightarrow$ 赤
- $\{(v_7, v_9), (v_8, v_9)\} \rightarrow$ 青

v_3, v_6 間の局所辺連結度は $\lambda_G(v_3, v_6) = 3$

v_3, v_6 を分ける最小辺カットは

- $\{(v_1, v_3), (v_2, v_3), (v_3, v_4)\} \rightarrow$ 赤
- $\{(v_2, v_6), (v_4, v_5), (v_4, v_6)\} \rightarrow$ 青
- $\{(v_2, v_6), (v_4, v_5), (v_5, v_6)\} \rightarrow$ 緑

グラフ全体の辺連結度は $\lambda(G) = 2$



実例（点連結度）

v_1, v_9 間の局所点連結度は $\kappa_G(v_1, v_9) = 1$

v_1, v_9 を分ける最小点カットは

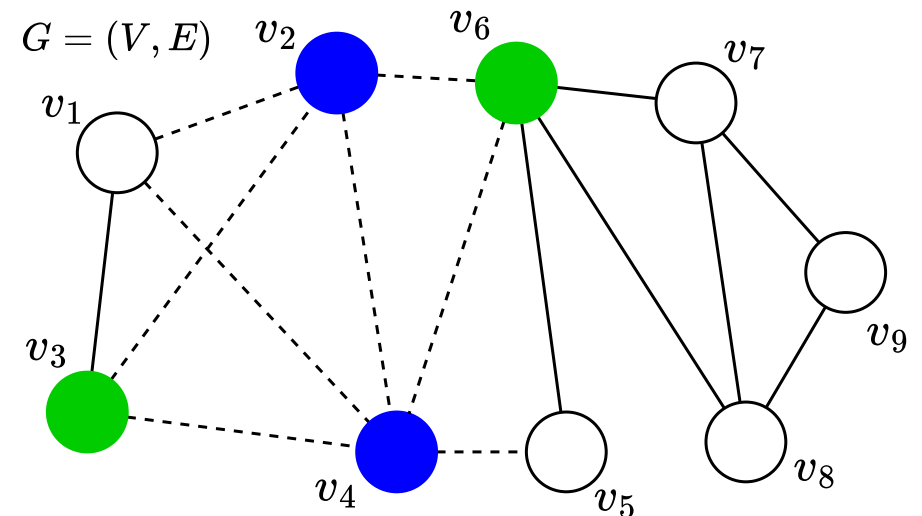
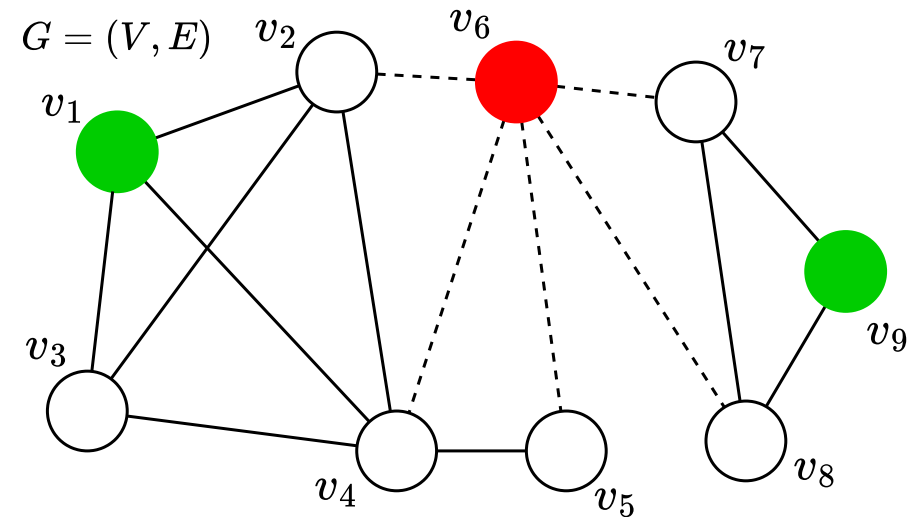
- $\{v_6\} \rightarrow$ 赤

v_3, v_6 間の局所辺連結度は $\kappa_G(v_3, v_6) = 2$

v_3, v_6 を分ける最小点カットは

- $\{v_2, v_4\} \rightarrow$ 青

グラフ全体の点連結度は $\kappa(G) = 1$



2.3 パスによる定義

辺連結度，点連結度は以下のように定義することもできる．

- $\lambda_G(u, v)$: u, v 間の，辺を互いに共有しない（**辺素な**）パスの最大数
- $\kappa_G(u, v)$: u, v 間の u, v 以外の頂点を互いに共有しない（**内部点素な**）パスの最大数

カットによる定義との同値性は，以下の定理により示すことができる．

メンガーの定理

1. 2頂点 $u, v \in V$ に対して， u, v を分ける辺カットの最小サイズは u, v 間の辺素なパスの最大数に等しい．
2. 互いに隣接しない2頂点 $u, v \in V$ に対して， u, v を分ける点カットの最小サイズは， u, v 間の内部点素なパスの最大数に等しい．

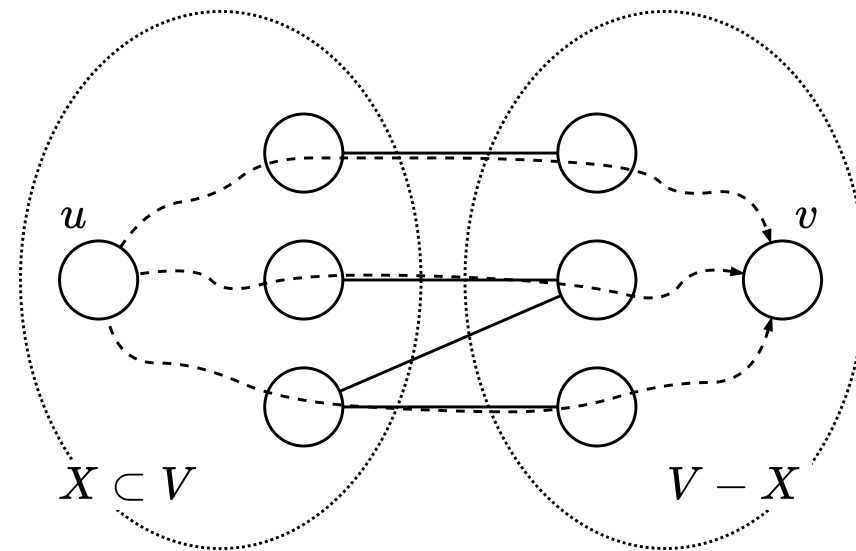
メンガーの定理の証明 (辺について) (1/2)

($\geq$)

頂点 u, v を分ける任意の辺カット E' に対して,
そのカットを通過するのに $|E'|$ 本の辺しか使えない
 $\rightarrow u, v$ の間に辺素なパスは高々 $|E'|$ 本しか存在しない

($\leq$)

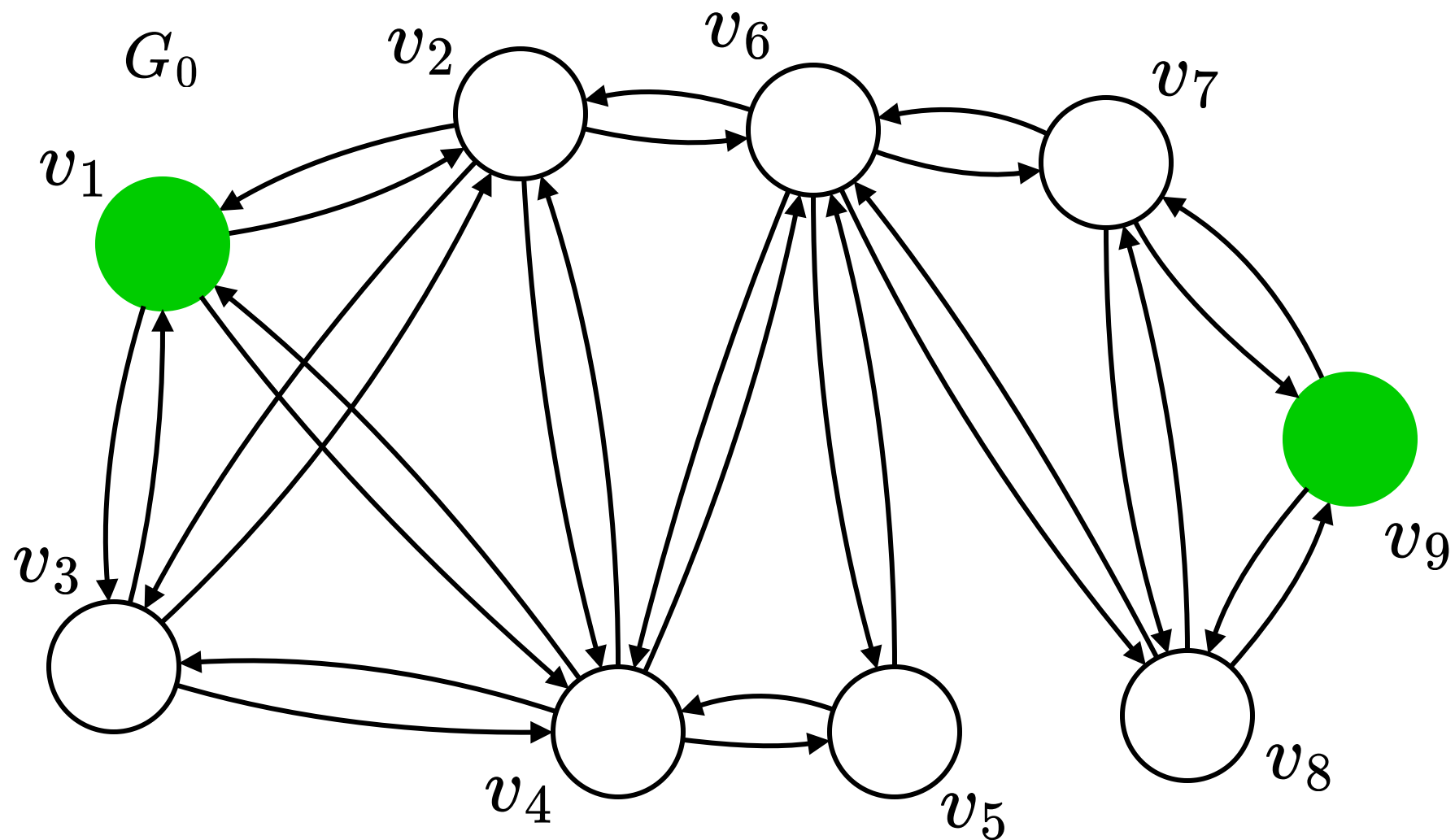
頂点 u, v の辺カットのサイズと同じ数の辺素なパスが存在することを構成的に示す.



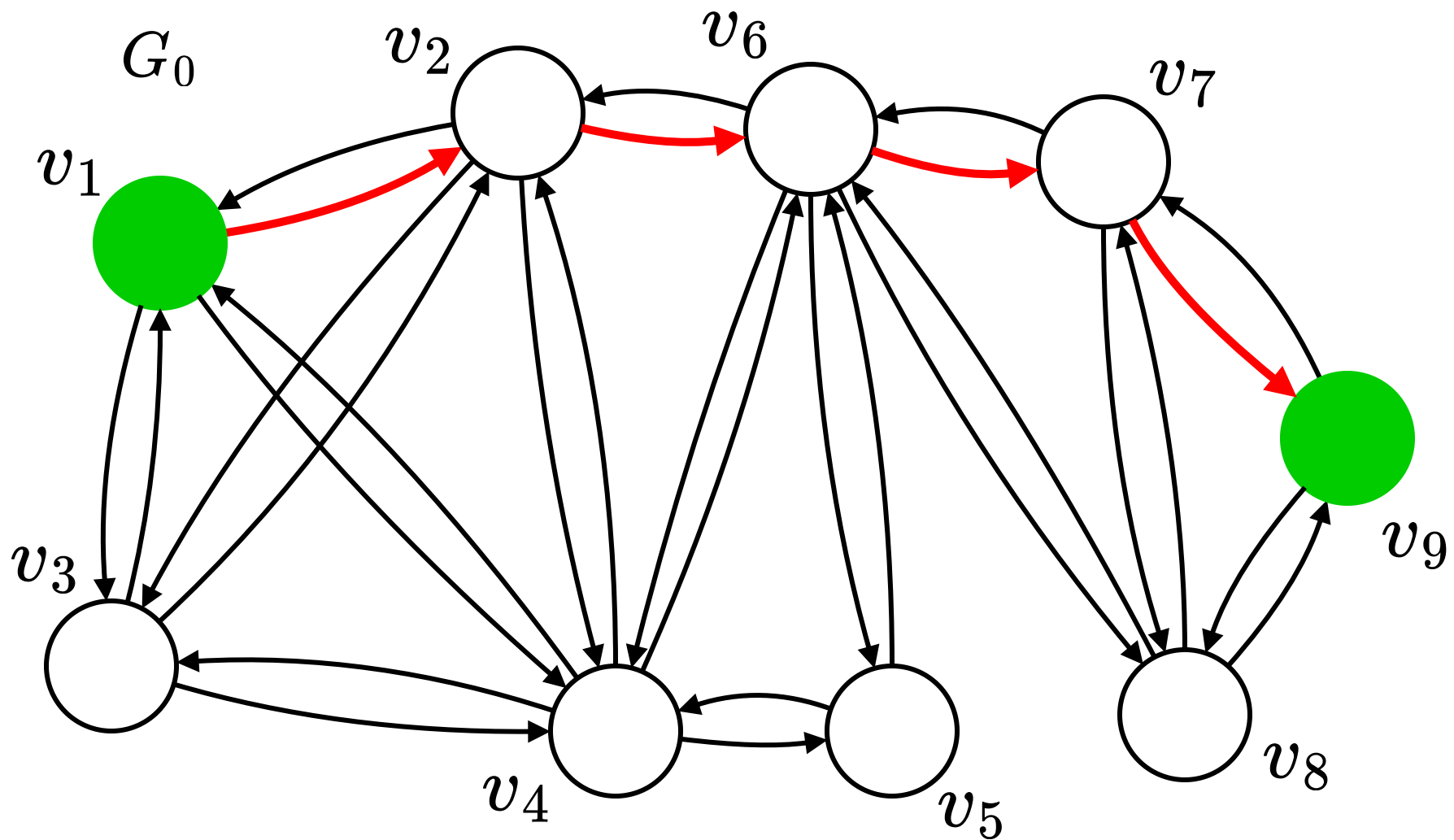
アルゴリズム

1. 各辺を互いに向きの異なる2つの有向辺に付け替える
2. その有向辺をたどって u, v 間のパスを探す
3. もし存在すれば, そのパス上の辺を逆向きに付け替える

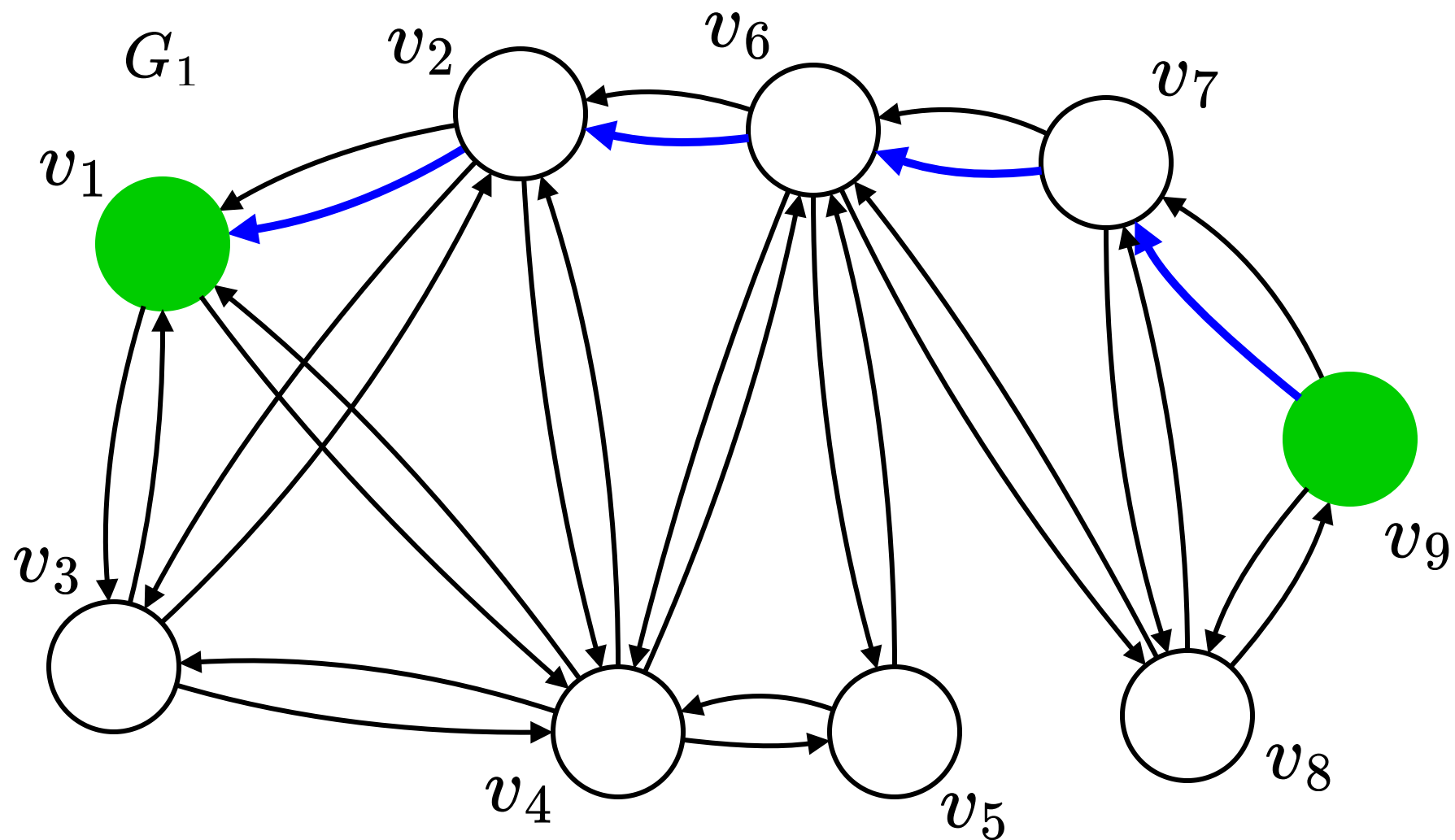
アルゴリズムの実例 (1/6)



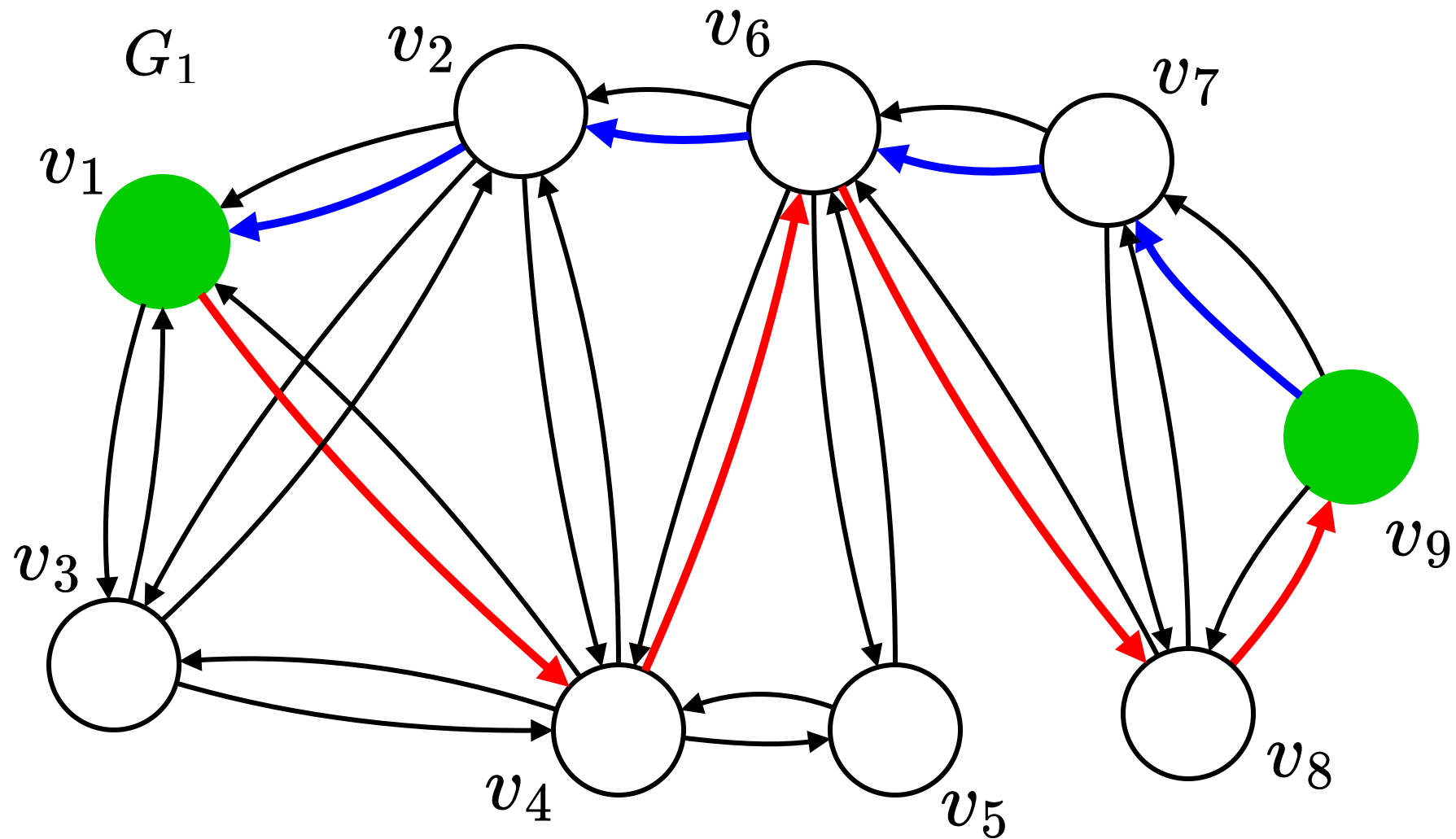
アルゴリズムの実例 (2/6)



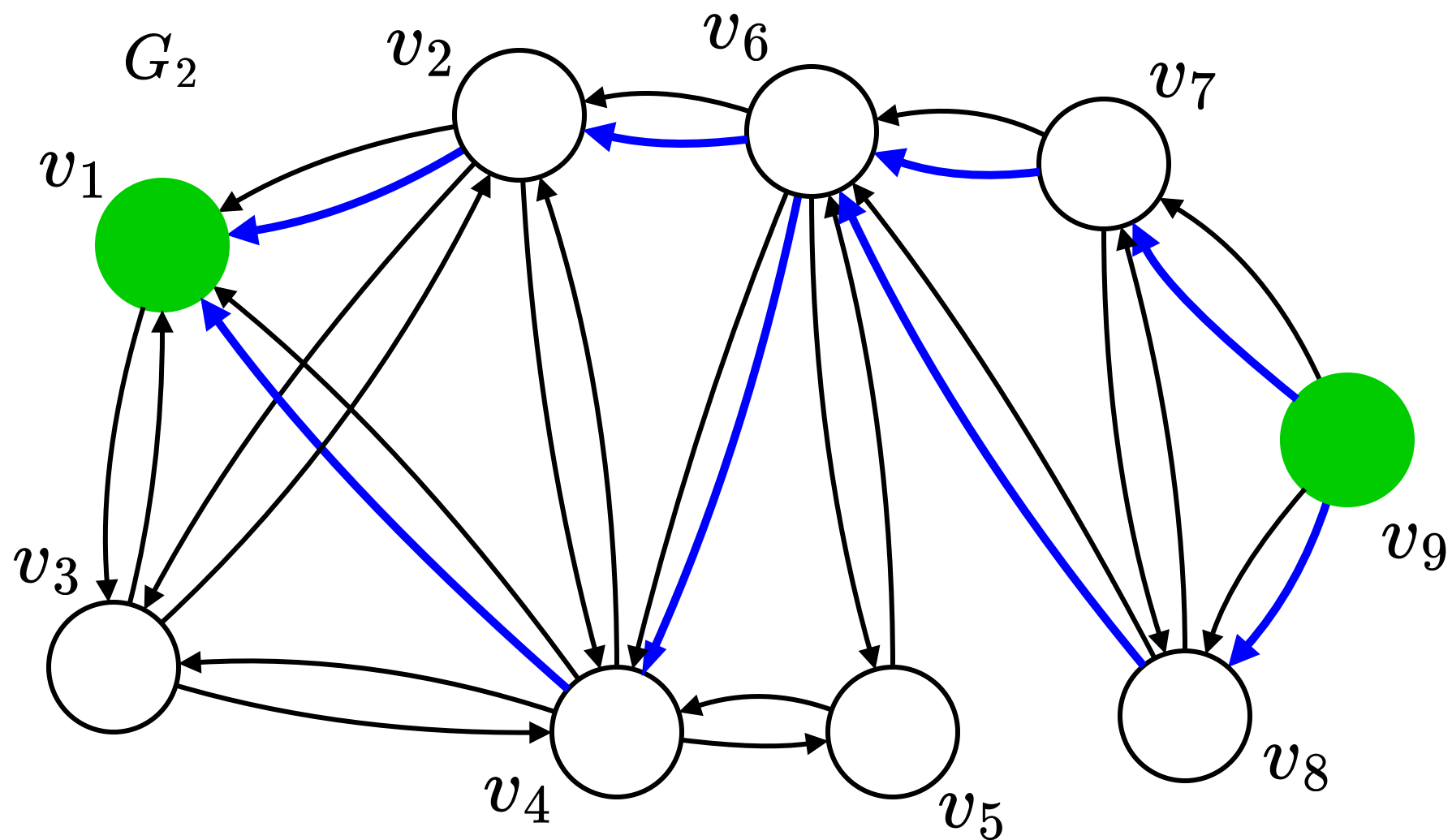
アルゴリズムの実例 (3/6)



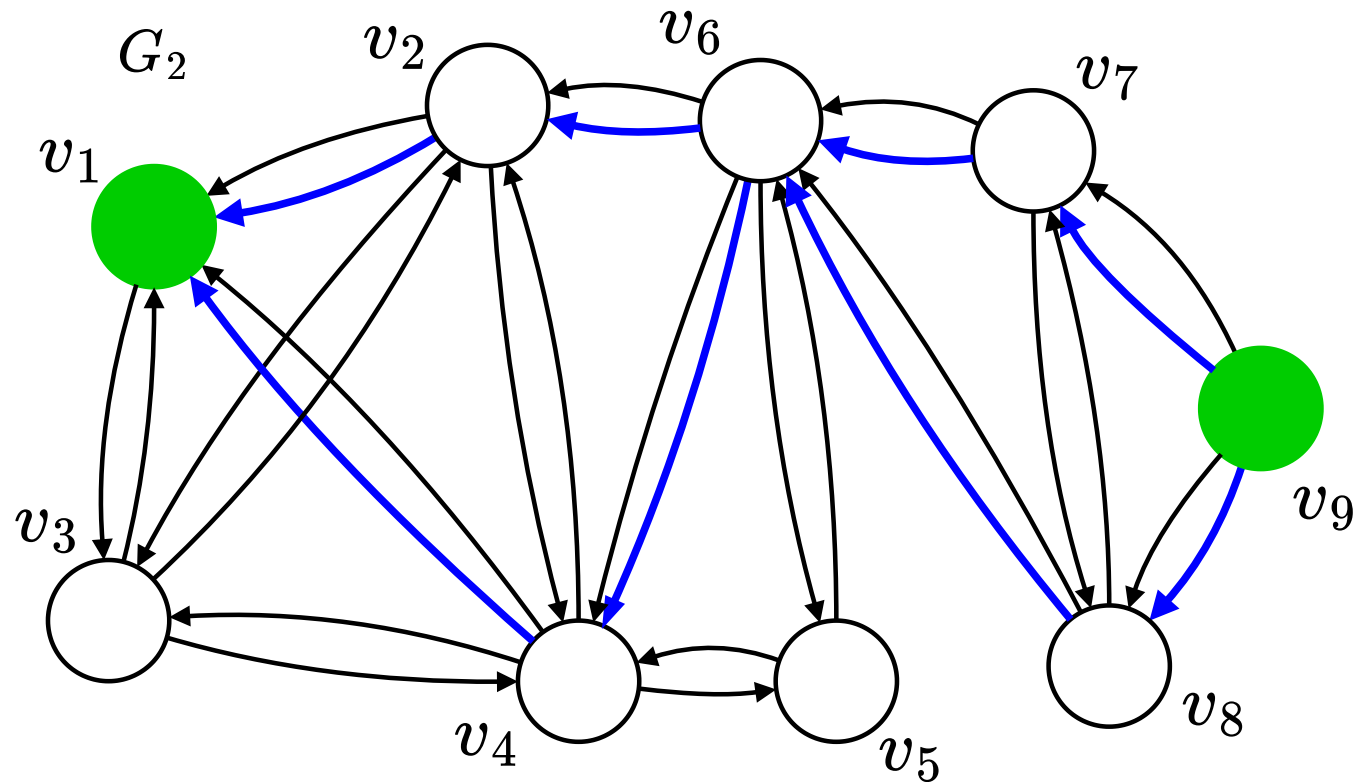
アルゴリズムの実例 (4/6)



アルゴリズムの実例 (5/6)



アルゴリズムの実例 (6/6)



G_2 には v_1 から v_9 へのパスは存在しないため終了. → 2つの辺素なパスが見つかった!

(これはFord-Fulkerson法とよばれるアルゴリズムである)

メンガーの定理の証明 (辺について) (2/2)

($\leq$) (つづき)

頂点 u から頂点 v までの辺素なパスを見つけていく過程で、すでに k 本のパスが見つまっている場合を考える．また、この状態での有向グラフを $\vec{G}_k$ とする．

このとき、 u, v を分ける辺カット $E_G(X, V - X)$ ($u \in X, v \in V - X$) に対して、元のグラフ G においてそのカットサイズが $k + 1$ 以上であったなら、 $\vec{G}_k$ には X から $V - X$ へ向く有向辺が少なくとも1本残っている．

($\because$ 存在しなければ、逆向きに辺を張る操作を $k + 1$ 回以上行ったことになり、矛盾)

つまり、 u, v を分ける辺カットの最小サイズ未満のパスしか見つからない状態ならば、必ず u から v へ到達する辺素なパスが見つかる．

よって、頂点 u から v に向かう $\lambda_G(u, v)$ 本の辺素なパスが存在する． $\square$

(厳密でない?)

計算量の解析（辺連結度）

局所辺連結度

先のアルゴリズムを用いると、辺素なパスの数を多項式時間で求めることができる。

（証明）

グラフ $G = (V, E)$ に対し、パスの数は辺の数を超えないから、高々 $|E|$ 個。

一つのパスは深さ優先探索などを用いることで $O(|V| + |E|)$ 時間で求めることができる。よって、全体でも $O(|E|(|V| + |E|))$ 時間で終了する。

→ 本には $O(|E|^2)$ と書かれているがなぜ？

グラフ全体の辺連結度

頂点のペアは $\frac{|V|(|V|-1)}{2}$ 個であるから、すべてのペアについて局所辺連結度を調べても、多項式時間で終了する。→ 実際にはもっと高速なアルゴリズムも存在する。

メンガーの定理の証明 (頂点について) (1/2)

メンガーの定理

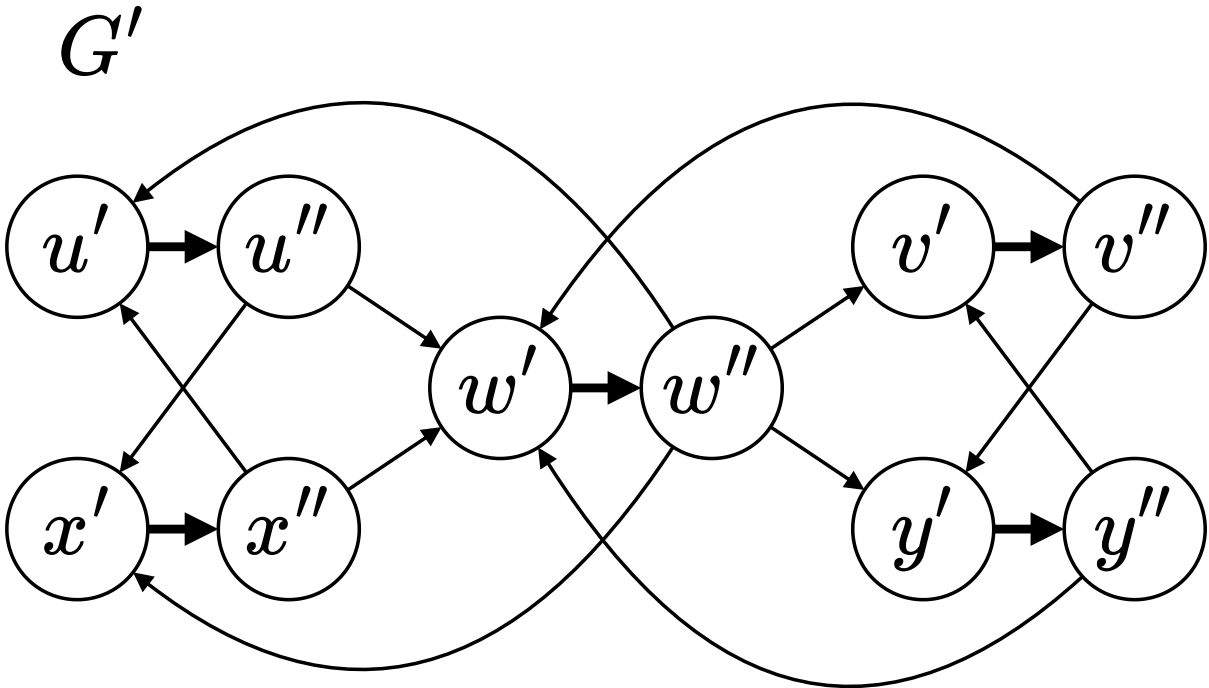
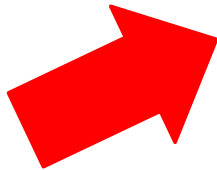
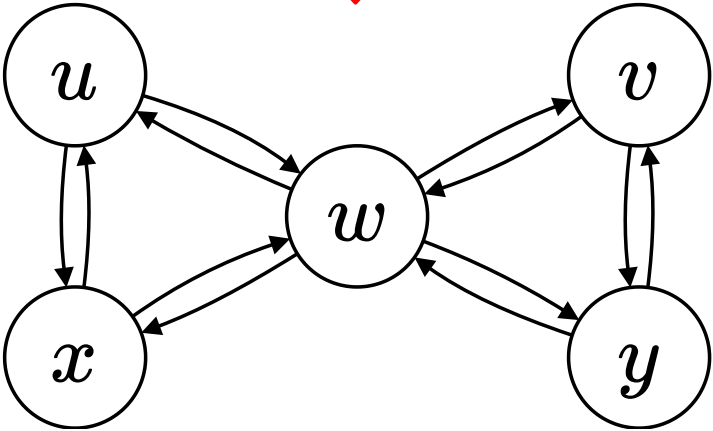
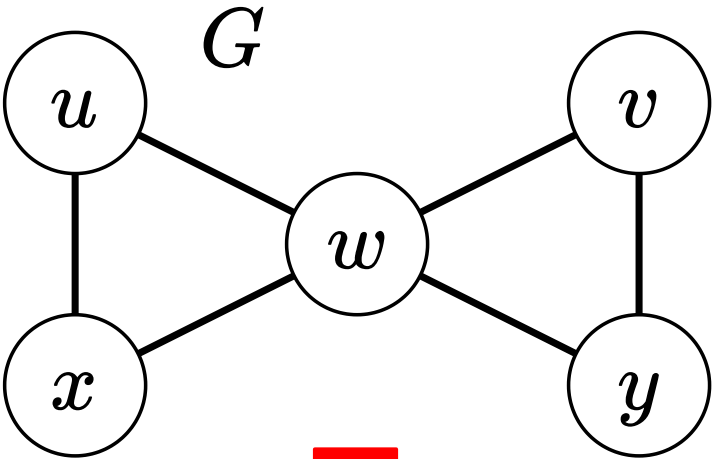
2. 互いに隣接しない2頂点 $u, v \in V$ に対して, u, v を分ける点カットの最小サイズは, u, v 間の内部点素なパスの最大数に等しい.

辺の場合と同様に, 各辺を異なる向きの2本の有向辺に変換する.

その後, 以下の変形を行い, グラフ G' に変換する. (変換 t)

- 各頂点 v を2頂点 v', v'' に分け, v' から v'' へ有向辺を1本加える
- u から v への有向辺 (u, v) を, 有向辺 (v'', u') に置き換える

変形の実例



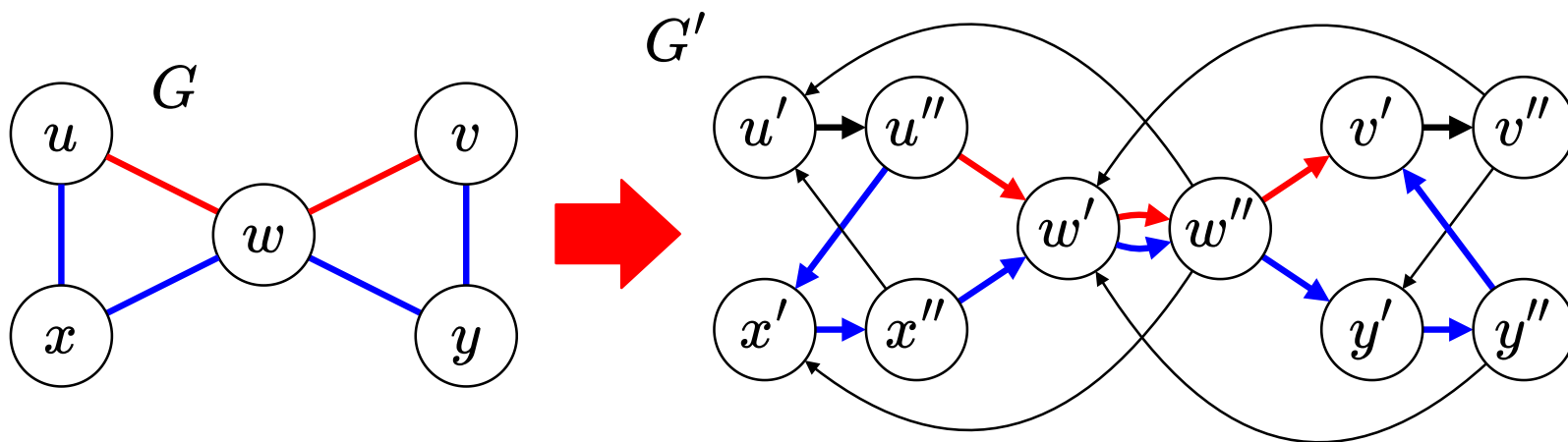
メンガーの定理の証明 (頂点について) (2/2)

この変換によって、元のグラフ G において頂点 w を通るパスは、変換後の有向グラフでは必ず有向辺 (w', w'') を通ることになる。

このとき、以下の2つの集合が対応付けられる⁽¹⁾

- G における u から v への内部点素なパスの集合 $\dots P_G$
- G' における u'' から v' への辺素なパスの集合 $\dots P_{G'}$

→ 先の辺についてのメンガーの定理に帰着させて考えることができる



(1) の対応付けが存在することの証明 (1/3)

1. グラフ G 上のパスの集合 P_G とグラフ G' 上のパスの集合 $P_{G'}$ の間に全単射 f が存在すること
2. グラフ G のパス $p_1, p_2 \in P_G$ が互いに内部点素であることと,
グラフ G' のパス $f(p_1), f(p_2)$ が互いに辺素であることが同値であること

を示せば良い.

(1) の対応付けが存在することの証明 (2/3)

1.

(1) $p \in P_G$ ($p = \{(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)\}$) に対し,

$f(p) = \{(v_1'', v_2'), (v_2', v_2''), (v_2'', v_3'), \dots, (v_{n-1}'', v_n')\}$ とすれば逆写像が存在し, 全単射となる. $\square$

(1) の対応付けが存在することの証明 (3/3)

2.

G における u から v へのパス p_1, p_2 を考える. p_1, p_2 が互いに内部点素でない場合, 変換後のパス $p'_1 = f(p_1), p'_2 = f(p_2)$ は共有している頂点 x に対して辺 (x', x'') を共有するため, 辺素でない. 逆に p'_1, p'_2 が辺素でない場合, G 上の頂点 x に対し G' 上の辺 (x', x'') または G 上の辺 (y, z) に対し, G' 上の辺 (y'', z') のいずれかを共有していることになるが, いずれにせよ同じ頂点を共有しているためパス p_1, p_2 は内部点素ではない. $\square$

辺連結度・点連結度の大小関係

辺連結度・点連結度の大小関係

1. 任意の $u, v \in V$ について, $\lambda_G(u, v) \geq \kappa_G(u, v)$.
2. $\lambda(G) \geq \kappa(G)$.

(証明)

任意のグラフ G と G 上のパス p に対し, 「 p が内部点素なパス $\Rightarrow p$ が辺素なパス」が成立する. すなわち 「内部点素なパスの集合 $\subseteq$ 辺素なパスの集合」が成り立つ.

さらに,

$$|u, v \text{間の内部点素なパスの集合}| \leq |u, v \text{間の辺素なパスの集合}| \Leftrightarrow \kappa_G(u, v) \leq \lambda_G(u, v)$$

また, 任意の2頂点 $u, v \in V$ について $\kappa_G(u, v) \leq \lambda_G(u, v)$ であることから, $\kappa(G) \leq \lambda(G)$

3. 連結度を考慮したネットワーク問題

これまでの連結度の議論をもとに,

1. ネットワーク解析問題
2. ネットワーク設計問題

の2つの問題を取り扱う.

3.1 ネットワーク解析問題

耐故障性の観点から、与えられたネットワークの性能や信頼性の解析を行う。

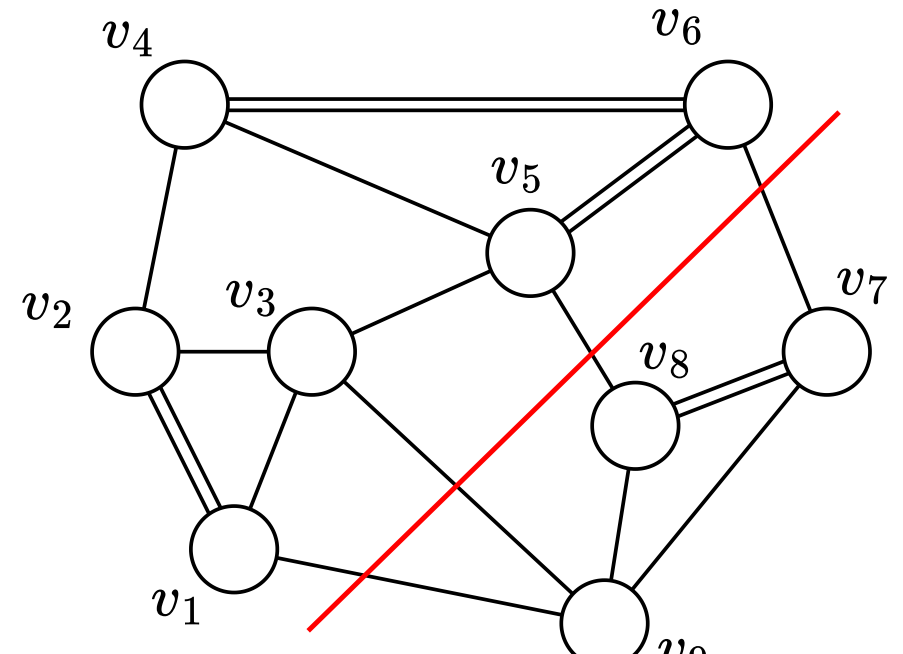
例) グラフ G のネットワークの辺連結度は 4 であるから、辺 3 本までの故障には耐えられることがわかる。

問題

現実には、辺連結度の値だけでなく
「**どの頂点間で局所連結度が最小になるか**」
を知りたい場合が多い。

(先のアルゴリズムでも最小辺カットを見つけることはできるが、**すべて**を見つけることは難しい)

図5



カクタスグラフ (1/4)

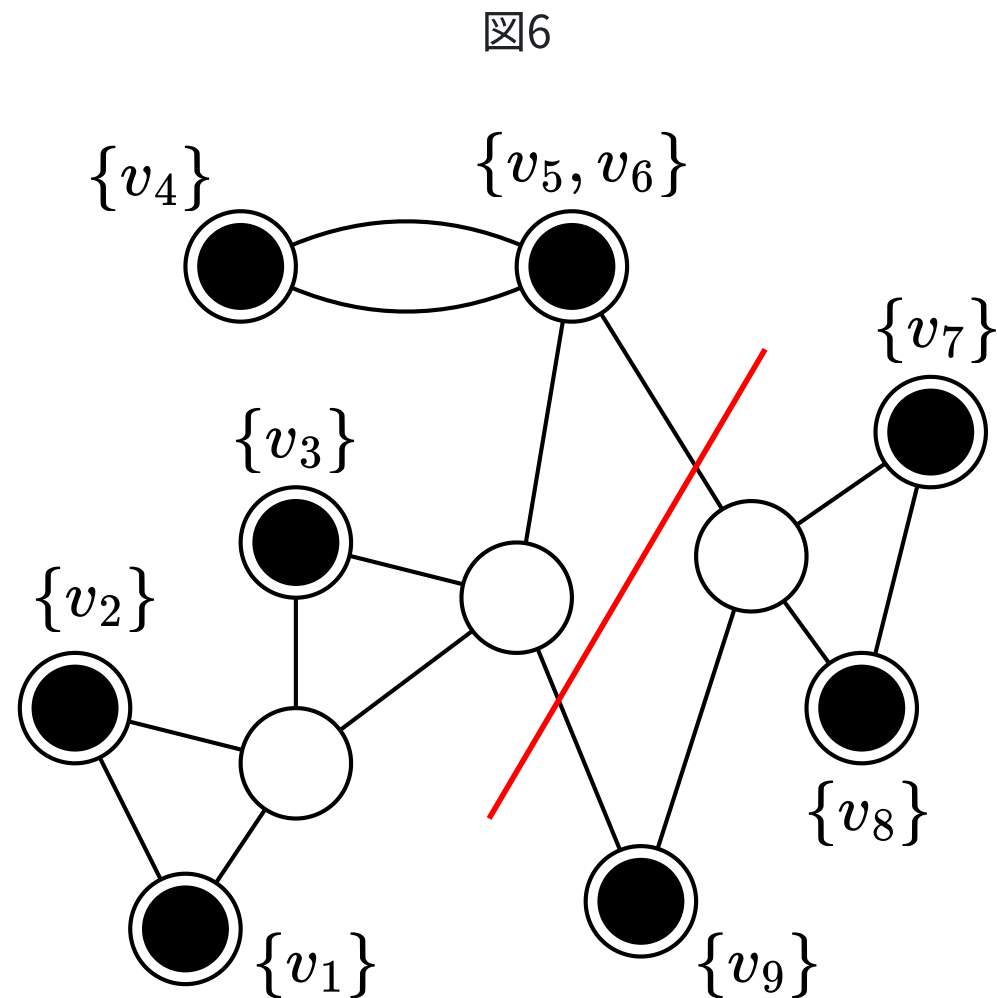
図5のグラフを別の形式に変換することにより、すべての最小辺カットを表現する。

図6：カクタスグラフ と呼ばれる表現 →
※カクタス：サボテンのこと

カクタスグラフ

任意の2つの閉路が高々1つの共有点しか持たないグラフ

カクタスグラフ上では，以下の手順で簡単に最小辺カットをみつけることができる。



カクタスグラフ (2/4)

グラフ G とそのカクタス表現 H の対応関係

1. H の任意の最小辺カット $E_H(Y, W - Y)$ に対して, Y に対応付けられている G の頂点集合を X_Y とすると, $E_G(X_Y, V - X_Y)$ は G の最小辺カットである.
2. G の任意の最小辺カットが H のある最小辺カットに対して1の変換により表現される.

カクタスは非常に簡潔なグラフで, 最小辺カットのサイズは2であるから, 簡単に最小辺カットを見つけることができる.

また, G の最小辺カットの総数は $O(|V|^2)$ である.

カクタスグラフ (3/4)

なぜ最小辺カットはカクタスのような閉路を繋げた形で表現できるのか？

→ 証明は長いので関連する性質を紹介

カクタス表現と関係がある最小辺カットの性質

グラフ G の辺カット $E_G(X, V - X)$ のサイズを $d_G(X)$ と表す．このとき，次の関係式が成り立つ．

$$d_G(X) + d_G(Y) \geq d_G(X \cap Y) + d_G(X \cup Y) \quad (1)$$

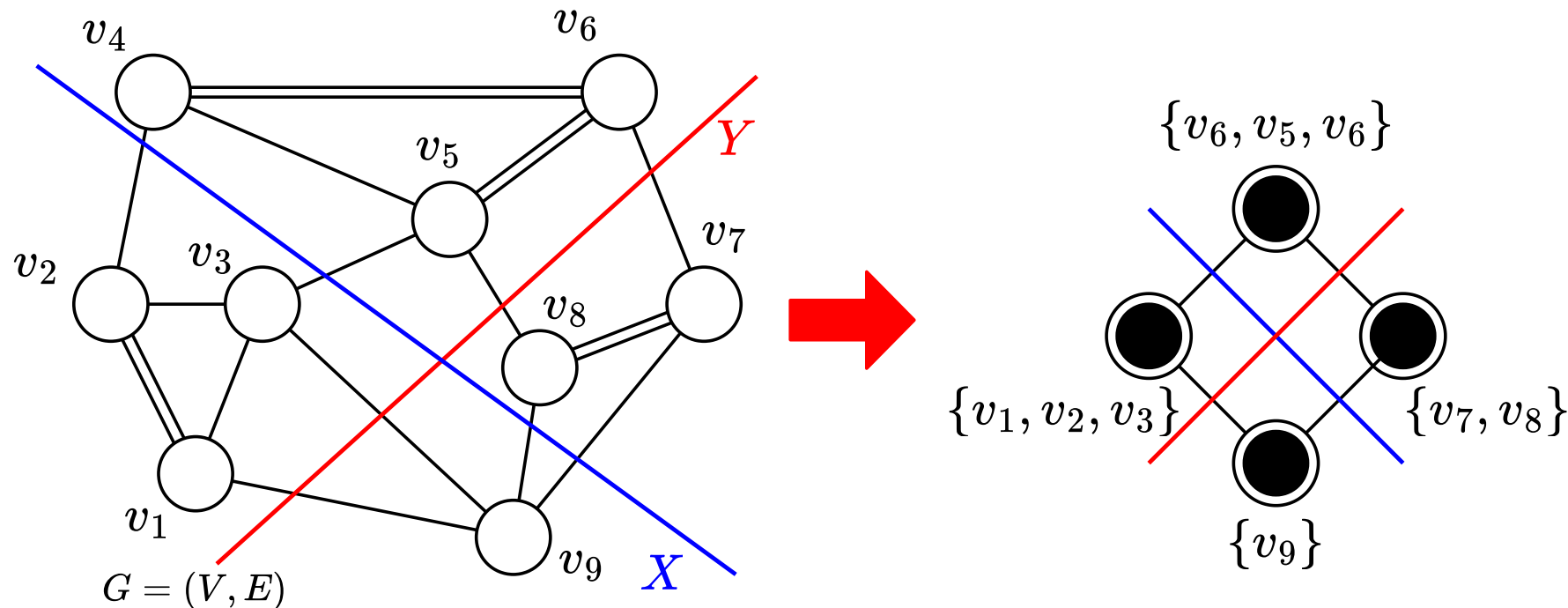
$$d_G(X) + d_G(Y) \geq d_G(X - Y) + d_G(Y - X) \quad (2)$$

カクタスグラフ (4/4)

$X = \{v_1, v_2, v_3, v_9\}, Y = \{v_7, v_8, v_9\}$ とする. $E_G(X, V - X), E_G(Y, V - Y)$ はともに最小辺カットであるが, 互いに交差するような形になっている. ここで,

$$\lambda(G) = d_G(X) + d_G(Y) = d_G(X \cap Y) + d_G(X \cup Y) = d_G(X - Y) + d_G(Y - X) = 8$$

であることから, $\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}, \{v_7, v_8\}, \{v_9\}$ をまとめてカクタス表現できる.



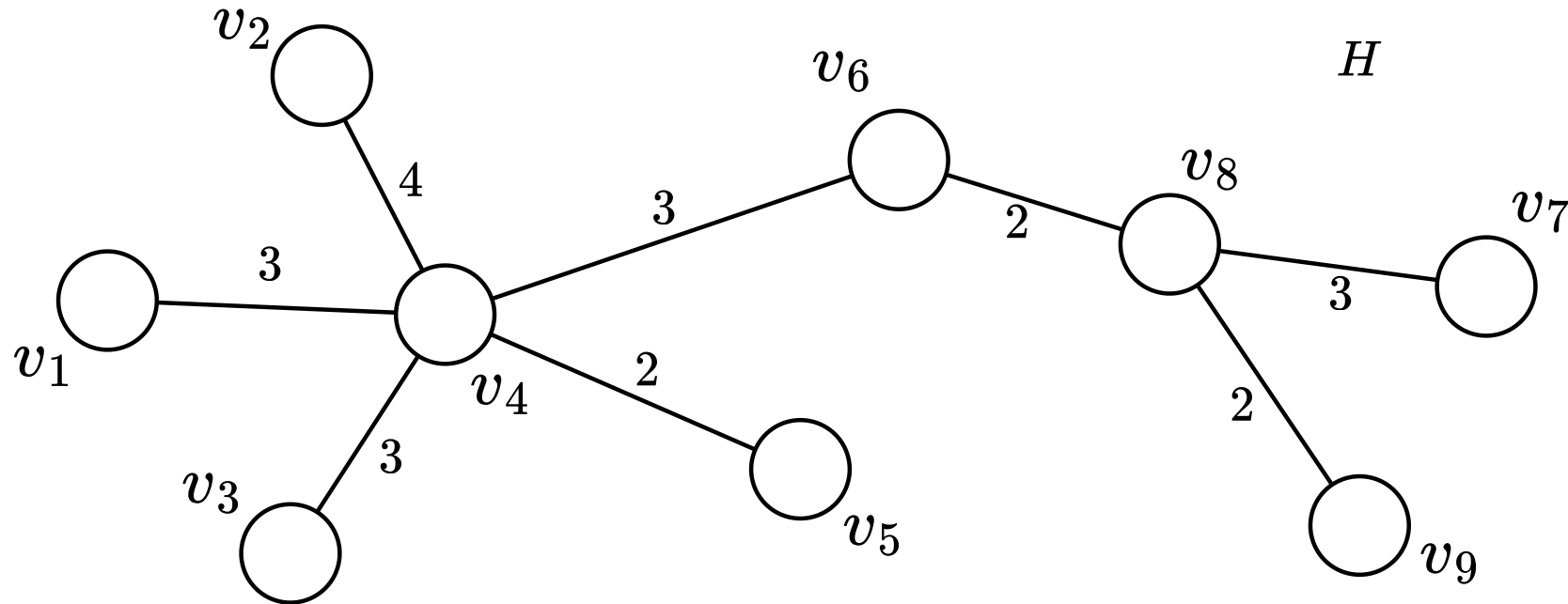
ゴモリー・フー木 (1/5)

ネットワークを完結に表現する手法としては他にも「ゴモリー・フー木」がある

下図の木 H の各頂点はグラフ G の各頂点に対応しており, 任意の2頂点 u, v に対し,

「 G における u, v 間の局所辺連結度」

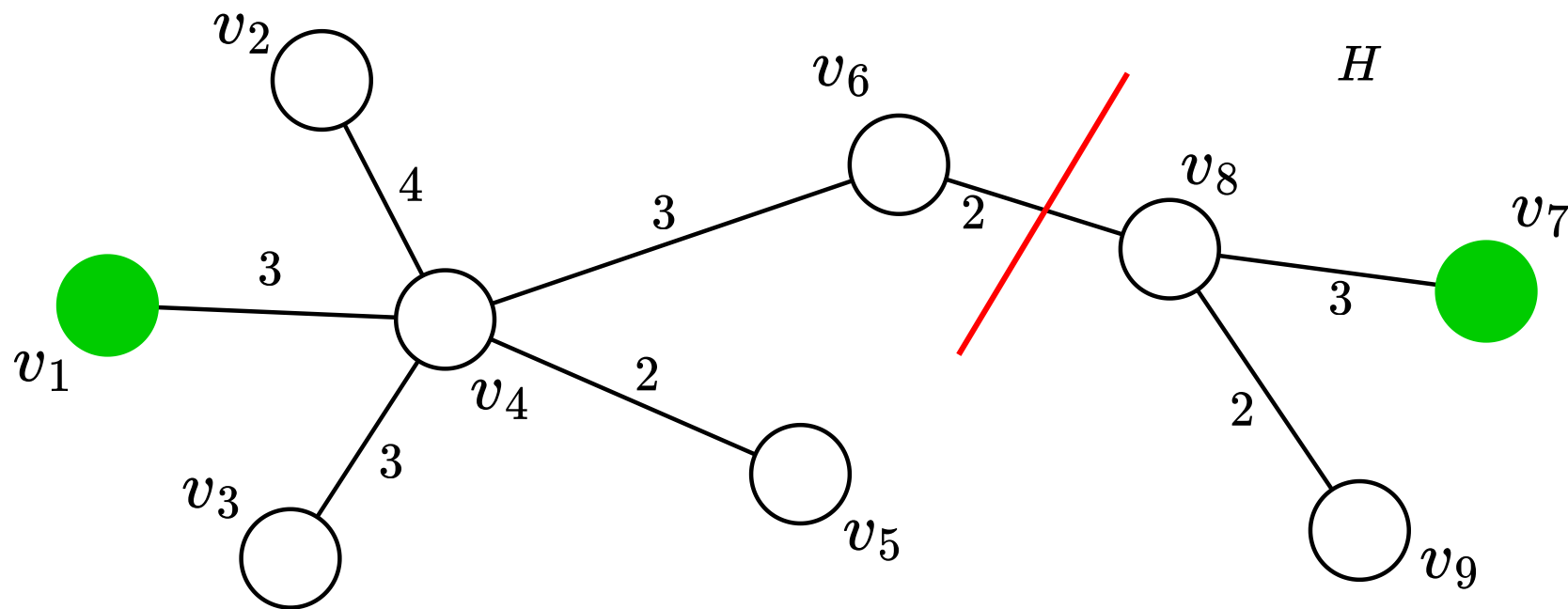
$\Leftrightarrow$ 「 H における u, v 間のパス上に付されている数値の最小値」



ゴモリー・フー木 (2/5)

さらに、「 H での u, v 間のパス上の最小の数値を持つ辺を削除したときに得られる連結な2つの頂点集合 $X, V - X$ について、 $E_G(X, V - X)$ は G において u と v を分ける最小辺カットである」という性質も満たす。

↓ 実際、 $X = \{v_7, v_8, v_9\}$, $V - X$ は G 上の最小辺カットである． ($\lambda_G(v_1, v_7) = 2$)



ゴモリー・フー木 (3/5)

ゴモリー・フー木の存在性については長いので省略．任意のグラフに対応するゴモリー・フー木を以下のアルゴリズムで構築できる．

ゴモリー・フーのアルゴリズム

1. $T_1 \leftarrow$ 1頂点からなるグラフ, $\mathcal{X} \leftarrow \{V(G)\}$, $k \leftarrow 1$ とおく．
2. $k = 1, 2, \dots, n - 1$ の順に, 以下の i - iv を反復する
 - i. $|X| \geq 2$ である $X \in \mathcal{X}$ を選択し, X に属する2点 s, t を選択する． T_k において X に対応する点 x に接続する各辺を $e_i = \{x, x_i\}$ と記し, e_i の先の頂点を縮約し, X_{e_i} と定める．
 - ii. どの X_{e_i} も横断しない G における最小 (s, t) -カット $\{S, V - S\}$ とその重み $\lambda_G(s, t)$ を計算する．
 - iii. $X' = X \cap S$ に対応する点 x' と $X'' = X \cap (V - S)$ に対応する点 x'' を新たに作る． T_k において, x を x', x'' に置き換え, この新しい2点の間に重み $\lambda_G(s, t)$ の辺を加える．また, 各 e_i について $X_{e_i} \subseteq S$ であれば, e_i を新しい辺 $e'_i = \{x, x_i\}$ に置き換える．この結果得られる木を T_{k+1} とする．
 - iv. $\mathcal{X}_{k+1} \leftarrow (\mathcal{X}_k - \{X\}) \cup \{X', X''\}$ ．
3. $T \leftarrow T_n$ を出力して終了．

ゴモリー・フー木 (4/5)

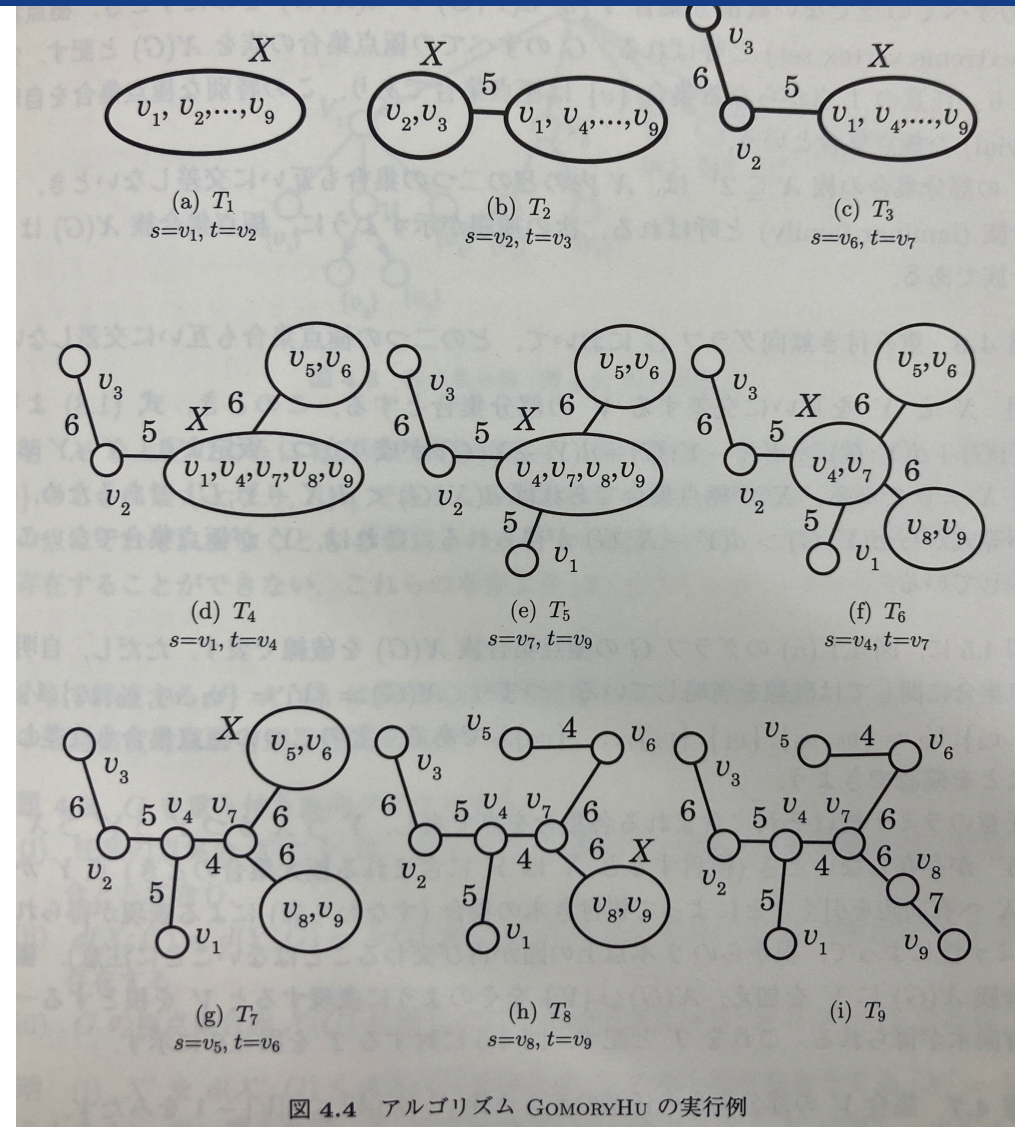
ゴモリー・フーのアルゴリズムの計算量解析

ゴモリー・フーのアルゴリズムで最も計算時間を要するのはステップ 2-i で s, t 間の最小辺カットを計算する部分である．よって，アルゴリズム全体では最小辺カットを $n - 1$ 回計算するだけの計算回数が必要になる．

ここで，最小カットは頂点，辺の数に関して多項式時間で計算できるため，ゴモリー・フー木も多項式時間で計算できる．

参考：「グラフ理論 -連結構造とその応用-」 p76-83 [2]

ゴモリー・フー木 (5/5)



ネットワーク解析問題のまとめ

辺カット

辺カット解析においては、カクタス表現，ゴモリー・フー木などを用いてグラフを単純な形に変形する手法が有効である．

また，任意のグラフについてカクタス表現もゴモリー・フー木も必ず存在し，かつ多項式時間で計算できることがわかっている．

点カット

1 ～ 2 程度の低い点連結度のグラフに関しては木表現などの手法を用いることが可能．

それ以上のグラフに関してはきれいな表現方法は知られていない．

3.2 ネットワーク設計問題

既存のネットワークの耐故障性を解析する問題を取り上げた。
より実践的な「グラフにどのように辺を追加すれば、連結度を向上させられるか」という問題を扱う。

連結度増大問題

入力

グラフ $G = (V, E)$, 目標の辺連結度 k

出力

$G + F$ の辺連結度が k 以上となる最小本数の辺集合 F
(ただし, $G + F = (V, E \cup F)$ とする)

(点連結度の場合もある)

連結度増大問題 (1/3)

ここでは、図5のグラフ G の連結度を4から5に増大させてみる。

解法1

1. 1本の辺の加え方を全通り
2. 2本の辺の加え方を全通り
3. ...

グラフ全体の連結度が5を超えるまで行う

→ かなり効率が悪い

見方を変えて、「何本の辺を加える必要があるか」を考える。

例) G の連結度は4であるから、少なくとも1本は辺を加える必要がある。

→ もし1本加えて辺連結度が5になれば、その解は最適である。

連結度増大問題 (2/3)

解の下界を見積もる解法

辺連結度を1増やすためには、各最小辺カット $E_G(X, V - X)$ に対して X と $V - X$ の間に少なくとも1本の辺を加える必要がある。→「 X の不足度は1である」と表現する。

G の最小辺カットは図6の H で表されているため、 H を見てみる。 H の次数2の頂点に対応する G の頂点集合 X に対して、 $E_G(X, V - X)$ は G の最小辺カットであることから、 $\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_7\}, \{v_8\}, \{v_9\}$ はいずれもその不足度は1である。

→ 不足度の合計は7。これを、辺を追加することで解消したい。

辺 (v_1, v_2) を追加すると、 v_1, v_2 の不足度が1ずつ上がる。同様に1本の辺を加えることで、多くて2の不足度を解消することができる。よって、 $\lceil \frac{7}{4} \rceil = 4$ 本の辺を追加する必要がある。

連結度増大問題 (3/3)

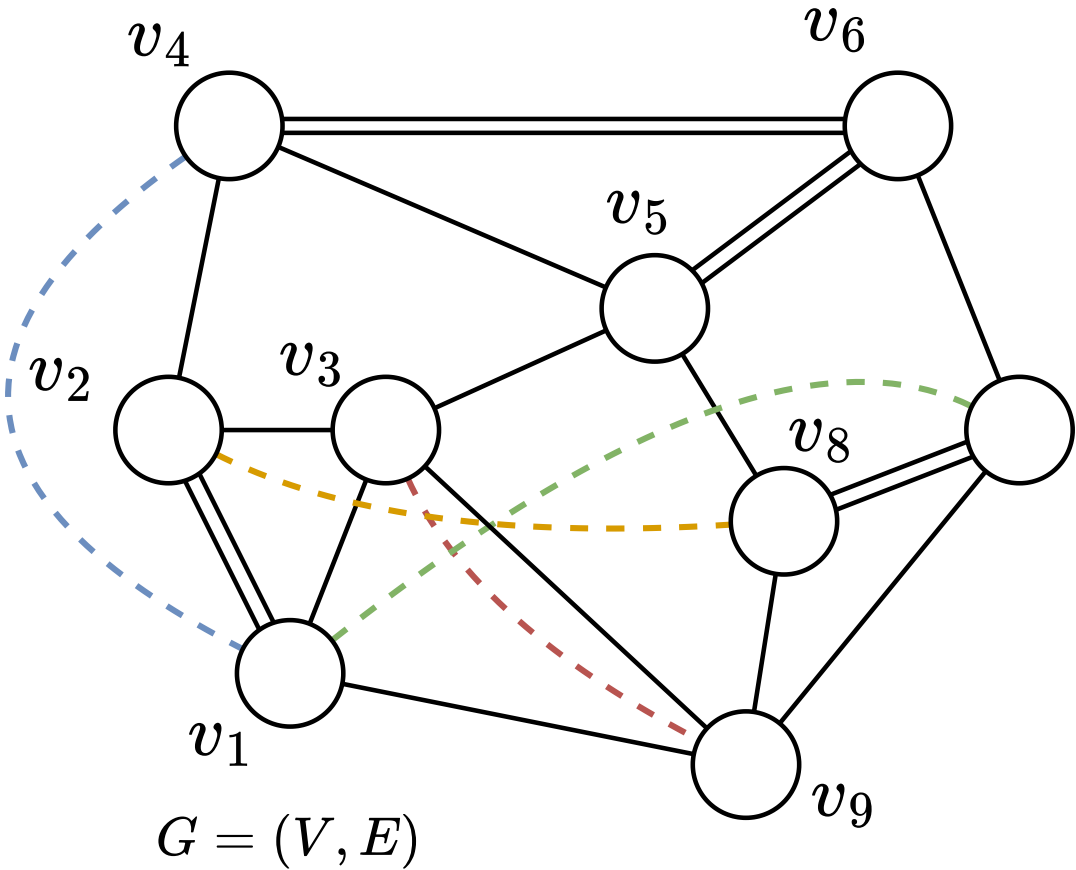
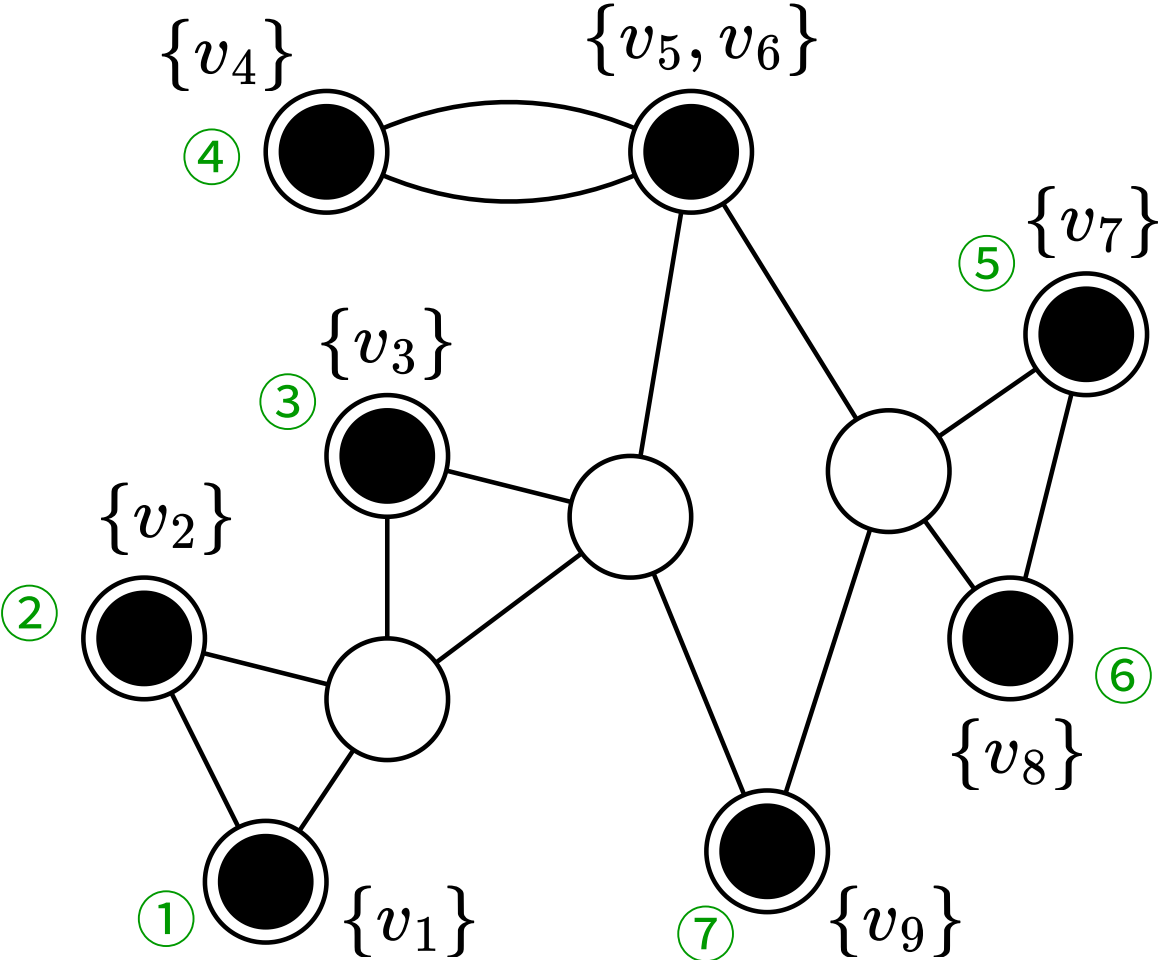
カクタス表現を用いた解法

1. 図6のカクタス表現において,
最小カットを持つ頂点を外側からなぞるようにたどり, 番号をつける.
2. たすき掛けのように互いに番号が $\lceil \frac{7}{2} \rceil$ ずれている頂点同士を辺で結ぶ.

図6は G のすべての最小辺カットを表現しているため, これらの辺に対して辺が1本以上加えられていることがわかる. → 最低限必要な本数の辺集合で解を構成できたため, この解は最適である.

すべての最小辺カットを表すカクタスグラフが多項式時間で構成できることから, この解は多項式時間で構成できる.

カクタス表現を用いた解法（実例）



ネットワーク設計問題のまとめ

辺カット

「解の下界値」を用いて最適解を多項式時間で求めることが可能

点カット

点連結度を増大させる問題に関しては、未解決な部分が多く、多項式時間で解けるかどうかもわかっていない。

- [1] 伊藤大雄・宇野裕之,「離散数学のすすめ」,現代数学社, 2010
- [2] 茨木俊秀・永持仁・石井利昌,「グラフ理論 -連結構造とその応用-」,朝倉書店, 2010