

ネットワーク設計における貪欲法

1. 頂点重みシュタイナー木

福永 拓郎

fukunaga@ise.chuo-u.ac.jp

今回のテーマ

NP 困難問題に対する貪欲アルゴリズム

集合被覆問題 → グラフ上の問題

1. 頂点重みシュタイナー木

スパイダーを使った貪欲アルゴリズム

2. Uncorssable 集合族被覆

スパイダーの拡張, LP を用いた解析

3. Buy-at-bulk ネットワーク設計問題

ジャンクション木を使った貪欲アルゴリズム

集合被覆問題

集合被覆問題

集合被覆問題

入力

- 有限集合 I ($|I| = n$ とする)
- I の部分集合 $S_1, S_2, \dots, S_m \subseteq I$
- 非負重み $w(j) \geq 0$ ($\forall j \in M := \{1, 2, \dots, m\}$)

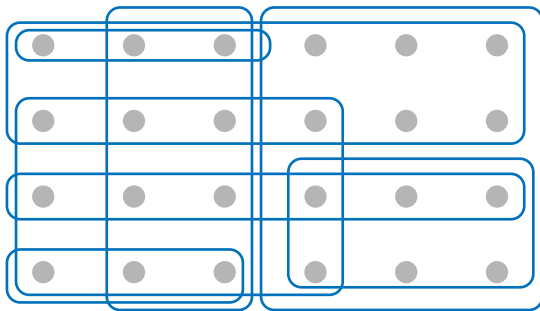
$N \subseteq M$ が**集合被覆** $\Leftrightarrow \bigcup_{j \in N} S_j = I$

重み最小の集合被覆 N を求めよ

$i \in I$ が S_j に含まれるとき、集合 S_j は要素 i を**被覆する**という。

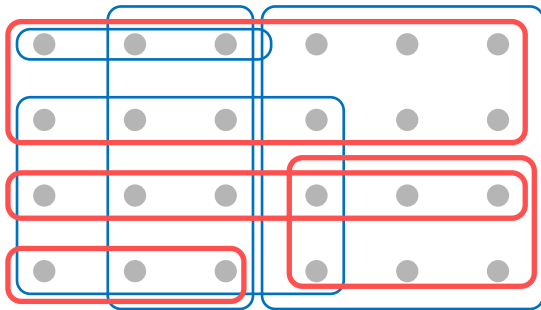
問題例

灰色の点が I の要素, 青い線が $S_1, \dots, S_m$ を表す.



問題例

灰色の点が I の要素, 青い線が $S_1, \dots, S_m$ を表す.



集合被覆の困難性

特殊ケースとして多くの問題を含んでいる

- 頂点被覆問題
 - 被覆すべきもの = 辺
 - 選ぶもの = 頂点
- 全域木問題
 - 被覆すべきもの = グラフのカット
 - 選ぶもの = 辺

定理 (Feige 1998)

$\text{NP} \subseteq \text{ZTIME}(n^{O(\log \log n)})$ でない限り、集合被覆問題では $\Omega(\log n)$ 近似より良い近似比を多項式時間では達成できない。

集合被覆問題に対する近似アルゴリズム

密度 $\text{density}(j) := \frac{w(j)}{S_j \text{を選ぶと新たに被覆される要素の数}}$

が最小の集合 S_j を選び続ける.

アルゴリズム

1. $i := 1, N_{i-1} := \emptyset$
2. while N_{i-1} が集合被覆でない:
 - $j_i := \arg \min \text{density}(j)$
 - $N_i := N_{i-1} \cup \{j_i\}, i := i + 1$
3. N_{i-1} を出力

近似比は $H_n = 1 + 1/2 + \cdots + 1/n$

集合被覆問題の一般化

ポテンシャル関数 $\phi: 2^M \rightarrow \mathbb{Z}_{\geq 0}$

- **単調:** $\phi(N) \leq \phi(N') \ (\forall N \subseteq N')$

- **劣モジュラ:**

$$\phi(N \cup \{i\}) - \phi(N) \geq \phi(N' \cup \{i\}) - \phi(N') \\ (\forall N \subseteq N' \subseteq M \setminus \{i\})$$

- 今回は簡単のため $\phi(\emptyset) = 0, \phi(M) = n$ とする.

集合被覆問題の一般化

劣モジュラ被覆問題

入力

- 有限集合 M
- 非負重み $w(j) \geq 0 \ (\forall j \in M)$
- ポテンシャル関数 $\phi: 2^M \rightarrow \mathbb{Z}_{\geq 0}$

制約

$N \subseteq M$ が劣モ被覆 $\Leftrightarrow \phi(N) = n$

重み最小の劣モ被覆 N を求めよ

集合被覆問題 $\rightarrow \phi(N) := |\bigcup_{i \in N} S_i|$

貪欲アルゴリズム（劣モジュラ被覆）

$$\text{密度 density}(j) := \frac{w(j)}{j \text{を選んだ際のポテンシャルの増分}}$$

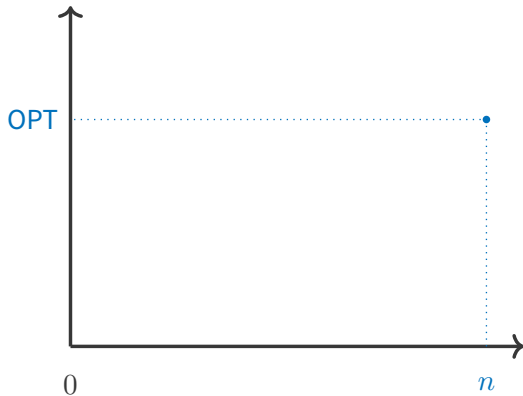
劣モジュラ被覆アルゴリズム

1. $i := 1, \phi_0 := 0$
2. while $\phi_i < n$:
 - $j_i := \arg \min \text{density}(j)$
 - $\phi_i := \phi(\{j_1, \dots, j_i\}), i := i + 1$
3. $\{j_1, \dots, j_i\}$ を出力

近似比 H_n

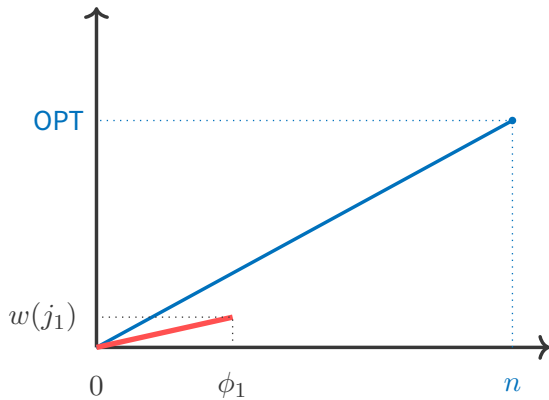
近似比の解析

$$\frac{w(j_i)}{\phi_i - \phi_{i-1}} \leq \frac{\text{OPT}}{n - \phi_{i-1}}$$



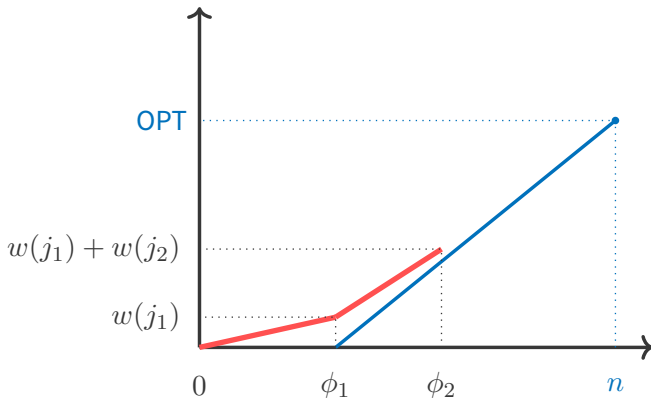
近似比の解析

$$\frac{w(j_i)}{\phi_i - \phi_{i-1}} \leq \frac{\text{OPT}}{n - \phi_{i-1}}$$



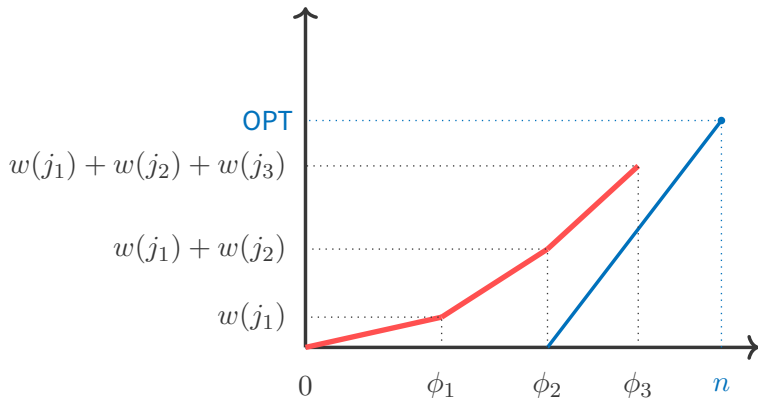
近似比の解析

$$\frac{w(j_i)}{\phi_i - \phi_{i-1}} \leq \frac{\text{OPT}}{n - \phi_{i-1}}$$



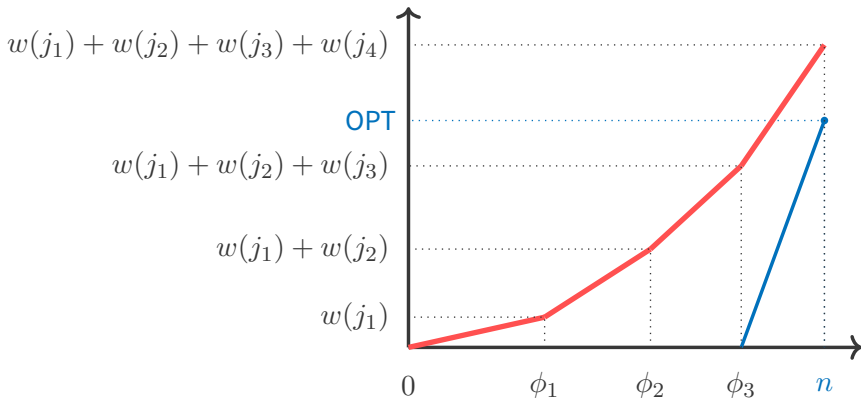
近似比の解析

$$\frac{w(j_i)}{\phi_i - \phi_{i-1}} \leq \frac{\text{OPT}}{n - \phi_{i-1}}$$



近似比の解析

$$\frac{w(j_i)}{\phi_i - \phi_{i-1}} \leq \frac{\text{OPT}}{n - \phi_{i-1}}$$



近似比

$w(\text{出力解})$

$$= w(j_1) + w(j_2) + w(j_3) + \dots$$

$$\leq \left(\frac{\phi_1 - 0}{n - 0} + \frac{\phi_2 - \phi_1}{n - \phi_1} + \frac{\phi_3 - \phi_2}{n - \phi_2} + \dots \right) \cdot \text{OPT}$$

$$\leq \left(\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{n-\phi_1} + \frac{1}{n-\phi_1-1} + \dots \right) \cdot \text{OPT}$$

$$= H_n \cdot \text{OPT}$$

主定理

$$\frac{w(j_i)}{\phi_i - \phi_{i-1}} \leq \frac{\text{OPT}}{n - \phi_{i-1}}$$

最適解 = $\{1, 2, \dots, k\}$, $S_{i-1} = \{j_1, \dots, j_{i-1}\}$ とする

$$\frac{\text{OPT}}{n - \phi_{i-1}} = \frac{w(1) + w(2) + \dots + w(k)}{\phi(S_{i-1} \cup \{1, \dots, k\}) - \phi(S_{i-1})} \quad \because \text{最適解の定義}$$

$$\geq \frac{w(1) + w(2) + \dots + w(k)}{\sum_{j=1}^k (\phi(S_{i-1} \cup \{j\}) - \phi(S_{i-1}))} \quad \because \phi \text{ が劣モ}$$

$$\geq \min_{j=1, \dots, k} \frac{w(j)}{\phi(S_{i-1} \cup \{j\}) - \phi(S_{i-1})} \quad \because \text{演習問題}$$

$$\geq \frac{w(j_i)}{\phi_i - \phi_{i-1}} \quad \because j_i \text{ の定義}$$

頂点重みシュタイナー木

問題設定

頂点重みシュタイナー木問題

入力

- 無向グラフ $G = (V, E)$
- 端点集合 $T \subseteq V$ ($k := |T|$ とする)
- 頂点重み $w(v) \geq 0$ ($v \in V$)

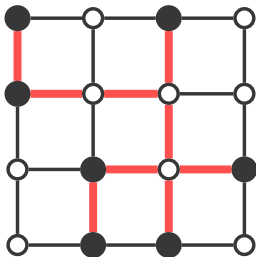
制約

シュタイナー木: T を連結にする G の部分グラフ F

出力

$\sum_{v \in V_F} w(v)$ を最小化するシュタイナー木 F を求めよ

問題例



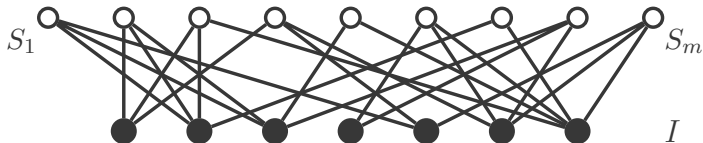
- シュタイナー点
- 端点

$\forall v \in V \setminus T : w(v) = 1 \rightarrow \text{頂点重み} = 3$

頂点重みシュタイナー木の近似困難性

定理

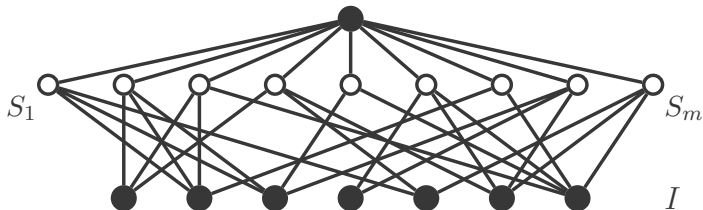
頂点重みシュタイナー木問題に対する α 近似アルゴリズム
 $\Rightarrow$ 集合被覆問題に対する α 近似アルゴリズム



頂点重みシュタイナー木の近似困難性

定理

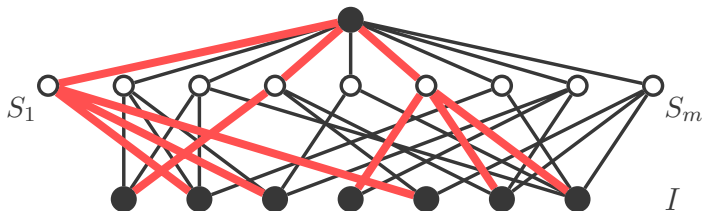
頂点重みシュタイナー木問題に対する α 近似アルゴリズム
 $\Rightarrow$ 集合被覆問題に対する α 近似アルゴリズム



頂点重みシュタイナー木の近似困難性

定理

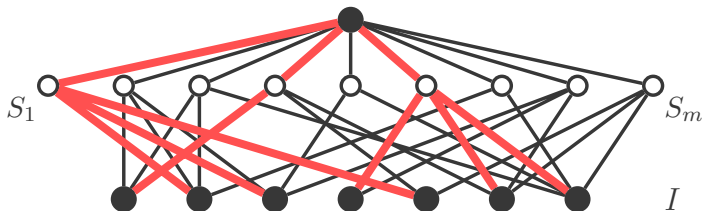
頂点重みシュタイナー木問題に対する α 近似アルゴリズム
 $\Rightarrow$ 集合被覆問題に対する α 近似アルゴリズム



頂点重みシュタイナー木の近似困難性

定理

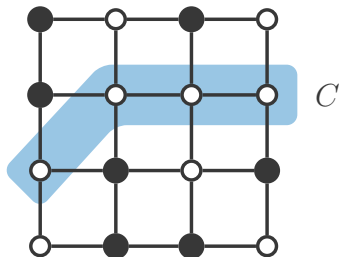
頂点重みシュタイナー木問題に対する α 近似アルゴリズム
 $\Rightarrow$ 集合被覆問題に対する α 近似アルゴリズム



頂点重みは複数の辺で共有できるので難しい

頂点重みシュタイナー木問題をどうやって解く？

ナイーブなアイデア: 集合被覆問題に帰着



頂点重みシュタイナー木問題

= 「Find 重み最小 $V' \subseteq V$ s.t. $V' \cap C \neq \emptyset$ for $\forall$ 頂点カット C 」

つまり,

- $I := \{\text{グラフの頂点カット}\}$
- $S_i := \text{グラフの頂点}$

貪欲アルゴリズムを単純に適用すると

貪欲アルゴリズム

- ポテンシャル $\phi :=$ 被覆した頂点カットの数
- $\text{density}(v) :=$
 $w(v)/(v \text{ を選択したときのポテンシャルの増分})$
効率性最小の頂点 v を選ぶことを繰り返す

問題点

- どうやって効率性最小の頂点を選ぶか？
- 近似比 $H_{\phi(V)} = \Omega(n)$ (頂点カットの数は 2^n ぐらい)

頂点重みシュタイナー木問題の解法

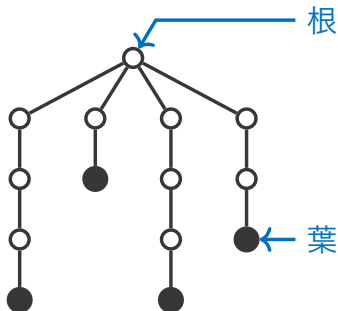
Klein & Ravi (1995) のアイデア

貪欲アルゴリズムを上手く適用する

- 単純に集合被覆には帰着しない
- 貪欲アルゴリズムの考え方をほぼそのまま利用する
 - ポテンシャル $\rightarrow k$ - 端点の数
 - 集合 $\rightarrow$ スパイダー

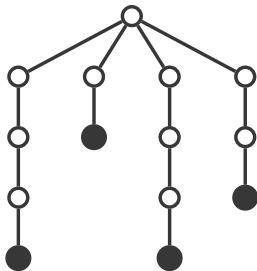
スパイダーとは？

- 根から葉へのパスの結合.
- 葉は全て端点. (葉以外の頂点は端点でもシュタイナー点でもどちらでもよい)
- 葉の数 ≥ 2 .
- 端点同士を結ぶ 1 本の辺も葉の数 = 2 のスパイダーと見なす (その場合はどちらかの端点を根とする).



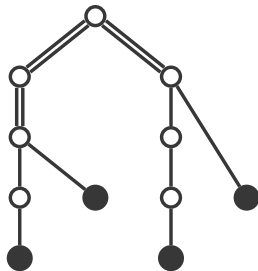
スパイダーと弱スパイダー

(強) スパイダー S



- 葉へのパスは、根以外は点素
- $w(S) := \sum_{v \in V_S} w(v)$

弱スパイダー S'



- パスが頂点・辺を共有してもいい
- $w(S') :=$ 根以外は重みを重複してカウント

スパイダー被覆アルゴリズム

- $\phi(G) := k - \text{端点の数}$
- 弱スパイダー S を選び縮約. 縮約でできた頂点は端点とする.
→ $\phi(G)$ は (S に含まれる端点数 -1) 増加

$$\text{density}(S) := \begin{cases} w(S)/(\text{Sの葉の数}) & \text{根が端点でない} \\ w(S)/(\text{Sの葉の数} + 1) & \text{根が端点} \end{cases}$$

密度の定義の分母をスパイダー S のポテンシャルと呼ぶ.
 S のポテンシャル $\neq S$ を縮尺したときの $\phi(G)$ の増分

スパイダー被覆アルゴリズム

スパイダー被覆アルゴリズム

1. while $\phi(G) < k - 1$ (i.e. 端点の数 > 1)
 - 密度最小の弱スパイダー S を計算
 - V_S を縮約し, 新しい頂点を端点とする
2. 計算された弱スパイダーに含まれる頂点と辺をすべて選択したものを解として出力

弱スパイダーをいかに見つけるか

密度の値は弱スパイダー S の重みと葉の数, 根が端点かどうかによりのみ依存.

密度最小の弱スパイダーの計算

1. For $\forall v \in V$

1.1 v から各端点への (頂点重み) 最短路を計算.

1.2 For $\forall k = 1, 2, \dots, |T|$

最短路を重みが小さい方から k 個を選び結合することで弱スパイダーを定義. ただし, v が端点でないなら $k \geq 2$ のみ考える.

2. ステップ 1.2 で計算した弱スパイダーのうち, 密度最小のものを出力.

近似比の解析

グラフ G における密度最小の弱スパイダー S は以下を満たす.

$$\frac{w(S)}{\phi(G/S) - \phi(G)} \leq \frac{2\text{OPT}}{k - 1 - \phi(G)}$$

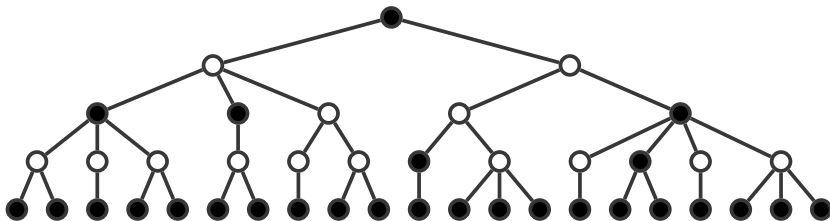
上の定理を認めれば, $O(H_{k-1}) = O(\log k)$ 近似.

スパイダー分解

スパイダー分解定理 [Klein, Ravi 1995]

任意のシュタイナー木は、次を満たすスパイダーの集合に分解できる.

- スパイダー同士は点素.
- 各端点はどれかのスパイダーの根もしくは葉.

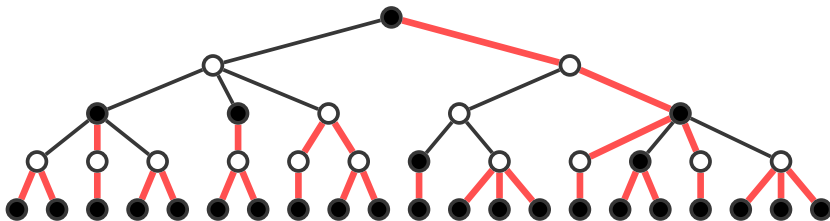


スパイダー分解

スパイダー分解定理 [Klein, Ravi 1995]

任意のシュタイナー木は、次を満たすスパイダーの集合に分解できる。

- スパイダー同士は点素.
- 各端点はどれかのスパイダーの根もしくは葉.



主定理の証明

S' を, 最適解を分解して得られるスパイダーのうち, 密度最小のものとする.

$$\begin{aligned}\frac{w(S)}{\phi(G/S) - \phi(G)} &\leq \frac{w(S)}{S \text{ のポテンシャル} - 1} && \because \text{葉はすべて端点} \\ &\leq \frac{2w(S)}{S \text{ のポテンシャル}} \\ &\leq \frac{2w(S')}{S' \text{ のポテンシャル}} && \because S \text{ は密度最小} \\ &\leq \frac{2 \sum_{S'' \in \text{スパイダー分解}} w(S'')}{\sum_{S'' \in \text{スパイダー分解}} S'' \text{ のポテンシャル}} && \because S' \text{ の定義} \\ &\leq \frac{2\text{OPT}}{k - 1 - \phi(G)}\end{aligned}$$

参考文献

1. U. Feige. A threshold of $\ln n$ for approximating set cover. Journal of the ACM 45, 634–652, 1998.
2. P.N. Klein, R. Ravi. A nearly best-Possible approximation algorithm for node-weighted Steiner trees. Journal of Algorithms 19(1), 104–115, 1995.
3. T. Fukunaga. Spider covering algorithms for network design problems. Combinatorial Optimization and Graph Algorithms: Communications of NII Shonan Meetings, 43–66, 2017.