グラフにおけるk-メディアン問題の高速な近似解法の研究

A Fast Approximate Algorithm for k-Median Problem On a Graph

情報認識学研究室
藤堂　佳介

背景

知り合いのネットワークやウェブのリンクなど世の中の様々な人や物のネットワークはグラフとして抽象化できる。

グラフ中の重要な頂点(中心)を探すという定式化は多くの応用を持つ。

中心性の指標:
- Degree-Centrality [Proctor & Loomis, 1951]：頂点の持つ次数による中心性
- Closeness-Centrality [Beauchamp, 1965]：他の頂点と平均してどのくらい近いかに基づく中心性
- Betweenness-Centrality [Freeman, 1977]：頂点を通る最短経路の数による中心性

世の中のネットワークは大きくなっている。

Facebookの全世界ユーザ数は23億8000万人、国内だけでも2600万人(2019年3月時点)

k-メディアン問題 [Hakimi, 1964,1965]

入力:無向グラフ$G = (V, E, \ell)$　　辺の重み関数$\ell: E \to (0, \infty)$

$|V| = n,\ |E| = m$

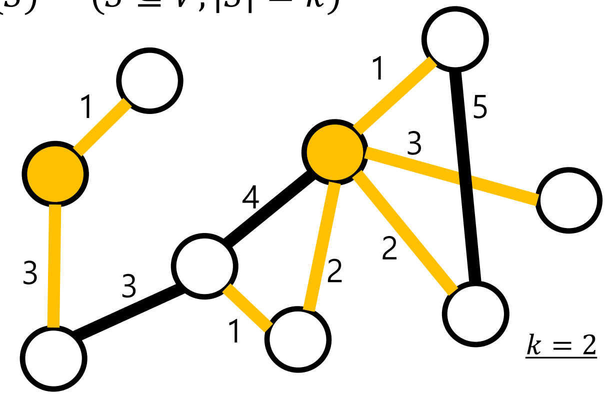
- 目的

各頂点からの移動コストの総和を最小にするk頂点集合(メディアン)の発見

$$Find\ S^* = \arg\min_S \sum_{v \in V} \min_{u \in S} L(u, v) = \arg\min_S \mathrm{NC}_G(S) \qquad (S \subseteq V, |S| = k)$$

$L(u, v)$: 頂点u, v を結ぶ最短経路のコスト

非中心性 $NC_G(S) = \sum_{v \in V} \min_{u \in S} L(u, v)$



$\mathrm{NC}_G(S) = 1 + 3 + 1 + 3 + 2 + 2 + 3 = 15$

<u>$k \geq 2$でNP困難[Kariv and Hakimi, 1979]</u>

従来研究

最もナイーブな解法では$O(n^{k+1})$の時間計算量がかかる(ただし$O(n^3)$はワーシャルフロイドのアルゴリズムで使用)

↳全ての組み合わせに対して、各頂点からの移動コストの総和を求める

分枝限定法を用いた厳密解法[Narula et al., 1977; Christfides, 1982]やメタヒューリスティクスを適用する手法[Chiyoshi and Galvao, 2000; Alp et al., 2003]が提案されている。

これらの従来手法は、まず全ての二頂点間の距離を求めることが必要。
(ワーシャルフロイドのアルゴリズム[Floyd, 1962]で$O(n^3)$時間、$O(n^2)$空間)

本研究の目的

グラフから全ての二頂点間の距離を求めるだけでも計算コストが重い。
(ワーシャルフロイドのアルゴリズム[Floyd, 1962]で$O(n^3)$時間、$O(n^2)$空間)

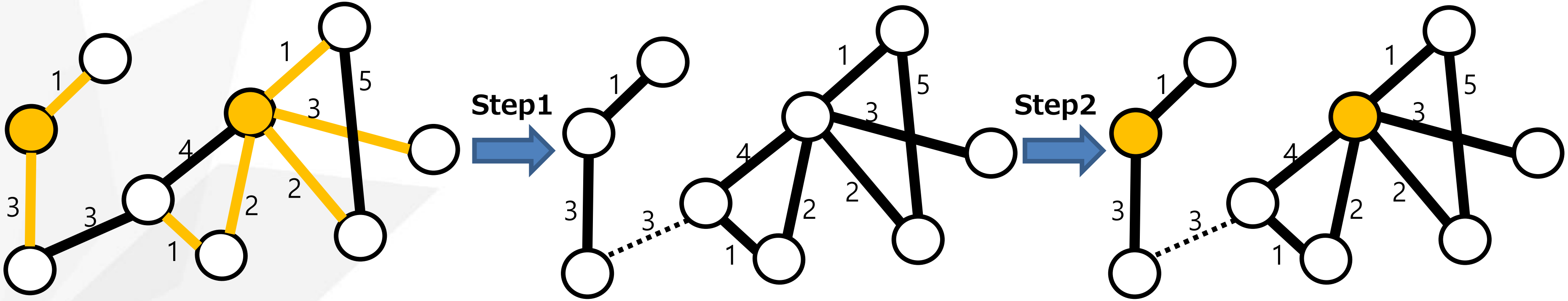
全体で$O(n^2)$の時間計算量の近似アルゴリズムの提案

従来手法[Maranzana, 1964]

以下の2つのStepを解の更新が止まるまで繰り返す。

Step1 現在の解Sからの距離によってグラフを分割。

Step2 各部分グラフで1-メディアンを求め、次の解候補S'を構成。



提案:アルゴリズムIDNC(Iteratively Decreasing NonCentrality)

[Todo, 2019]

初期解としてランダムにk頂点集合を選びSとし、以下のステップで解を更新する。

Step1 現在の解Sから最短経路森(k個の最短経路木の集合)を構成する。

Step2 各最短経路木の中心を求め、次の解候補S'とする。

S'によって解が更新される間、Step1,2を繰り返す。

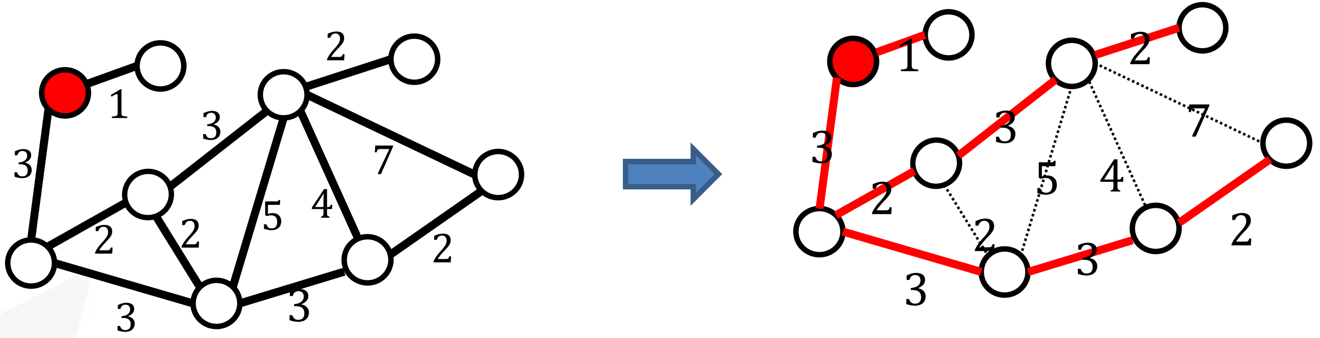
<u>一回の更新で必ず非中心性を減少させる</u>

提案:アルゴリズムIDNC(Iteratively Decreasing NonCentrality)

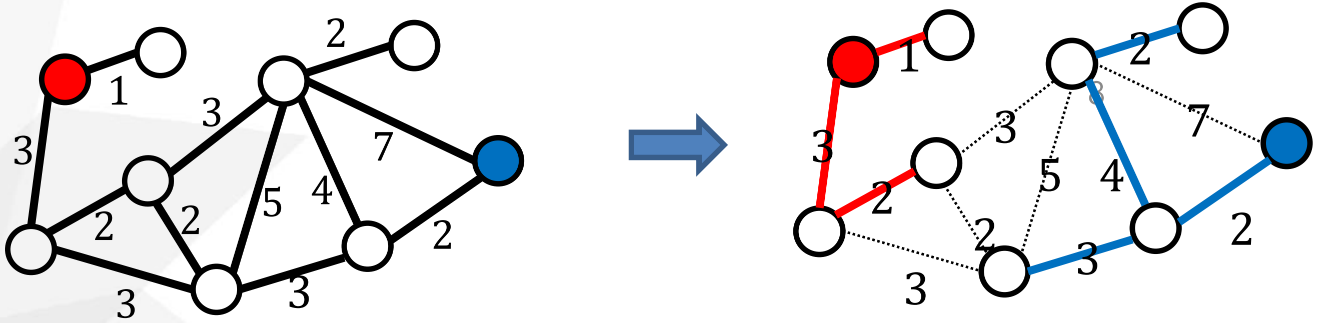
Step1 現在の解Sから最短経路森(k個の最短経路木の集合)を計算する

ダイクストラ法[Dijkstra, 1959]を任意の始点1個から始点k個へと拡張

$k = 1$



$k = 2$



提案:アルゴリズムIDNC(Iteratively Decreasing NonCentrality)

Step2 各最短経路木の中心を求める

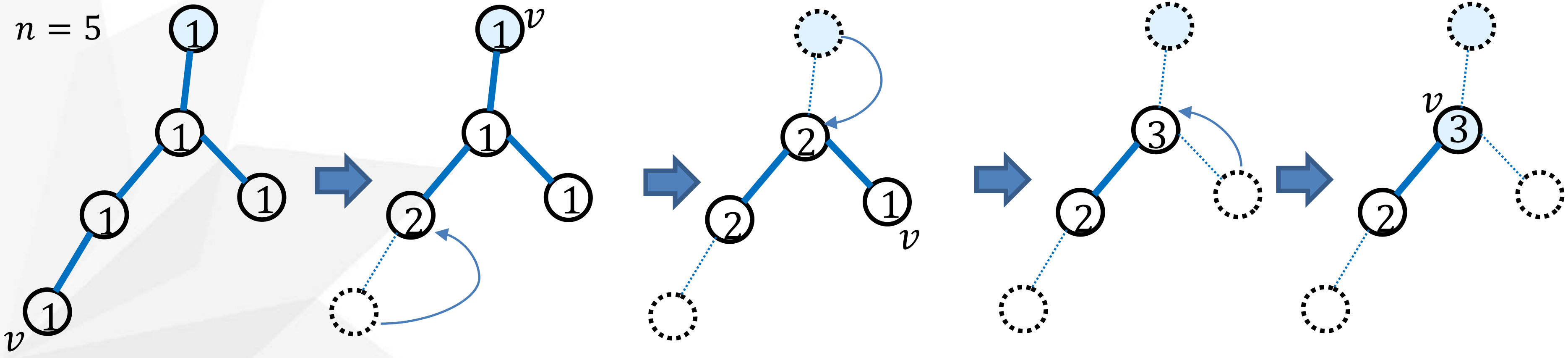
<u>GoldmanのアルゴリズムGoldman, 1971]</u>

→木に対し、以下の三つの手順で中心を探す

1. 木が一つの頂点vのみの場合、vが中心
2. 葉の一つの頂点vに着目し、vの重みが$n/2$を超える場合、vが中心
3. 頂点vの重みをつながる頂点に加え、vを木から削除する

中心が見つかるまで、1~3を繰り返す

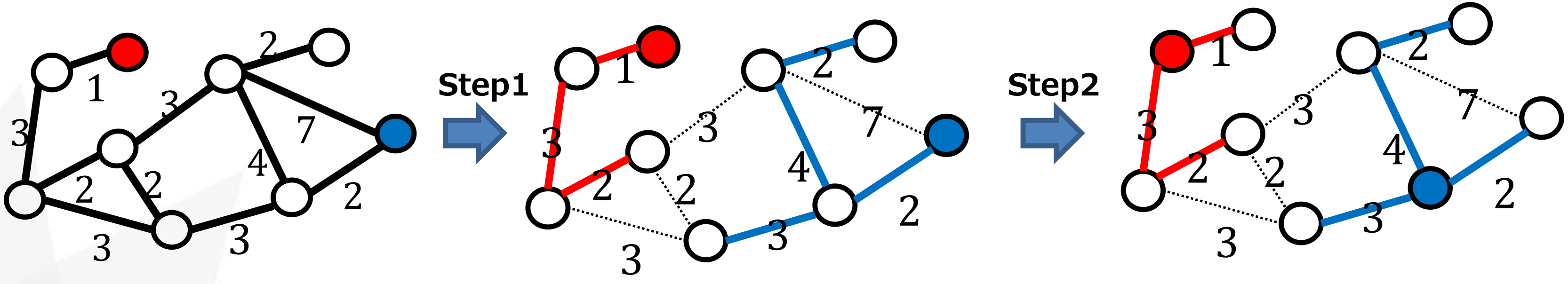
$n = 5$



提案:アルゴリズムIDNC(Iteratively Decreasing NonCentrality)

更新により非中心性が減少する(増加しない)

→常に$\mathrm{NC}_G(S') \leq \mathrm{NC}_G(S)$となる



非中心性NCは各最短経路木の根から他頂点への移動コストの総和

$\mathrm{NC}_G(S) = \underbrace{1+4+6}_{11}+\underbrace{2+5+6+8}_{21} = 32$

$\mathrm{NC}_G(S') = \underbrace{1+3+5}_{9}+\underbrace{2+3+4+6}_{15} = 24$

Step2によって各最短経路木内の移動コストの和は減少する(もしくは不変)

よって$\mathrm{NC}_G(S') \leq \mathrm{NC}_G(S)$となる

提案:アルゴリズムIDNC(Iteratively Decreasing NonCentrality)

各ステップの計算量(頂点数$|V| = n$, 辺数$|E| = m$)

Step1

ダイクストラのアルゴリズムによって$O(m + n\log n)$の時間計算量、$O(m)$の空間計算量

Step2

Goldmanのアルゴリズムによって$O(n)$の時間、空間計算量

アルゴリズム全体で、

$O\big(r(m + n\log n)\big)$の時間計算量、 $O(m)$の空間計算量　r:Stepの繰り返し回数

実験

実験１：小規模データセットによる計算時間・近似比比較

実験２：実験1より大規模なデータセットによる計算時間・非中心性比較

実験1

使用データセット

OR-Library[Beasley, 1990]のpmedデータセット

- 頂点数$n = \{100, 200, \dots, 900\}$
- 辺数$m = n^2/50$本 (ランダムに与えられており、重みは1～100の整数値)
- 各データは異なるグラフであり、与えられたkに対する最適解が既知

	頂点数n	辺数m	k
Pmed 1, 2, 3, 4, 5	100	200	5, 10, 10, 20, 33
Pmed 6, 7, 8, 9, 10	200	800	5, 10, 20, 40, 67
Pmed 11, 12, 13, 14, 15	300	1800	5, 10, 30, 60, 100
Pmed 16, 17, 18, 19, 20	400	3200	5, 10, 40, 80, 133
Pmed 21, 22, 23, 24, 25	500	5000	5, 10, 50, 100, 167
Pmed 26, 27, 28, 29, 30	600	7200	5, 10, 60, 120, 200
Pmed 31, 32, 33, 34	700	9800	5, 10, 70, 140
Pmed 35, 36, 37	800	12800	5, 10, 80
Pmed 38, 39, 40	900	16200	5, 10, 90

実験1

実験条件

初期解：ランダムに選択したk頂点集合

各データに対し、初期解を変え1000回実行

kは各データセットにおいて与えられた値で実行

比較

比較手法[Maranzana, 1964]

部分グラフの1-メディアンにより、解の更新を行う

ワーシャル・フロイドのアルゴリズム(WF)

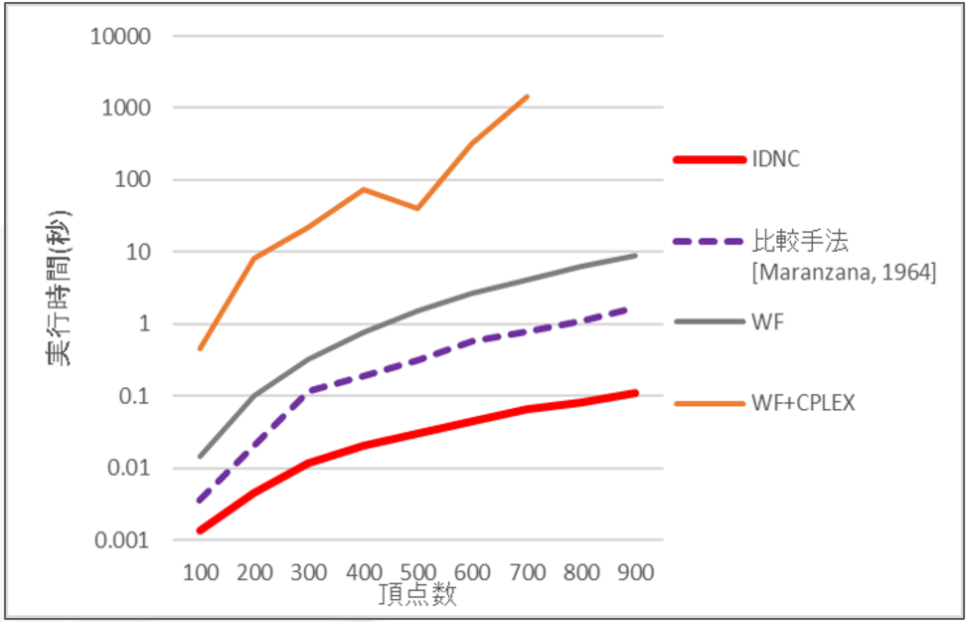
全ての二頂点間の距離を求めるのにかかる時間を測定

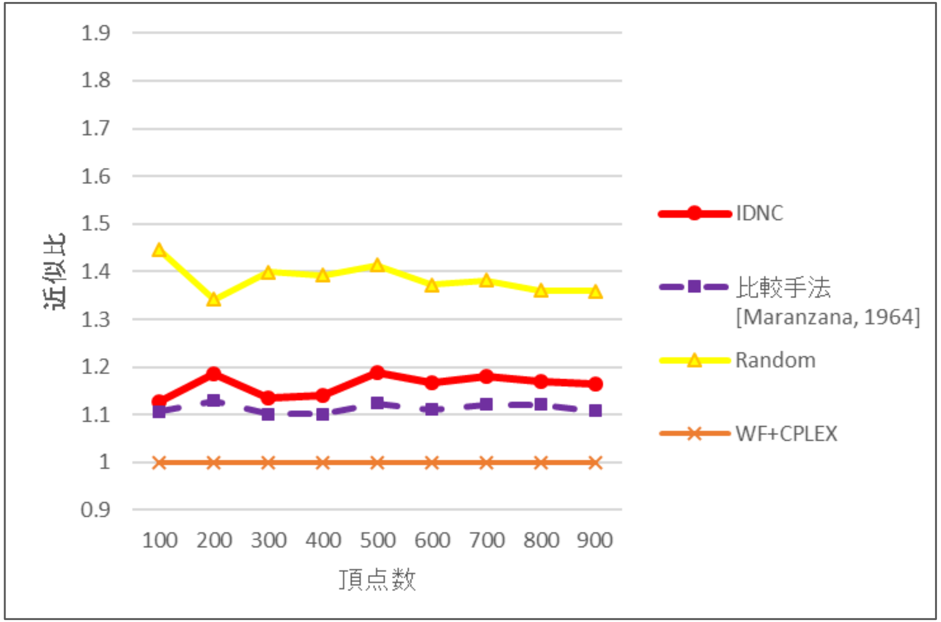
WF+CPLEX

全ての二頂点間の距離を求めた後、最適化問題ソルバーCPLEXを用いて解く

実験1 頂点数の変化による結果$(k = 5)$

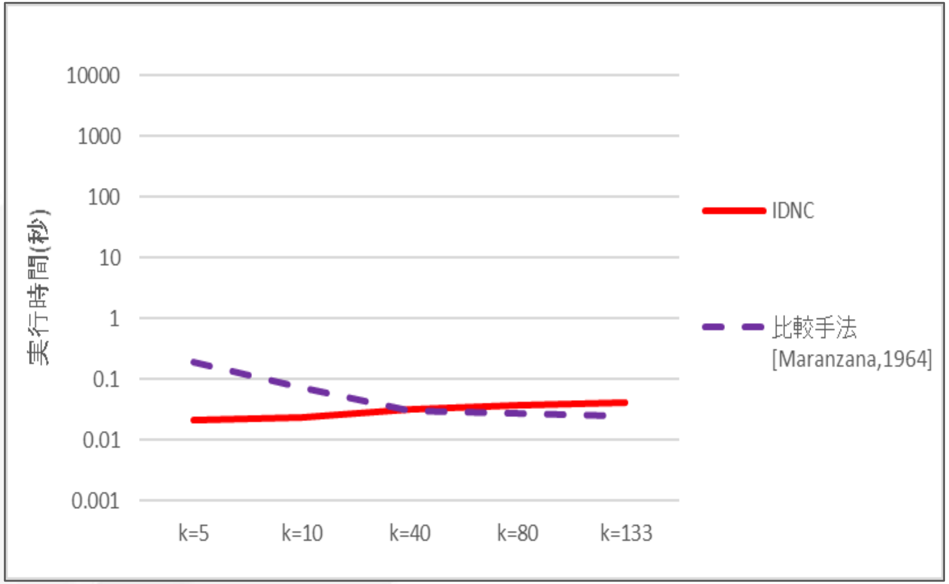
提案手法の時間計算量：$O\big(r(m + n\log n)\big)$　頂点数$|V| = n$, 辺数$|E| = m$, r:Stepの繰り返し回数

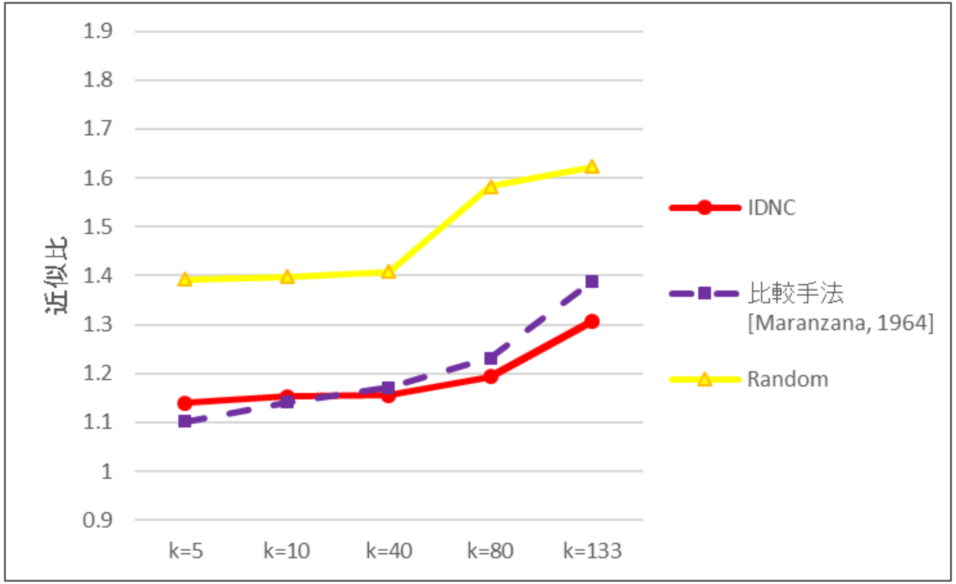




実験1 kの変化による結果(頂点数$n = 400$)

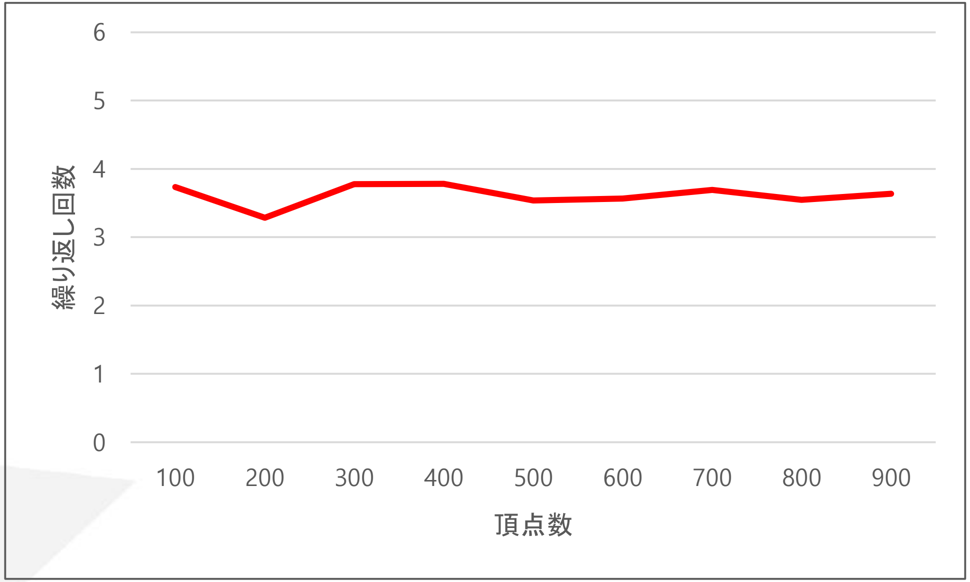
提案手法の時間計算量：$O(r(m + n\log n))$　頂点数$|V| = n$, 辺数$|E| = m$, r:Stepの繰り返し回数





実験1 繰り返し回数

提案手法の時間計算量：$O\big(r(m+n\log n)\big)$　頂点数$|V|=n$, 辺数$|E|=m$, r:Stepの繰り返し回数



実験2

使用データセット

□疎なランダムグラフ

頂点数$n = \{1000, 10000, 50000, 100000\}$　辺数$m = n^2/50$本

□Barabási-Albertモデルに基づくグラフ

→スケールフリー性を持つ

頂点数$n = \{1000, 10000, 50000, 100000\}$　辺数mは疎なランダムグラフとほぼ同数

□密なランダムグラフ

頂点数$n = \{1000, 10000, 50000\}$　辺数$m = 0.7 * \frac{n(n-1)}{2}$ (完全グラフの辺数の7割)

辺の重みはいずれも1~100の整数値

実験2

実験条件

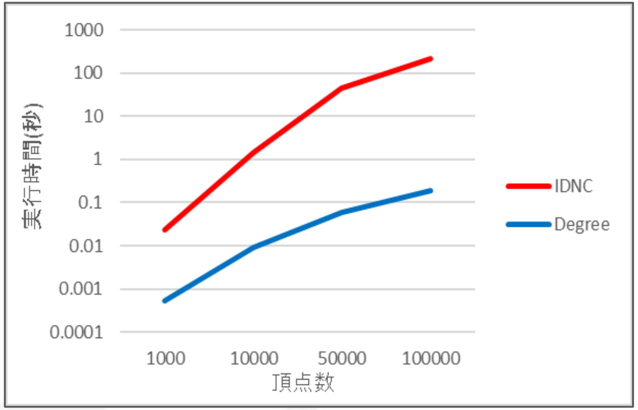
初期解：ランダムに選択したk頂点集合
各データに対し、初期解を変え50回実行
全てのデータに対し、$k = 5$で固定

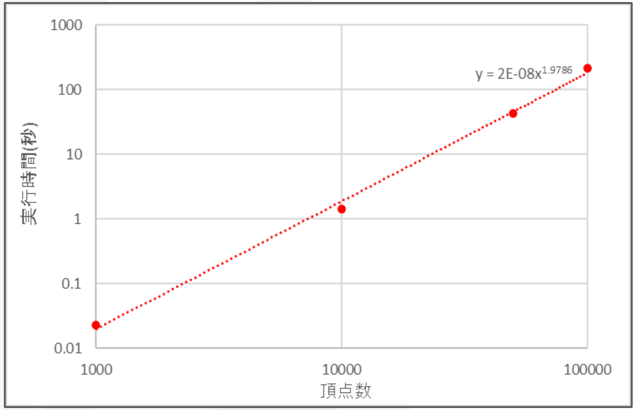
比較

Degree-Centrality [Proctor & Loomis, 1951]
頂点の持つ次数による中心性の指標

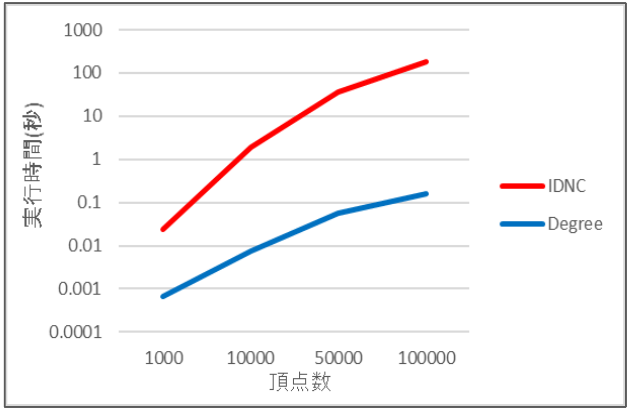
実験2 実行時間

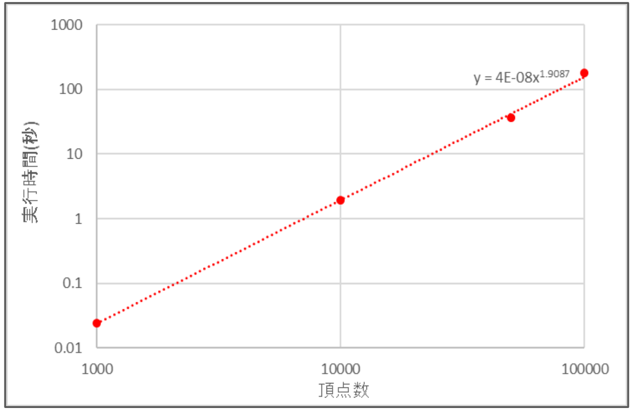
疎なランダムグラフ



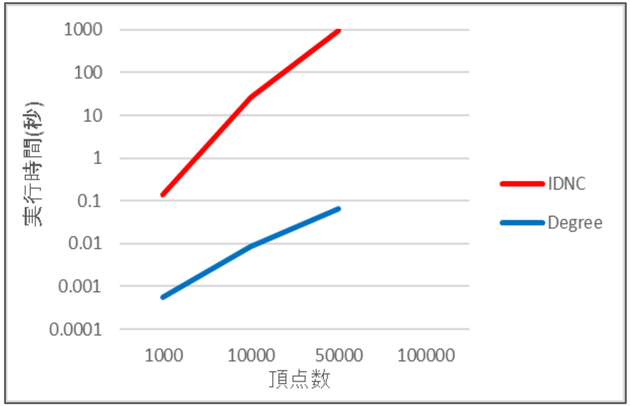


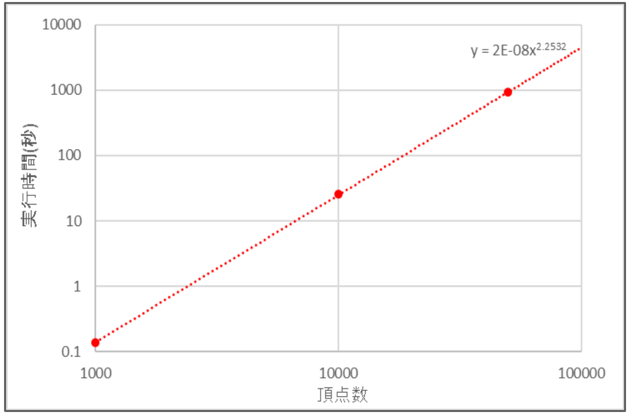
Barabási-Albertモデル





密なランダムグラフ

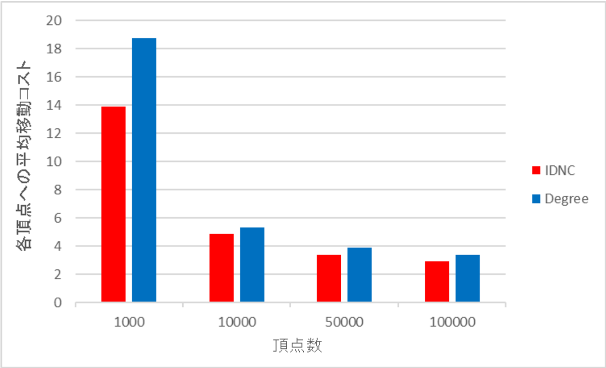




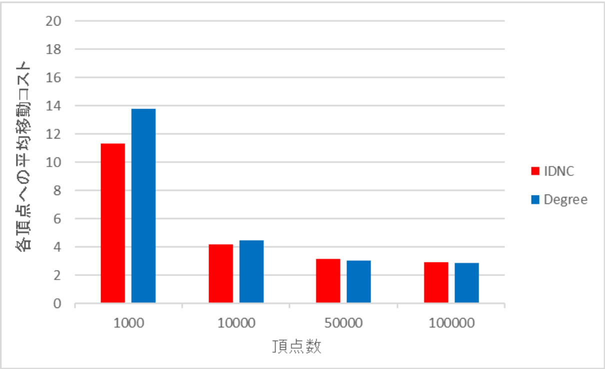
実験2 非中心性

$$\text{各頂点への平均移動コスト} = \frac{\text{非中心性（移動コストの総和）}}{\text{頂点数}}$$

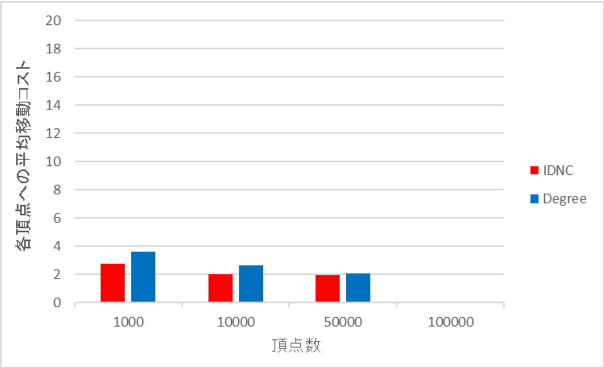
疎なランダムグラフ



Barabási-Albertモデル

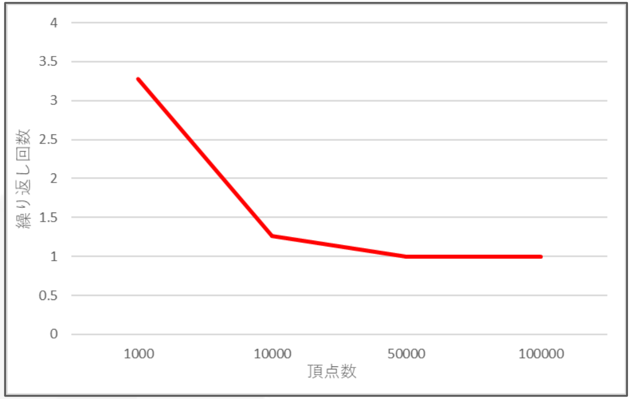


密なランダムグラフ

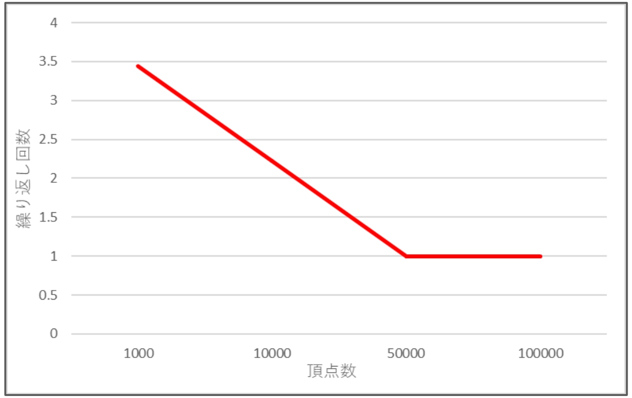


実験2 繰り返し回数

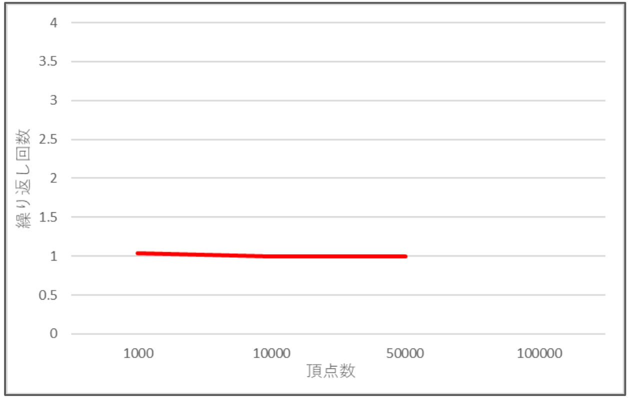
疎なランダムグラフ



Barabási-Albertモデル



密なランダムグラフ



結論

✓グラフにおけるk－メディアン問題に対して、 $O\left(r(m+n\mathrm{log}n)\right)$の時間計算量、O($m$)の空間計算量の近似アルゴリズムを提案した。

結果として

反復回数rは、頂点数の増加によらずほぼ定数かつnに比べて小さい

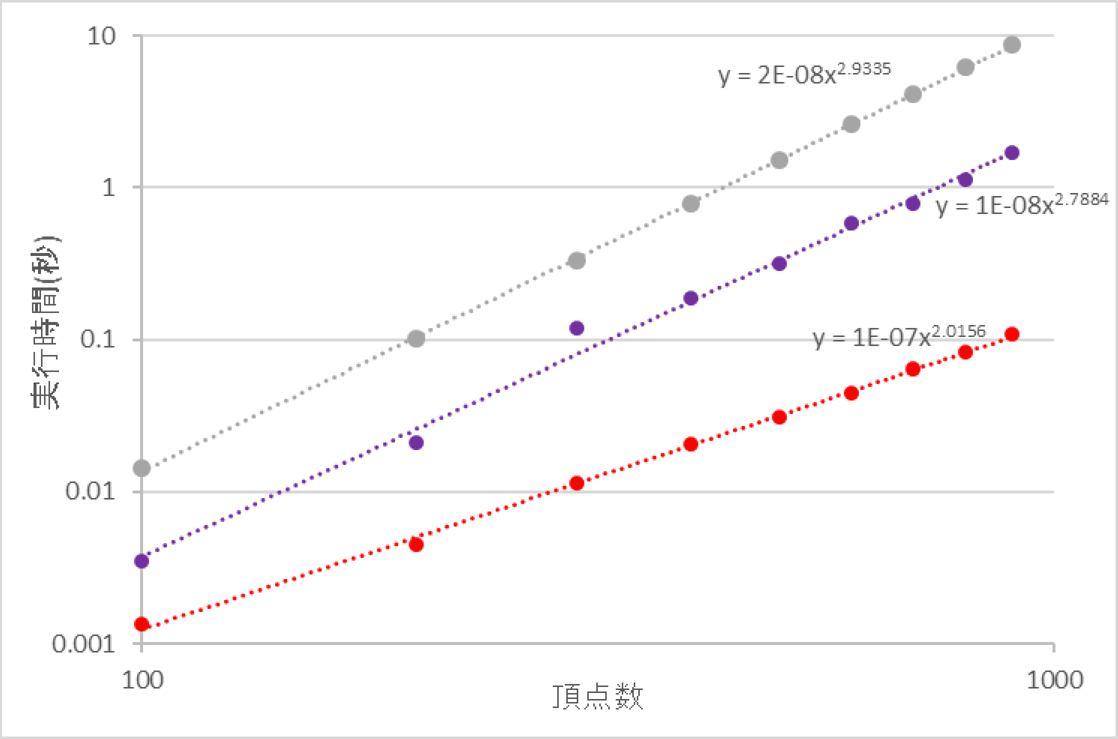
→全体として$O\left(n^2\right)$の時間計算量

(様々な性質を持った様々なサイズのグラフで検証が必要)

発表業績

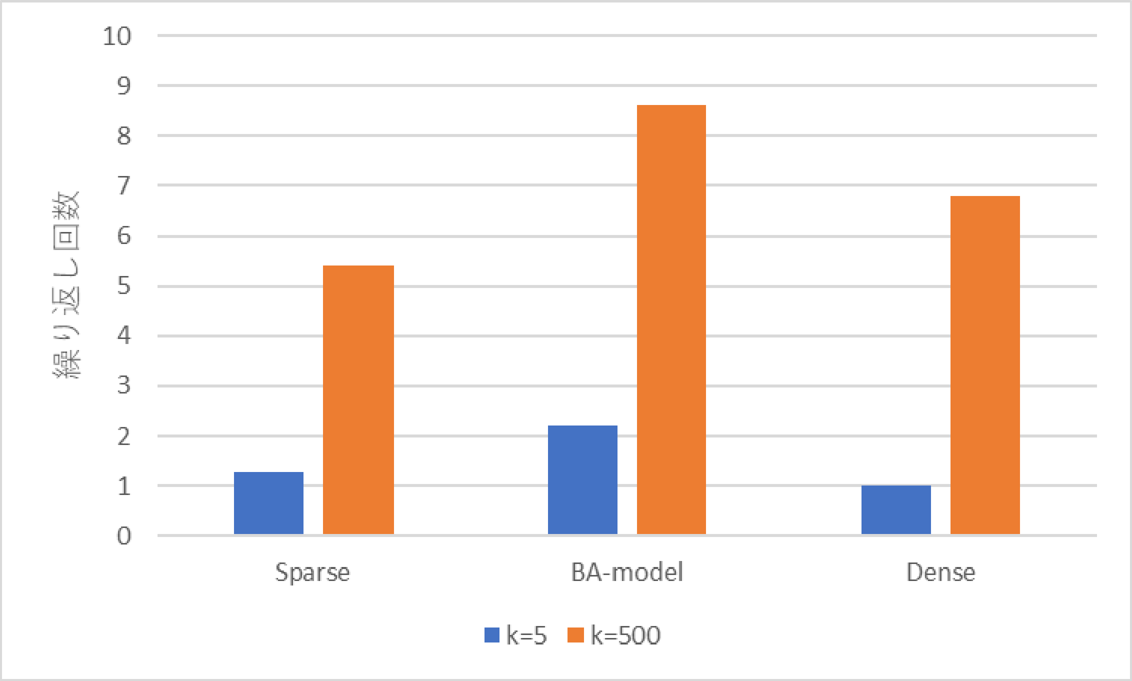
K Todo, A Nakamura, M Kudo, “A Fast Approximate Algorithm for k-Median Problem on a Graph.” , 15th International Workshop on Mining and Learning with Graphs(2019), Anchorage, Alaska.

付録



頂点数400のグラフに対する実行時間の両対数グラフ

付録



頂点数10000のグラフにおけるkによる繰り返し回数の差