

# Paper Reading:

## **Streaming Graph Partitioning for Large Distributed Graphs**

Isabelle Stanton  
University of California Berkeley  
Berkeley, CA  
isabelle@eecs.berkeley.edu

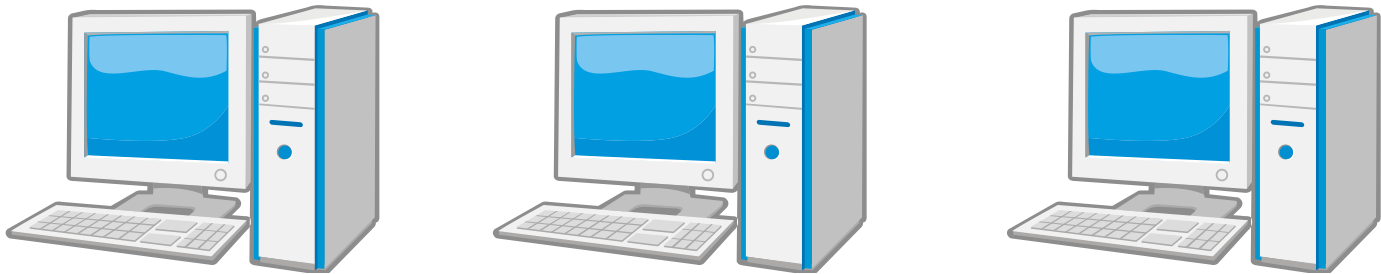
Gabriel Kliot  
Microsoft Research  
Redmond, WA  
gkliot@microsoft.com

from KDD 2012

speaker: Kazuhiro Inaba

# 内容

- 巨大グラフは1台のメモリには乗らない
- 複数台に分割して格納・処理する必要がある



- どのようにグラフを分割するのが良いか？

# 問題

- 入力

(実験での値)

- $G = (V, E)$  有向or無向グラフ

$|V| \sim 40M, |E| \sim 1G$

- $k$  マシンの台数

$2 \sim 100$

- $\epsilon$  許容アンバランス度

$2 \sim 5\%$  (?)

- 出力

- $V_1, V_2, \dots, V_k$   $V$ のdisjointな分割

$|V_i| \leq C = (1+\epsilon)|V|/k$  or  $\sum \text{deg}(V_i) \leq C = (1+\epsilon)(\sum \text{deg}(V))/k$

# 問題

- 入力

(実験での値)

- $G = (V, E)$  有向or無向グラフ

$|V| \sim 40M, |E| \sim 1G$

- $k$  マシンの台数

$2 \sim 400$

- $\varepsilon$  許容アンバランス度

$2 \sim 5\%$  (?)

- 出力

- $V_1, V_2, \dots, V_k$   $V$ のdisjointな分割

$|V_i| \leq C = (1+\varepsilon)|V|/k$  or  $\sum \text{deg}(V_i) \leq C = (1+\varepsilon)(\sum \text{deg}(V))/k$

評価基準:

$\{(u,v) \in E \mid u \in V_a, v \in V_b, a \neq b\}$  が少ないほど良い

# Streaming

- 入力
  - $G = (V, E)$  有向or無向グラフ
  - $k$  マシンの台数
  - $\epsilon$  許容アンバランス度
- 出力
  - $V_1, V_2, \dots, V_k$   $V$ のdisjointな分割



- 入力は  $v \in V$  が1つずつ順に渡される
  - 各  $v$  を受け取るたびに、(ほぼ)すぐにその partition 番号を出力しなければならない
  - $v$  の近傍に関する情報のみ使ってよい

# この論文の内容

- ストリームの順番
  - 3 種類 (DFS, BFS, RANDOM)
- パーティション番号を割り当てるヒューリスティクス
  - 8 種類 (後述)

の組み合わせで評価実験

# ヒューリスティクス

1. Balanced : 現在最小のパーティションに

$$ind = \arg \min_{i \in [k]} \{|P^t(i)|\}$$



2. Chunking : 端から順に詰める

$$ind = \lceil t/C \rceil$$



3. Hashing : ランダム

# ヒューリスティクス

## 4. Deterministic Greedy

- 近傍に多いパーティションを選ぶ

$$ind = \arg \max_{i \in [k]} \{ |P^t(i) \cap \Gamma(v)| w(t, i) \}$$

ただし既に  $|P(i)|=C$  な  $i$  を除く  
(と思われる)(以下同様)

where

- 4-1.  $w(t, i) = 1$  unweighted
- 4-2.  $w(t, i) = 1 - |P^t(i)|/C$  linear
- 4-3.  $w(t, i) = 1 - \exp(-|P^t(i)|/C)$  exponential



# ヒューリスティクス

## 5. Randomized Greedy

- 近傍に多いパーティションに行く確率を高める

$$Pr(i) = |P^t(i) \cap \Gamma(v)|w(t, i) / Z$$

where

- 5-1.  $w(t, i) = 1$  **unweighted**
- 5-2.  $w(t, i) = 1 - |P^t(i)|/C$  **linear**
- 5-3.  $w(t, i) = 1 - \exp(-|P^t(i)|/C)$  **exponential**

# ヒューリスティクス

## 6. Triangle

–  $v$ を足したら三角形が最も増えるパーティション

$$\arg \max_{i \in [k]} \left\{ \frac{|E(P^t(i) \cap \Gamma(v), P^t(i) \cap \Gamma(v))|}{\binom{|P^t(i) \cap \Gamma(v)|}{2}} w(t, i) \right\}$$

where

– 6-1.  $w(t, i) = 1$  **unweighted**

– 6-2.  $w(t, i) = 1 - |P^t(i)|/C$  **linear**

– 6-3.  $w(t, i) = 1 - \exp(-|P^t(i)|/C)$  **exponential**

# ヒューリスティクス

## 7. Balance Big

- 閾値より degree の高いノードは “Balanced”

$$ind = \arg \min_{i \in [k]} \{|P^t(i)|\}$$

- 閾値より degree の低いノードは  
Deterministic Greedy

$$ind = \arg \max_{i \in [k]} \{|P^t(i) \cap \Gamma(v)|w(t, i)\}$$

# (Buffering) ヒューリスティクス

C 個まで入力ノードを貯めておくことを許す

## 8. Prefer Big

バッファに C 個のノード ID を読み込む

loop {

if バッファに閾値より degree の高いノードがある then

そのノードを “Balanced” で分配

バッファに 1 個新たに読み込む

else

C個のノードを “Deterministic Greedy” で分配

バッファに新たに C 個のノード ID を読み込む

}

# (Buffering) ヒューリスティクス

C 個まで入力ノードを貯めておくことを許す

## 9. Avoid Big

C個のバッファの範囲で低degreeノードを先に処理。“Deterministic Greedy”で

## 10. Greedy EvoCut

C個のバッファの範囲でクラスタリング  
by EvoCut [Andersen&Peres STOC'09]  
クラスタ単位で “Deterministic Greedy”

# 実験

Finite  
Element  
Mesh

Social

Watts &  
Strogatz  
Nature  
1998

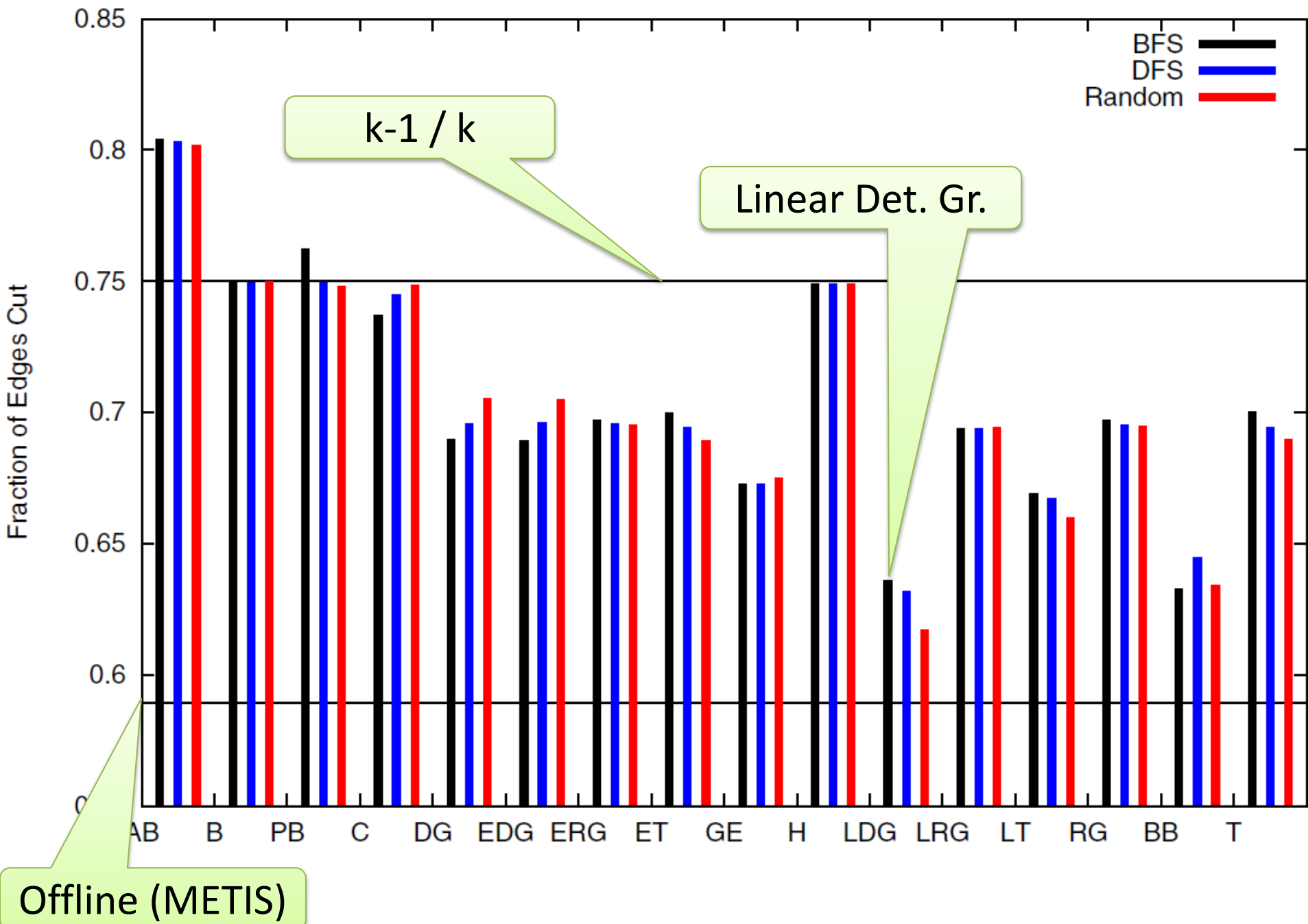
Holme&Kim  
Phys.Rev.E  
2002

Social

Name	$ V $	$ E $	Type	Source
3elt	4720	13,722	FEM	[5]
4elt	15606	45,878	FEM	[5]
vibrobox	12,328	165,250	FEM	[5]
celegans	297	2,148	Protein	[30]
astro-ph	18,772	396,160	Citation	[27]
Slashdot0811	77,360	504,230	Social	[22]
wiki-Vote	7,115	99,291	Social	[22]
Marvel	6,486	427,018	Social	[7]
web-ND	325,729	1,497,134	Web	[22]
BA	1,000	9,900	Synth.	[13]
BA	10,000	129,831	Synth.	[13]
BA	50,000	1,249,375	Synth.	[13]
RMAT	1,000	9,175	Synth.	[23]
RMAT	10,000	129,015	Synth.	[23]
RMAT	50,000	1,231,907	Synth.	[23]
WS	1,000	5,000	Synth.	[30]
WS	10,000	120,000	Synth.	[30]
WS	50,000	3,400,000	Synth.	[30]
PL	1,000	9,878	Synth.	[18]
PL	10,000	129,763	Synth.	[18]
PL	50,000	1,249,044	Synth.	[18]
LiveJournal	$4.6 \cdot 10^6$	$77.4 \cdot 10^6$	Social	[26]
Twitter	$41.7 \cdot 10^6$	$1.468 \cdot 10^9$	Social	[21]

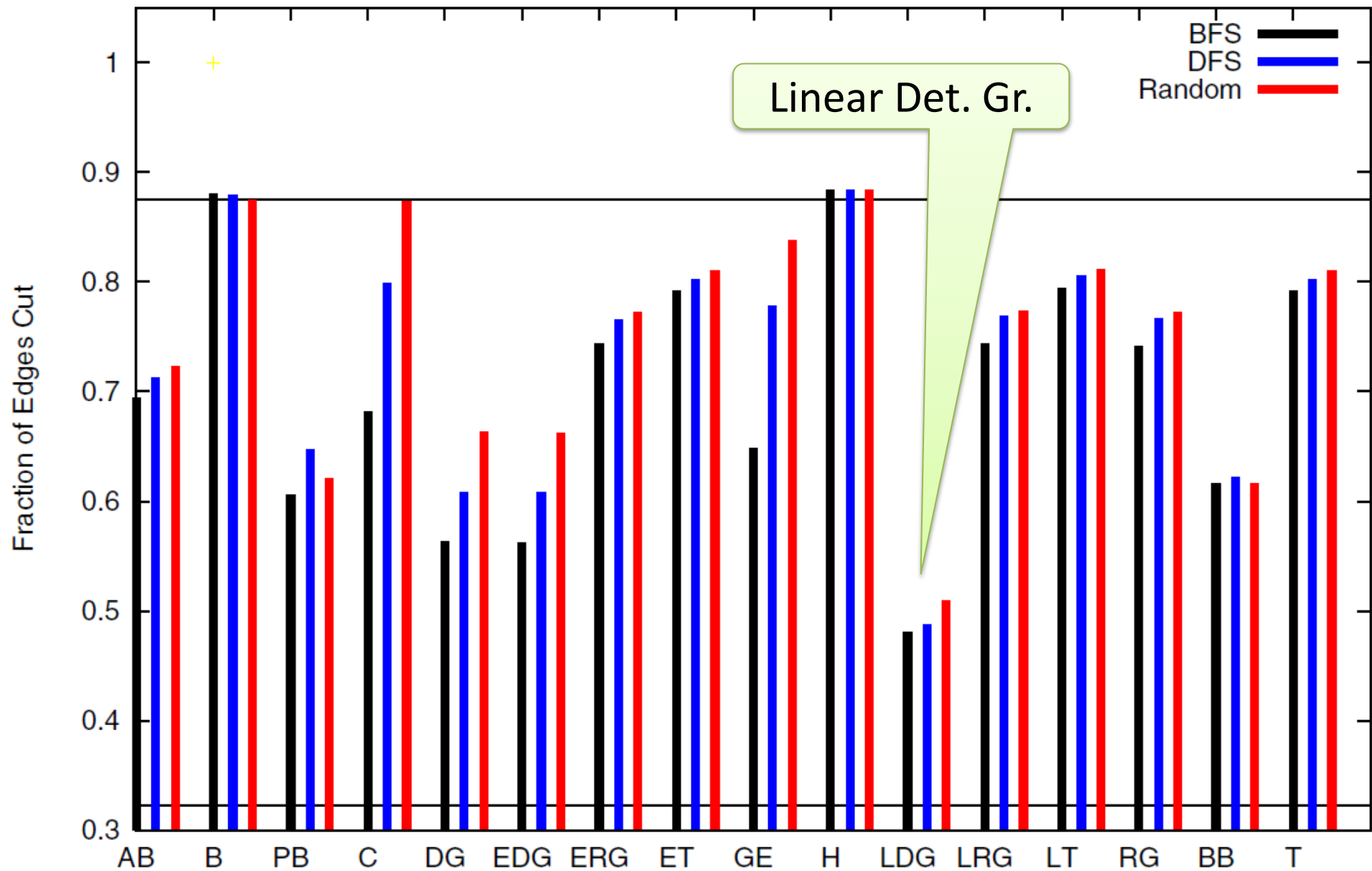
# 実験：ヒューリスティクス間の比較

PL1000 results, 4 partitions

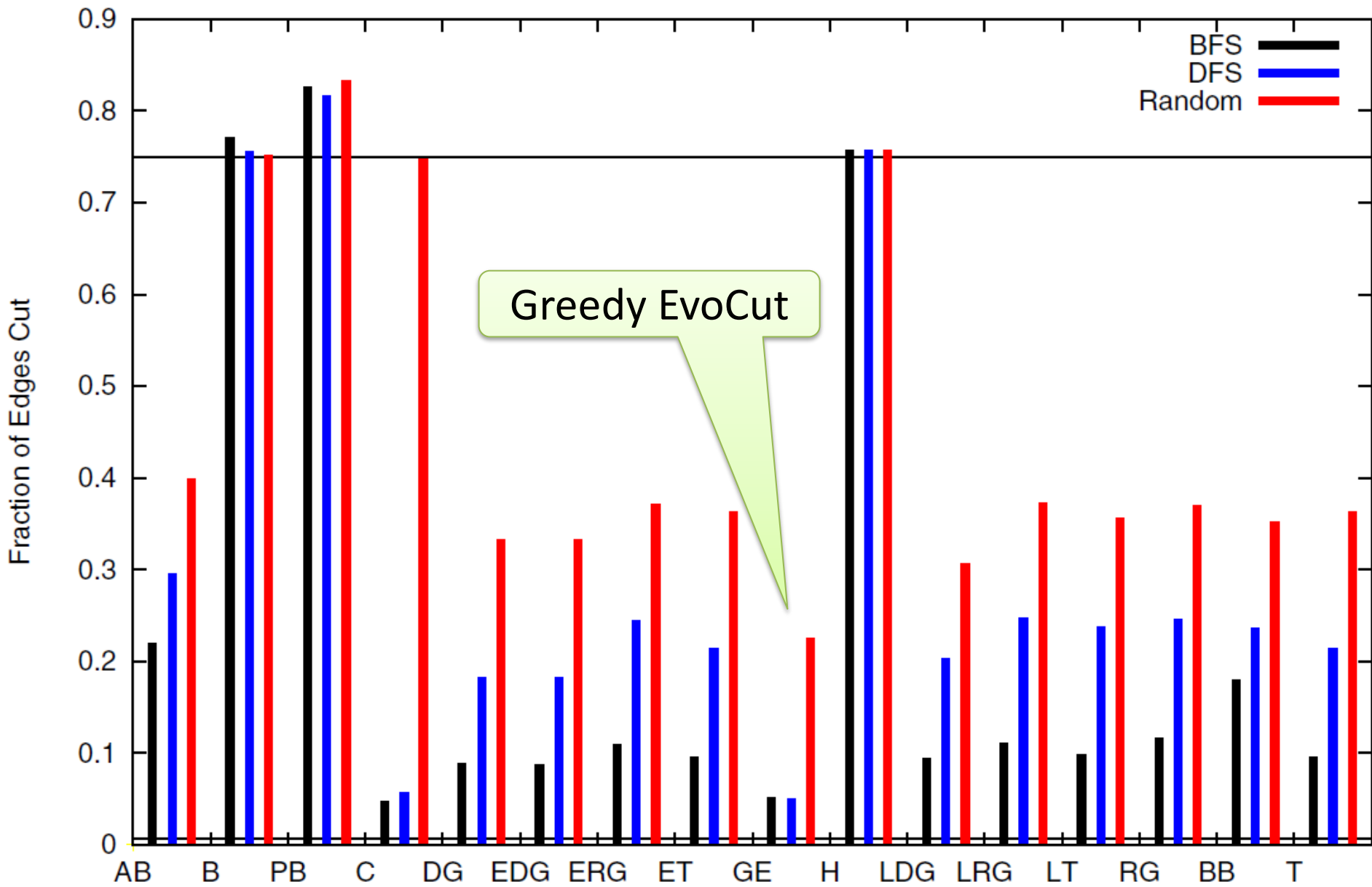




Marvel results, 8 partitions



4elt results, 4 partitions



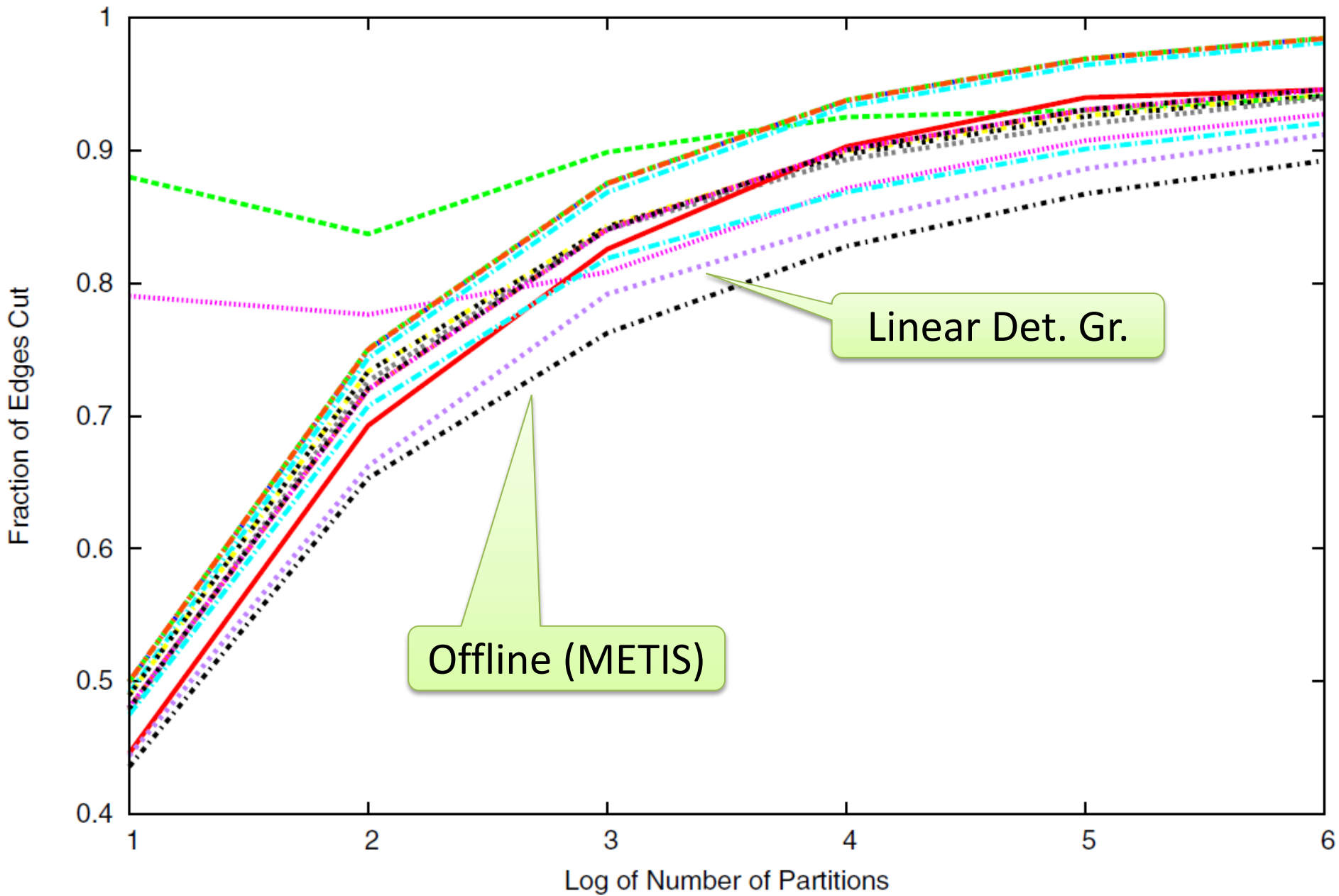
$\left(\frac{\text{random-heuristic}}{\text{random-METIS}}\right)$  の全データセットでの平均

Heuristic		<i>BFS</i>	<i>DFS</i>	<i>Random</i>
Avoid Big	AB	-27.3	-38.6	-46.4
Balanced	B	-1.5	-1.3	-0.2
Prefer Big	PB	-9.5	-18.6	-23.1
Chunking	C	37.6	35.7	0.7
Deterministic Greedy	DG	57.7	54.7	45.4
Exp. Det. Greedy	EDG	59.4	56.2	47.5
Exp. Rand. Greedy	ERG	45.6	45.6	38.8
Exp. Triangles	ET	50.7	49.3	41.6
Greedy EvoCut	GE	60.3	58.6	43.1
Hashing	H	-1.9	-2.1	-1.7
Linear Det. Greedy	LDG	76	73	75.3
Linear Rand. Greedy	LRG	46.4	44.9	39.1
Linear Triangles	LT	55.4	54.6	49.3
Randomized Greedy	RG	45.5	44.9	38.7
Balance Big	BB	67.8	68.5	63.3
Triangles	T	49.7	48.4	40.2



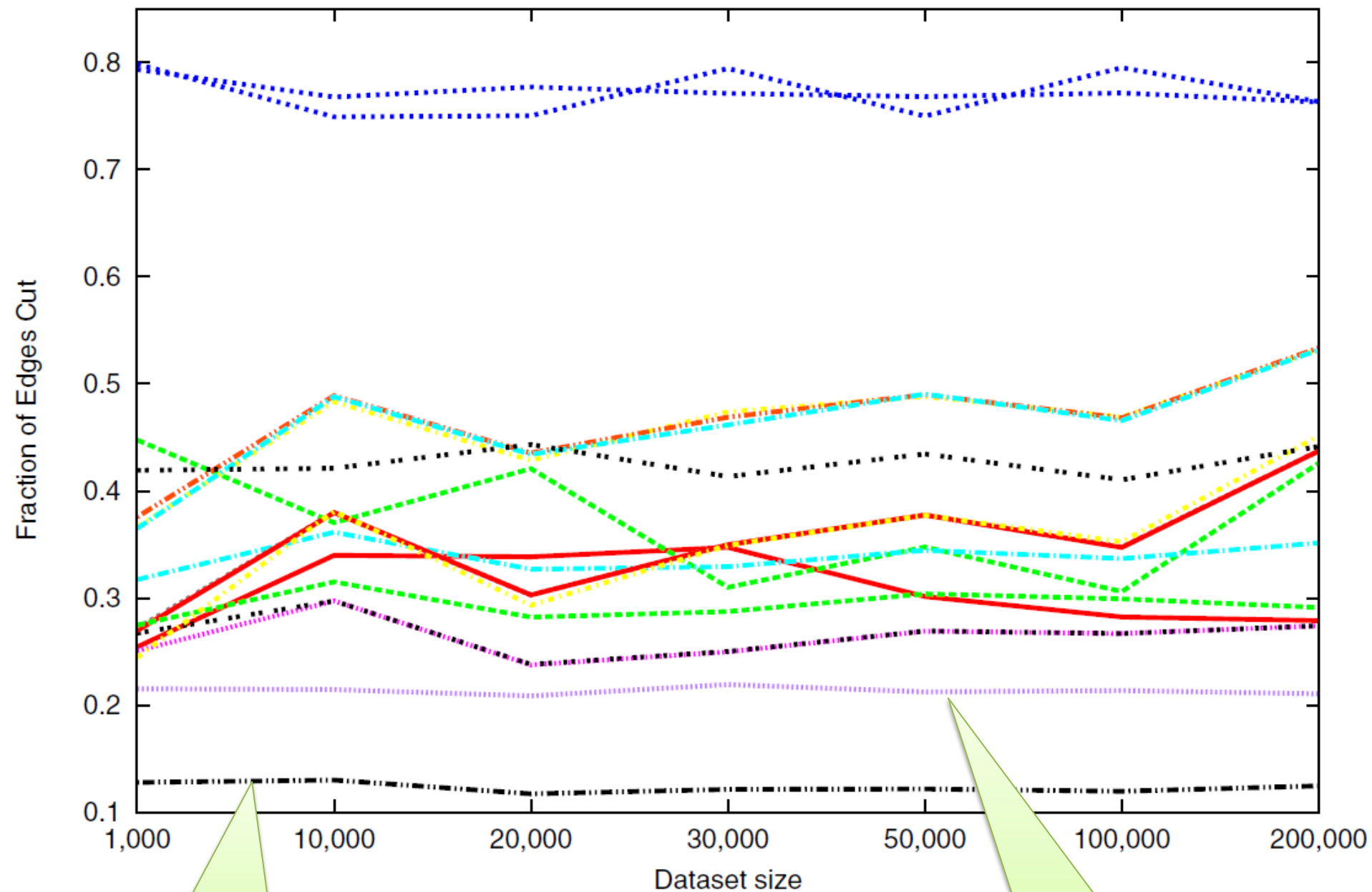
# 実験：分割数・グラフサイズの影響

Heuristic performance over the PL Graph with 50,000 nodes



Heuristic performance over Watts-Strogatz Graphs

4 partition



Offline (METIS)

Linear Det. Gr.

**実験：実際の計算速度への効果**

# PageRank 計算のパフォーマンス

- SPARK Framework ( <http://spark-project.org/> )
  - Naive実装と Partition 間の通信を減らす実装
- $|E|=77M$   $k=100$   $\epsilon=2\%$

0.99

0.61

	LJ Hash	LJ Streamed
Naïve PR Mean	296.2s	181.5s
Naïve PR STD	5.5 s	2.2 s
Combiner PR Mean	155.1 s	110.4 s
Combiner PR STD	1.5 s	0.8 s



# PageRank 計算のパフォーマンス

- $|E|=1.3G$   $k=400$   $\epsilon=2\%$

0.997

0.913

	Twitter Hash	Twitter Streamed
Naïve PR Mean	1199.4 s	969.3 s
Naïve PR STD	81.2 s	16.9 s
Combiner PR Mean	599.4 s	486.8 s
Combiner PR STD	14.4 s	5.9 s

# Further Reading

- <http://arxiv.org/abs/1212.1121>

Random order + Linear Deterministic Greedy が  
Random order + Linear Random Greedy  
に勝る理由の分析をしている

– [McSherry 2001] のモデルで LDG はクラスタを復元するが LRG はしない

- [McSherry 2001]

- Erdos-Reny の拡張。ノードが  $k$  色に彩色されている
- 辺を貼る確率は  $P = k * k$  matrix で与える
- このdraftでは  $p = P_{ii} > P_{ik} = q$  のケースを解析

# まとめ

- 著者曰く
  - BFS がよい
  - Linear Deterministic Greedy がよい
- 感想
  - ...