Eccentricity Heuristics through Sublinear Analysis Lenses

Tal Wagner

MIT

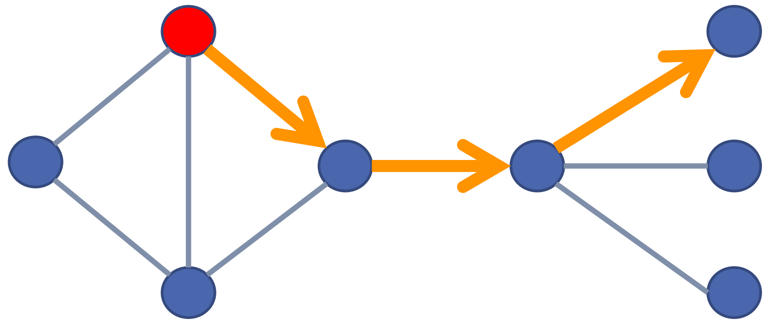
Graph Eccentricities

- Let $G(V, E)$ by a graph
- Shortest-path metric: $\Delta: V \times V \rightarrow \mathbb{R}$
- **<u>Eccentricities:</u>**

$$\boldsymbol{e}(v) = \max_{u \in V} \Delta(v, u)$$

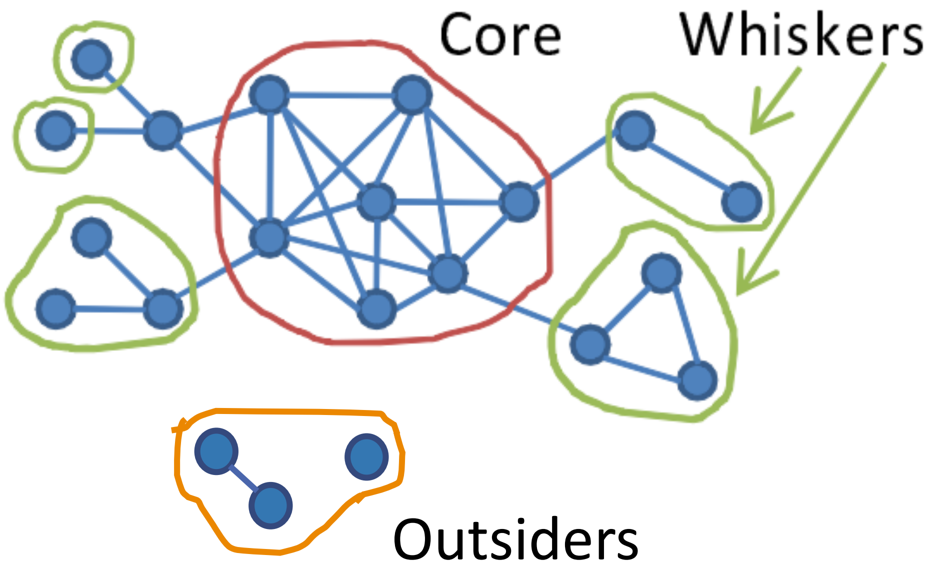
- Max $\boldsymbol{e}(v)$ = ***diameter***; Min $\boldsymbol{e}(v)$ = ***radius***
  90th percentile $\boldsymbol{e}(v)$ = "*effective diameter*" (excludes outliers)

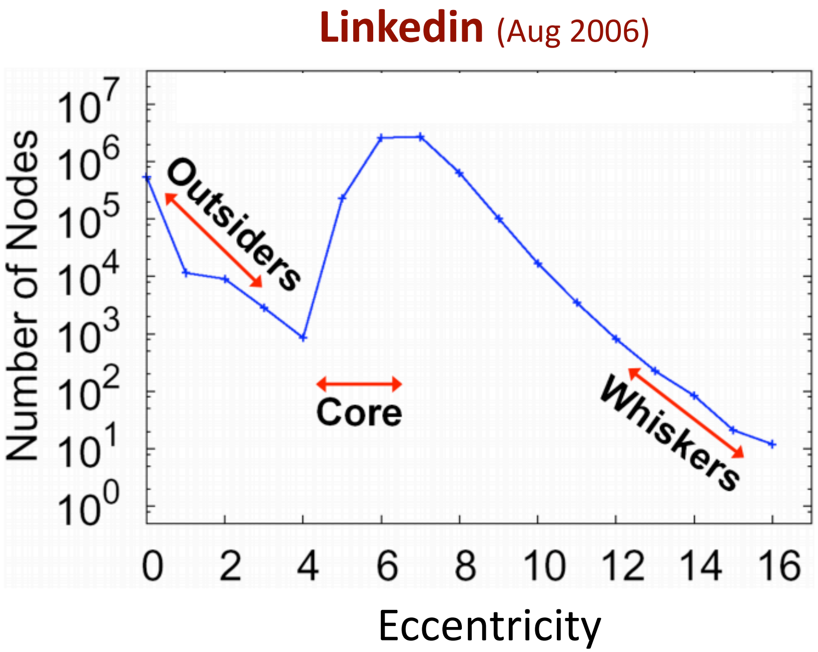
$\boldsymbol{e}(v) = 3$



<u>Applications</u>: Network topology analysis (computers, social, biological), hardware verification, sparse linear system solving, …

Eccentricity Distribution of Large Graphs

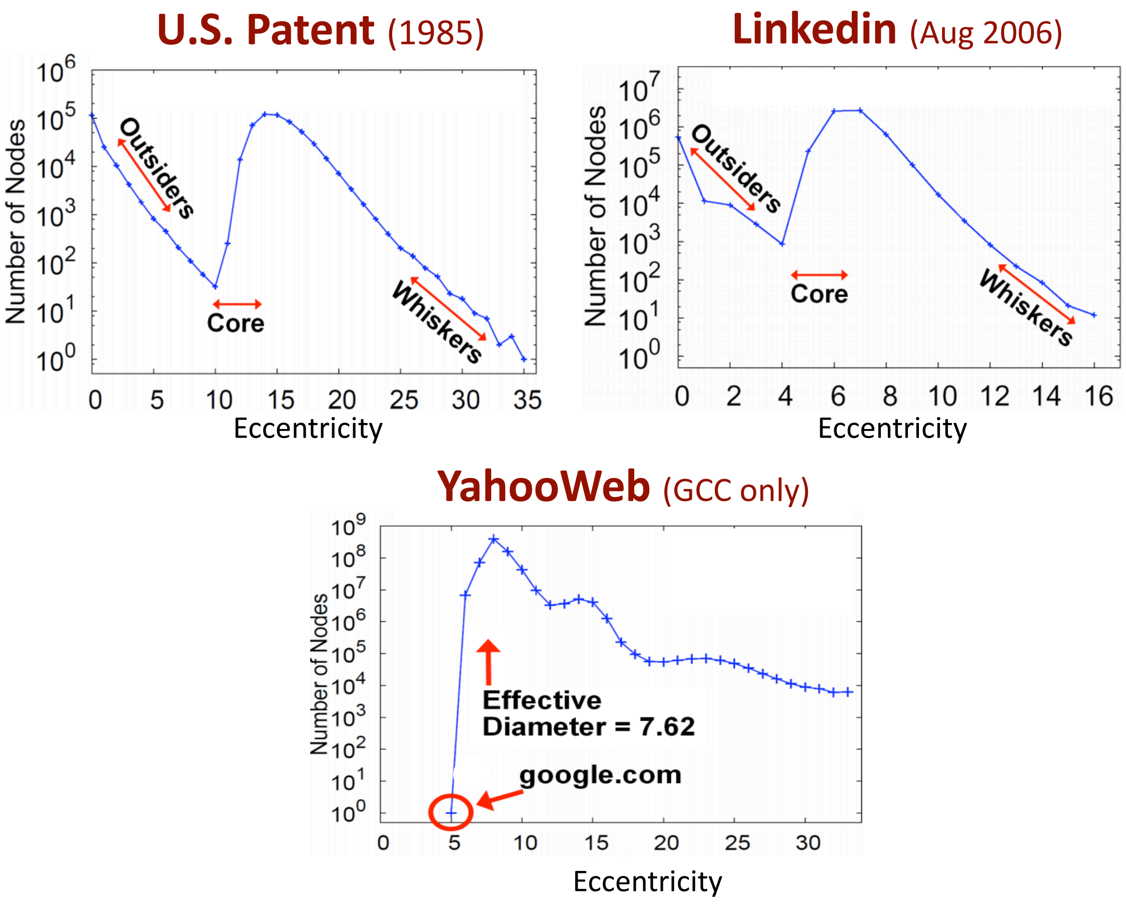




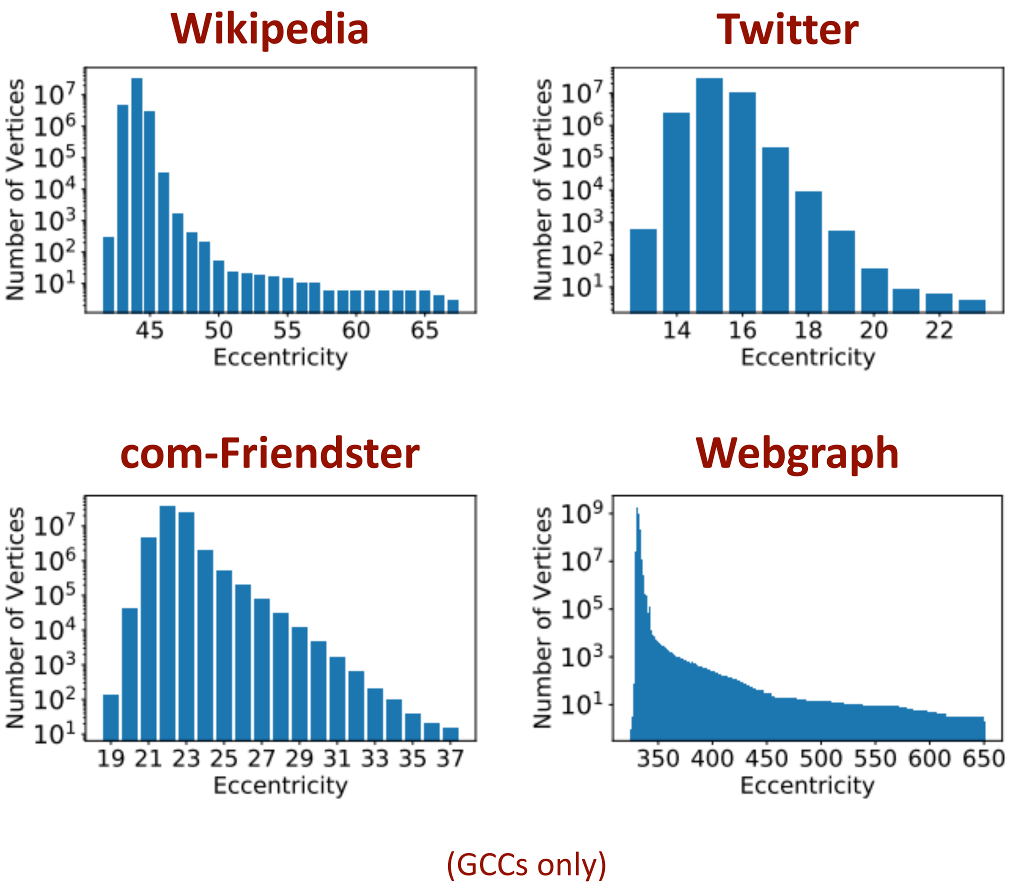
Leskovec et al. WWW 2008

Kang et al. TKDD 2011

Eccentricity Distribution of Large Graphs



Kang et al. TKDD 2011



Iwabuchi et al. CLUSTER 2018

Computing All Eccentricities

- Exact computation: $O(mn)$ (e.g. BFS from each node)
- Approximate algorithms
  - **Theoretical**:

4-approx.	$O(m)$ time	[One BFS]
$(2+\delta)$-approx.	$\tilde{O}(m/\delta)$ time	[Backurs-Roditty-Segal-V.Williams-Wein'18]
$(5/3)$-approx.	$\tilde{O}(m^{1.5})$ time	[Chechik-Larkin-Roditty-Schoenebeck-Tarjan-V.Williams'14]

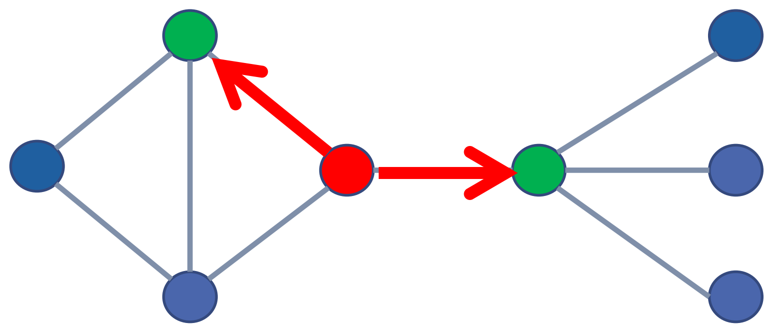
Tight under SETH

  - **Empirical**: [Kang et al. '11], [Boldi et al. '11], [Takes & Kosters '13], [...], **[Shun'15]**

Parallel k-BFS Heuristics [Shun'15]

k-BFS$_1$:

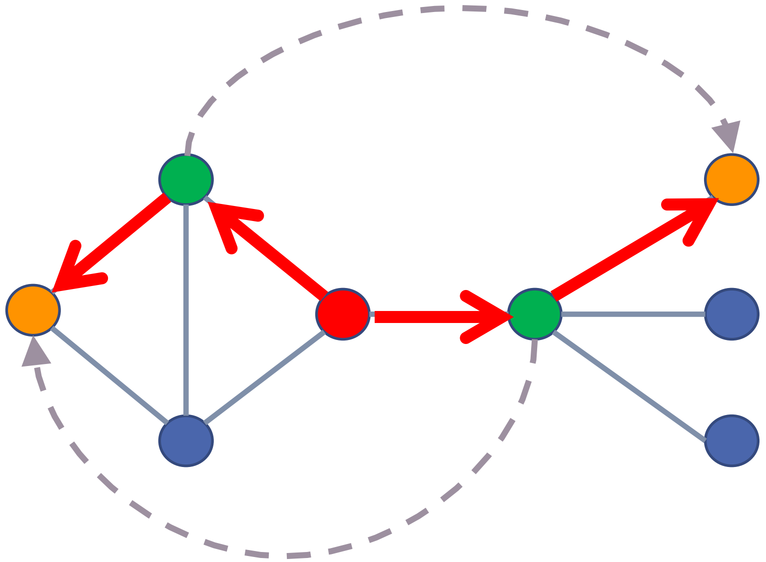
- $S_1 \leftarrow k$ random nodes
- Compute BFS from each $u \in S_1$
- $\hat{\boldsymbol{e}}_{\boldsymbol{1}}(v) \leftarrow$ max distance from S_1



Parallel k-BFS Heuristics [Shun'15]

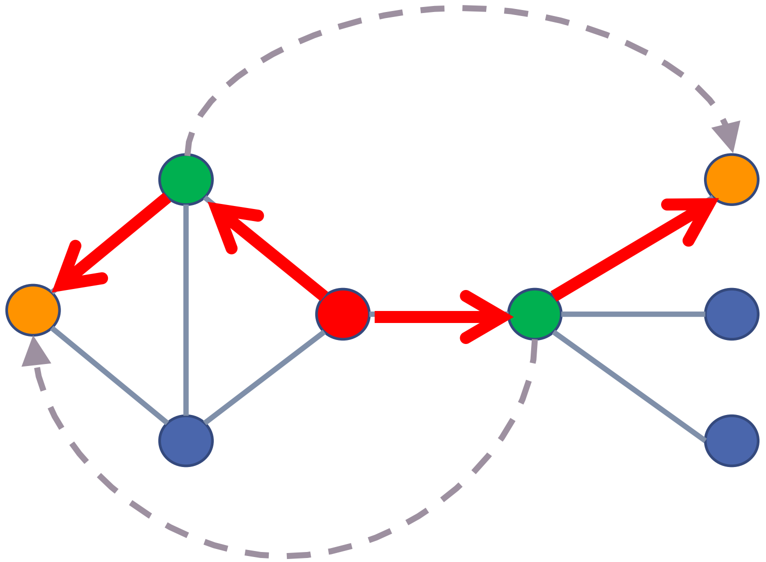
k-**BFS$_1$**:
- $S_1 \leftarrow k$ random nodes
- Compute BFS from each $u \in S_1$
- $\hat{\boldsymbol{e}}_1(v) \leftarrow$ max distance from S_1

k-**BFS$_2$**:
- $S_2 \leftarrow k$ furthest nodes from S_1
- Compute BFS from each $u \in S_2$
- $\hat{\boldsymbol{e}}_2(v) \leftarrow$ max distance from $S_1 \cup S_2$



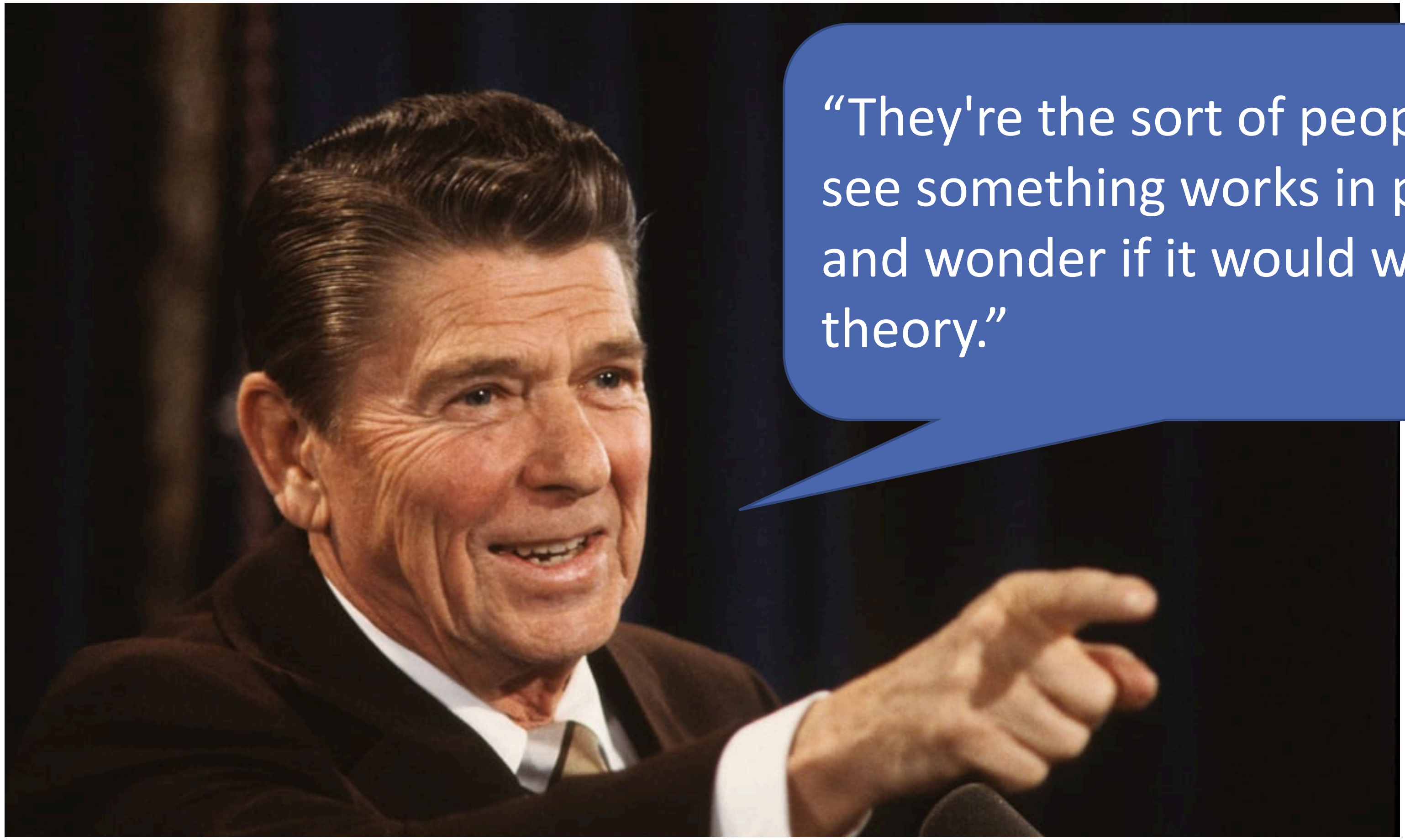
Empirical Results in [Shun'15]

- k-**BFS**$_1$ performs reasonable well
  - E.g., median average relative error 7.55%
- k-**BFS**$_2$ **beats all other methods by orders of magnitude**
  - Often computes all eccentricities **exactly**



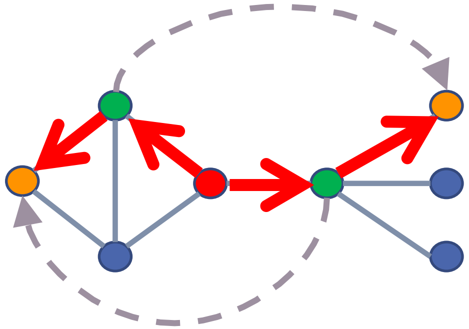
Why?

Reagan’s Principle



“They're the sort of people who see something works in practice and wonder if it would work in theory.”

This Work



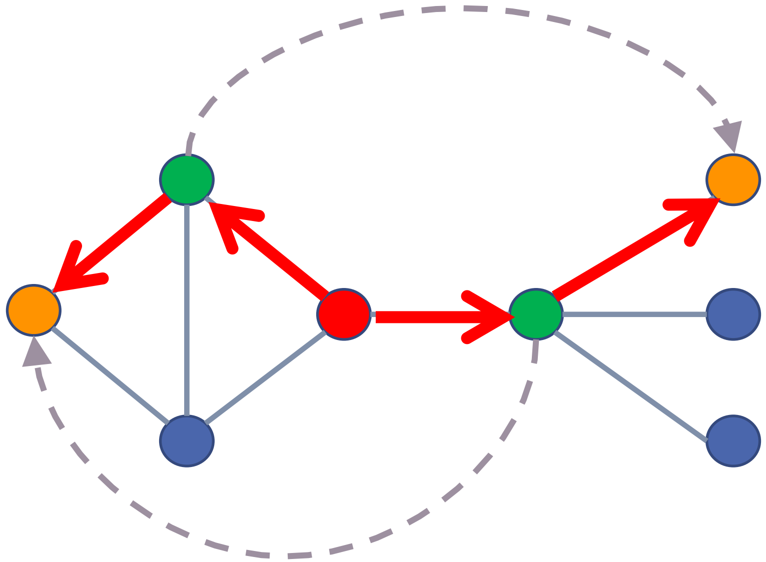
- **Analyze** heuristics in order to **explain** and **improve**
  - Will get **provable** variants with **better** empirical performance
  - Need to go beyond worst-case (due to SETH-hardness)

- k-**BFS$_2$**: Connection to **Streaming Set Cover**
  - [Demaine, Indyk, Mahabadi, Vakilian ’14]

- k-**BFS$_1$**: Connection to **Diameter Property Testing**
  - [Parnas & Ron ’02]

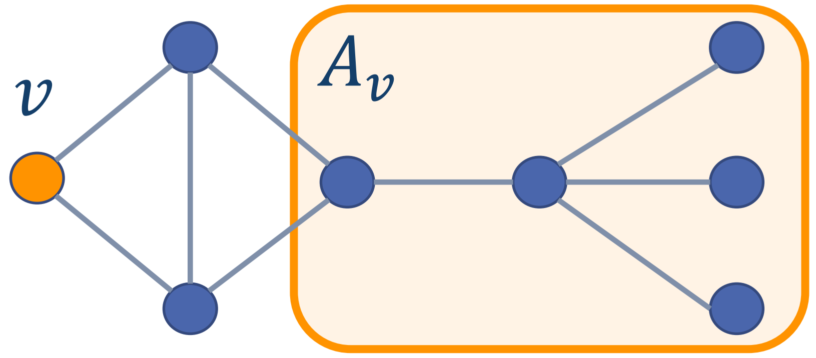
Empirical validation of theory-based algorithms

k-BFS$_2$ by Streaming Set Cover



Set Cover Formulation

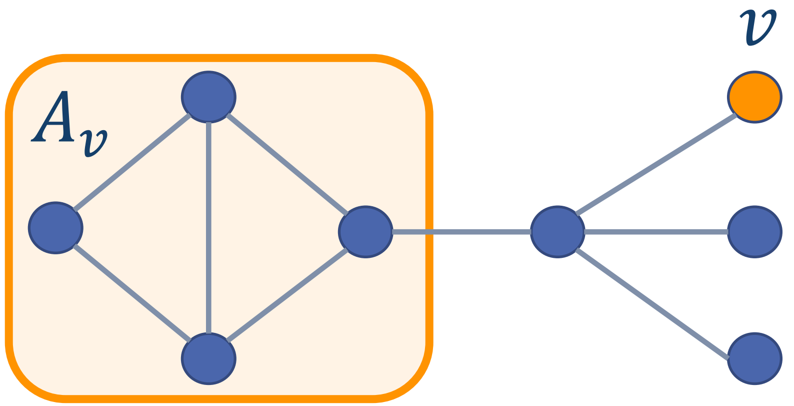
- **Set Cover**: Given elements V and subsets $\mathcal{S} \subset 2^V$, find smallest cover $C \subset \mathcal{S}$ of V.
- **Eccentricities as Set Cover:**
  - Nodes are elements
  - Nodes are sets: $\mathcal{S} = \{A_v : v \in V\}$



$$A_v = \{u \in V : \boldsymbol{e}(u) = \Delta(v, u)\}$$

Set Cover Formulation

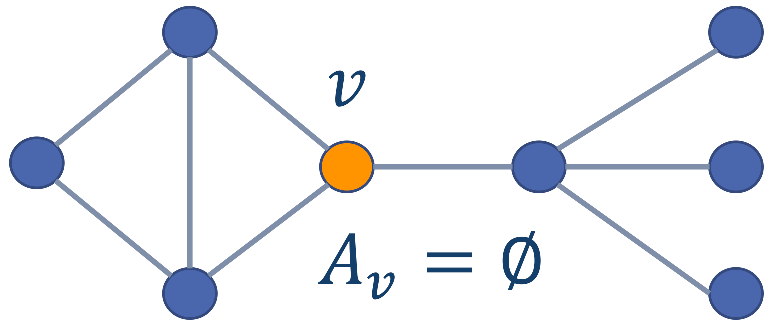
- **<u>Set Cover</u>**: Given elements V and subsets $\mathcal{S} \subset 2^V$, find smallest cover $C \subset \mathcal{S}$ of V.
- **<u>Eccentricities as Set Cover:</u>**
  - Nodes are elements
  - Nodes are sets: $\mathcal{S} = \{A_v : v \in V\}$



$$A_v = \{u \in V : \boldsymbol{e}(u) = \Delta(v, u)\}$$

Set Cover Formulation

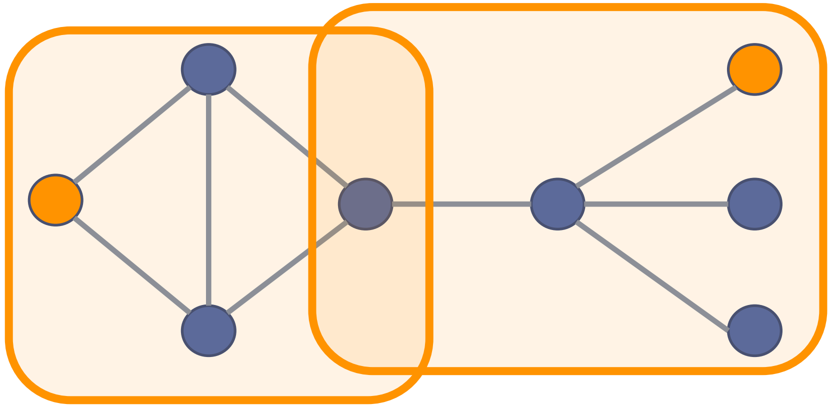
- **<u>Set Cover</u>**: Given elements V and subsets $\mathcal{S} \subset 2^V$, find smallest cover $C \subset \mathcal{S}$ of V.
- **<u>Eccentricities as Set Cover:</u>**
  - Nodes are elements
  - Nodes are sets: $\mathcal{S} = \{A_v : v \in V\}$



$$A_v = \{u \in V : \boldsymbol{e}(u) = \Delta(v, u)\}$$

Set Cover Formulation

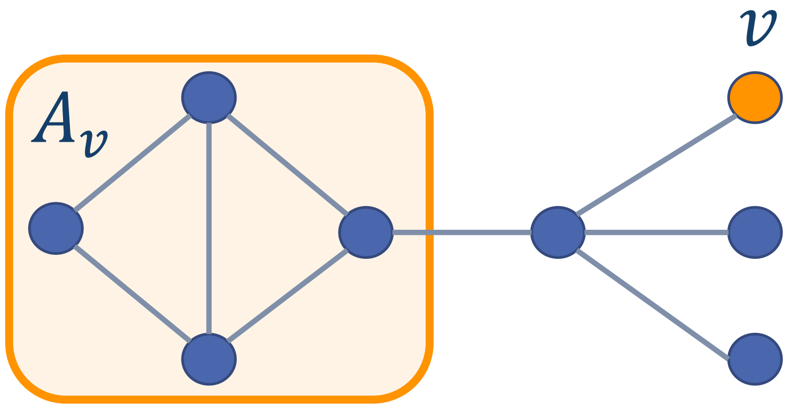
- **<u>Set Cover</u>**: Given elements V and subsets $\mathcal{S} \subset 2^V$, find smallest cover $C \subset \mathcal{S}$ of V.
- **<u>Eccentricities as Set Cover:</u>**
  - Nodes are elements
  - Nodes are sets: $\mathcal{S} = \{A_v : v \in V\}$
- Cover computes all eccentricities
- Optimal cover = "eccentric cover", $\boldsymbol{\kappa}$



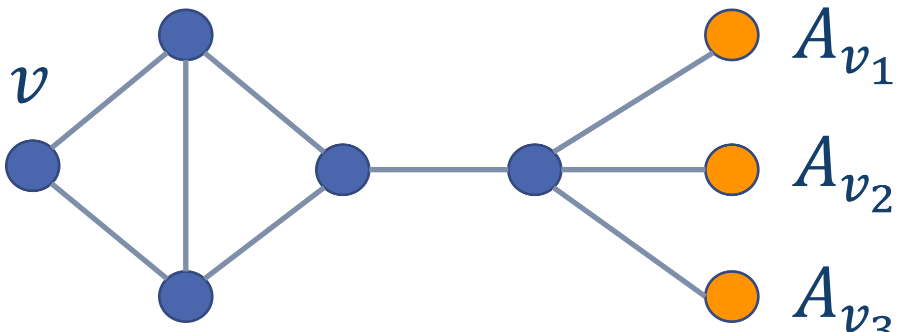
$$A_v = \{u \in V : \boldsymbol{e}(u) = \Delta(v, u)\}$$

Computational Constraints

- Computing a set A_v is **prohibitive**
  - $O(mn)$ work



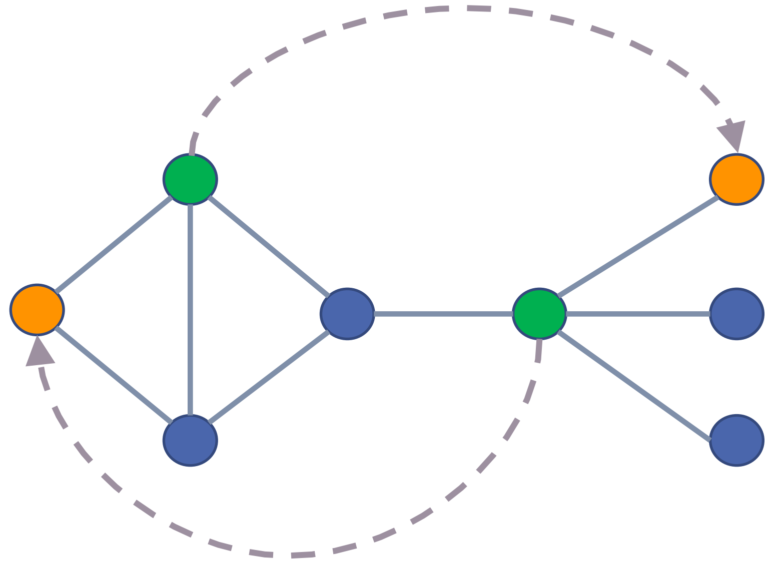
- Computing which sets cover v is **expensive**
  - Single BFS, $O(m)$ work



- Known Set Cover algorithms? **Yes**

Streaming Set Cover [Demaine-Indyk-Mahabadi-Vakilian'14]

- $S_1 \leftarrow k$ random elements
- $C \leftarrow$ Cover for sample (e.g. greedy)



Element Sampling Lemma:

If global optimum is small, C covers almost all elements.

k-BFS$_2$ vs. DIMV

k-BFS$_2$		Streaming Set Cover [DIMV'14]
$S \leftarrow$ Random sample	$\Longleftrightarrow$	$S \leftarrow$ Random sample
Compute BFS from each $v \in S$	$\Longleftrightarrow$	Compute covering sets for each $v \in S$
$C \leftarrow k$ nodes with max $\Delta(v, S)$	$\not\approx$	$C \leftarrow$ Greedy cover for S

k-BFS$_{SC}$

k-BFS$_2$		Streaming Set Cover [DIMV'14]
$S \leftarrow$ Random sample	$\Longleftrightarrow$	$S \leftarrow$ Random sample
Compute BFS from each $v \in S$	$\Longleftrightarrow$	Compute covering sets for each $v \in S$
~~$C \leftarrow k$ nodes with max $\Delta(v, S)$~~	$\not\approx$	$C \leftarrow$ Greedy cover for S

$C \leftarrow$ Parallel greedy cover for S

[Blelloch-Peng-Tangwongsan'11]

[Blelloch-Simhadri-Tangwongsan'12]

k-BFS$_{\text{SC}}$

Theorem:

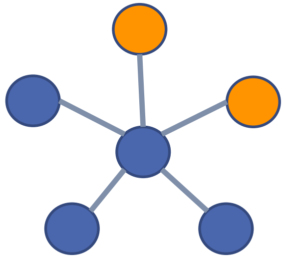
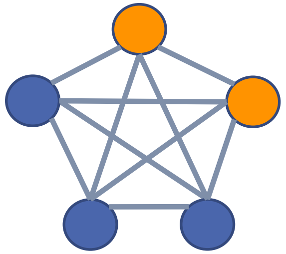
Suppose $G(V,E)$ has eccentric cover size $\boldsymbol{\kappa}$.

k-**BFS$_{\text{SC}}$** with $k = \tilde{O}(\boldsymbol{\kappa} \cdot \epsilon^{-1} \log n)$ satisfies:

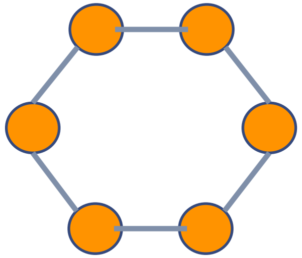
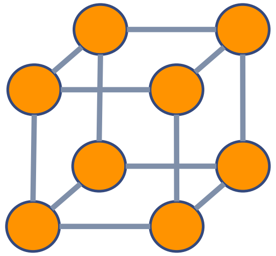
- Expected work: $O(km)$, expected depth: $\tilde{O}(\mathrm{diam}(G))$
- Computes ***exact eccentricities*** of all but an ϵ-fraction of nodes w.h.p.

Eccentric Cover: Warm-Up

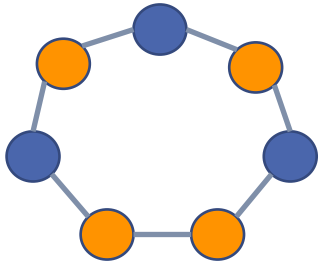
- Path, star, clique: $\boldsymbol{\kappa} = 2$

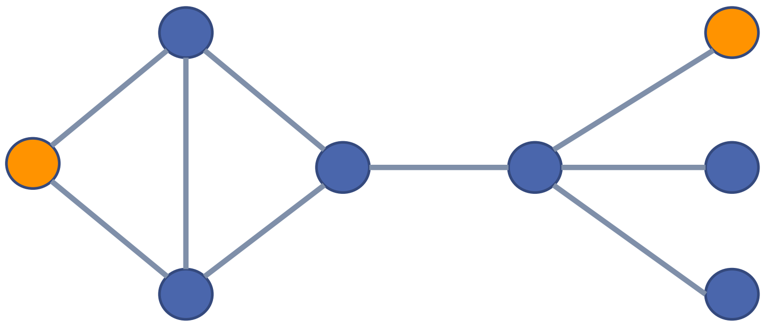
- Even cycle, hypercube: $\boldsymbol{\kappa} = n$

- Odd cycle: $\boldsymbol{\kappa} = \frac{1}{2}(n + 1)$



Eccentric Cover in the Wild



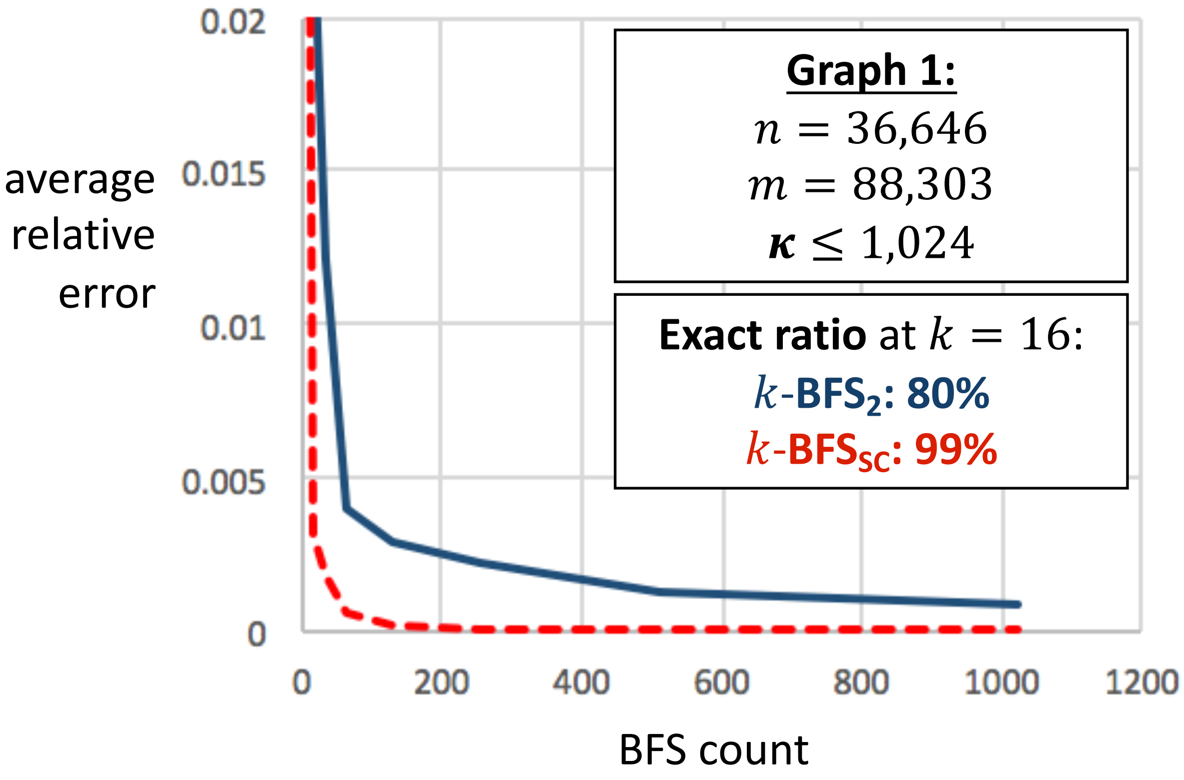
- 8 real-world graphs in [Shun'15]
- **1M-4M** nodes each
- Upper bounds on eccentric cover size:
  - 2 graphs: $\boldsymbol{\kappa} \leq \mathbf{128}$
  - 5 graphs: $\boldsymbol{\kappa} \lesssim \mathbf{1,000}$
  - 1 graph: $\boldsymbol{\kappa} \lesssim \mathbf{10,000}$

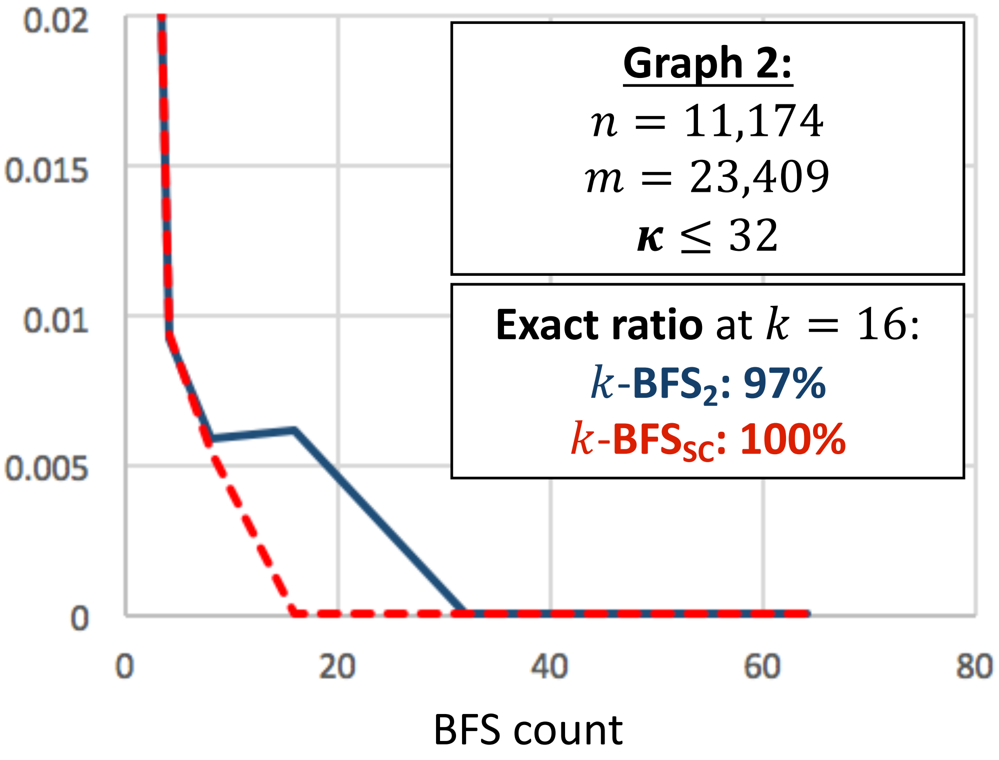
Real-world graphs have small eccentric covers

Experiments

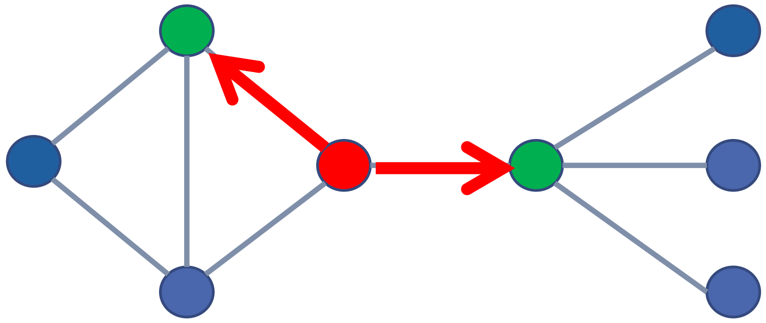
k-**BFS$_2$** vs. k-**BFS$_{SC}$**

(Real-world graphs from Stanford Network Analysis Project)





k-BFS$_1$ by Property Testing

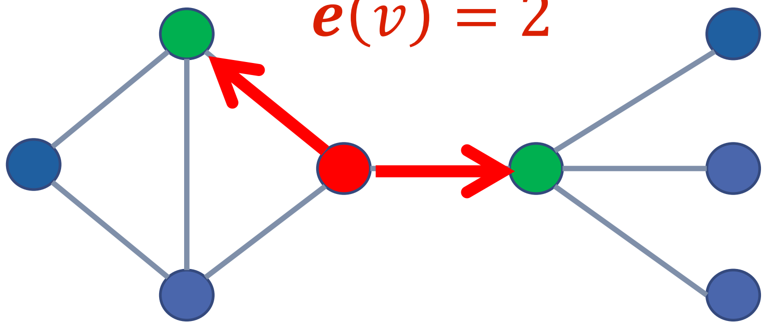


Property Testing Approximation

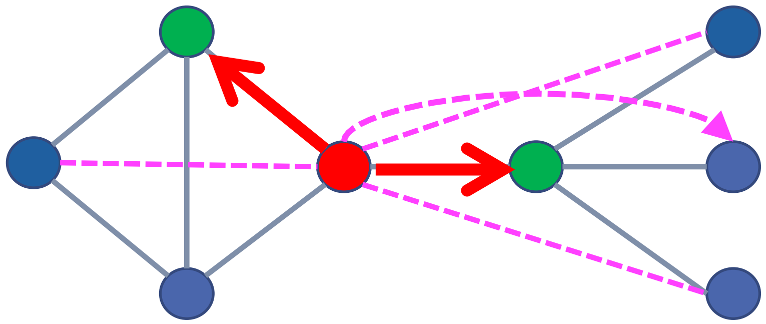
- **Usual approximation**: $\hat{e}(v)$ is close to $e(v)$
- **Property testing approximation**: $\hat{e}(v)$ is exact on some $\hat{G}$ close to G
  - Graphs are ϵ-close if up to $\epsilon \cdot m$ edges can be added/removed to get $\hat{G}$ from G
  - No sparsity/density assumption ("General Graph Model")
- **Notation**: $\hat{e}(v) \leq e(v) \preccurlyeq_\epsilon \hat{e}(v)$

$\hat{e}(v) = 1$

$e(v) = 2$



$\hat{G}$ is $\frac{4}{9}$-close to G



k-BFS$_1$ vs. Diameter Testing

k-**BFS$_1$** with $k = O(\epsilon^{-1} \log n)$ satisfies $\hat{\boldsymbol{e}}(v) \leq \boldsymbol{e}(v) \preccurlyeq_\epsilon \hat{\boldsymbol{e}}(v)$ for all v.

- Work: $\tilde{O}(\epsilon^{-1} m)$, depth: $\tilde{O}(\mathrm{diam}(G))$
- Algorithm: Start BFS at k random nodes

But this is linear time

<u>**Theorem**</u> [Parnas & Ron]: Given a graph G, compute a diameter estimate $\widehat{\boldsymbol{D}}$ such that $\widehat{\boldsymbol{D}} \leq \mathrm{diam}(G) \preccurlyeq_\epsilon 2\widehat{\boldsymbol{D}} + 2$.

- Time: $poly(\epsilon^{-1})$
- Algorithm: Start **truncated** BFS at k random nodes

Eccentricity Testing

Aux. Theorem: Given G and v, compute $\hat{\mathbf{e}}(v)$ s.t. $\hat{\mathbf{e}}(v) \leq \mathbf{e}(v) \preccurlyeq_{\epsilon} \hat{\mathbf{e}}(v)$ in time $poly(\epsilon^{-1})$.

- Corollary – Diameter testing: $\widehat{\boldsymbol{D}} \leq \mathrm{diam}(G) \preccurlyeq_{\epsilon} 2\widehat{\boldsymbol{D}}$ (shaved off +2)
- Corollary – Radius testing: $\widehat{\boldsymbol{R}} \leq \mathrm{radius}(G) \preccurlyeq_{\epsilon} \widehat{\boldsymbol{R}} + \mathbf{1}$

Implies variant of k-**BFS**$_1$: k-**BFS**$_{\mathbf{TST}}$

k-BFS$_{\text{TST}}$

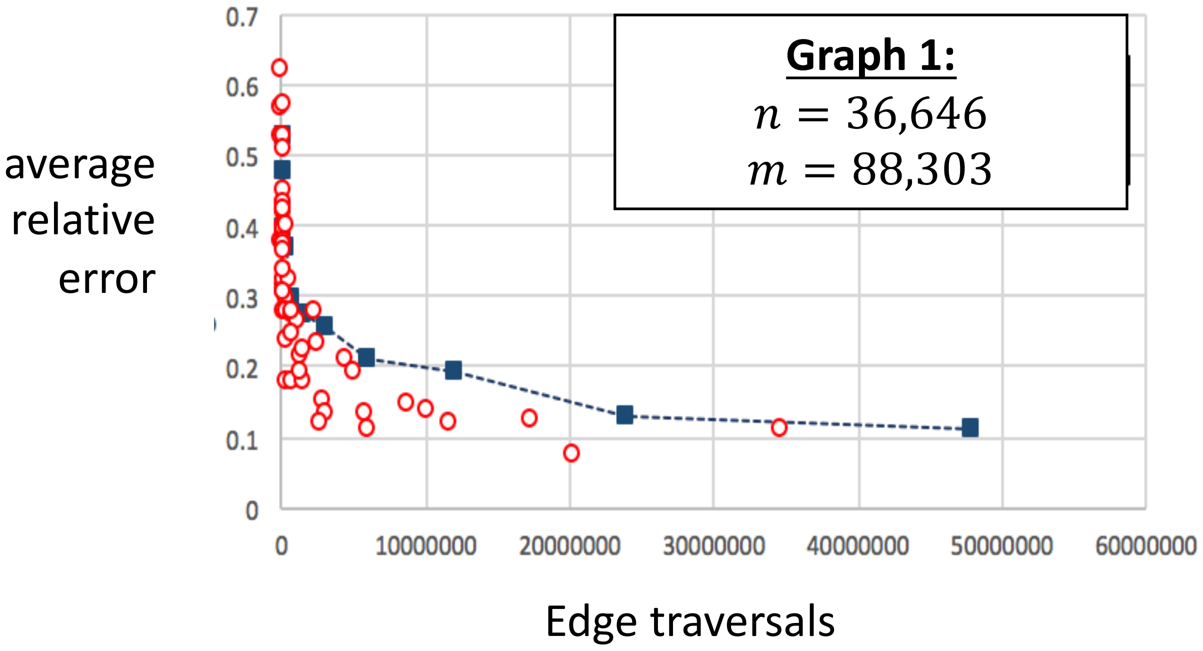
Theorem: k-**BFS**$_{\textbf{TST}}$ satisfies $\hat{\boldsymbol{e}}(v) \le \boldsymbol{e}(v) \preccurlyeq_\epsilon \hat{\boldsymbol{e}}(v)$ for all v.

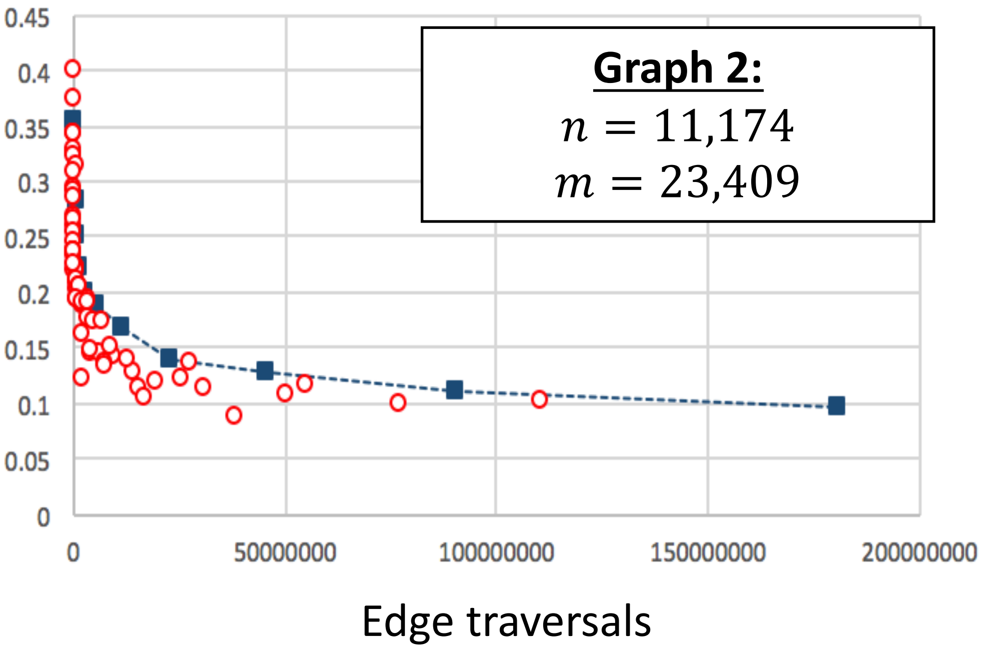
- Work: $O(\epsilon^{-2} n)$, depth: $\tilde{O}(\epsilon^{-1} \log n)$
- Algorithm: truncated BFS
  - $S_1 \leftarrow k$ random nodes
  - From each $u \in S_1$, start a BFS up to first level ℓ_u where $\tilde{O}(\epsilon^{-1})$ nodes are seen. All unseen nodes are considered at "distance" $\ell_u + 1$ from u.
  - $\hat{\boldsymbol{e}}_{\textbf{TST}}(v) \leftarrow$ max "distance" from S_1

Same guarantee as k-**BFS**$_{\mathbf{1}}$ but in **sublinear** work and depth, independent of graph.

Experiments

k-**BFS$_1$** vs. k-**BFS$_{TST}$** (with different BFS cutoffs)





Conclusion

- **Explain** and **improve** high-performing heuristics
  - Practical algorithm -> “fit” analysis -> practical improvement with guarantees
- Inter-connections of **parallel**, **streaming**, **sketching**, and **property testing** algorithms
  - All “point to same direction”

Thank you