# Querying Minimal Steiner Maximum-Connected Subgraphs from Large Graphs

**Jiafeng Hu**, Xiaowei Wu, Reynold Cheng, Siqiang Luo and Yixiang Fang

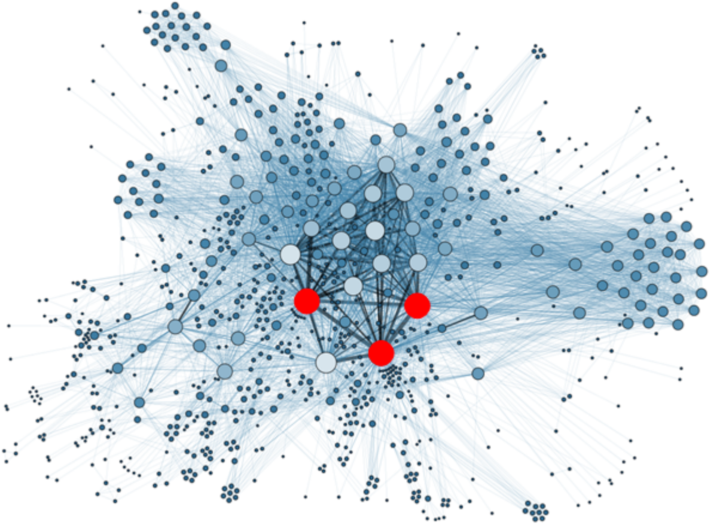
Department of Computer Science

The University of Hong Kong

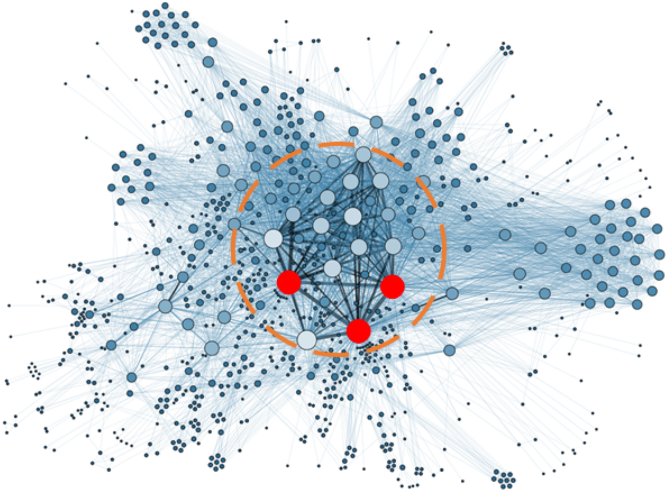
October 27, 2016

# Community Search

# Community Search

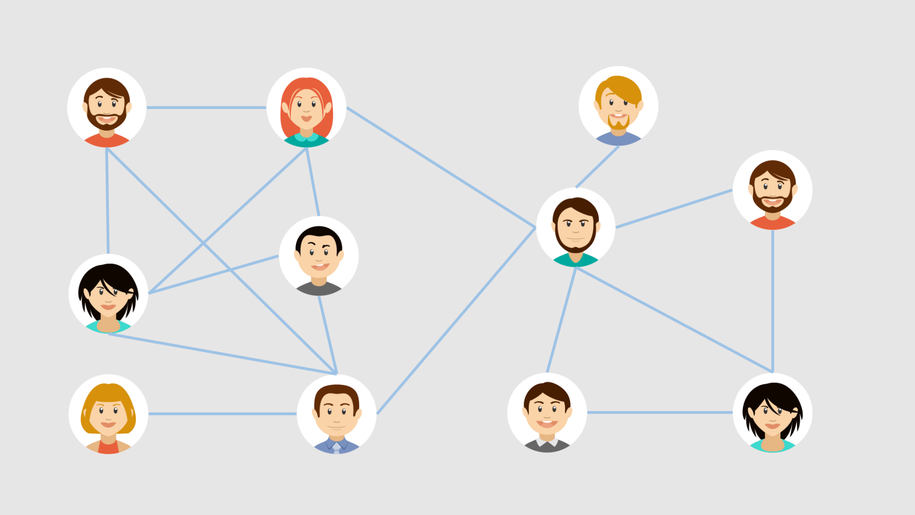


# Community Search

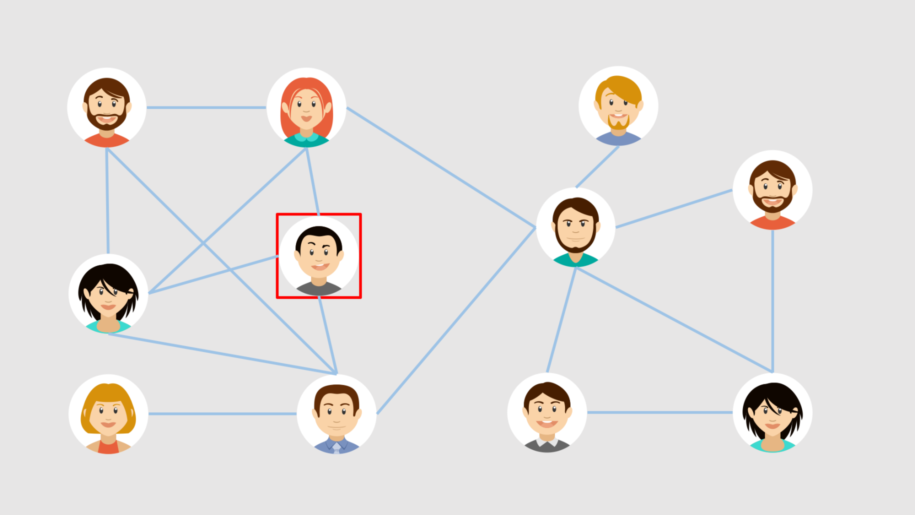


Find a subgraph: 1) contains query nodes; 2) the cohesiveness is maximized

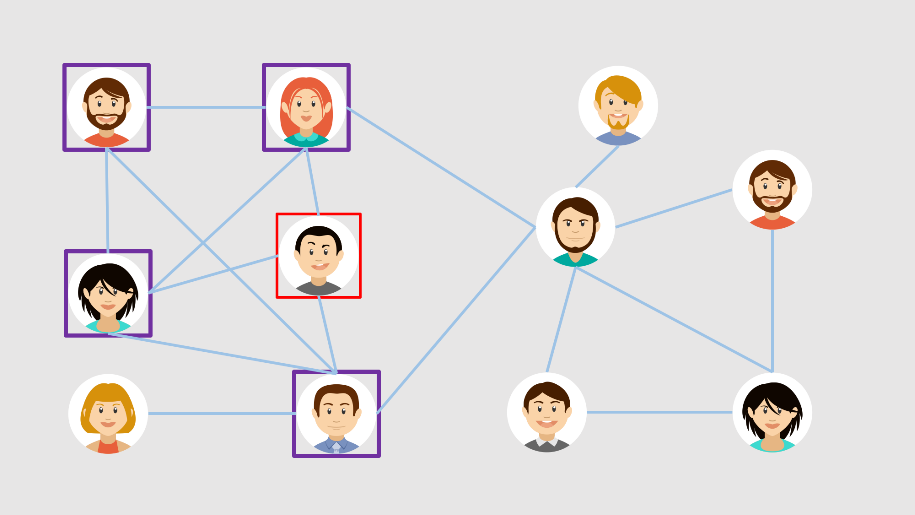
# Application (Social Network)



# Application (Social Network)

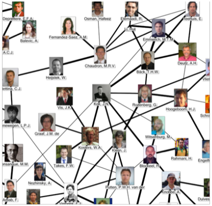


# Application (Social Network)



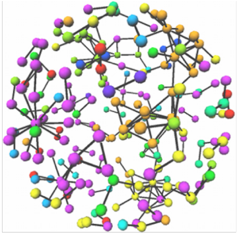
# Other Applications


Social Media Marketing



DBLP Google scholar Microsoft Academic Search Beta

Research Team Assembling



Protein-Protein interaction Network

# Cohesiveness

Metrics to measure cohesiveness:

# Cohesiveness

Metrics to measure cohesiveness:

- $k$-core [KDD'10, SIGMOD'14, DMKD'15, VLDB'16]
  - the minimum degree among all nodes in the subgraph $\geq k$

# Cohesiveness

Metrics to measure cohesiveness:

- $k$-core [KDD'10, SIGMOD'14, DMKD'15, VLDB'16]
  - the minimum degree among all nodes in the subgraph $\geq k$
- $k$-truss [SIGMOD'14, VLDB'15]
  - every edge is contained in at least $k - 2$ triangles within the subgraph

# Cohesiveness

Metrics to measure cohesiveness:

- $k$-core [KDD'10, SIGMOD'14, DMKD'15, VLDB'16]
  - the minimum degree among all nodes in the subgraph $\geq k$
- $k$-truss [SIGMOD'14, VLDB'15]
  - every edge is contained in at least $k-2$ triangles within the subgraph
- connectivity [SIGMOD'15]

# Connectivity [Gibbons, Graph Theory, 1985]

Given an undirected graph $G$:

# Connectivity [Gibbons, Graph Theory, 1985]

Given an undirected graph $G$:

- $\lambda(G)$: the minimum number of edges whose removal disconnects $G$.

# Connectivity [Gibbons, Graph Theory, 1985]

Given an undirected graph $G$:

- $\lambda(G)$: the minimum number of edges whose removal disconnects $G$.

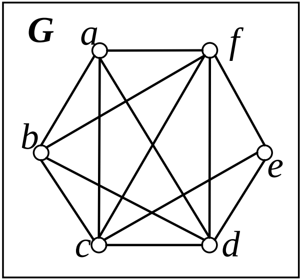



Figure : An example of connectivity

$$\lambda(G)=3$$

# Connectivity [Gibbons, Graph Theory, 1985]

Given an undirected graph $G$:

- $\lambda(G)$: the minimum number of edges whose removal disconnects $G$.

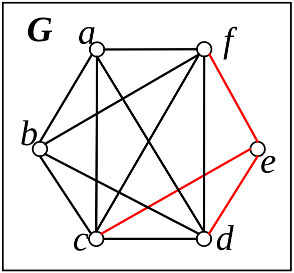



Figure : An example of connectivity

$$\lambda(G)=3$$

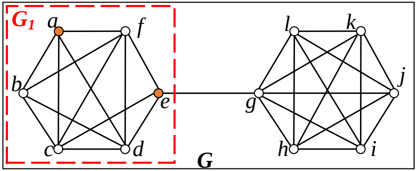
# Comparison with $k$-core and $k$-truss

Connectivity is more appropriate to model cohesiveness:

# Comparison with $k$-core and $k$-truss

Connectivity is more appropriate to model cohesiveness:

- $k$-core only restricts degrees of nodes in subgraphs without any structure constraint.
  - E.g., in Figure(a), $Q = \{a, e\}$
  - return $G$ ($k$-core) VS $G_1$ (connectivity ✓)

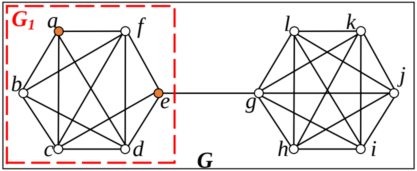



(a) $Q = \{a, e\}$

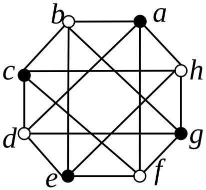
# Comparison with $k$-core and $k$-truss

Connectivity is more appropriate to model cohesiveness:

- $k$-core only restricts degrees of nodes in subgraphs without any structure constraint.
  - E.g., in Figure(a), $Q = \{a, e\}$
  - return $G$ ($k$-core) VS $G_1$ (connectivity ✓)
- $k$-truss is too restrictive on the triangle structure.
  - E.g., in Figure(b), $Q = \{a\}$
  - return $\emptyset$ ($k$-truss) VS the whole graph (connectivity ✓)




(a) $Q = \{a, e\}$




(b) $Q = \{a\}$

# Steiner Maximum-Connected Subgraph (SMCS) Problem

## Definition

Given an undirected graph $G$ and a set $Q$ of query nodes, the SMCS is a subgraph $G_Q$ of $G$ which contains $Q$ and the **connectivity** $\lambda(G_Q)$ of $G_Q$ is **maximized**.

# Steiner Maximum-Connected Subgraph (SMCS) Problem

## Definition

Given an undirected graph $G$ and a set $Q$ of query nodes, the SMCS is a subgraph $G_Q$ of $G$ which contains $Q$ and the **connectivity** $\lambda(G_Q)$ of $G_Q$ is **maximized**.

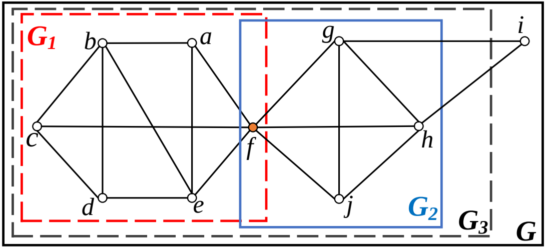



Figure : An example of SMCS, with $Q = \{f\}$

- SMCSs: $\{G_1, G_2, G_3\}$
- $sc(Q) = 3$ (the connectivity of any SMCS of $Q$)

# Maximum SMCS [Chang et al., SIGMOD'15]

## Definition

An SMCS $G_Q$ of $Q$ such that the number of nodes in $G_Q$ is **maximized**.

# Maximum SMCS [Chang et al., SIGMOD'15]

## Definition

An SMCS $G_Q$ of $Q$ such that the number of nodes in $G_Q$ is **maximized**.

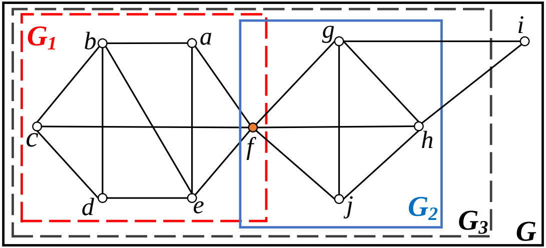



Figure : An example of maximum SMCS, with $Q = \{f\}$

- maximum SMCS:$\{G_3\}$

# Maximum SMCS [Chang et al., SIGMOD'15]

## Definition

An SMCS $G_Q$ of $Q$ such that the number of nodes in $G_Q$ is **maximized**.

**Drawback**: the returned subgraph is too large
E.g., on the DBLP dataset ($803K$ Nodes, $3M$ Edges), a maximum SMCS has over $400K$ nodes in average.

# Maximum SMCS [Chang et al., SIGMOD'15]

## Definition

An SMCS $G_Q$ of $Q$ such that the number of nodes in $G_Q$ is **maximized**.

**Drawback**: the returned subgraph is too large

E.g., on the DBLP dataset ($803K$ Nodes, $3M$ Edges), a maximum SMCS has over $400K$ nodes in average.

| Name | $size$ (Max-SMCS) |
|---|---|
| Jiawei Han, Jian Pei | 12,459 |
| Michael Stonebraker, Jennie Duggan | 171,435 |
| Reynold Cheng | 26,223 |
| Jiafeng Hu | 414,499 |
| Siqiang Luo | 130,228 |

# Minimum SMCS

## Definition

An SMCS $G_Q$ of $Q$ such that the number of nodes in $G_Q$ is **minimized**.

## Definition

An SMCS $G_Q$ of $Q$ such that the number of nodes in $G_Q$ is **minimized**.

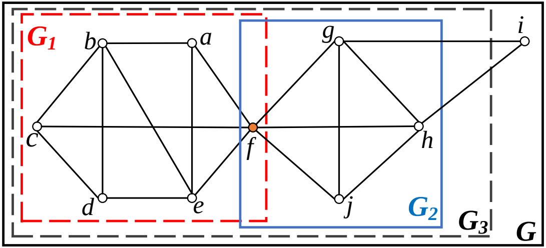



Figure : An example of minimum SMCS, with $Q = \{f\}$

- minimum SMCS: $\{G_2\}$

# Minimum SMCS

**Definition**

An SMCS $G_Q$ of $Q$ such that the number of nodes in $G_Q$ is **minimized**.

**Theorem (Inapproximability)**

*The minimum SMCS problem is* **NP***-hard. Moreover, there does not exist any polynomial-time algorithm that approximates the minimum SMCS problem within any constant ratio.*

# Minimal SMCS

## Definition

An SMCS $G_Q$ of $Q$ such that any proper induced subgraph of $G_Q$ containing $Q$ is not an SMCS of $Q$.

## Definition

An SMCS $G_Q$ of $Q$ such that any proper induced subgraph of $G_Q$ containing $Q$ is not an SMCS of $Q$.

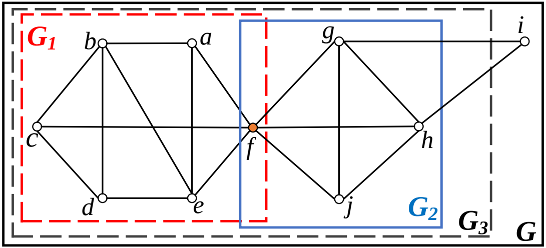



Figure : An example of minimal SMCS, with $Q = \{f\}$

- minimal SMCSs: $\{G_1, G_2\}$

# Our Contributions

- Propose the minimum and minimal SMCS problems.
- Prove the hardness of the minimum SMCS problem.
- Devise the Expand-Refine algorithm and its approximate version with accuracy guarantees.

# Our Contributions

- Propose the minimum and minimal SMCS problems.
- Prove the hardness of the minimum SMCS problem.
- Devise the Expand-Refine algorithm and its approximate version with accuracy guarantees.

# Expand-Refine Framework

**Input**: a graph $G$, and a set $Q$ of query nodes
**Output**: a minimal SMCS of $Q$

# Expand-Refine Framework

**Input**: a graph $G$, and a set $Q$ of query nodes
**Output**: a minimal SMCS of $Q$

1. Compute the Steiner-connectivity of $Q$
   - the connectivity of any SMCS of $Q$ (existing solution, in $\mathcal{O}(|Q|)$)

# Expand-Refine Framework

**Input**: a graph $G$, and a set $Q$ of query nodes
**Output**: a minimal SMCS of $Q$

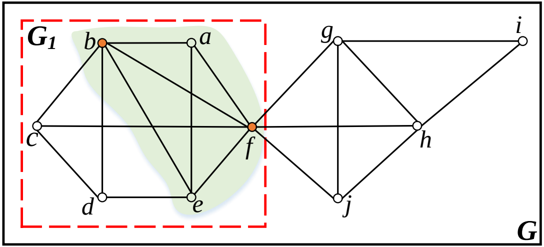
1. Compute the Steiner-connectivity of $Q$
   - the connectivity of any SMCS of $Q$ (existing solution, in $\mathcal{O}(|Q|)$)
2. Perform **Expand** operation to generate an SMCS of $Q$
   - Iteratively expand the candidate node set and test whether there exists an SMCS.

# Expand-Refine Framework

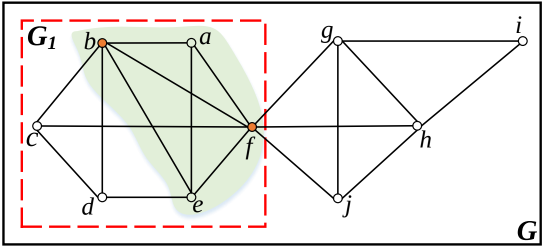
**Input**: a graph $G$, and a set $Q$ of query nodes
**Output**: a minimal SMCS of $Q$

1. Compute the Steiner-connectivity of $Q$
   - the connectivity of any SMCS of $Q$ (existing solution, in $\mathcal{O}(|Q|)$)
2. Perform **Expand** operation to generate an SMCS of $Q$
   - Iteratively expand the candidate node set and test whether there exists an SMCS.
3. Execute **Refine** operation on the SMCS returned by step 2 to find a minimal SMCS of $Q$

# A Toy Example (Graph $G$, query $Q = \{b, f\}$)

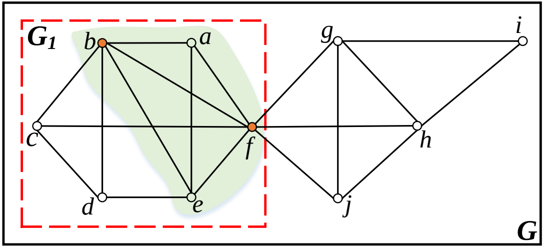



1. Compute the Steiner-connectivity, $sc(Q)$=3.



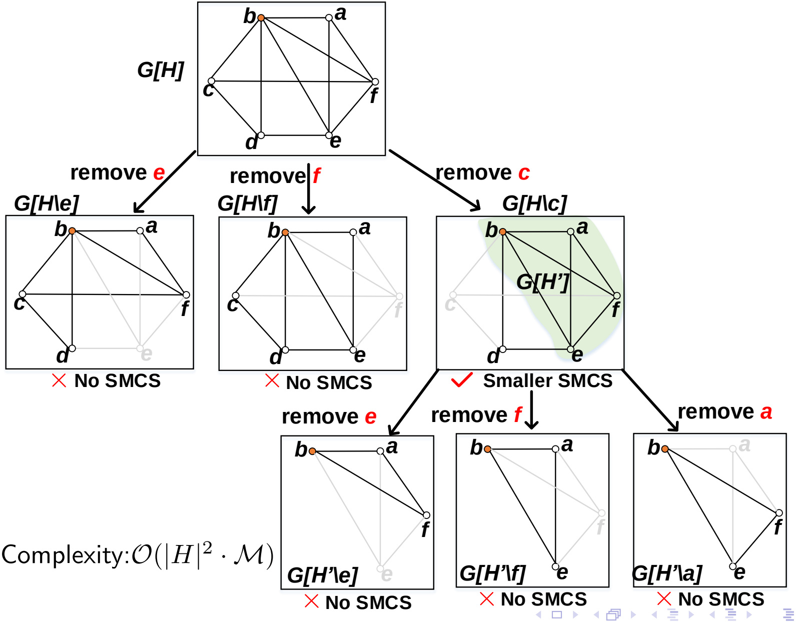

1. Compute the Steiner-connectivity, $sc(Q)$=3.
2. Generate a candidate SMCS of $Q$, e.g., $G_1$.

# A Toy Example (Graph $G$, query $Q = \{b, f\}$)



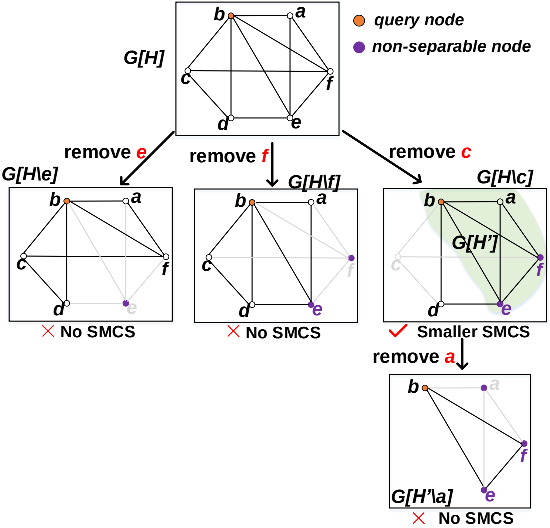

1. Compute the Steiner-connectivity, $sc(Q)$=3.
2. Generate a candidate SMCS of $Q$, e.g., $G_1$.
3. Refine $G_1$ to get a minimal SMCS: the green-shaded graph, $\{a, b, e, f\}$.

# The Refine Operation (Step 3) [Basic refinement]



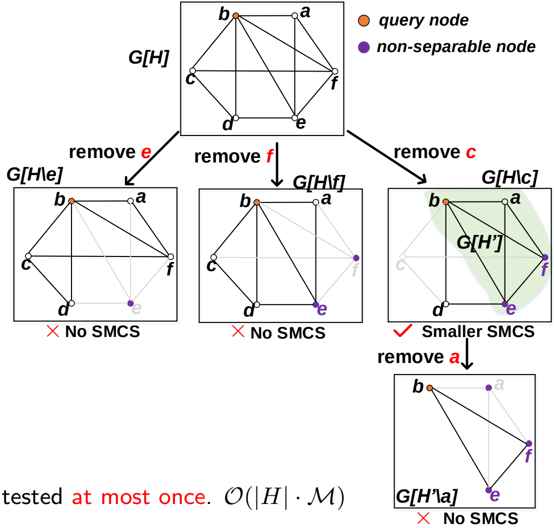

Time Complexity: $\mathcal{O}(|H|^2 \cdot \mathcal{M})$

**[Separable/Non-separable]** Given any SMCS $G[H]$ of $Q$, node $u$ is **separable** for $Q$ if there exists any SMCS of $Q$ in $G[H \backslash u]$; otherwise, $u$ is **non-separable** for $Q$.



# Advanced Refinement

**[Separable/Non-separable]** Given any SMCS $G[H]$ of $Q$, node $u$ is **separable** for $Q$ if there exists any SMCS of $Q$ in $G[H \backslash u]$; otherwise, $u$ is **non-separable** for $Q$.




All nodes will be tested at most once. $\mathcal{O}(|H| \cdot \mathcal{M})$

# Relaxation

- Connectivity Relaxation (Early Stop in the **Expand** Step)
  - Stop expanding the candidate node set $S$ if $|S| > \theta$
  - Extract a maximal SMCS from $S$

- Minimality Relaxation (Approximation in the **Refine** Step)

# Minimality Relaxation (Approximation in the Refine Step)

**Main idea**:

- Sample nodes uniformly at random.
- Record the number of non-separable nodes sampled consecutively.
- Halt when $\omega = \lceil \frac{\log \frac{1}{\delta}}{\log r} \rceil$ non-separable nodes are sampled consecutively.

# Minimality Relaxation (Approximation in the Refine Step)

**Main idea**:

- Sample nodes uniformly at random.
- Record the number of non-separable nodes sampled consecutively.
- Halt when $\omega = \lceil \frac{\log \frac{1}{\delta}}{\log r} \rceil$ non-separable nodes are sampled consecutively.
  - Approximation ratio $r$: $\frac{|\text{returned SMCS}|}{|\text{minimal SMCS}|}$

# Minimality Relaxation (Approximation in the Refine Step)

**Main idea**:

- Sample nodes uniformly at random.
- Record the number of non-separable nodes sampled consecutively.
- Halt when $\omega = \lceil \frac{\log \frac{1}{\delta}}{\log r} \rceil$ non-separable nodes are sampled consecutively.
  - Approximation ratio $r$: $\frac{|\text{returned SMCS}|}{|\text{minimal SMCS}|}$
  - Failure probability $\delta$: the probability that the approximation ratio is larger than $r$ in practice.

# Minimality Relaxation (Approximation in the Refine Step)

**Main idea**:

- Sample nodes uniformly at random.
- Record the number of non-separable nodes sampled consecutively.
- Halt when $\omega = \lceil \frac{\log \frac{1}{\delta}}{\log r} \rceil$ non-separable nodes are sampled consecutively.
    - Approximation ratio $r$: $\frac{|\text{returned SMCS}|}{|\text{minimal SMCS}|}$
    - Failure probability $\delta$: the probability that the approximation ratio is larger than $r$ in practice.

**Lemma (Approximated Minimal SMCS)**

*Given an SMCS $G[H]$ of $Q$, for any constant $\delta \in (0, 1)$ and $r > 1$, the proposed algorithm returns an $r$-approximation of a minimal SMCS of $Q$ in $G[H]$ with probability at least $(1 - \delta)$.*

# Experiments

Table : Dataset statistics (**K**$=10^3$ and **M**$=10^6$)

| ID | Dataset | #Nodes | #Edges | $\overline{d}$ | $sc_{max}$ |
|---|---|---|---|---|---|
| D1 | ca-CondMat | 21**K** | 91**K** | 8.55 | 25 |
| D2 | soc-Epinions1 | 75**K** | 405**K** | 10.69 | 67 |
| D3 | DBLP | 803**K** | 3.2**M** | 8.18 | 118 |
| D4 | wiki-Talk | 2.3**M** | 4.6**M** | 3.90 | 131 |
| D5 | as-Skitter | 1.6**M** | 11**M** | 13.09 | 111 |
| D6 | uk-2002 | 18**M** | 261**M** | 28.34 | 943 |

All algorithms are implemented in C++.

Queries:

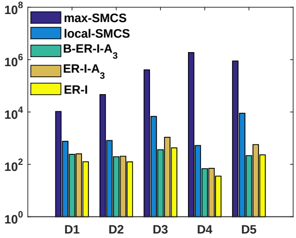
- size $|Q|$ (default: 3)
- inter-distance $l$: maximum distance between any nodes in $Q$ (default: 2)
- #Queries: 500

# Effectiveness

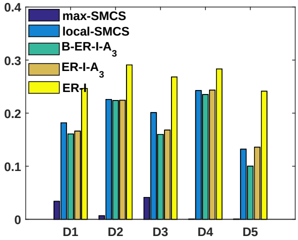
Metrics:

- *Size*: The number of nodes in the result graph
- *Edge Density* $\rho$: the ratio of the number of edges of a graph to that of its complete version ($\frac{2|E(g)|}{|V(g) \times (|V(g)|-1)|}$)

**Exp-1: Quality Evaluation** (500 queries, $|Q|$ from 1 to 16, $l$=2)




(a) Size




(b) $\rho$

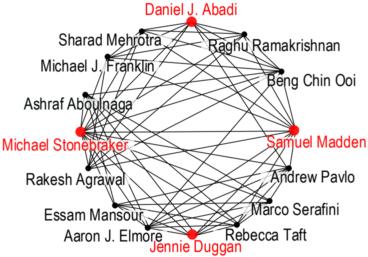
# Effectiveness (Cont.)

**Exp-2: DBLP case study**

$Q$={"Michael Stonebraker", "Samuel Madden", "Daniel J. Abadi", "Jennie Duggan"}

| | Size | $\rho$ |
|---|---|---|
| `max-SMCS` | 171,435 | $10^{-4}$ |
| `Local-SMCS` | 129 | 0.21 |
| `B-ER-I-`$A_3$ | 17 | 0.54 |
| `ER-I-`$A_3$ | 17 | 0.54 |
| `ER-I` | 15 | 0.58 |

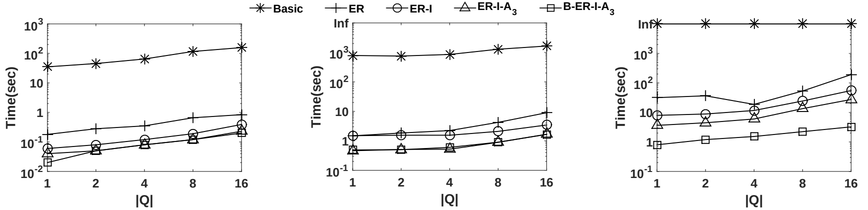
(a) Quality Measure



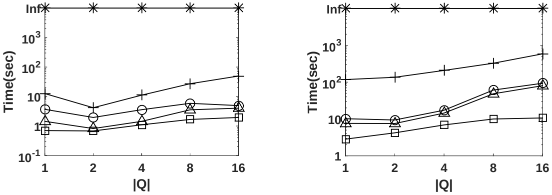

(b) The minimal SMCS

# Efficiency

## Exp-3: Effect of queries (Varying the query size $|Q|$)




(a) D1 (b) D2 (c) D3




(d) D4 (e) D5

Inf: the running time exceeds 1 hour

# Conclusion

- Propose the minimal SMCS problem and prove the hardness of the minimum SMCS problem.
- Develop an efficient Expand-Refine algorithm and its approximate version with accuracy guarantees.
- Detailed evaluation on real graph datasets demonstrates the effectiveness and efficiency of our solution.

# Thanks!

Jiafeng Hu

jhu@cs.hku.hk