Community Structure in Complex Networks

Mojtaba Rezvani

Australian National University

April 27, 2018

Outline

- Introduction
- Hierarchical community structure
- Structural hole spanners
- Overlapping community structure
- Community search
- Conclusion

The problems:

- Hierarchical community detection
- Identifying top-k structural hole spanners
- Overlapping community detection
- Community search

The thesis

- Formalises these problems, and
- Proposes efficient algorithms for them, and
- Presents extensive experimental results on real datasets

Outline

- Introduction
- Hierarchical community structure
- Structural hole spanners
- Overlapping community structure
- Community search
- Conclusion

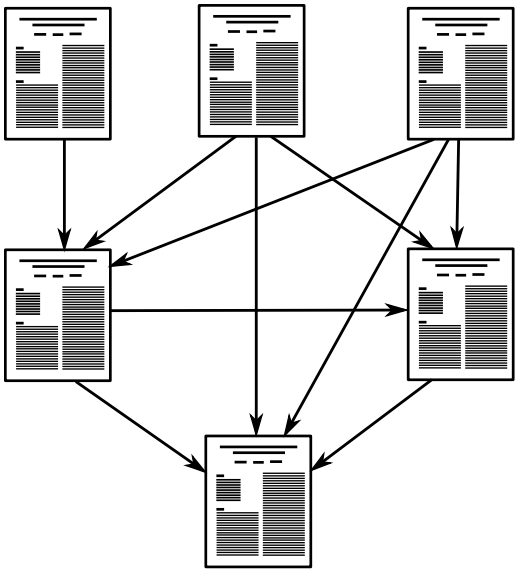
Complex networks

Appear in a wide range of contexts,

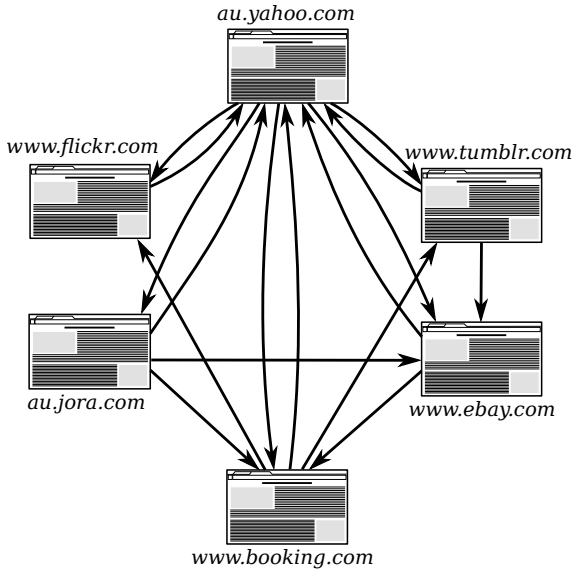
- Biology
- Sociology
- Citation networks
- Collaboration networks
- World Wide Web network

Australian National University

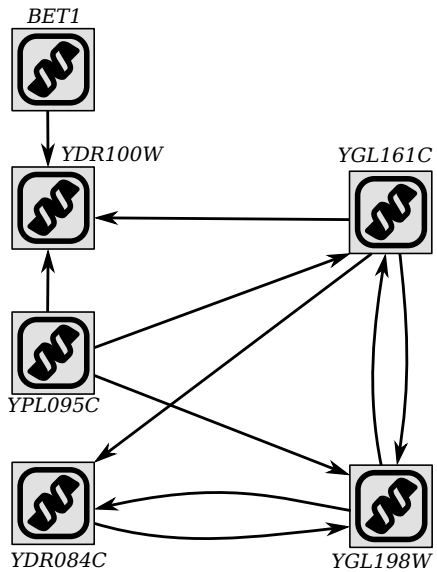
Examples of complex networks



(a) Citation network



(b) Hyperlink network



(c) Protein network

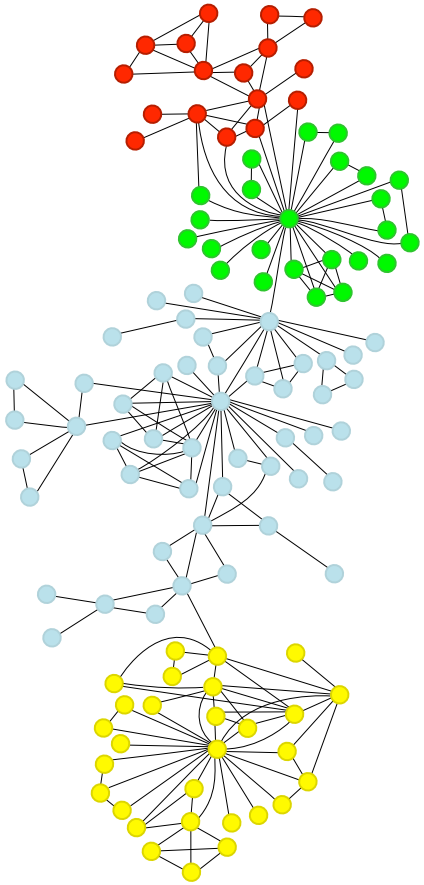
Community structure

Edges appear with high density among groups of vertices, while with less density between groups,

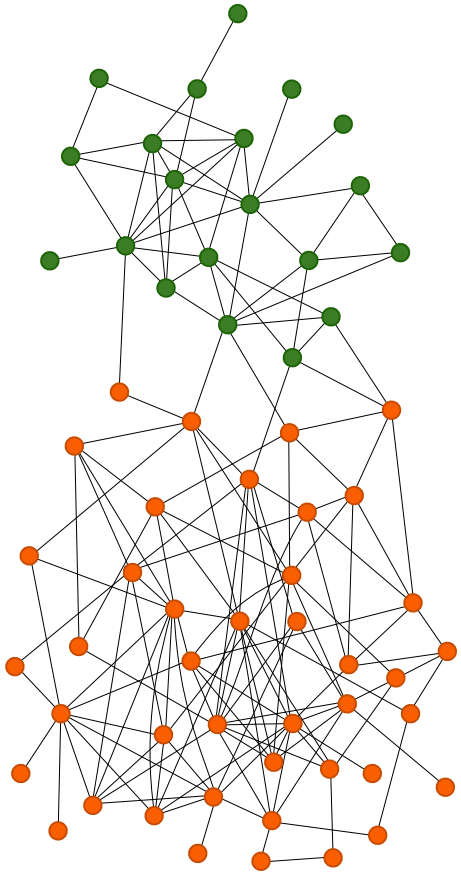
- Social networks
- Network of bloggers
- Collaboration networks
- World Wide Web network

Australian National University

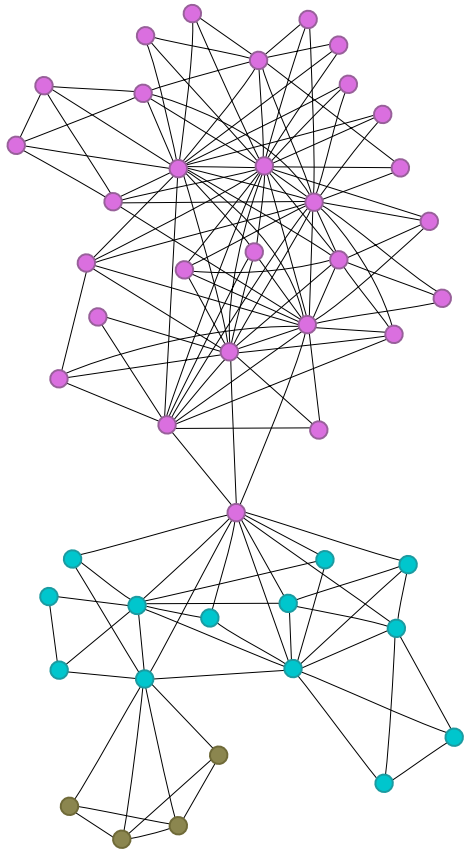
Examples of community structure



(d) Collaboration network



(e) Network of dolphines



(f) PPI network

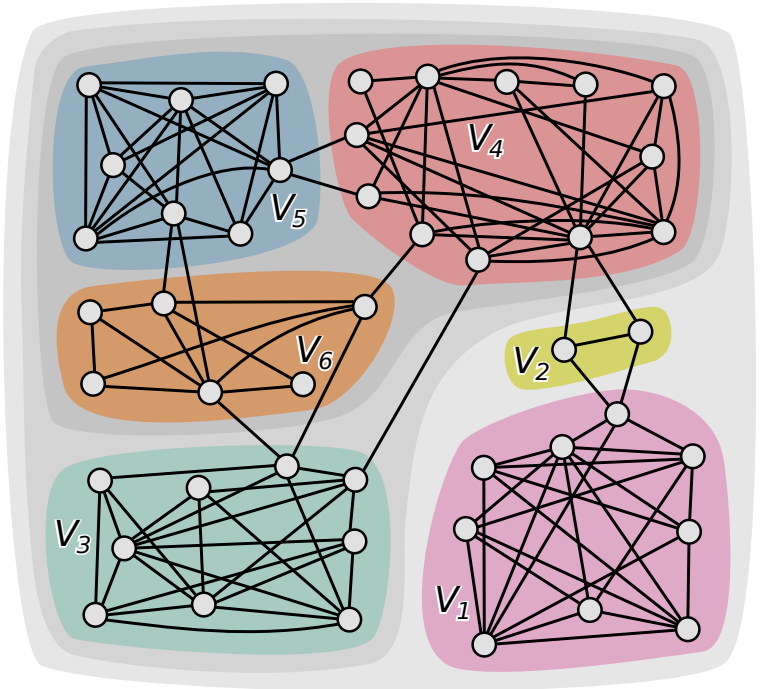
Outline

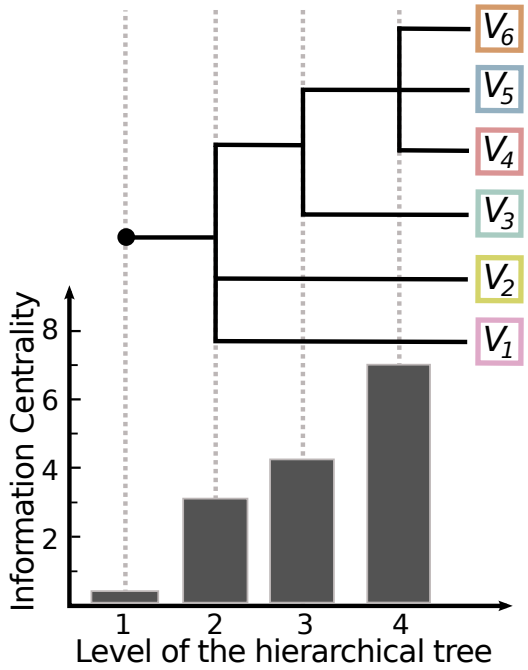
- Introduction
- Hierarchical community structure
- Structural hole spanners
- Overlapping community structure
- Community search
- Conclusion

Australian National University

Hierarchies among communities

Communities can be successively partitioned into denser and denser communities.





(*) Information centrality of a subgraph is the inverse of average distance in the subgraph.

Existing approaches

Top-down: Isolate communities by removing edges based on:

- Betweenness centrality of edges
- Impact on the information centrality
- Impact on the modularity
- Edge-cuts, vertex-cuts and triangles

Bottom-up: Expand communities using fitness metrics from:

- Random vertices
- Maximal cliques
- Maximum degree vertices

Major challenges

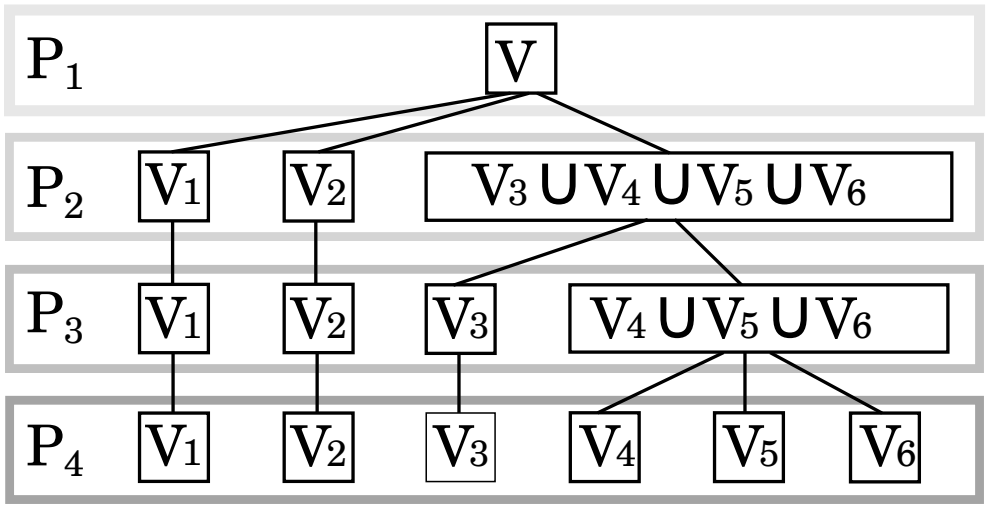
- Scalability of the algorithms
- Accuracy of the modelling

Australian National University

Cohesive hierarchies

Communities in lower levels of the hierarchy are more densely connected to each other.

In a **Cohesive Hierarchy** $\mathcal{P}$, communities at level k are connected with no larger than k edges.



Hierarchical community detection problem

Given a network G, find a cohesive hierarchy $\mathcal{P}$ of communities, where the sum of information centralities of the subgraphs in $\mathcal{P}$ is maximized, i.e.,

$$\text{maximize} \sum_{P_i \in \mathcal{P}} \sum_{V' \in P_i \setminus P_{i-1}} \mathcal{D}(G[V']).$$

Theorem. The hierarchical community detection problem is NP-hard.

Basic algorithm (CHD)

1. Let $P_k = \emptyset$ be the set of communities at level k;
2. Let $P_1 \leftarrow V$, $k \leftarrow 1$;
3. While $(\mathcal{D}(P_k) > \mathcal{D}(P_{k-1}))$
4. $\quad$ Let $k \leftarrow k + 1$;
5. $\quad$ Remove all k-cuts in each $V_i \in P_{k-1}$;
6. Return $\mathcal{P} = \{P_1, \cdots, P_k\}$;

Sparse certificate

We can use a $(k+1)$-sparse certificate of G for finding k-cuts.

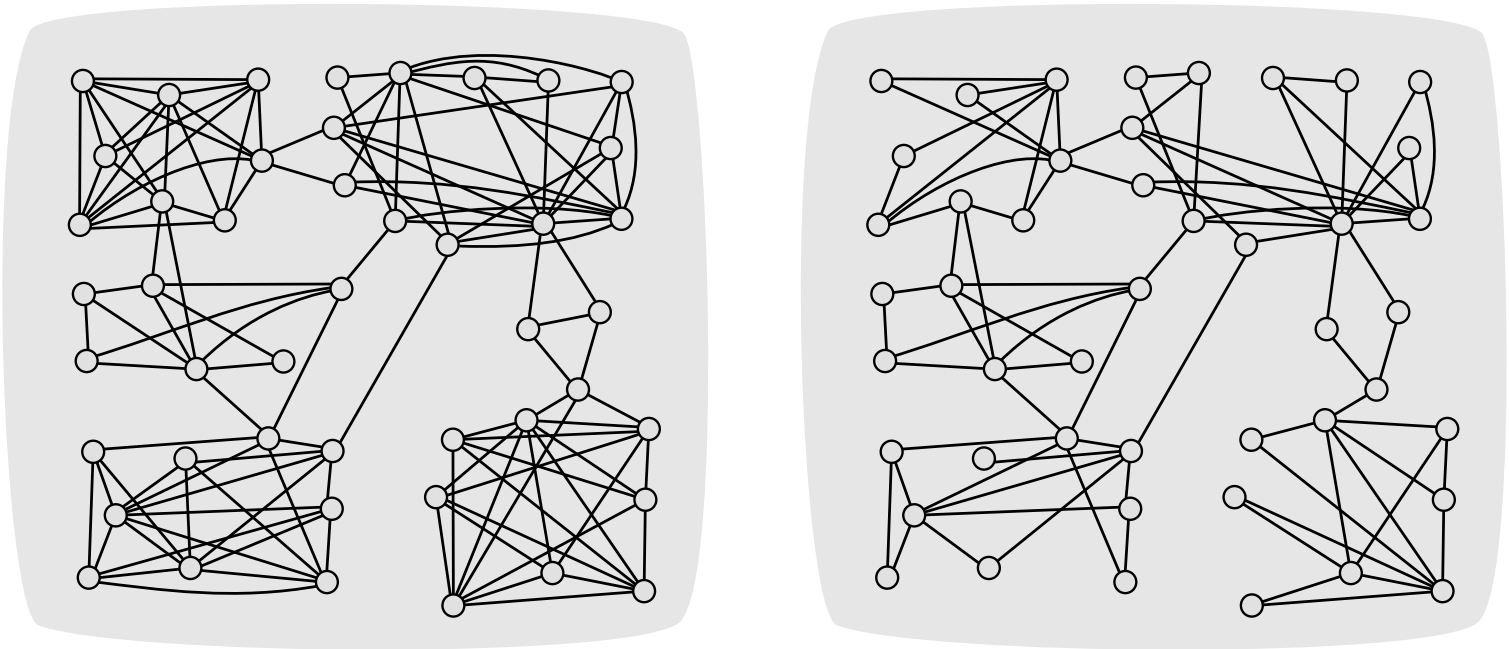


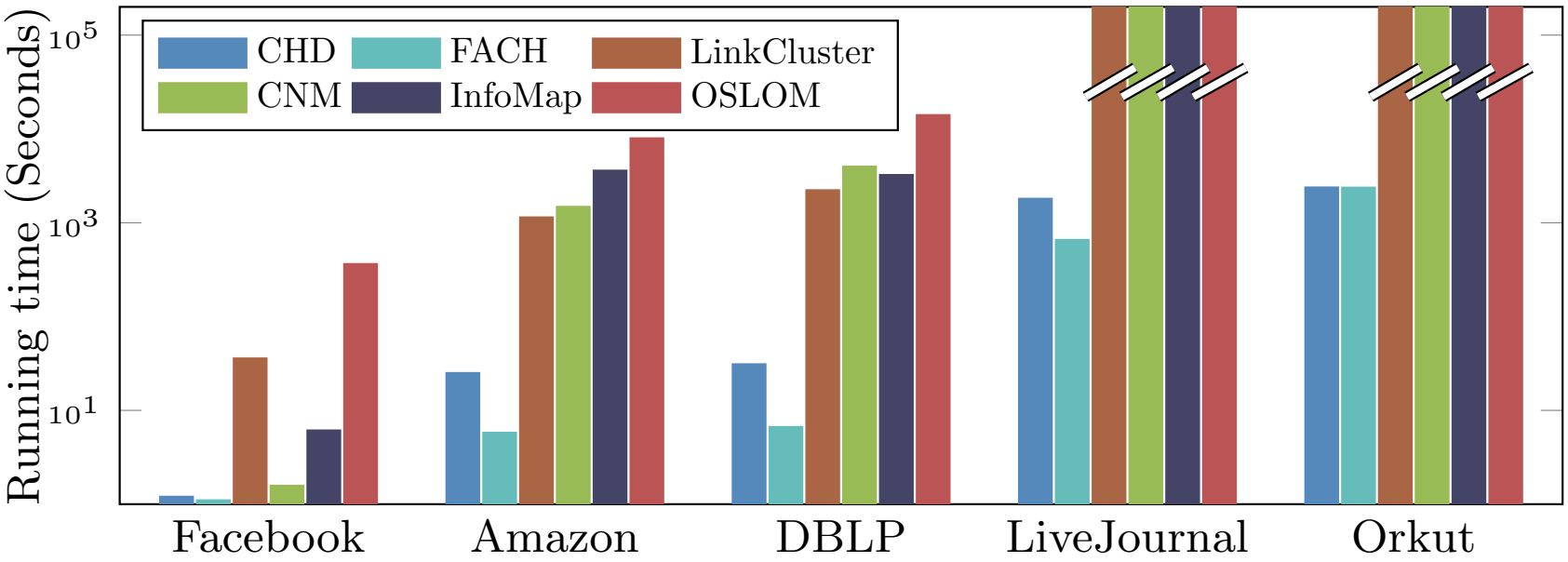
Figure 1 : Graph G and its 2-sparse certificate.

The fast algorithm (FACH)

1. Let $P_k = \emptyset$ be the set of communities at level k;
2. Let $P_1 \leftarrow V$, $k \leftarrow 1$;
3. While $(\mathcal{D}(P_k) > \mathcal{D}(P_{k-1}))$
4. Let $k \leftarrow k + 1$;
5. Find $(k+1)$-sparse certificate of G;
6. Remove all k-cuts in each $V_i \in P_{k-1}$;
7. Return $\mathcal{P} = \{P_1, \cdots, P_k\}$;

Australian National University

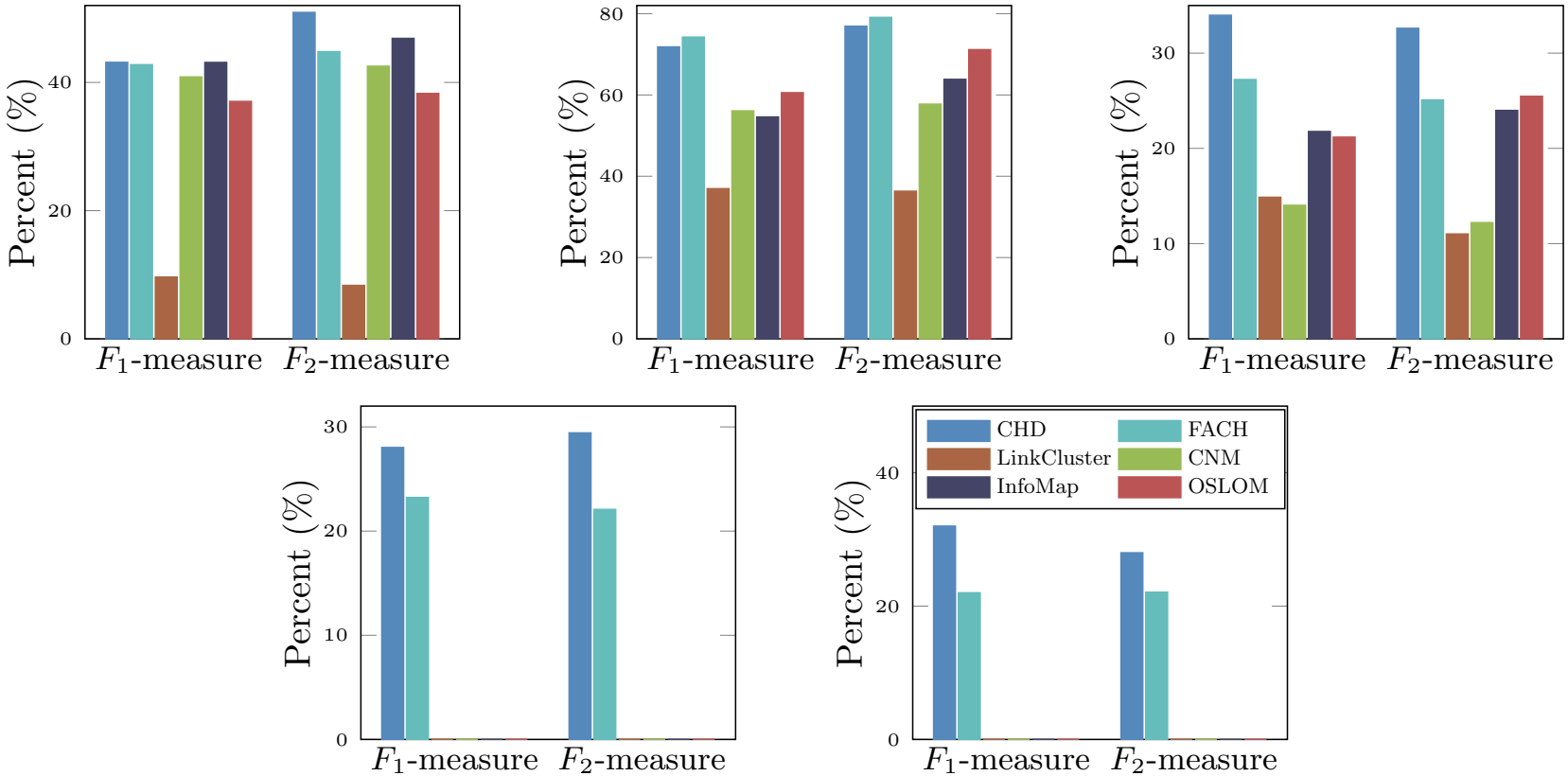
Running Time



CHD and FACH are the proposed algorithms, where CHD does not have the sparsification step and FACH uses the sparsification.

Australian National University

Accuracy



F_1-measure and F_2-measure are calculated by applying a factor α_i to the accuracy of communities at level i.

Outline

- Introduction
- Hierarchical community structure
- Structural hole spanners
- Overlapping community structure
- Community search
- Conclusion

Structural holes

Vertices that can bridge diverse groups and bring benefits to the beholder. Having the following properties,

- *Few neighbours*
- *Many communities*
- *Large communities*

Applications of structural hole spanners

Individuals who fill the structural holes, useful for,

- *Community detection*
- *Information spread control*
- *Disease spread control*
- *Strategic network formation*
- *Graph compression*

Existing approaches

Community-based models:

- Ground-truth communities are not available
- The result is sensitive to the found communities
- Definition of a community varies
- Finding communities in a network is difficult

Structural-based models:

- Vertices that reside on more shortest paths
- Vertices that reside on more shortest paths of length 2

Problem definition

The top-k structural hole spanners are a set V_S of k vertices such that their removal will result in the maximum increase on the sum of all pairs of shortest paths, i.e.,

$$\max_{V_S \subset V} \sum_{u,v \in V \setminus V_S} d_{G \setminus V_S}(u, v) \tag{1}$$

Theorem. The problem of identifying top-k structural hole spanners is NP-hard.

Removal of a vertex that is

- within community doesn't increase the distances significantly
- a structural hole spanners greatly changes the distances
- a structural hole spanners that bridge **more communities** changes the distances more considerably
- a structural hole spanner that bridge **larger communities** is more significant in distances

Algorithm based on closeness centrality

$$C(G \setminus \{v\}) = C(G) - \boxed{2\sum_{u \in V} d_{uv}^{G}} + \sum_{u,w \in V} (d_{uw}^{G \setminus \{v\}} - d_{uw}^{G})$$

Algorithm 1 ICC(G,k)

Input: $G = (V, E), k$

Output: k structural hole spanners

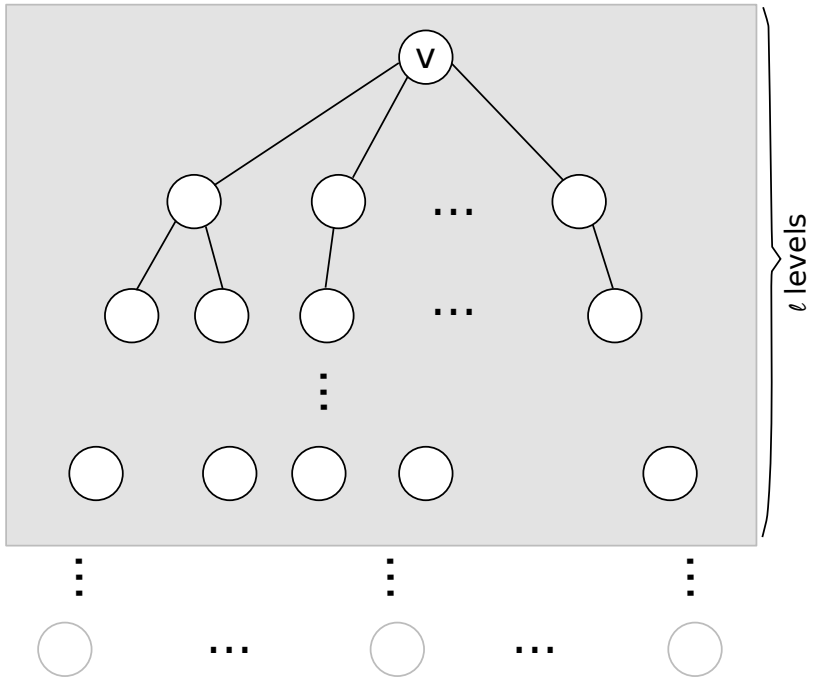
1: Calculate $c(v)$ for every vertex $v \in V$;

2: Find k vertices with smallest $c(v)$ from V;

Australian National University

Bounded inverse closeness centrality

$T_\ell(v)$



$$c_\ell(v) = \sum_{u \in T_\ell(v)} d_{uv}^G/(n-1).$$

Bounded inverse closeness centrality

Algorithm 2 BICC(G,k,K,ℓ)

Input: $G = (V, E), k, K, \ell$

Output: k structural hole spanners

1: Calculate $c_\ell(v)$ for every vertex $v \in V$;
2: Find K vertices, H, with **largest** $c_\ell(v)$, from V;
3: Calculate $c(v)$ for every vertex $v \in V$;
4: Find k vertices with smallest $c(v)$ from H;

Algorithm based on articulation points

Having a virtual edge with weight n^3 between every disconnected pair of vertices. Let $c'(v)$ be the sum of weight of virtual edges in the resulting graph after removal of v from G.

Algorithm 3 AP_BICC(G,k,K,ℓ)

Input: $G = (V, E), k, K, \ell$

Output: k structural hole spanners

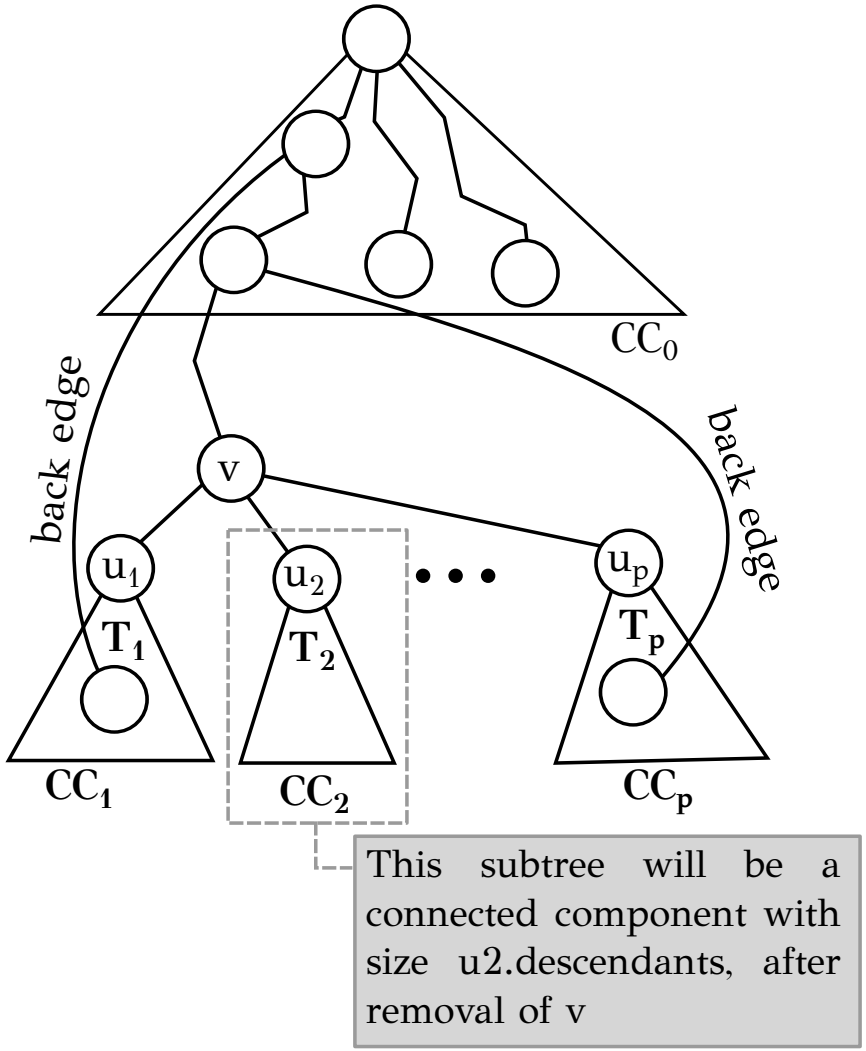
1: Calculate $c'(v)$ for each vertex $v \in V$;
2: **if** number of articulation points k' is larger than k **then**
3: **output** top-k articulation points with the largest $c'(\cdot)$;
4: **else**
5: **output** k' articulation points and $k - k'$ candidates of BICC.

Australian National University

Algorithm based on articulation points

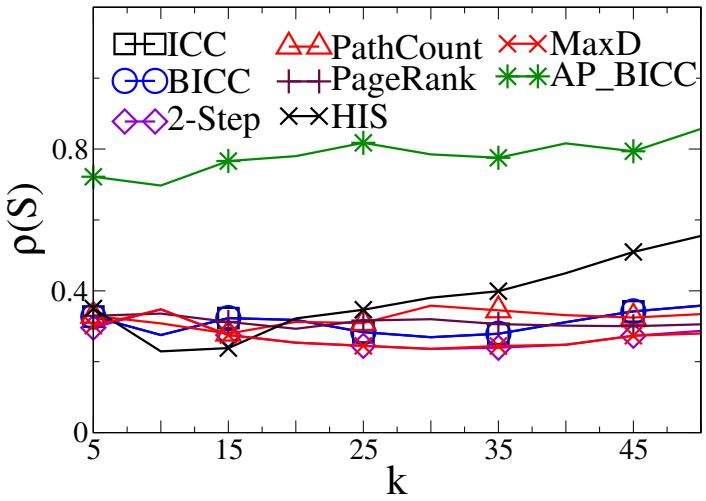
Lemma

Given a graph $G = (V, E)$, the value of $c'(v)$ for all vertices $v \in V$ can be calculated in $O(n + m)$ time.

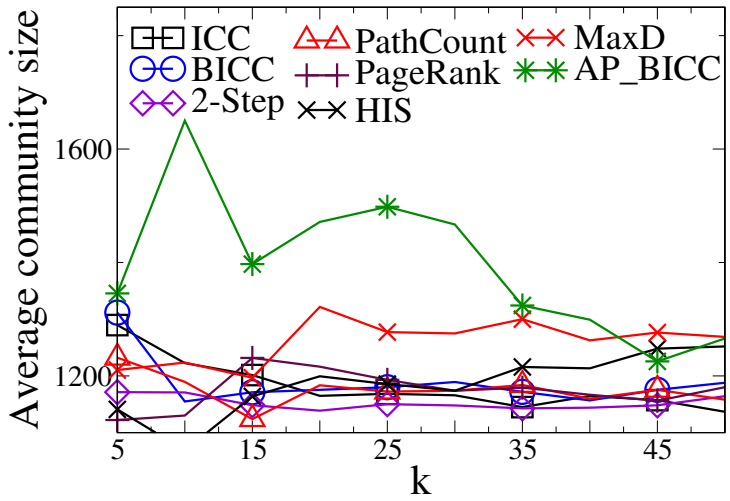


Australian National University

Effectiveness of the model using DBLP



(a) Community membership

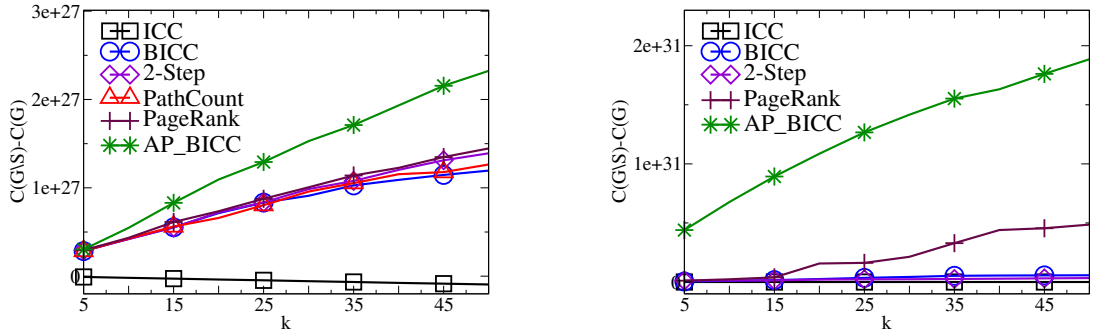


(b) Average community size

Figure 2 : Effectiveness of different algorithms on dataset DBLP using different quality metrics.

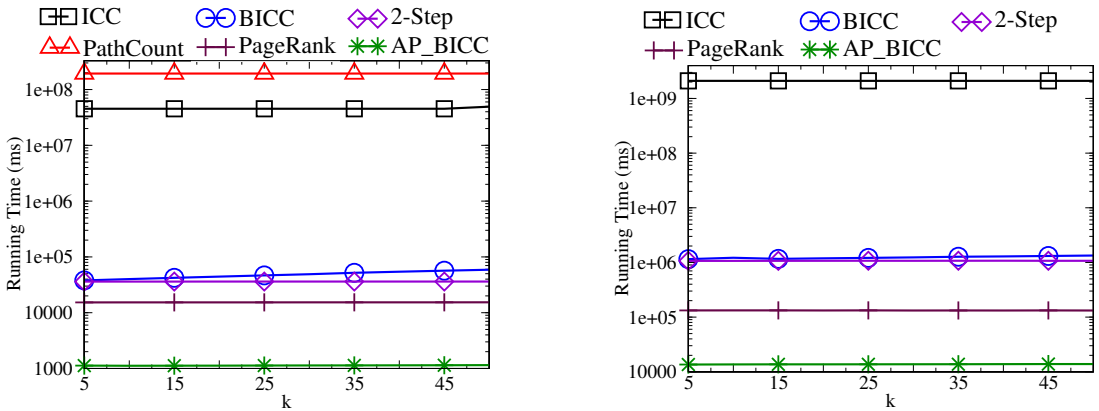
$$\rho(S) = \frac{\sum_{v \in S} (\#\text{ of communities that } v \text{ is connected to})/\deg(v)}{|S|}$$

Performance on real datasets



(a) DBLP-2011 (b) LiveJournal

Figure 3 : Performance of various algorithms, using different datasets.

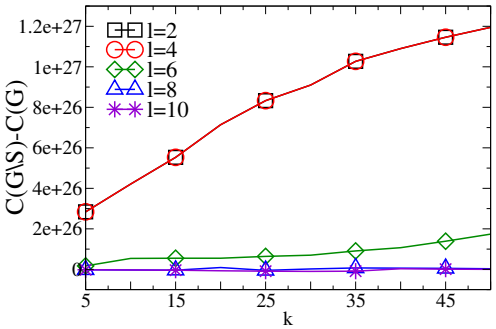


(a) DBLP-2011 (b) LiveJournal

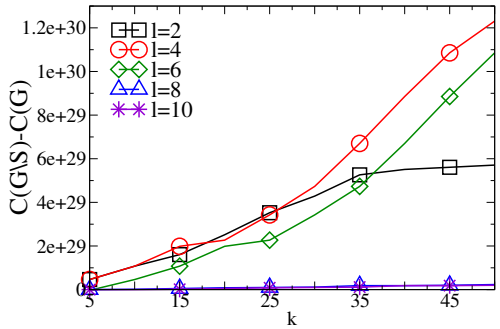
Figure 4 : Running time of various algorithms, using different datasets.

Australian National University

Impact of parameter ℓ on the performance

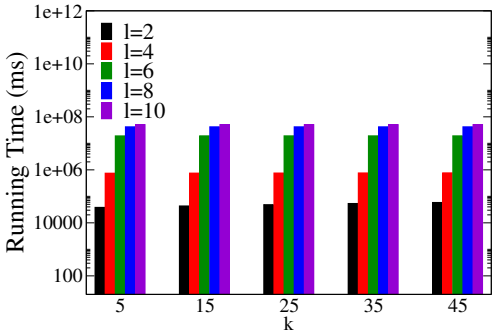


(a) DBLP-2011

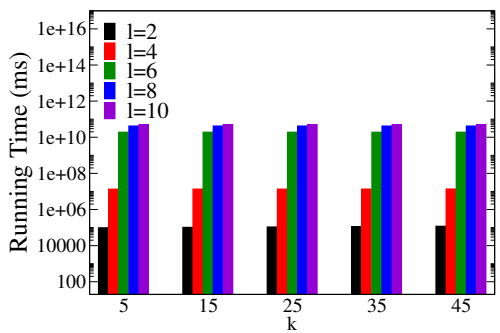


(b) LiveJournal

Figure 5 : Performance of various algorithms, varying parameter ℓ.



(a) DBLP-2011

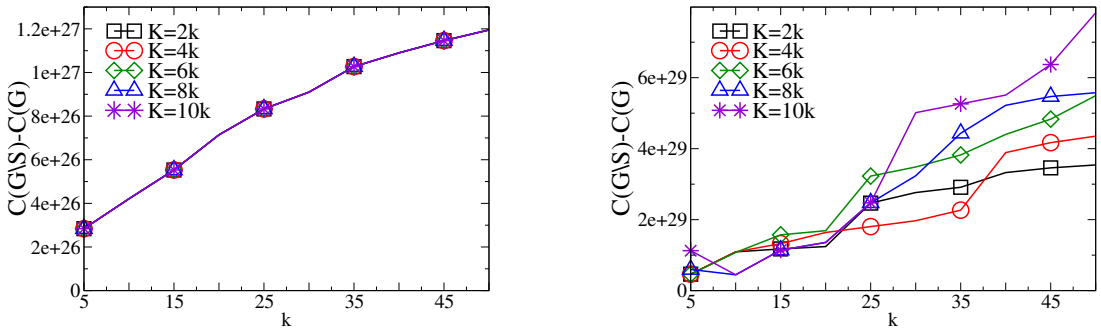


(b) LiveJournal

Figure 6 : Running time of various algorithms, varying parameter ℓ.

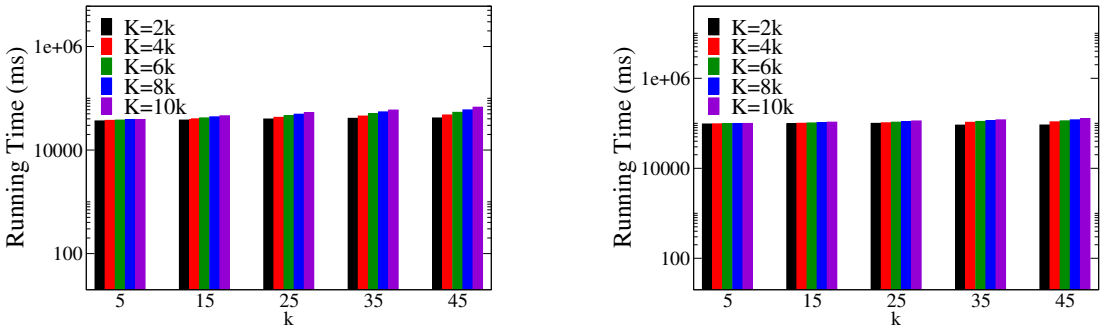
Australian National University

Impact of parameter K on the performance



(a) DBLP-2011 (b) LiveJournal

Figure 7 : Performance of various algorithms, varying parameter K.



(a) DBLP-2011 (b) LiveJournal

Figure 8 : Running time of various algorithms, varying parameter K.

Outline

- Introduction
- Hierarchical community structure
- Structural hole spanners
- Overlapping community structure
- Community search
- Conclusion

Overlapping communiy structure

Communities in complex networks overlap with each other:

- Members of social networks join more than 1 communities
- Researchers work in multidisciplinary areas
- Contents of pages may cover more than one topic

Existing approaches

Fuzzy community detection:

- Non-negative matrix factorization
- Non-linear constrained optimization

Agent-based methods:

- Speaker-Listener Propagation Approach (SLPA)
- Demon

Local expansion methods:

- Based on personal page-rank method and fitness metrics
- Clique expansion methods

Free rider and separation effects

- Free rider effect: when there are two communities A & B,

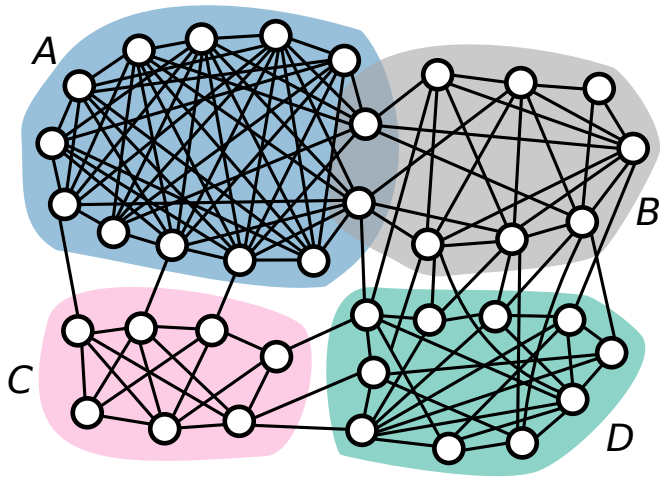
$$f(A) \geq f(A \cup B) \quad \wedge \quad f(B) < f(A \cup B).$$

- Separation effect: when for any two communities A & B,

$$f(A) + f(B) < \max\{f(A) + f(B - A \cap B), f(B) + f(A - A \cap B)\}.$$

Australian National University

Free rider and separation effects



Fitness metrics	Communities	Remarks
Classic density (CD)	A, $A \cup B$	Free rider effect
Relative density (RD)	A, $A \cup B$	Free rider effect
Subgraph modularity (SM)	A, $A \cup B$	Free rider effect
Local modularity (LM)	B, $A - B$	Separation effect
Conductance (CN)	A, $B - A$	Separation effect

Australian National University

Overlapping community fitness metric

- More triangles in a community imply a better community
- Community triangles: all its vertices belong to one community
- Fewer triangles between communities imply cohesiveness
- Cut triangle: two of its vertices are in one community, and one vertex is in another community

$$\tau(C) = \frac{(\#\text{ of community triangles in } C)}{(\#\text{ of cut triangles in } C) + (\text{overlap size}) + |C|}$$

Theorem. The overlapping community detection problem is NP-hard.

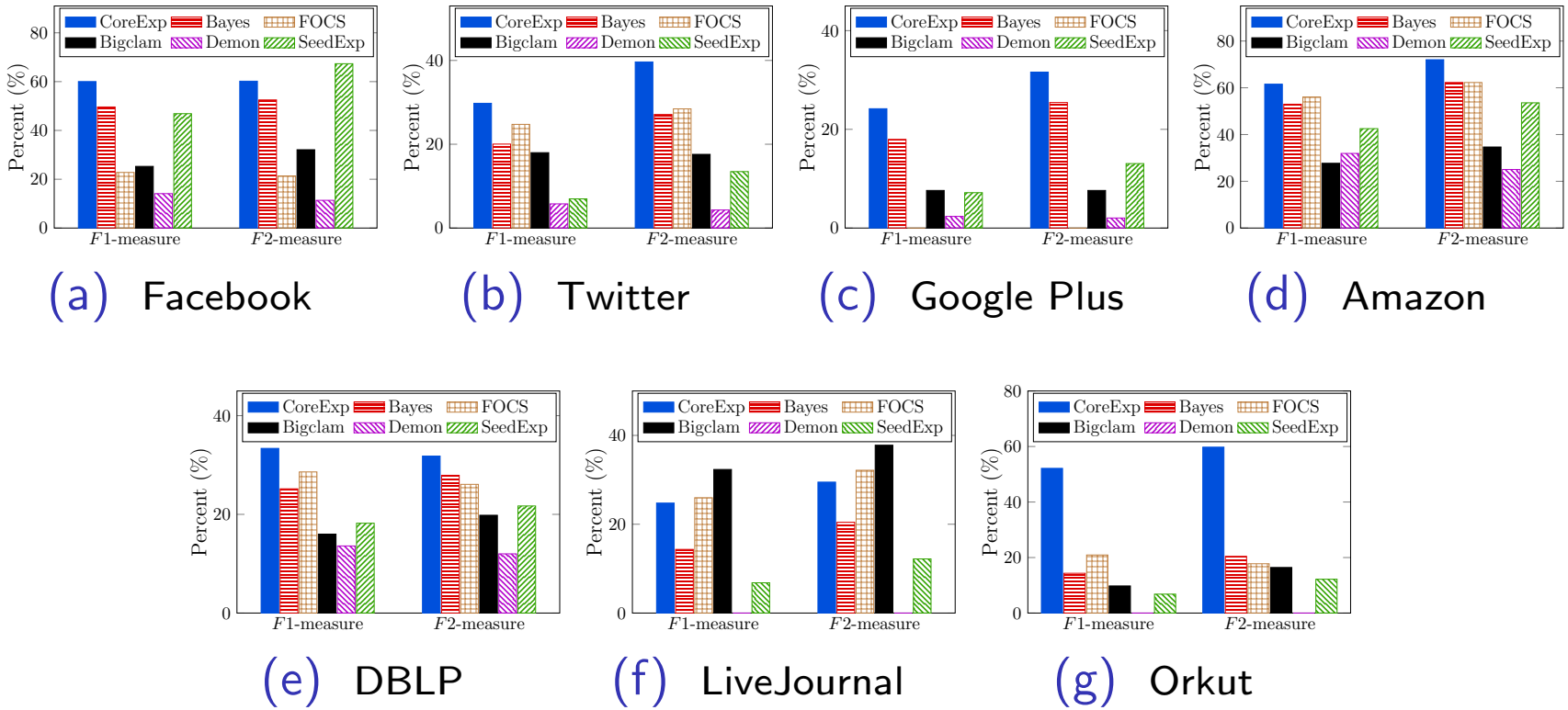
Overlapping community detection algorithm

Algorithm 4 OverlappingCommunityDetection(G)

$\mathcal{C}_0 \leftarrow \{V\}$; $k \leftarrow 0$;
/* Find core communities */
while there is an edge with support no larger than k **do**
    for each community $C \in \mathcal{C}_k$ **do**
        Update C in $\mathcal{C}_{k+1}$ if removal of edges increases τ;
    $k \leftarrow k + 1$;
/* Expand core communities in $\mathcal{C}_k$ */
for each core $C \in \mathcal{C}$ **do**
    for each neighbor u of vertex v **do**
        if $\tau(C \cup \{u\}) \geq \tau(C)$ **then**
            Add the neighbor u to C;
return $\mathcal{C}$;

Australian National University

Empirical evaluations



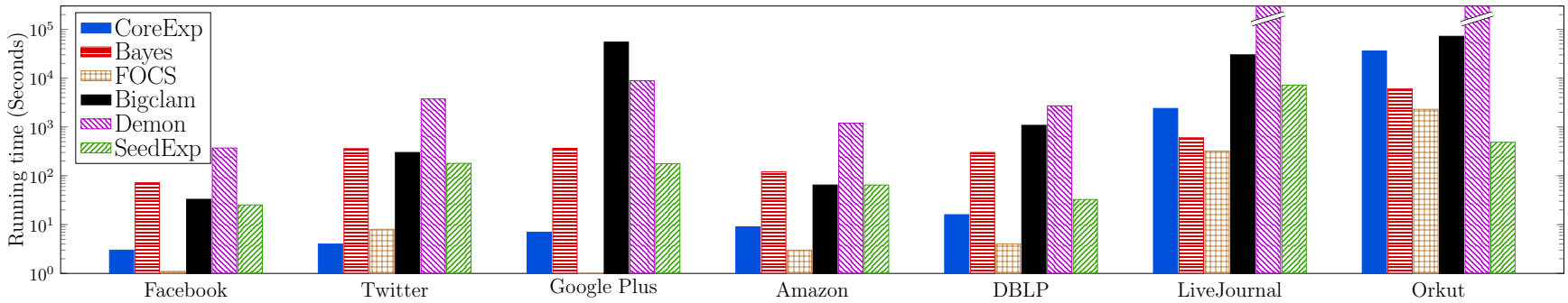
(a) Facebook (b) Twitter (c) Google Plus (d) Amazon

(e) DBLP (f) LiveJournal (g) Orkut

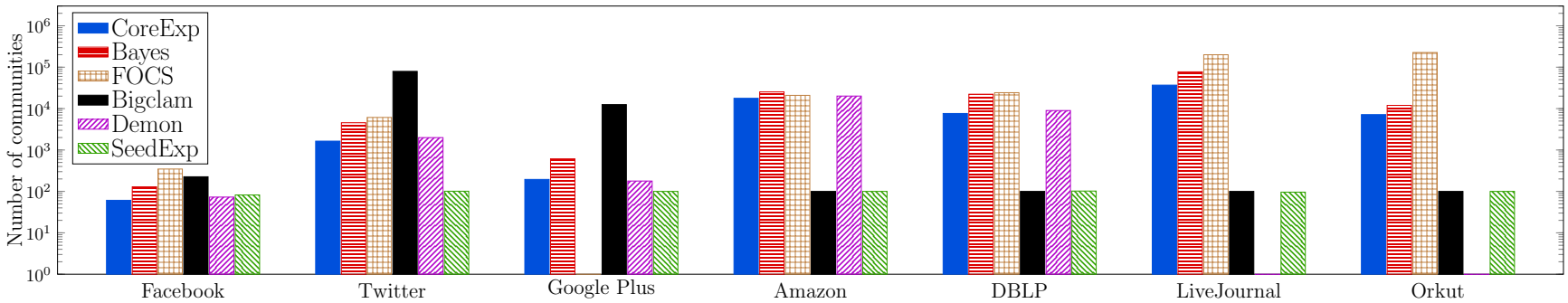
Figure 9 : The quality of overlapping communities found.

Australian National University

Empirical evaluations



(a) Running times of different algorithms



(b) Number of communities found

Outline

- Introduction
- Hierarchical community structure
- Structural hole spanners
- Overlapping community structure
- Community search
- Conclusion

Community search vs. community detection

- Given a set of query vertices, we are interested in finding communities of those vertices.
- In community detection, we are interested in finding all communities of a network.

Existing approaches

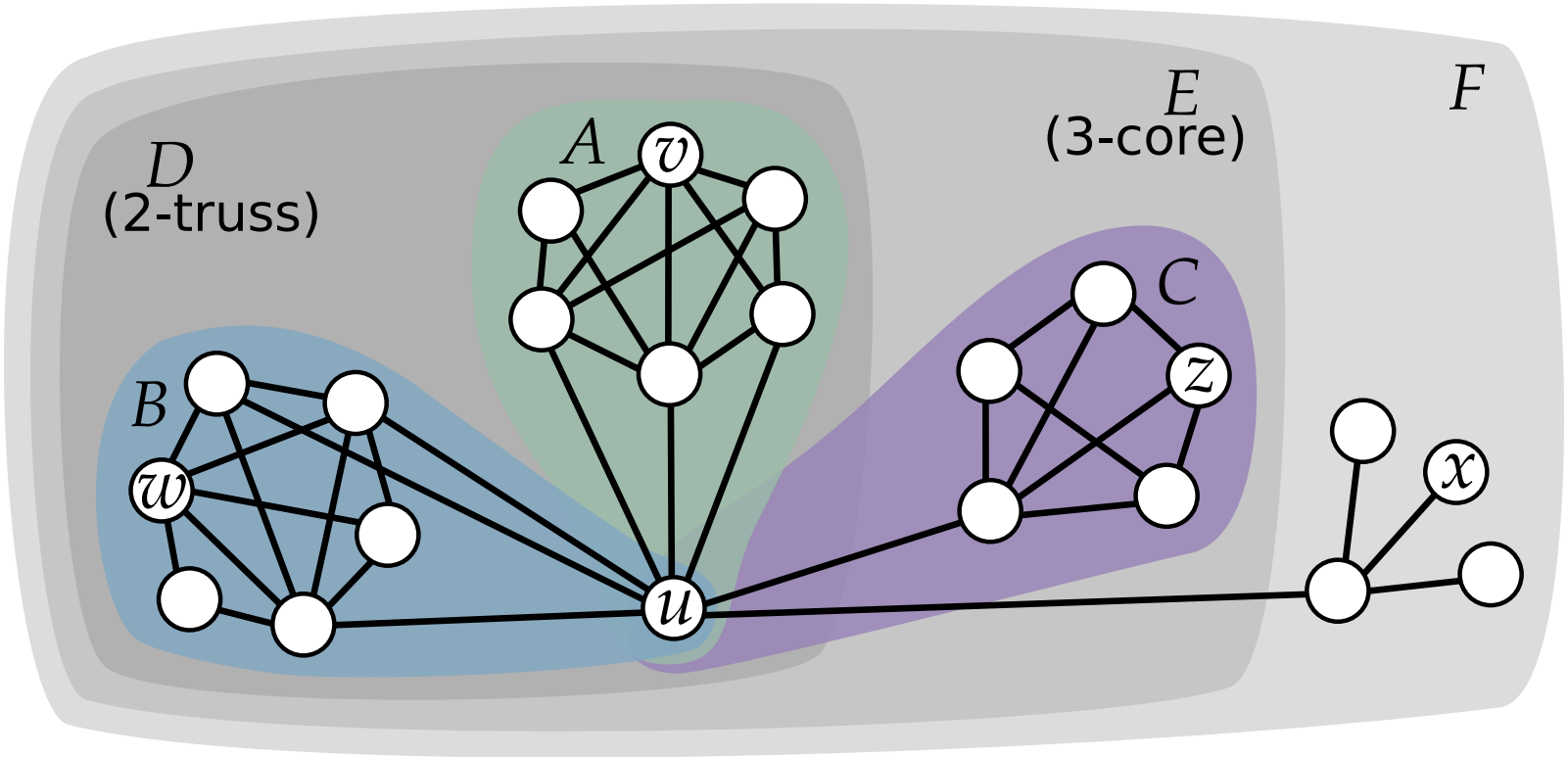
Find only one community that includes the query vertices:

- k-core
- k-truss
- k-truss with minimum diameter
- γ-quasi k-cliques
- γ-quasi k-cliques with k triangles

Australian National University

Illustrative example

Query vertices may belong to different communities:



Propinquity

The propinquity value between vertices u and v, i.e. $\Delta(u, v)$, represents the closeness between them.

Propinquity measures can have properties such as,

- Reflexive
- Symmetric
- Transitive (with respect to a value k)

Community search problem

Find a minimum cover $\mathcal{C} = \{V_1, \cdots, V_t\}$ $(t \leq |Q|)$ such that:

- the propinquity between every two vertices u and v in V_i $(V_i \in \mathcal{C})$ is no less than k, i.e. $\Delta(V_i) \geq k$,
- every community $V_i \in \mathcal{C}$ is maximal, and
- for every pair of query vertices $q_i, q_j \in Q$,

$$\Delta(q_i, q_j) \geq k \Rightarrow \exists\ V_p \in \mathcal{C}\ \ (\{q_i, q_j\} \subseteq V_p).$$

Theorem. The problem of identifying top-k structural hole spanners is NP-hard.

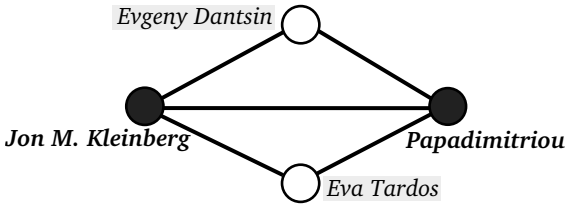
A novel propinquity measure

- A small length of shortest path between vertices denotes the closeness between them
- Hubs may obscure the accuracy of shortest paths
- The sum of top-k shortest paths between two vertices denotes propinquity between two vertices

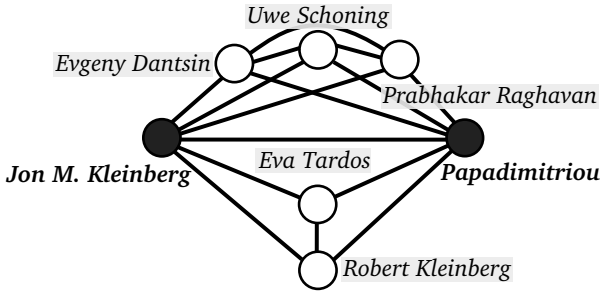
The number of top-k shortest paths between u and v whose sum is no larger than ℓ.

Australian National University

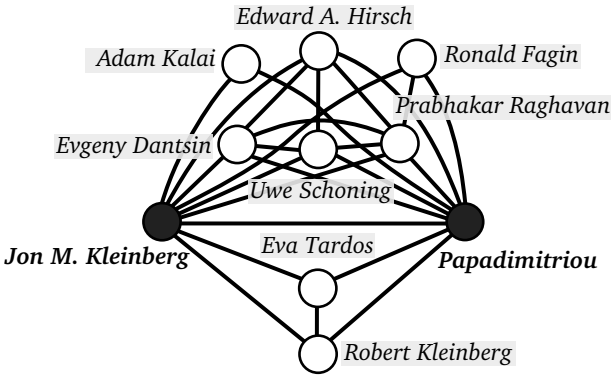
An illustrative example



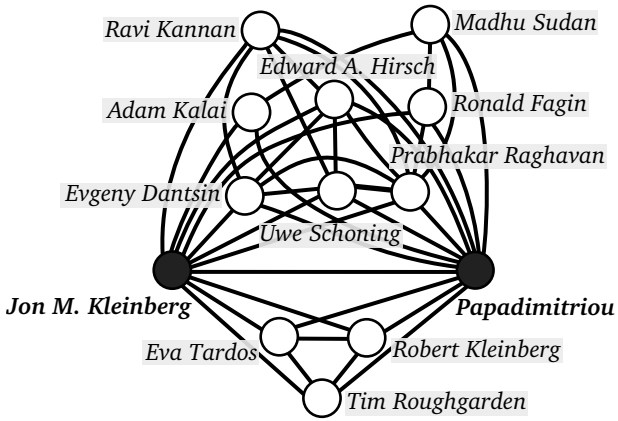
(c) $\ell = 5$



(d) $\ell = 11$



(e) $\ell = 17$



(f) $\ell = 23$

Clique-based algorithm for community search

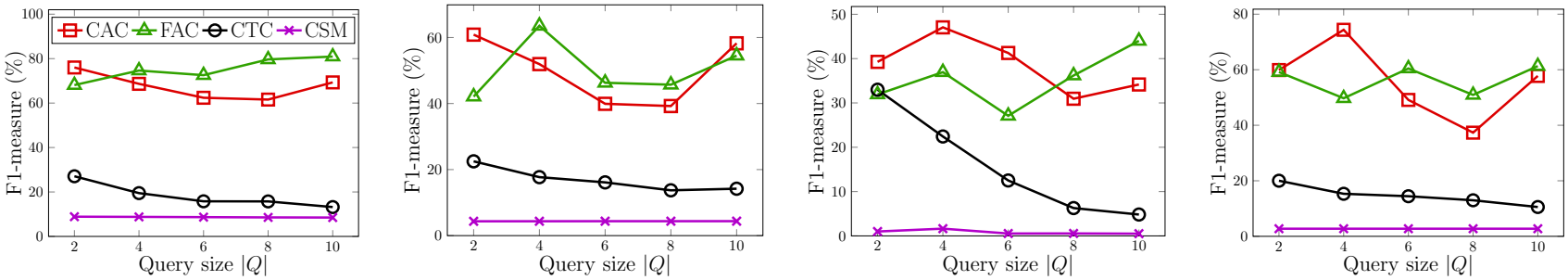
1. Find the *query relevance graph* (QRG);
2. Find **cliques** of QRG;
3. For each **clique** in QRG:
4. Discover the top-k shortest paths between query vertices in the clique;
5. Return the expanded communities;

Fast algorithm for community search

1. Find the *query relevance graph* (QRG);
2. Find **connected components** of QRG;
3. For each **connected component** in QRG:
4. Discover the top-k shortest paths between query vertices in the clique;
5. Return the expanded communities;

Australian National University

Empirical evaluations



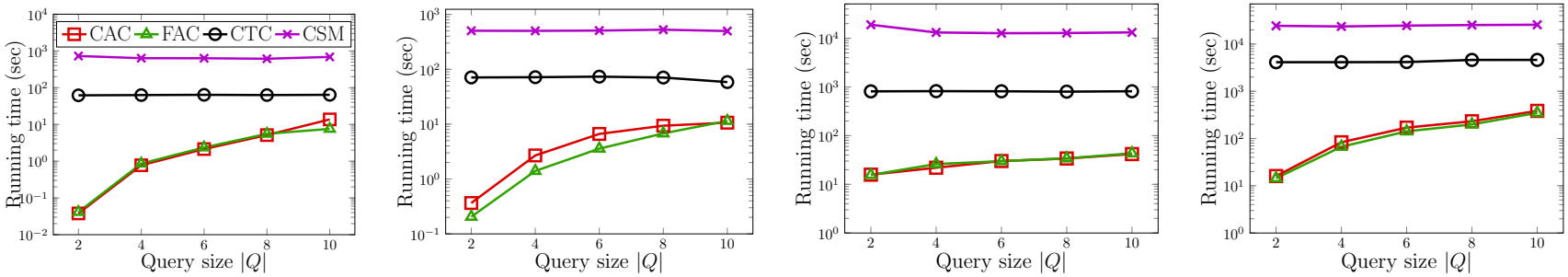
(g) Amazon (h) DBLP (i) Youtube (j) LiveJournal

(k) Amazon (l) DBLP (m) Youtube (n) LiveJournal

Figure 10 : F1-measure and F2-measure of different algorithms on random queries of various sizes.

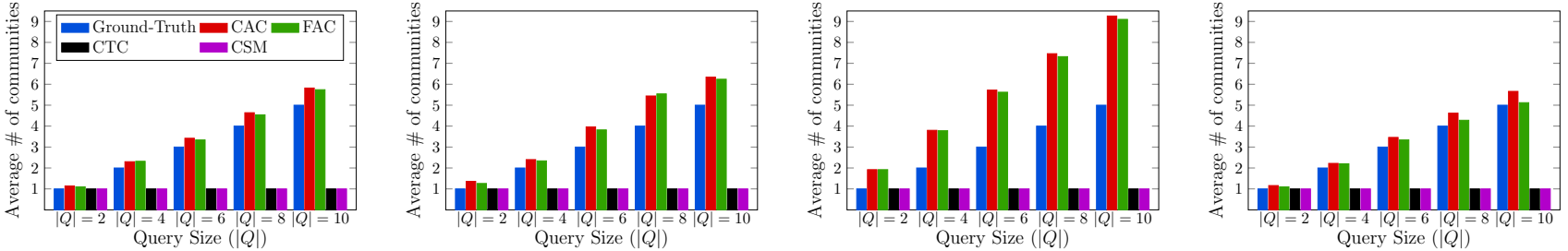
Australian National University

Empirical evaluations



(a) Amazon (b) DBLP (c) Youtube (d) LiveJournal

Figure 11 : Running time of different algorithms on random queries.

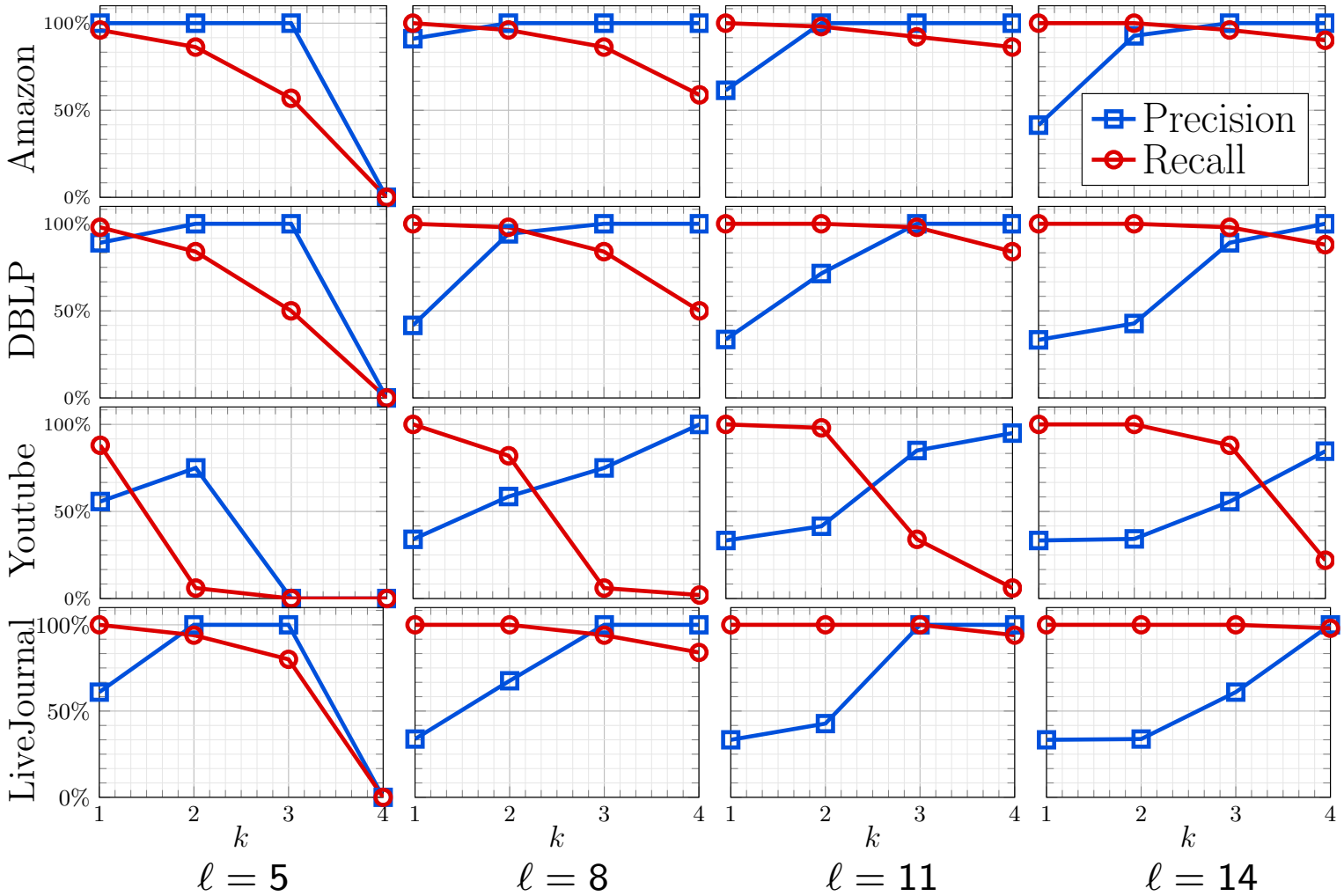


(a) Amazon (b) DBLP (c) Youtube (d) LiveJournal

Figure 12 : Average number of communities detected for each queries.

Australian National University

Parameter evaluations



Outline

- Community structure in complex networks
- Hierarchical community structure
- Structural hole spanners
- Overlapping community structure
- Community search
- Conclusion

Conclusion & future works

Conclusion

- The thesis formally defined each problem
- The thesis devised scalable algorithms for them
- The thesis evaluated the performance of the proposed algorithms by comparing against benchmark algorithms

Future works

- Extension to weighted and directed graphs
- Using different edge and vertex weighted methods
- Extension to dynamically evolving networks

Thank You