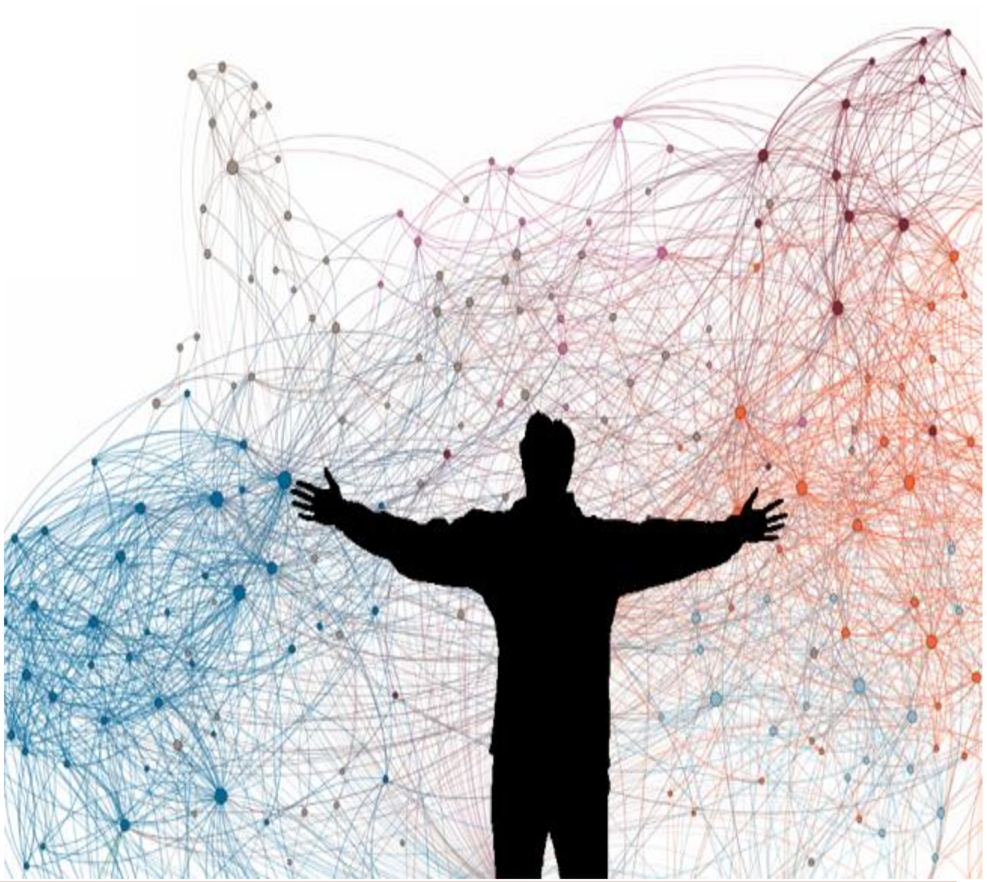
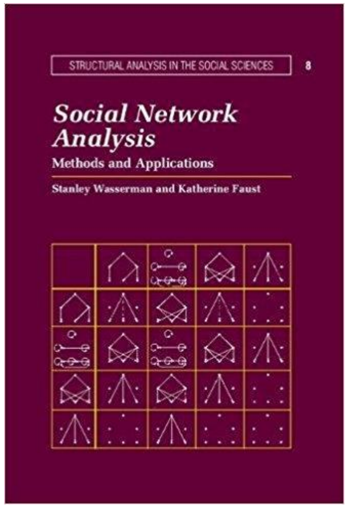
Searching Communities in Large Social Networks

Jeffrey Xu Yu (于旭)
The Chinese University of Hong Kong
yu@se.cuhk.edu.hk, http://www.se.cuhk.edu.hk/~yu

Social Networks



Cohesive Subgraphs



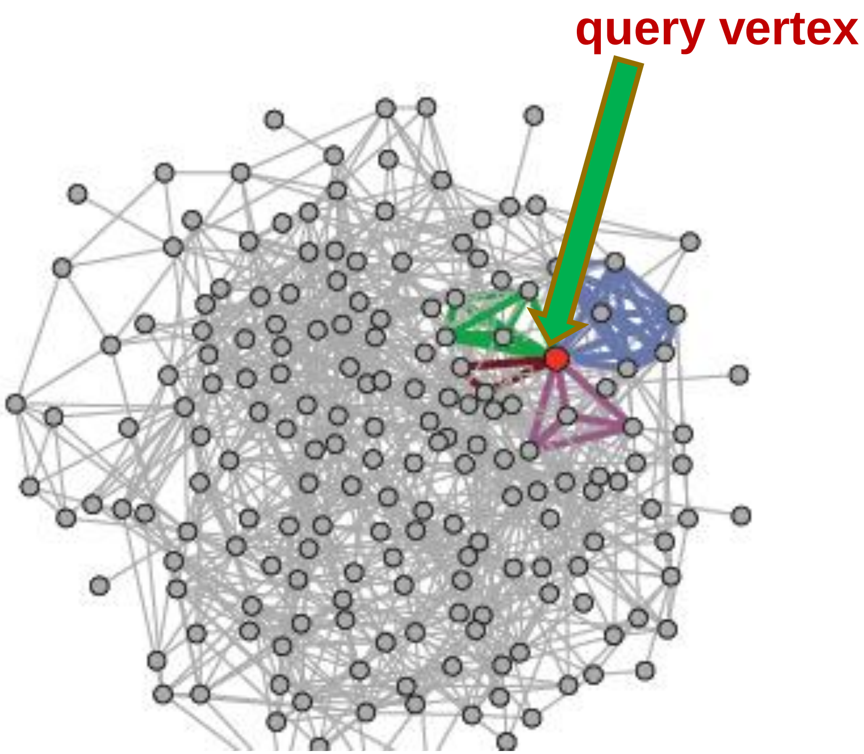
- One of the major issues in social networks is to find cohesive subgraphs.
- Cohesive subgraphs are subsets of people who have relative strong, direct, intense, frequent, or positive ties.
- The role of social cohesiveness is discussed in social explanations.
- By Collins (1988): *"The more tightly that individuals are tied into a network, the more they are affected by group standards", "how many ties an individual has to the group and how close the entire group is to outsiders".*

Some Dense Subgraphs

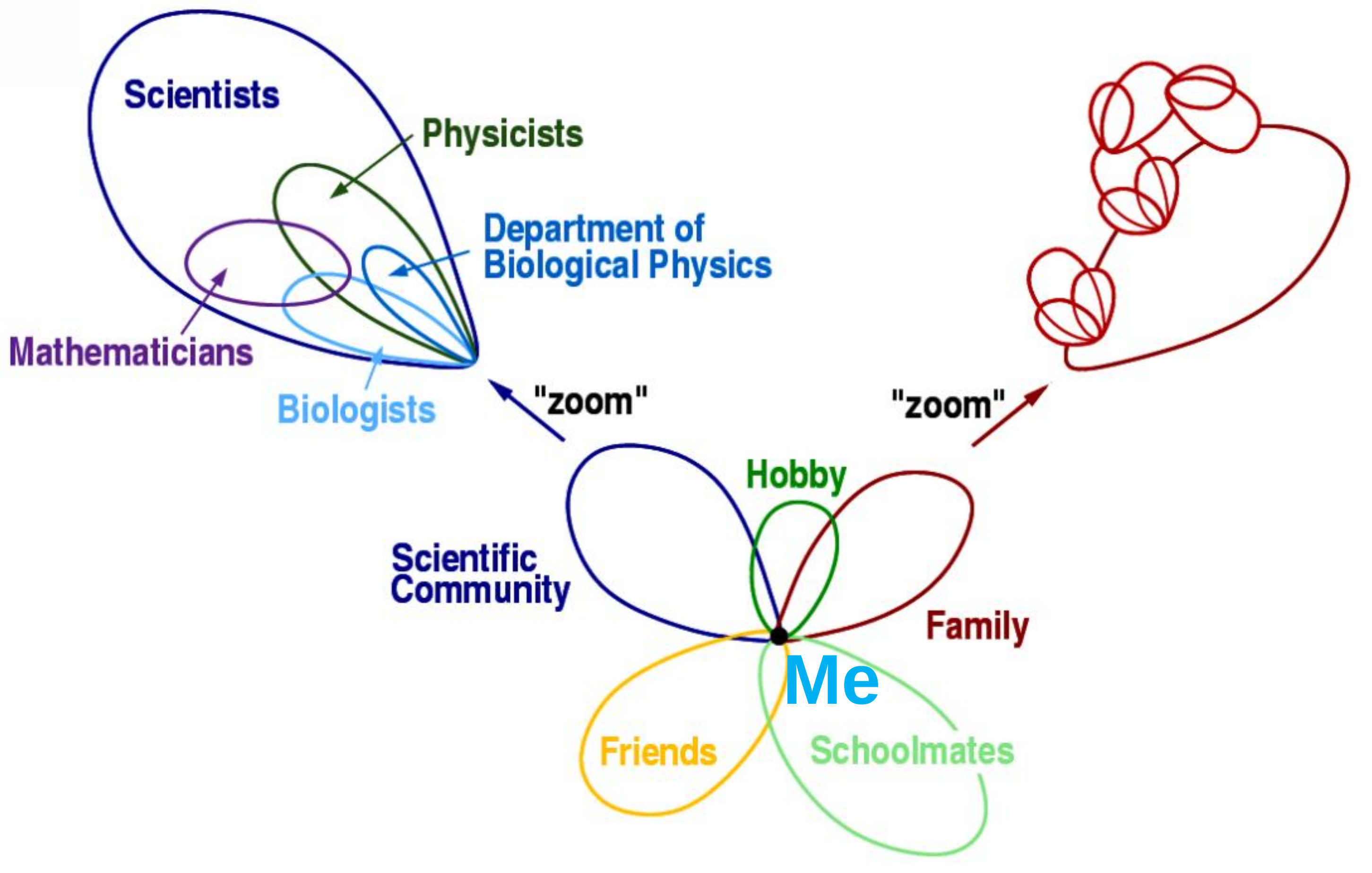
- k-clique: a complete subgraph of k nodes.
  - Maximal Clique Enumeration
  - Maximum Clique Problem
- k-core: The maximal subgraph in which every node is with k-degree.
- k-truss: The maximal subgraph in which every edge is contained in at least $(k-2)$ triangles.
- k-edge-connected: The maximal subgraph which is connected by removing $(k-1)$ edges.
- ……

Community Search/Detection

- Community Detection:
  - Find all communities with a global criterion
  - Expensive computation
  - Graphs evolve
- Community Search:
  - Find communities for particular persons
  - Less expensive
  - Online and dynamic



Overlapping Communities

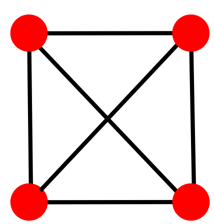


- An individual belongs to many social circles

OCS Method [Cui et al., SIGMOD'13]

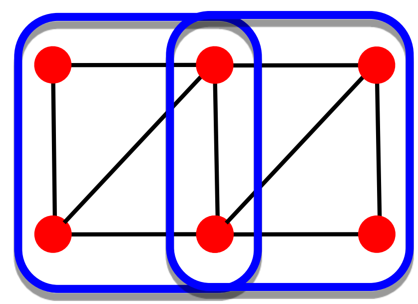
- **α-adjacency–γ-quasi-k-clique community model**
  - **α-quasi-k-clique:** a k-node graph with at least $\lfloor \gamma k(k-1)/2 \rfloor$ edges.
  - **α-adjacency-γ-quasi-k-clique:** overlap α vertices, where $\alpha \leq k-1$.

k-clique: a complete graph of k nodes with $k(k-1)/2$ edges.



k-cliques (k=4)

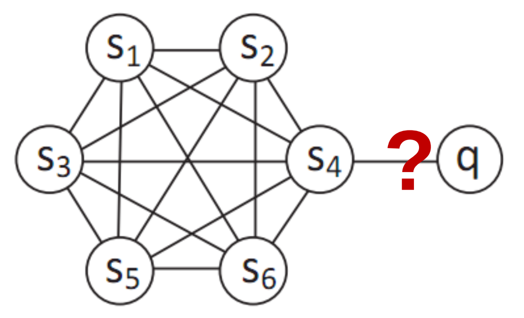
γ-quasi-k-cliques (γ=0.8, k=4)



α-adjacency-γ-quasi-k-cliques (α=2, γ=0.8, k=4)

OCS Method [Cui et al., SIGMOD'13]

- Given a query vertex q in graph G, the problem is to find all α-adjacency-γ-quasi-k-clique containing q.
- Limitations:
  - No cohesive guarantee
  - Three parameters
  - NP-hard problem



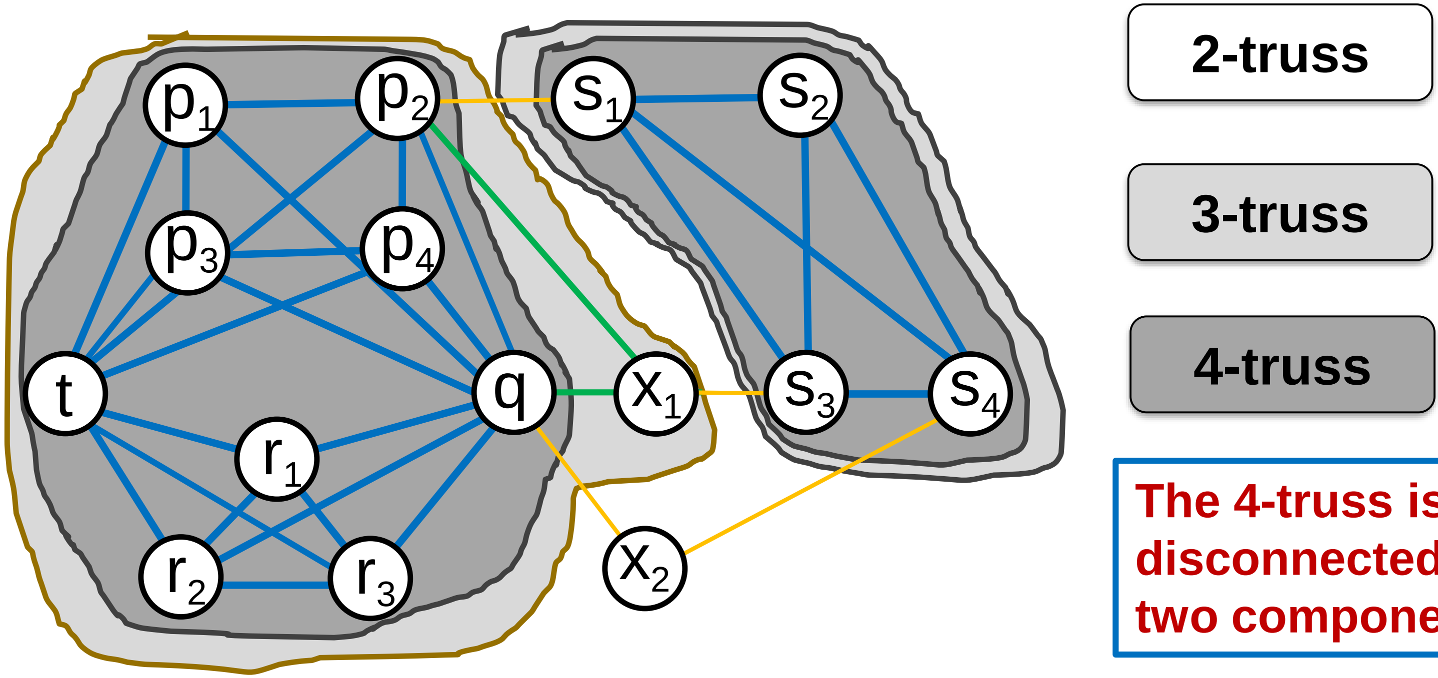
A 0.8-quasi-7-clique containing q

Querying K-Truss Community in Large and Dynamic Graphs [SIGMOD'14]

Xin Huang, Hong Cheng, Lu Qin, Wentao Tian, Jeffrey Xu Yu

K-Truss

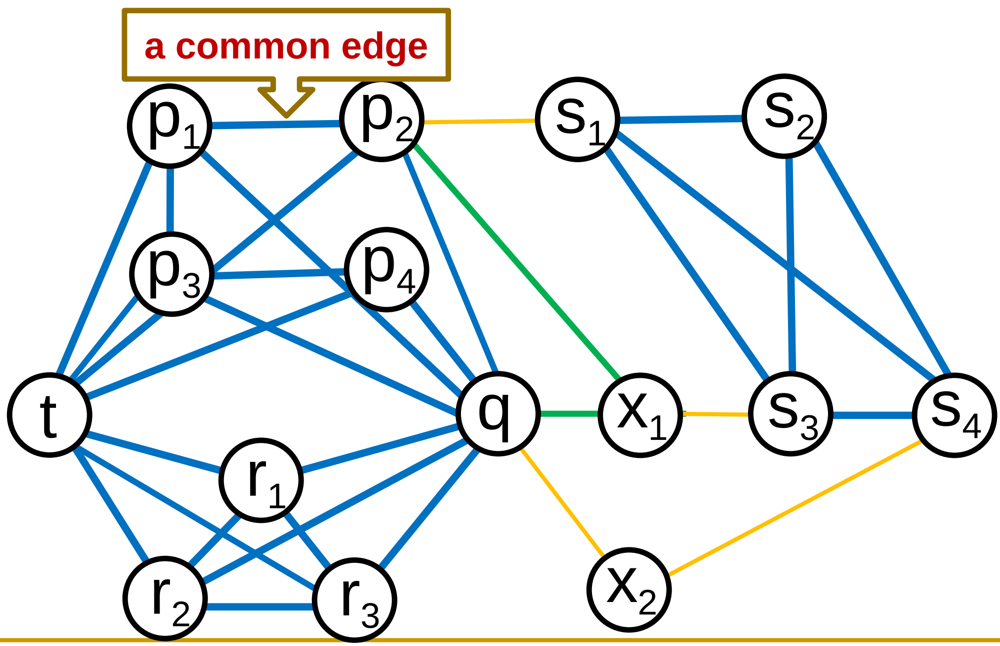
- k-truss of graph G: the largest subgraph H s.t. every edge in H is contained in *at least* $(k-2)$ *triangles* within H.



The 4-truss is disconnected with two components

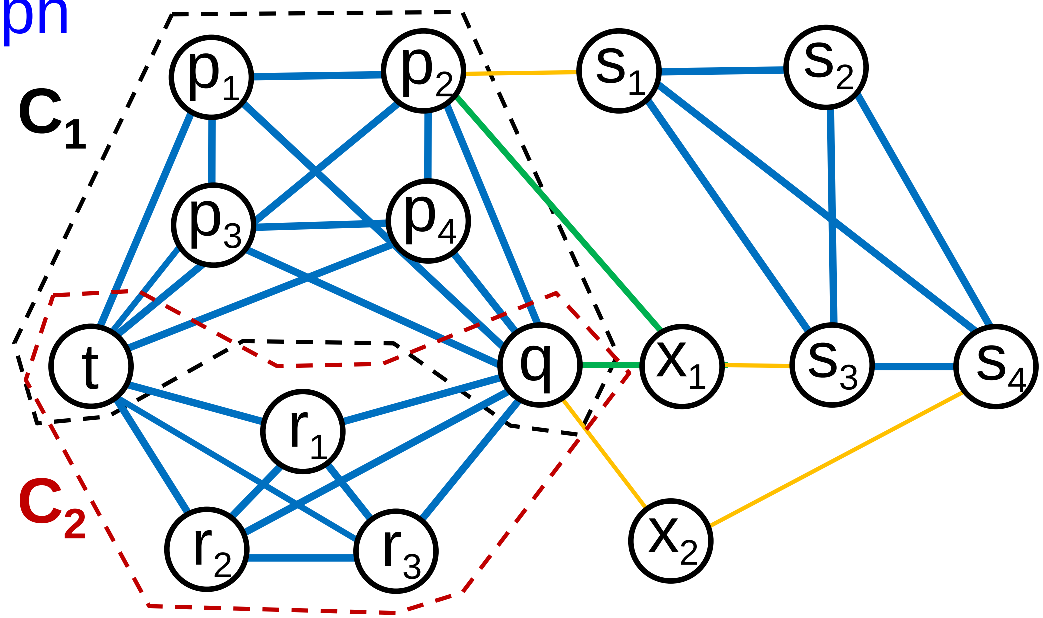
Edge Connectivity

- Triangle adjacency: $\Delta_1 \cap \Delta_2 \neq \emptyset$
- Edge connectivity in graph G':
  - $e_1 \in \Delta_1$, $e_2 \in \Delta_2$
  - $\Delta_1 = \Delta_2$ or Δ_1 is triangle connected with Δ_2 in graph G'.



A K-Truss Community Model

- A k-truss community satisfies:
  - (1) K-truss: each edge within *at least $(k-2)$ triangles*
  - (2) Edge Connectivity: all pairs of edges
  - (3) Maximal Subgraph

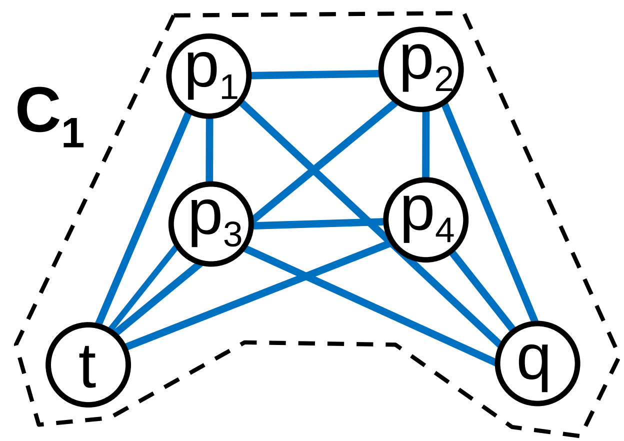


Two 4-truss communities for q

Why K-Truss Community?

- Cohesive structure
  - Bounded diameter
    - A k-truss community with $|C|$ vertices, the diameter is no larger than $\lfloor (2|C|-2)/k \rfloor$.
  - $(k-1)$-edge-connected graph
- Only one parameter to set
- Polynomial time

A 4-truss community

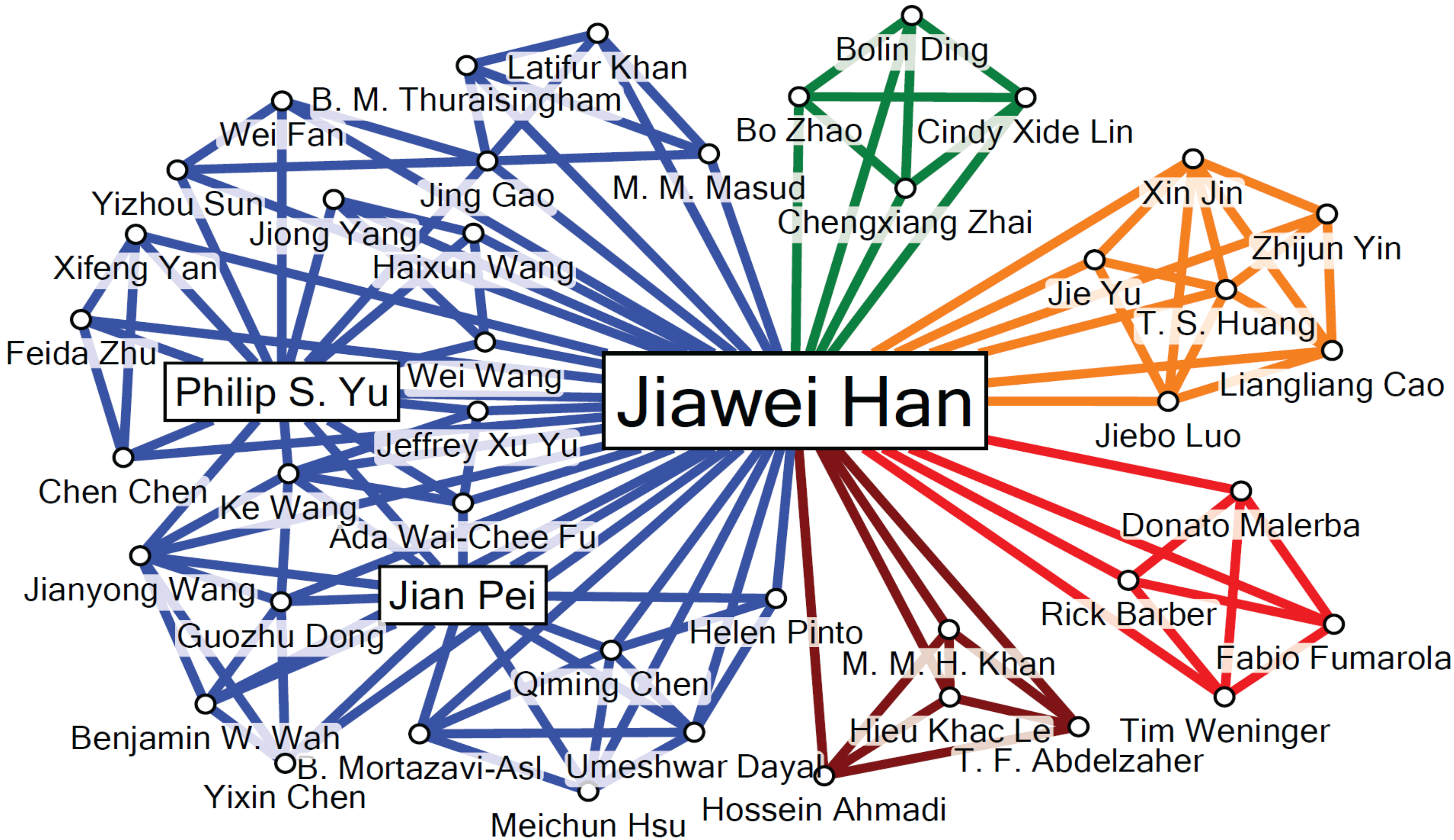


Diameter upper bound
$\lfloor (2 \times 6 - 2)/4 \rfloor = 2$

Problem Formulation

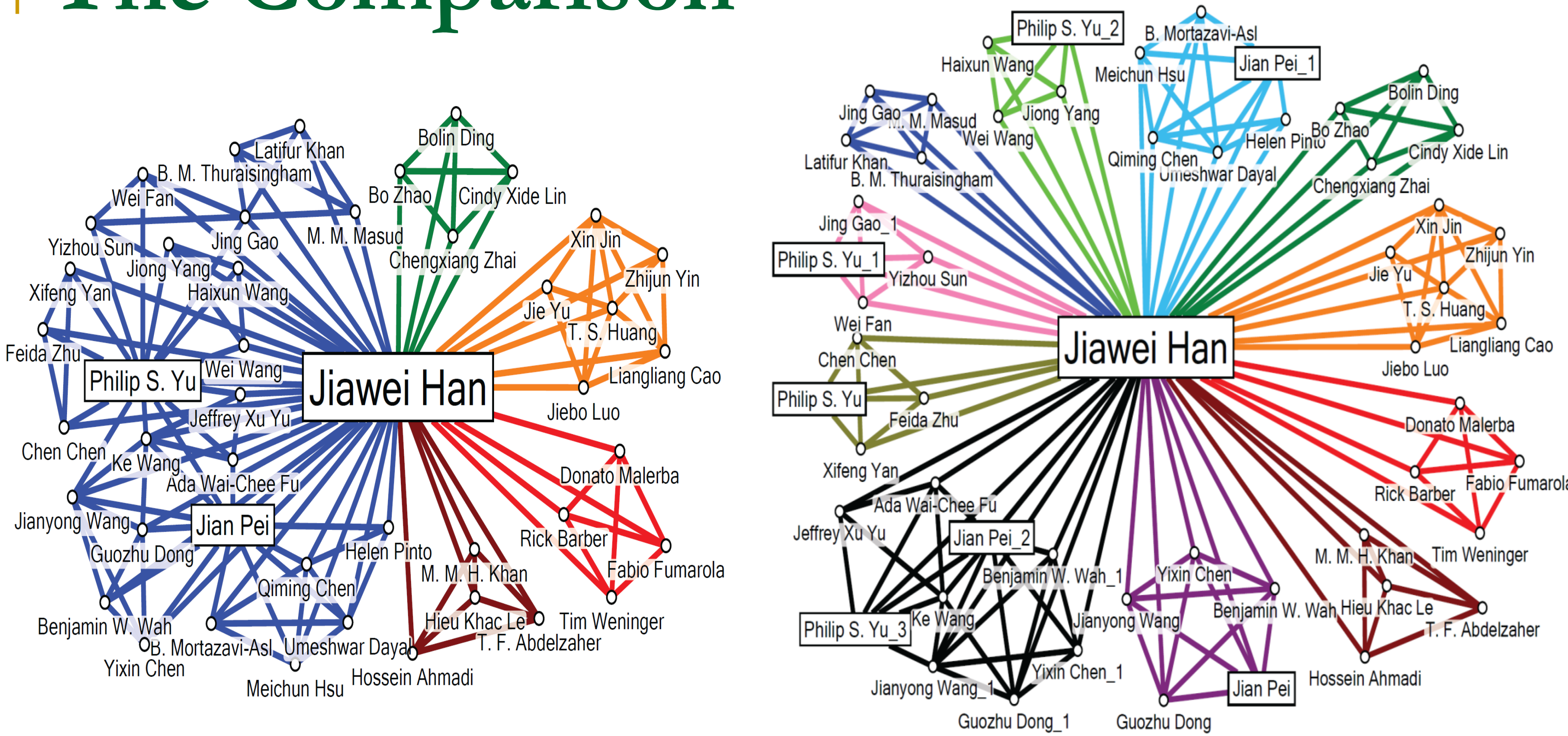
- Given a graph $G(V, E)$, a query vertex q and an integer $k \geq 3$, find all k-truss communities containing q.

Community Search: An Example



5-truss communities containing “Jiawei Han” in DBLP collaboration network

The Comparison

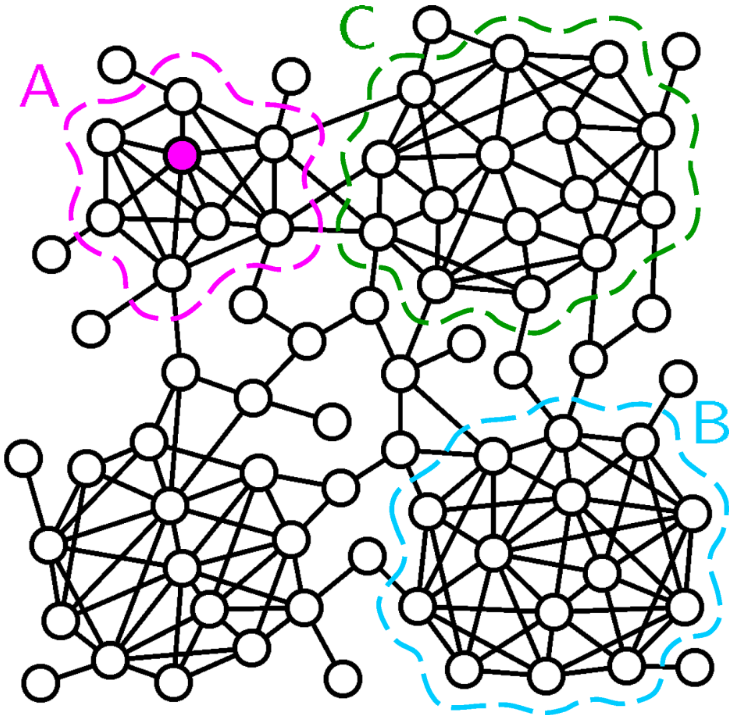


- 5-truss community (left)
- 11 4-adjacency-1.0-quasi-5-clique communities (right)
- The largest 5-truss (blue) community is decomposed into 7 smaller communities

Local Community Detection [Y. Wu, et al. PVLDB15]

- Input:
  - Graph $G(V,E)$
  - A set of query nodes Q
  - A goodness metric $f(S)$

- Output: Subgraph $G[S]$ such that:
  - S contains Q ($Q \subseteq S$)
  - $f(S)$ is maximized

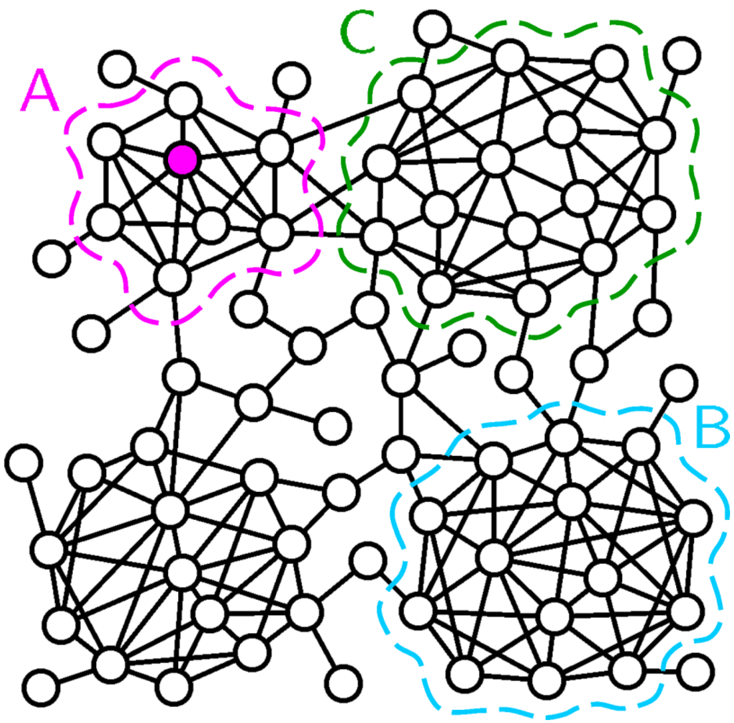


Local Community Detection [Y. Wu, et al. PVLDB15]

Intuitions	Goodness metrics	Formulas $f(S)$	
Internal denseness	Classic density	$e(S)/\|S\|$	
	Edge-surplus	$e(S) - \alpha h(\|S\|)$	concave $h(x)$
			$h(x) = \binom{x}{2}$
	Minimum degree	$\min_{u \in S} w_S(u)$	
Internal denseness & external sparseness	Subgraph modularity	$e(S)/e(S, \bar{S})$	
	Density-isolation	$e(S) - \alpha\, e(S, \bar{S}) - \beta\|S\|$	
	External conductance	$e(S, \bar{S})/\min\{\phi(S), \phi(\bar{S})\}$	
Boundary sharpness	Local modularity	$e(\delta S, S)/e(\delta S, V)$	

Local Community Detection [Y. Wu, et al. PVLDB15]

- Extend one query node to multiple nodes.
- Avoid free rider effect

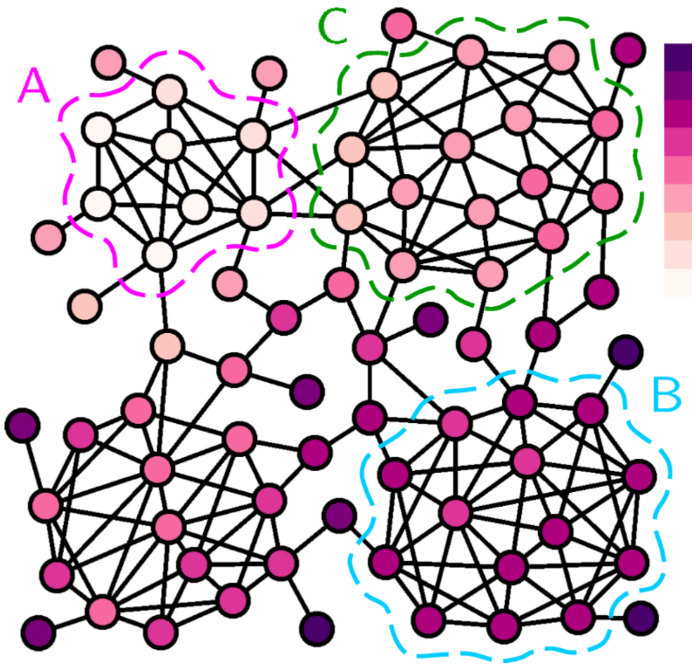


Classic density: $|E|/|V|$

Goodness metrics	A	A ∪ B	A ∪ C
Classic density	2.50	**2.95**	2.83
Edge-surplus	15.3	**26.5**	22.8
Minimum degree	4	4	4
Subgraph modularity	2.0	3.6	**4.6**
Density-isolation	-2.6	**3.8**	1.5
Ext. conductance	0.25	0.14	**0.11**
Local modularity	0.63	0.70	**0.78**

Query Biased Density [Y. Wu, et al. PVLDB15]

- Compute the proximity value of each node with regard to the query nodes, denoted $r(\cdot)$.
- The reciprocal of the proximity value is used as the node weight, denoted $\pi(u) = 1/r(u)$.
- The query biased density is $\rho(S) = \frac{e(S)}{\pi(S)}$ where S is a set of nodes.
- Find query biased densest connected subgraph with max $\rho(S)$, where $Q \subseteq S$ and $G[S]$ is connected.



Approximate Closet Community Search [PVLDB'16]

Xin Huang, Laks V.S. Lakshmanan, Jeffrey Xu Yu, Hong Cheng

Our Approach

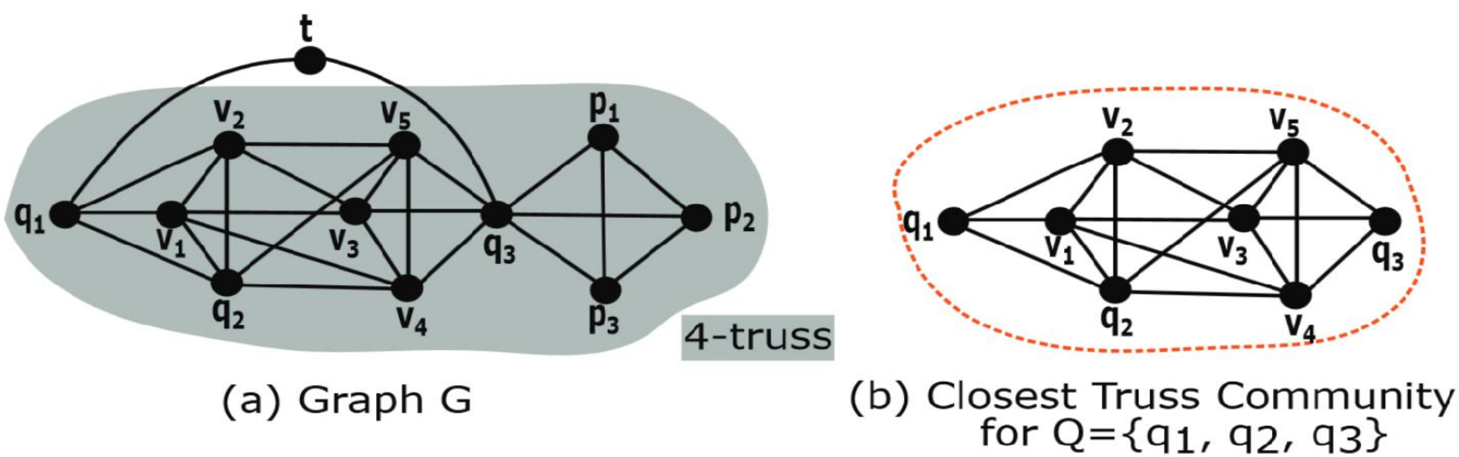
- Graph Diameter of G: $\text{diam}(G) = \max_{u,v \in G}\{\text{dist}_G(u,v)\}$
- Query Distance for a vertex v and a subgraph H in G:

  $$\text{dist}_G(v,Q) = \max_{q \in Q} \text{dist}_G(v,q)$$

  $$\text{dist}_G(H,Q) = \max_{u \in H} \text{dist}_G(u,Q) = \max_{u \in H, q \in Q} \text{dist}_G(u,q)$$

- Lower and upper bounds of graph diameter:

  $$\text{dist}_G(G,Q) \leq \text{diam}(G) \leq 2\text{dist}_G(G,Q)$$

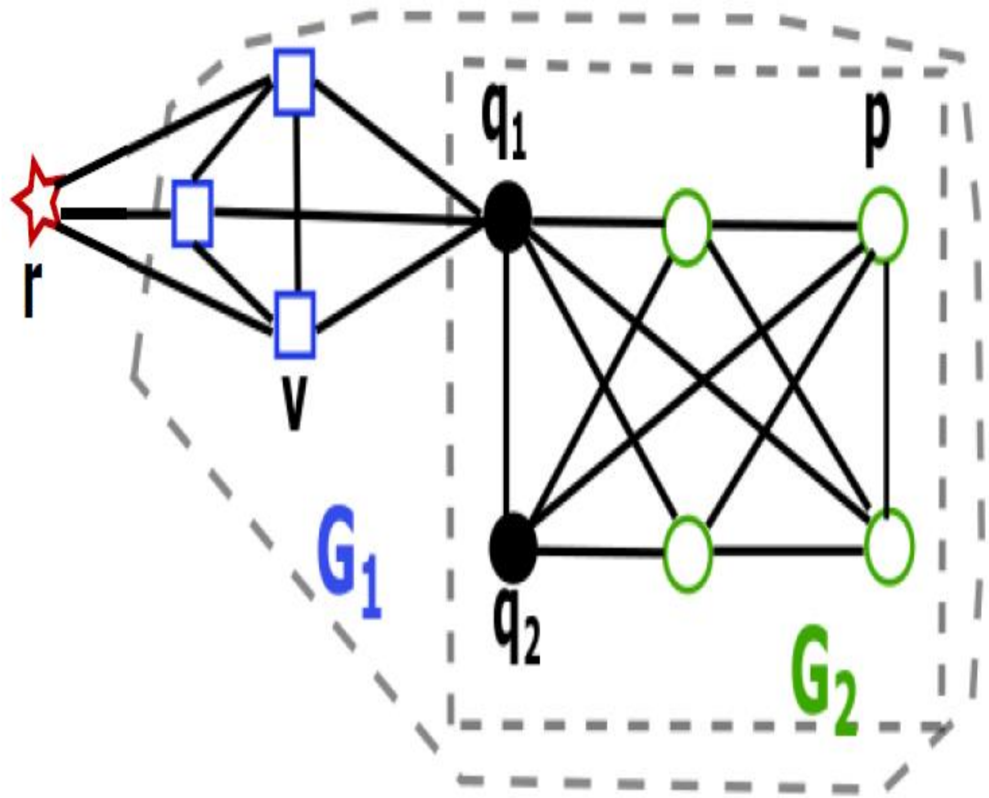


(a) Graph G

(b) Closest Truss Community for Q={q1, q2, q3}

An Example

- Consider a query with two query nodes, $Q = \{q_1, q_2\}$.
- G, G_1, and G_2 are 4-trusses containing Q.
- The query distance of r in G is 3. The query distance of G is 3.
- The query distance of v in G_1 is 2. But, the diameter of G_1 is 3 (between v and p).
- The query distance in G_2 is 2. The diameter of G_2 is 2.



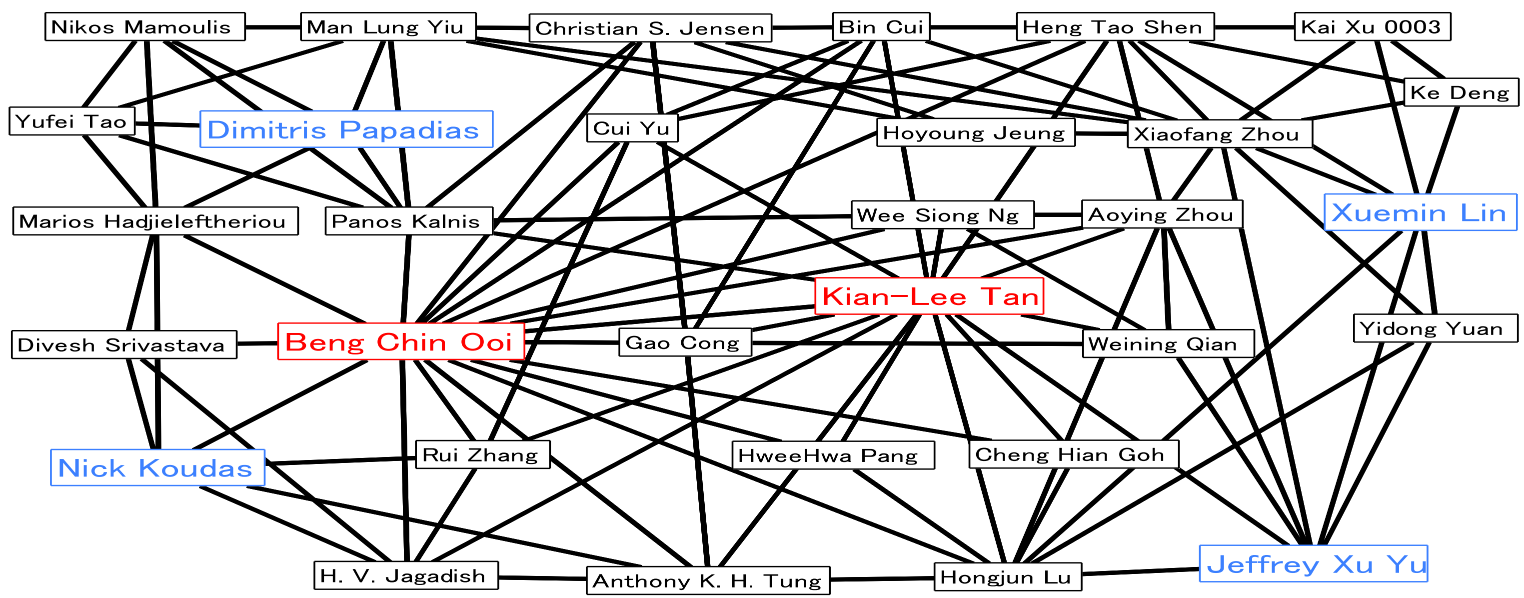
Our Problem Definition

- Input:
  - graph G
  - a set of query nodes Q

- Output: a connected subgraph H containing Q such that
  - H *is a* k*-truss with the largest* k
  - H *is with the smallest diameter*

A Case Study: DBLP network

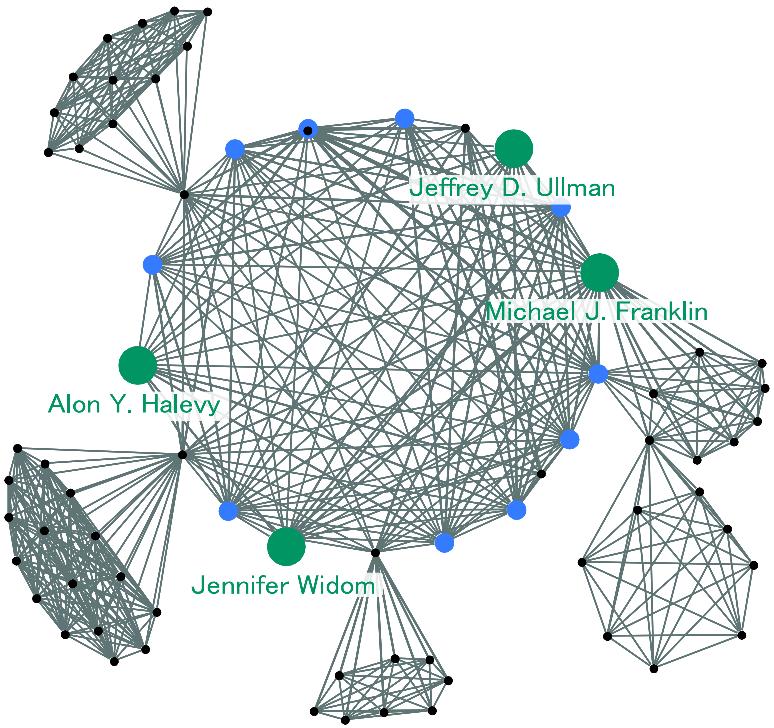
(a) QDC



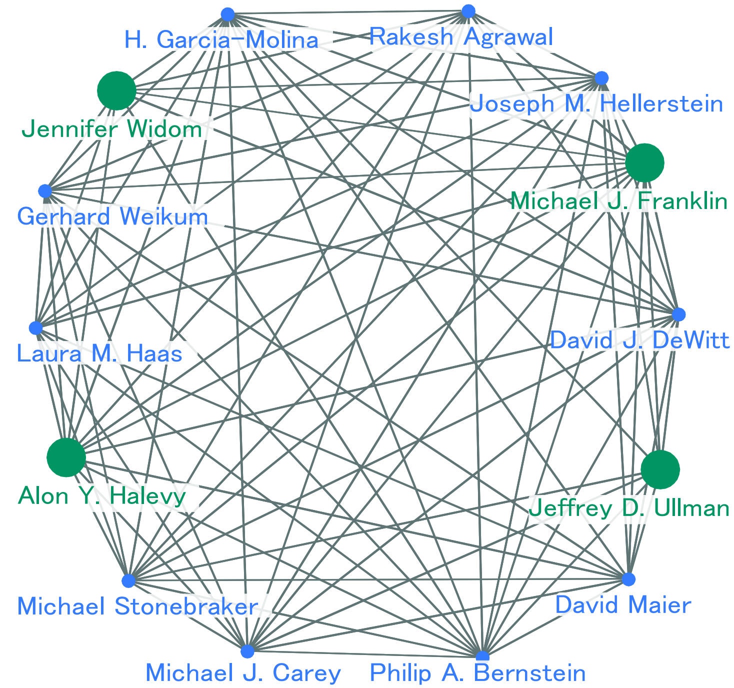
(b) Closest Truss community

Community search on DBLP network using query Q={ **"Xuemin Lin", "Jeffrey Xu Yu", "Nick Koudas", "Dimitris Papadias"** }

A Case Study



(a) 9-truss



(b) Closest Truss community

Community search on DBLP network using query Q={ **"Alon Y. Halevy", "Michael J. Franklin", "Jeffrey D. Ullman", "Jennifer Widom"** }

More to Explore Next

- There are many large networks.
  - Online Social Networks
  - Location Based Social Networks
  - Road/Transportation Networks
- There are issues related to social commerce and online shopping
  - It is possible to know where you are and when/what you call/buy.
- There are many research opportunities.