HIERARCHICAL DENSE SUBGRAPH DISCOVERY: MODELS, ALGORITHMS, APPLICATIONS

A. Erdem Sariyuce

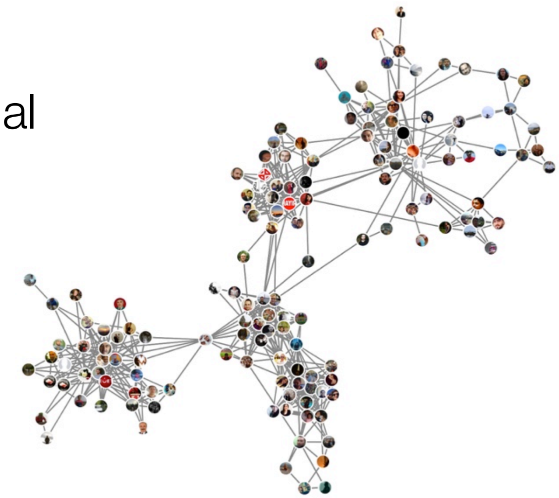
University of Waterloo

Nov. 19, 2018

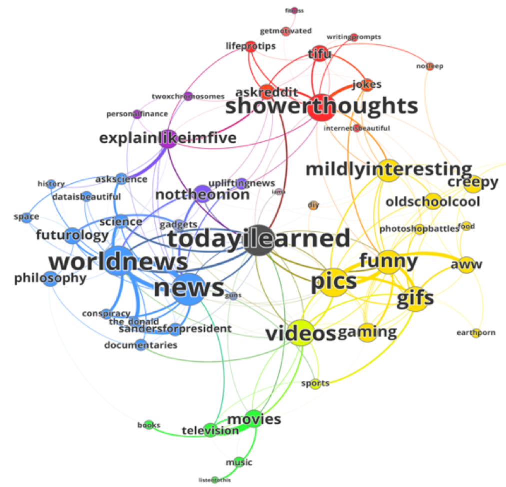
University at Buffalo The State University of New York

Graphs all around

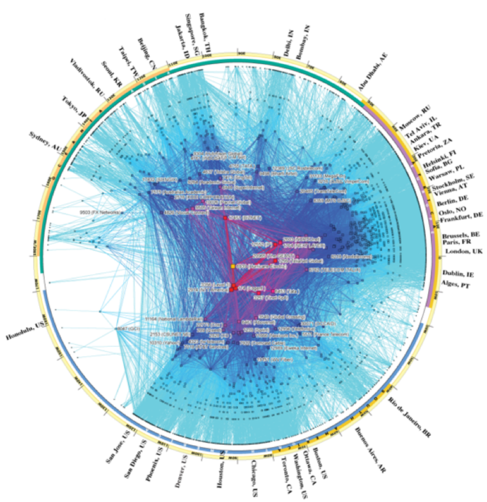
Social



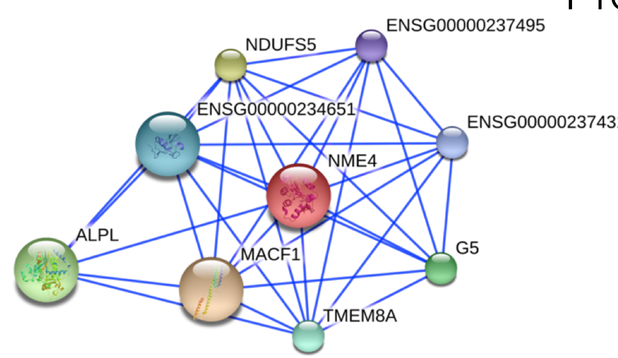
Information



Routers

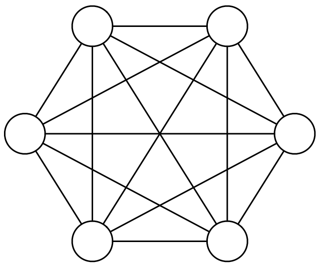


Protein-interaction



Dense subgraph discovery

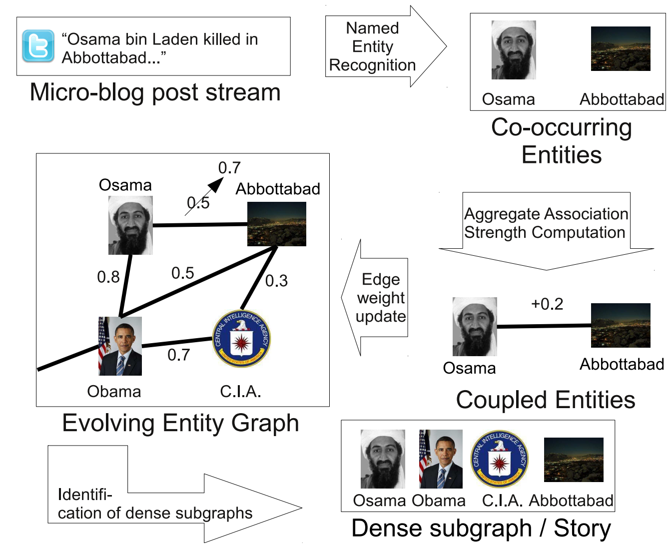
- Measure of connectedness on edges
  - # edge / # all possible
    - |E| / (|V| choose 2), 1.0 for a clique

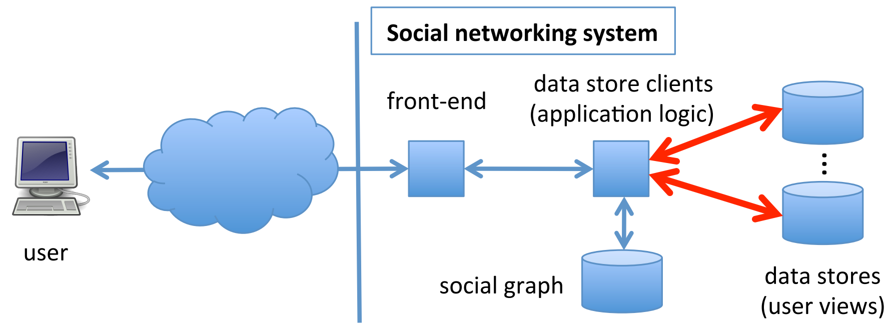


- Globally sparse, locally dense
  - $|E| << |V|^2$, but vertex neighborhoods are dense
    - High clustering coefficients – density of neighbor graph
- Many nontrivial subgraphs with high density
  - And relations among them
- Not clustering: Absolute vs. relative density

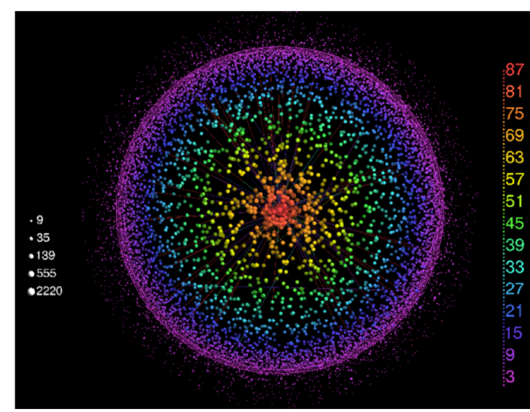
Dense subgraphs matter in many applications

- Significance or anomaly
  - Spam link farms [Gibson et al., '05]
  - Real-time stories [Angel et al., '12]



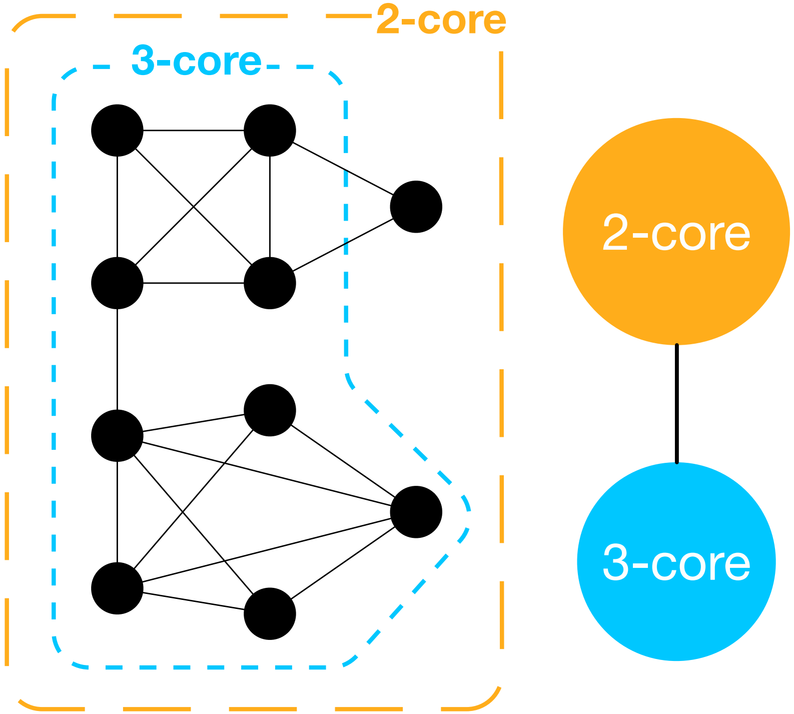


- Computation & summarization
  - System throughputs [Gionis et al., '13]
  - Graph visualization [Alvarez et al., '06]

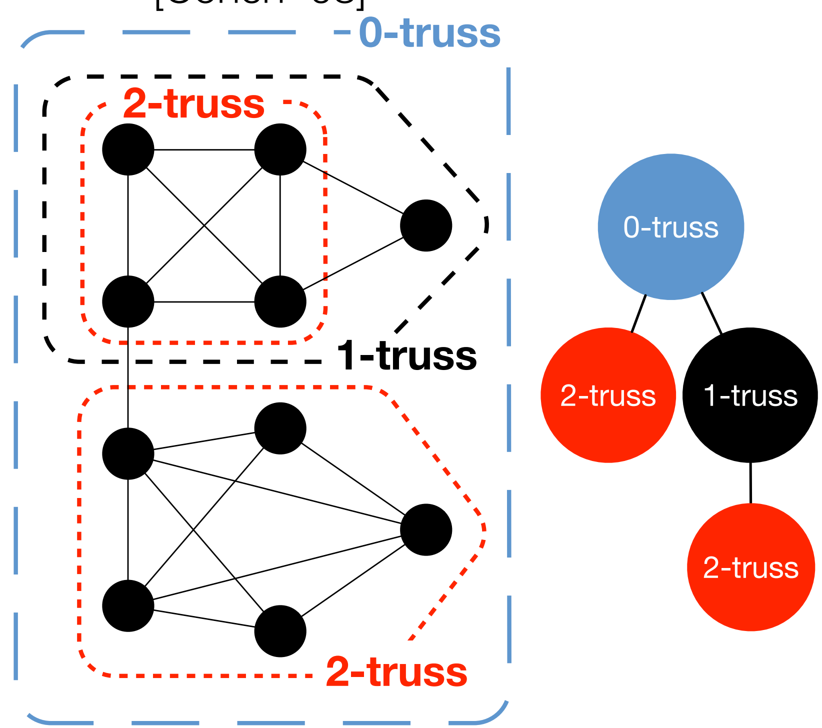


Two effective algorithms to find dense subgraphs with hierarchical relations

- ***k*-core:** Every vertex has at least *k* edges
  - [Seidman, ‘83], [Matula & Beck, ‘83]



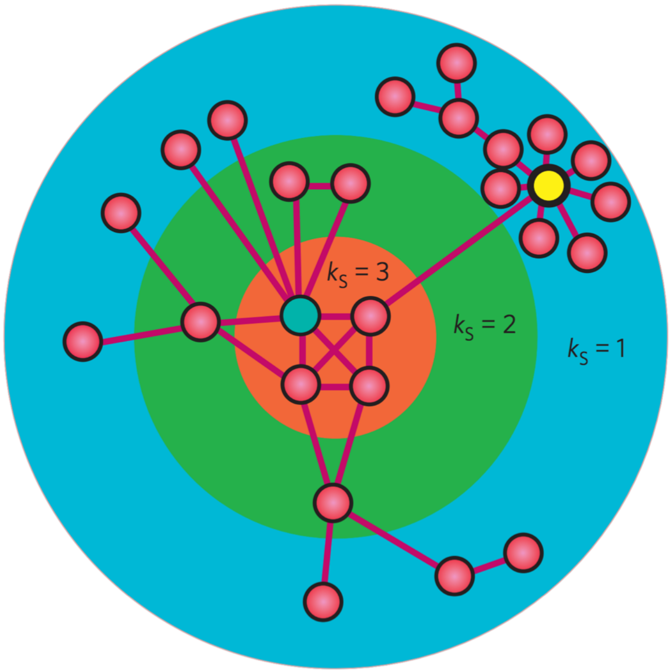
- ***k*-truss:** Every edge has at least *k* triangles
  - [Cohen ‘08]



Why core/truss decompositions?

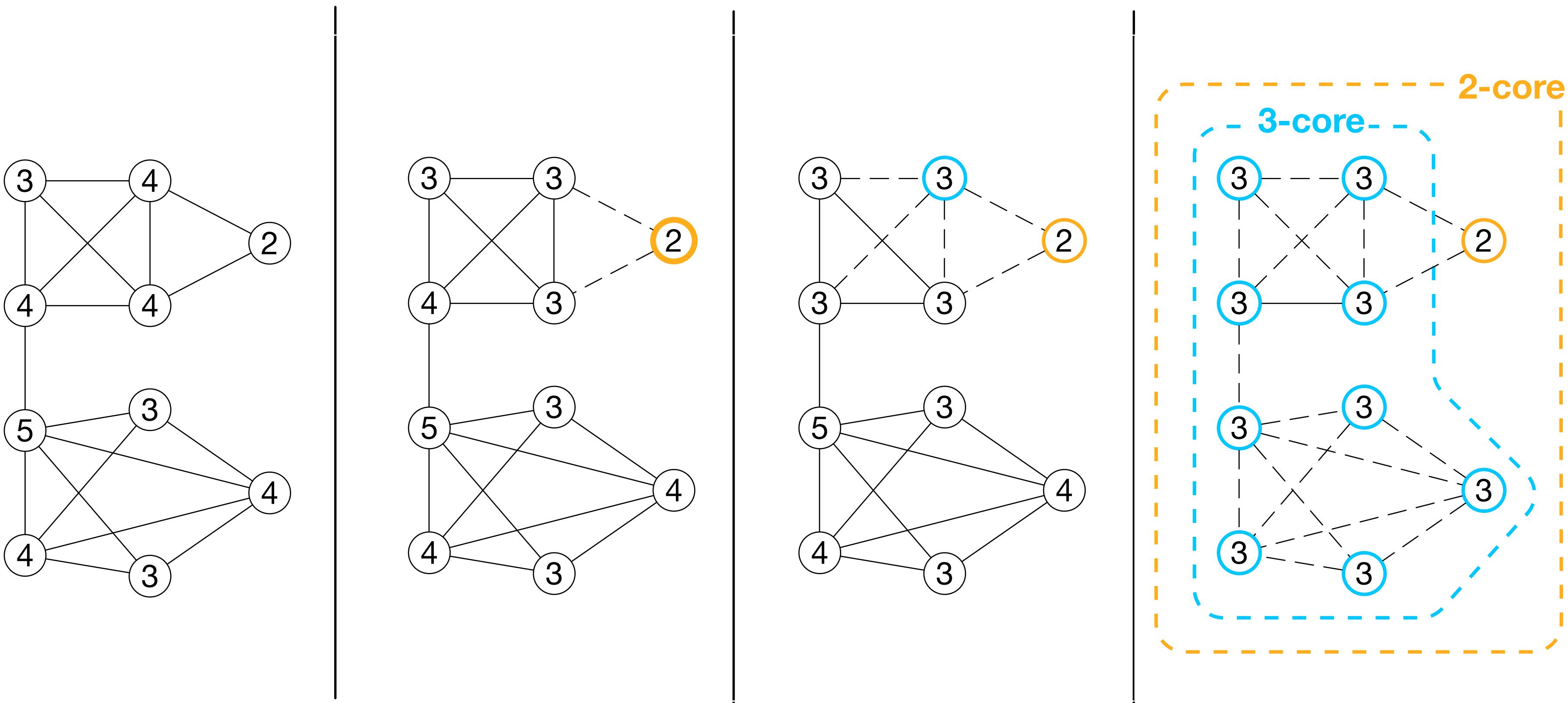
- Fundamental building block
  - Densest subgraph: 2-approximation [Charikar'00]
    - O(m.n.log(n).log(m)) -> O(m)
  - Maximal clique finding [Rossi'15]

- Identifying influential spreaders
  - Hubs are not always influential
  - Isolated star problem [Kitsak'10]



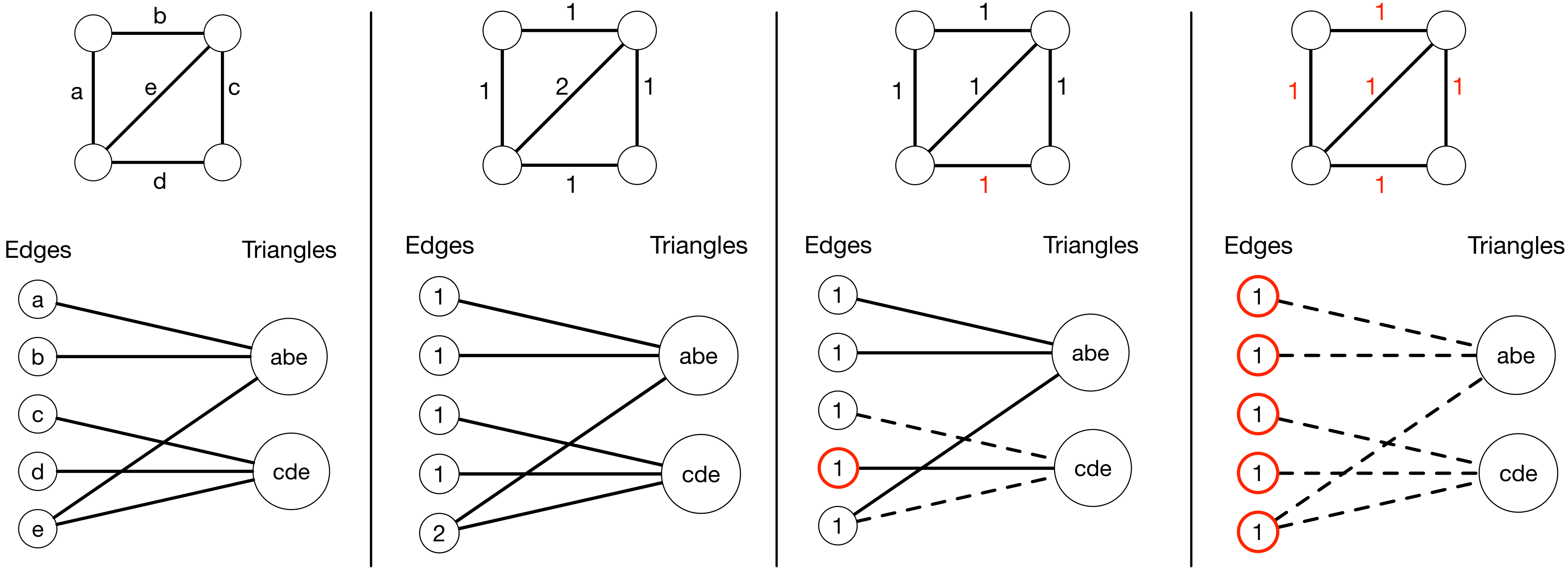
Peeling algorithm finds the *k*-cores & *k*-trusses

- Core numbers of vertices. O(|E|) [Matula & Beck, ‘83]
- Truss numbers of edges. $O(\sum_{u \in G} {d_u}^2)$ [Cohen ‘08]



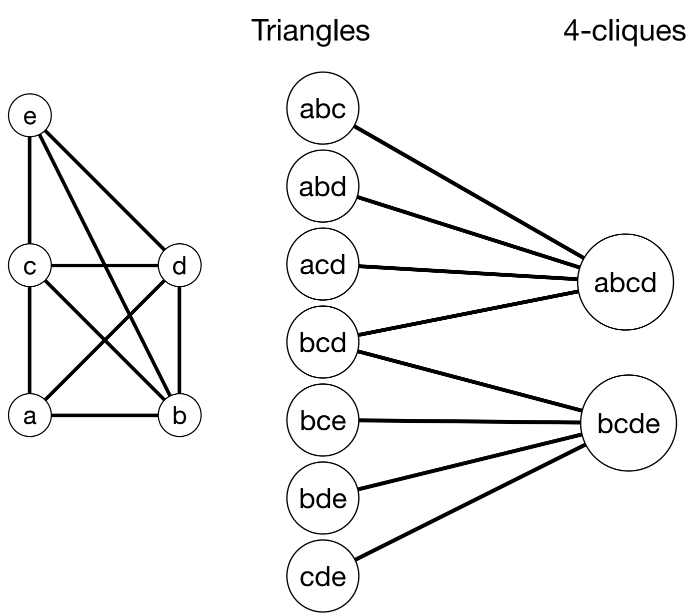
Observation: *k*-truss IS just *k*-core on the edge-triangle graph!

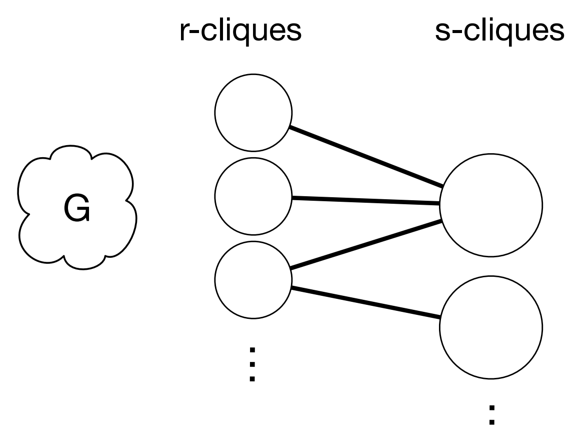
- Edge and triangle relations
  - Build a bipartite graph!
  - Not a binary relation – three edges in a triangle



Why limit to triangles?

- Small cliques in larger cliques
  - 1-cliques in 2-cliques (vertices and edges)
  - 2-cliques in 3-cliques (edges and triangles)
- Generalize for any clique: r-cliques in s-cliques ($r < s$)
- Convert to bipartite: r-cliques on left, s-cliques on right
  - Connect if right contains left





Nucleus decomposition generalizes *k*-core and *k*-truss algorithms

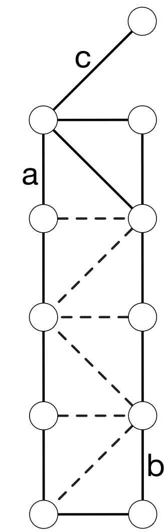
- k-(r, s) nucleus:
  - Every r-clique takes part in at least k number of s-cliques

Sariyuce, Seshadhri, Pinar, Catalyurek, WWW'15 (Best paper runner-up)

Nucleus decomposition generalizes *k*-core and *k*-truss algorithms

- k-(r, s) nucleus:
  - Every r-clique takes part in at least k number of s-cliques
    - Each r-clique pair is connected by series of s-cliques

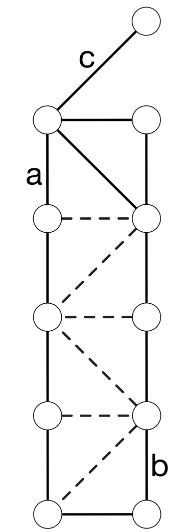
***r*=1, *s*=2**	***r*=2, *s*=3**
***k*-core**	***k*-truss**
	(stronger conn.)
d(v) ≥ k	△(e) ≥ k
Simply connected	Triangle connected



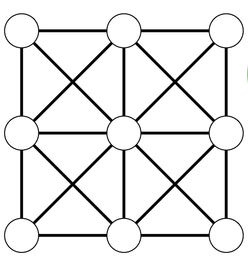
Sariyuce, Seshadhri, Pinar, Catalyurek, WWW'15 (Best paper runner-up)

Nucleus decomposition generalizes *k*-core and *k*-truss algorithms

- k-(r, s) nucleus:
  - Every r-clique takes part in at least k number of s-cliques
    - Each r-clique pair is connected by series of s-cliques

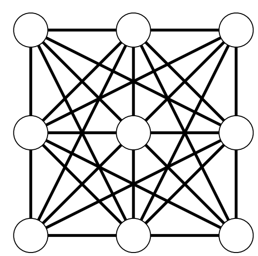


r=1, *s*=2	*r*=2, *s*=3
***k*-core**	***k*-truss**
	(stronger conn.)
d(v) ≥ k	△(e) ≥ k
Simply connected	Triangle connected



2 - 2 3 nucleus

Each edge has at least two triangles



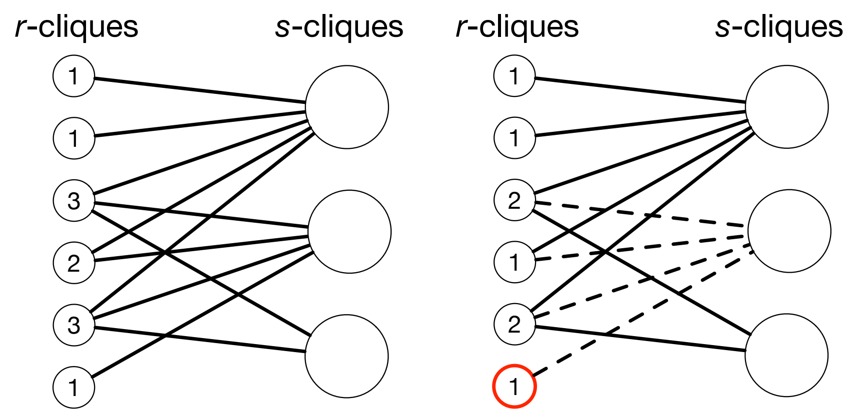
2 - 2 4 nucleus

Each edge has at least two 4-cliques

Sariyuce, Seshadhri, Pinar, Catalyurek, WWW'15 (Best paper runner-up)

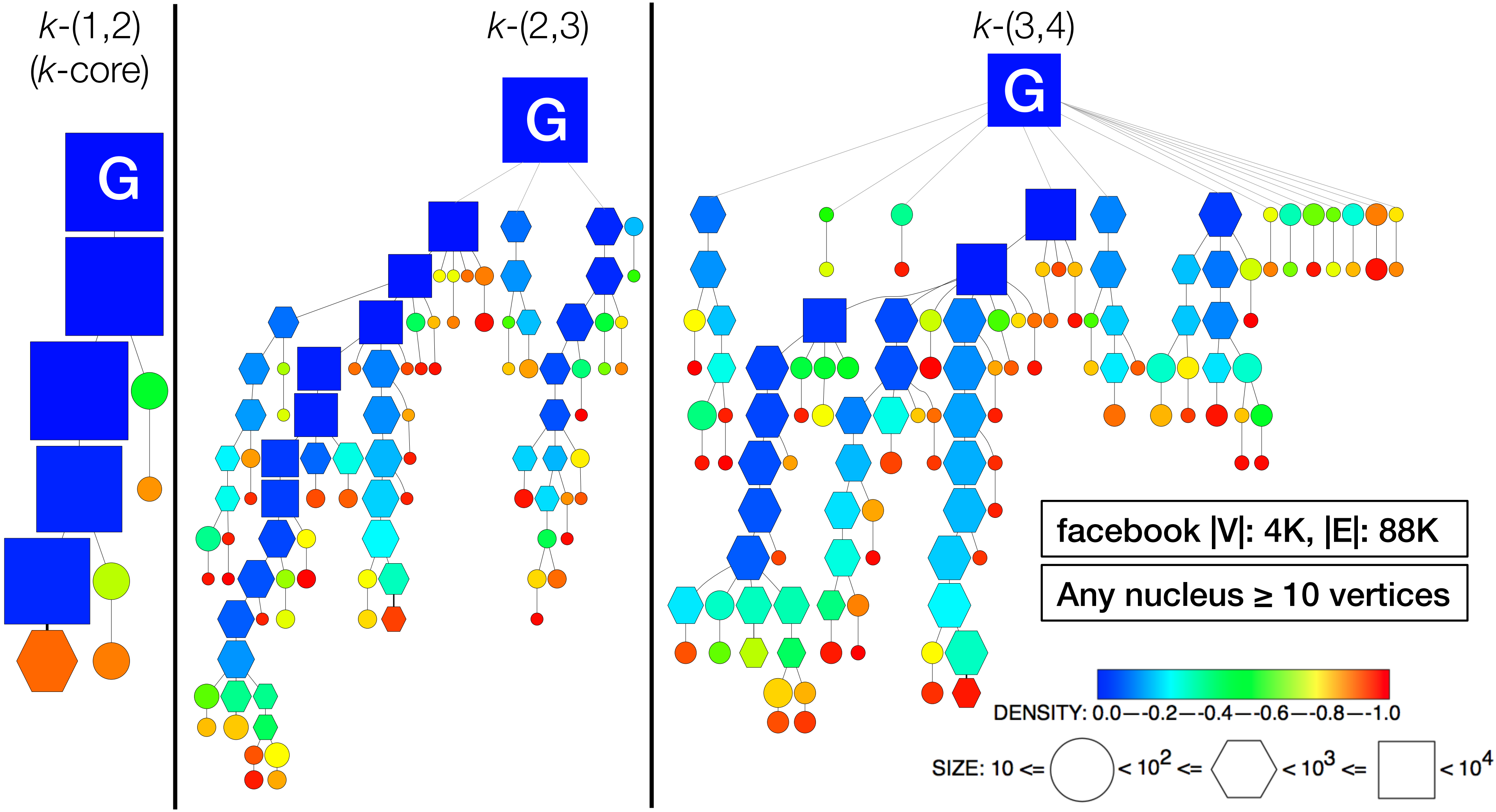
Peeling works for nucleus decomposition as well!

- On the bipartite graph
  - For the set of *r*-cliques
  - Degree based

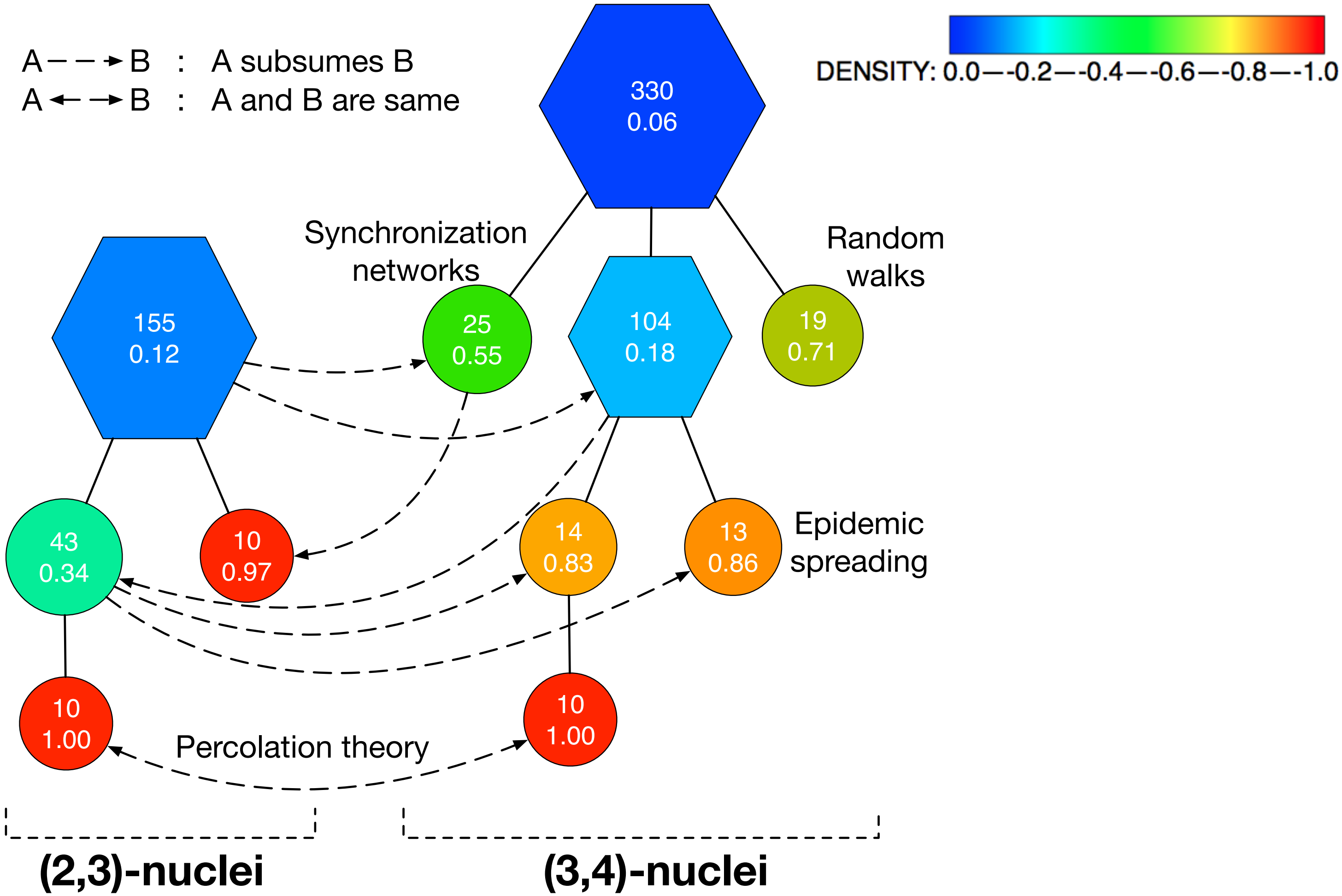


- Sounds expensive?
  - Yes, in theory
  - *r*=3, *s*=4: $O(\sum_v cc(v)d(v)^3)$
  - But practical
    - Clustering coefficients decay with the degree in many real-world networks
  - Can be scaled to tens of millions of edges

Comparing hierarchies for different nucleus decompositions



APS Citation Network Analysis

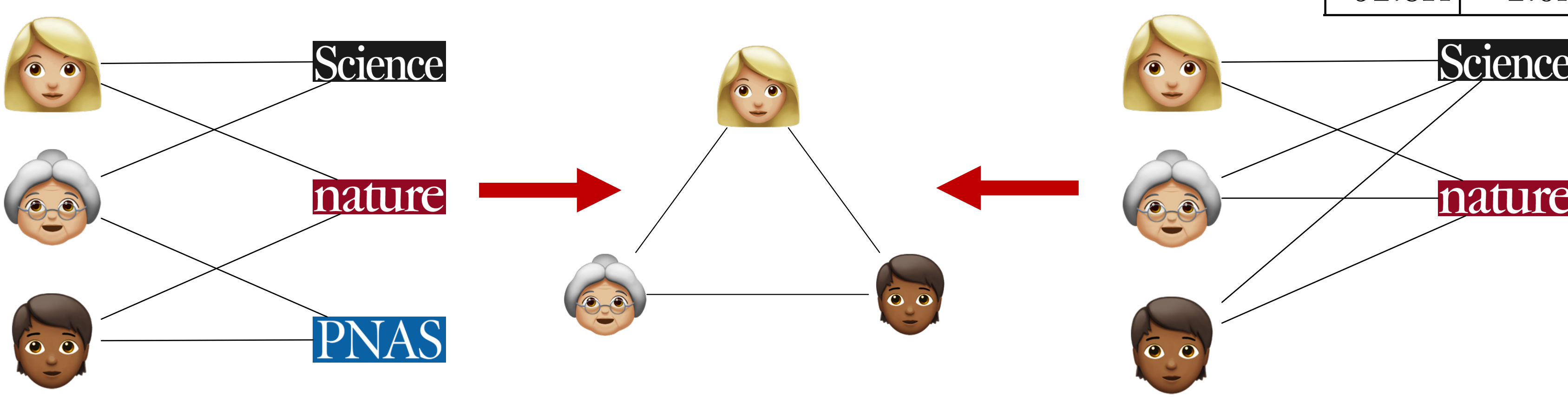


Sariyuce, Seshadhri, Pinar, Catalyurek, TWEB 11(3), 16

What about other graph types? Bipartite networks (One-to-many relations)?

- Author-paper, word-document, actor-movie…
  - Bipartite in nature, no triangle
- Usually project bipartite to unipartite
  - Author-paper → Co-authorship

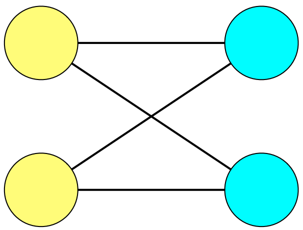
$\lvert E\rvert$	$\lvert E_p\rvert$
58.6K	95.1K
30.7K	84.8K
440.2K	44.5M
96.7K	336.5K
5.6M	157.5M
92.8K	2.0M

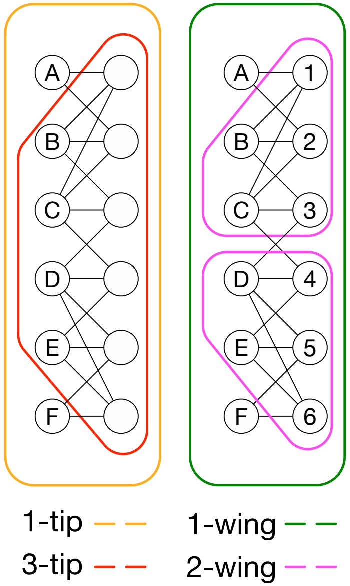


  - |E| explodes! Information lost (even for weighted)!
- Find dense regions directly on bipartite graph!

What is the "triangle" in a bipartite network?

- Focus on the smallest non-trivial structure
  - (2, 2)-biclique, or butterfly



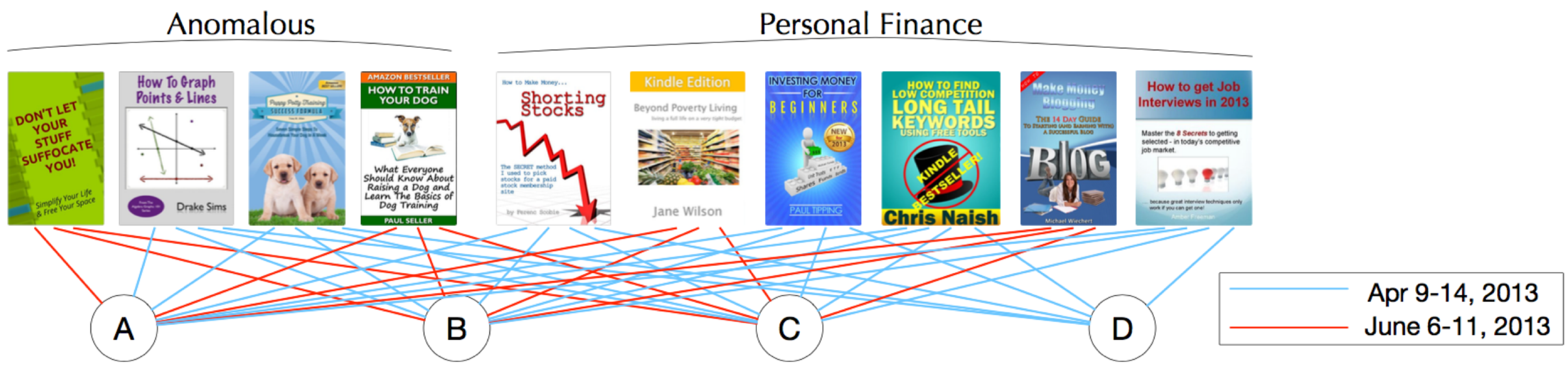


- Vertex-butterfly, edge-butterfly relations
  - *k*-tip: Each **vertex** participates in ≥ *k* butterflies
  - *k*-wing: Each **edge** participates in ≥ *k* butterflies
    - Can overlap

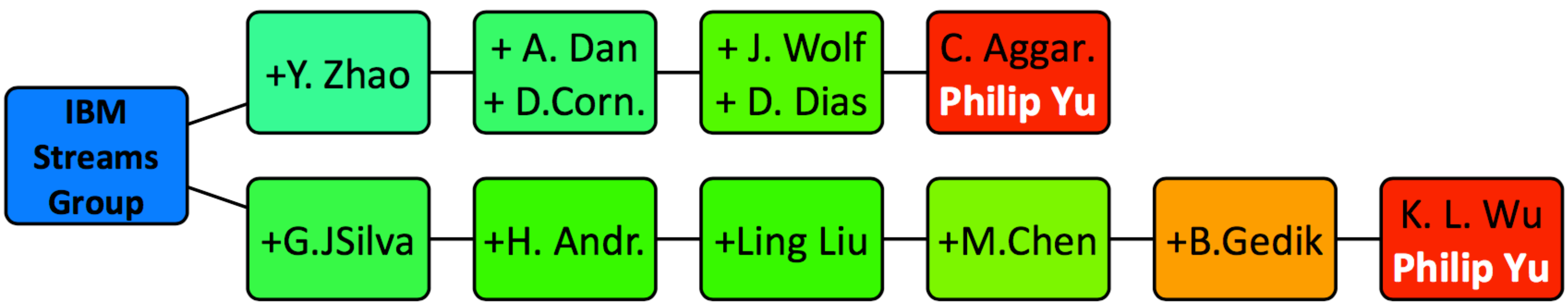
Sariyuce & Pinar, WSDM'18

Applications

- Amazon Kindle dataset (users rate books)

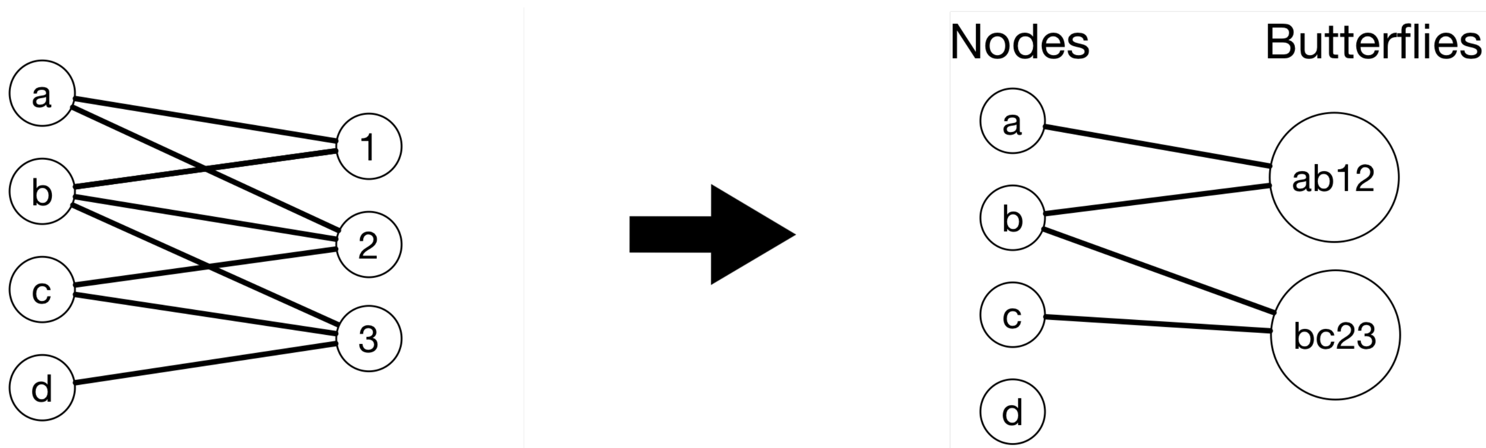


- Author-paper relations at top DB conferences



Peeling also works for tip and wing decompositions

- On the bipartite graph
  - Nodes & butterflies
  - Edges & butterflies



Challenge: Peeling needs global graph information

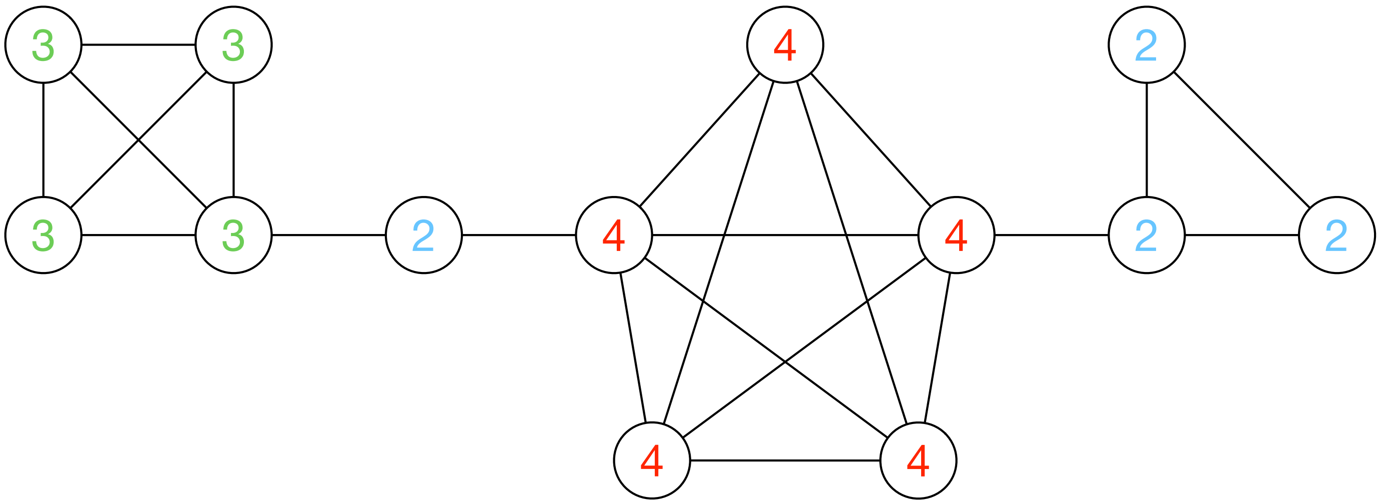
- Inherently sequential
  - Iterative processing

- Where is the vertex with the minimum degree?
- Independent computations not possible
  - Nothing is local

- Densest parts not revealed until the end
  - No sense of approximation, all or none

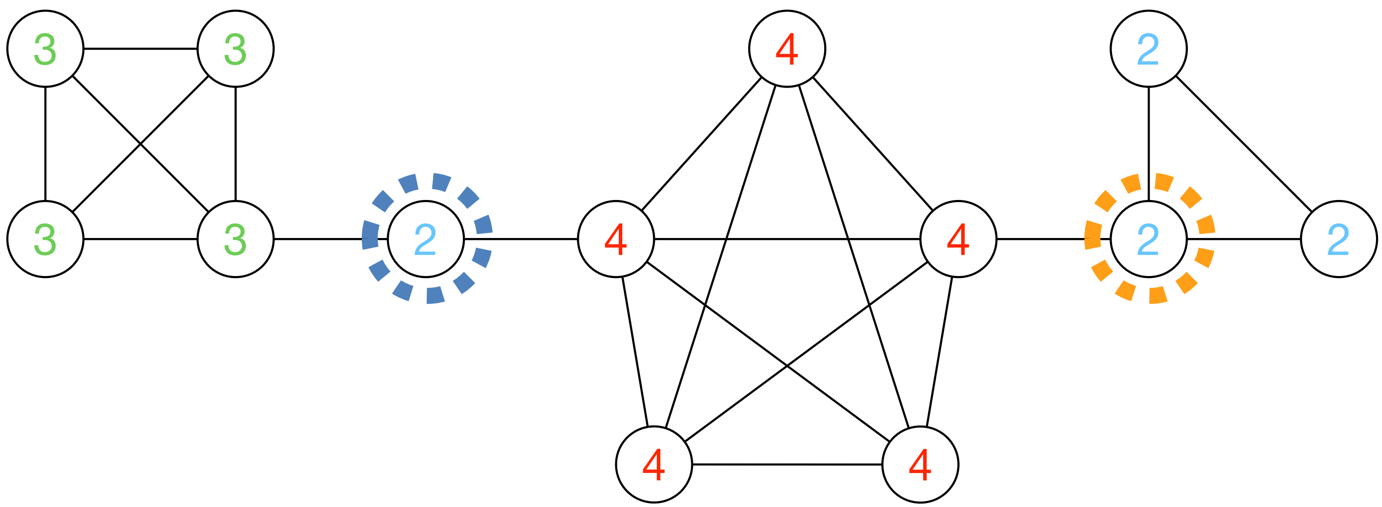
Any local information to infer the core numbers?

- Core numbers of neighbors



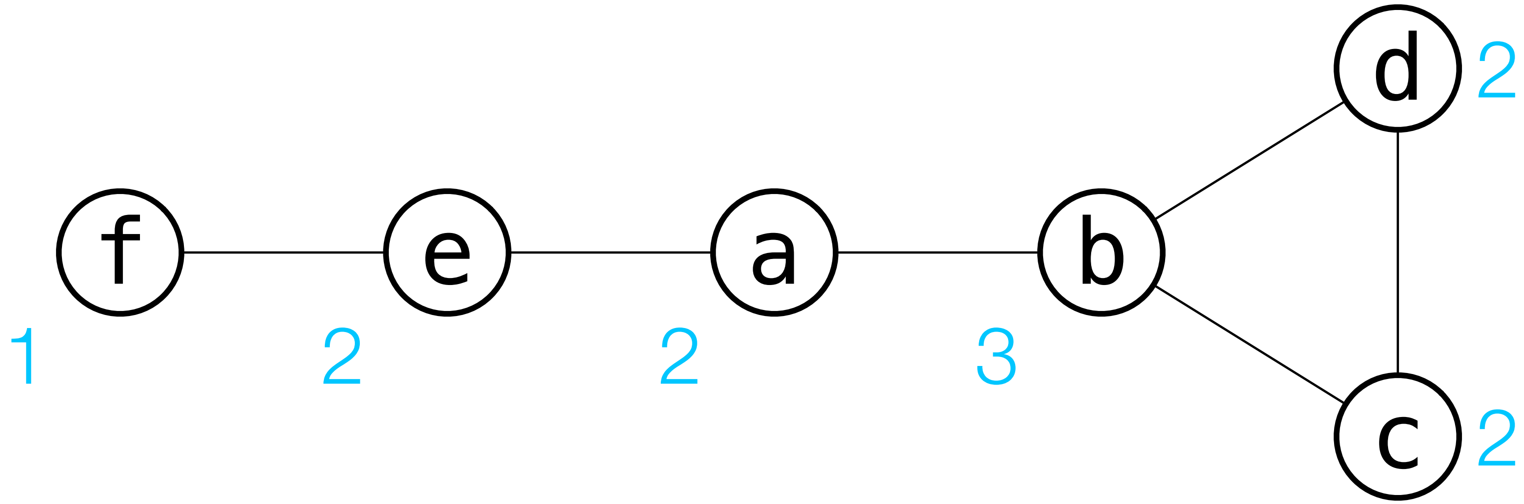
Any local information to infer the core numbers?

- Core numbers of neighbors

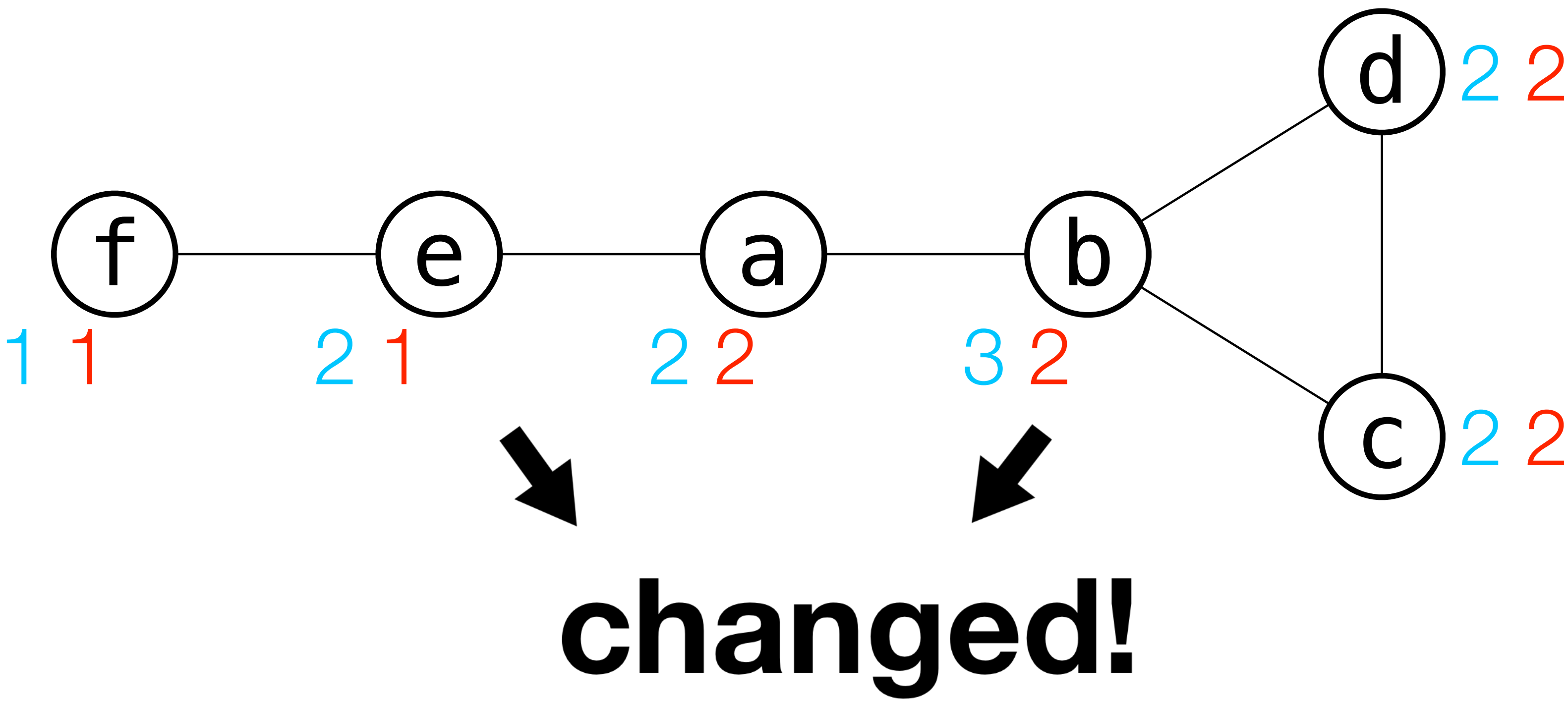


- h-index computation!
  - h{3,4}=2, h{2,2,4}=2
  - Start from degrees, repeat until no change
  - Degrees converge to core numbers [Lu et al.'16]
  - Generalizable for nucleus decomposition

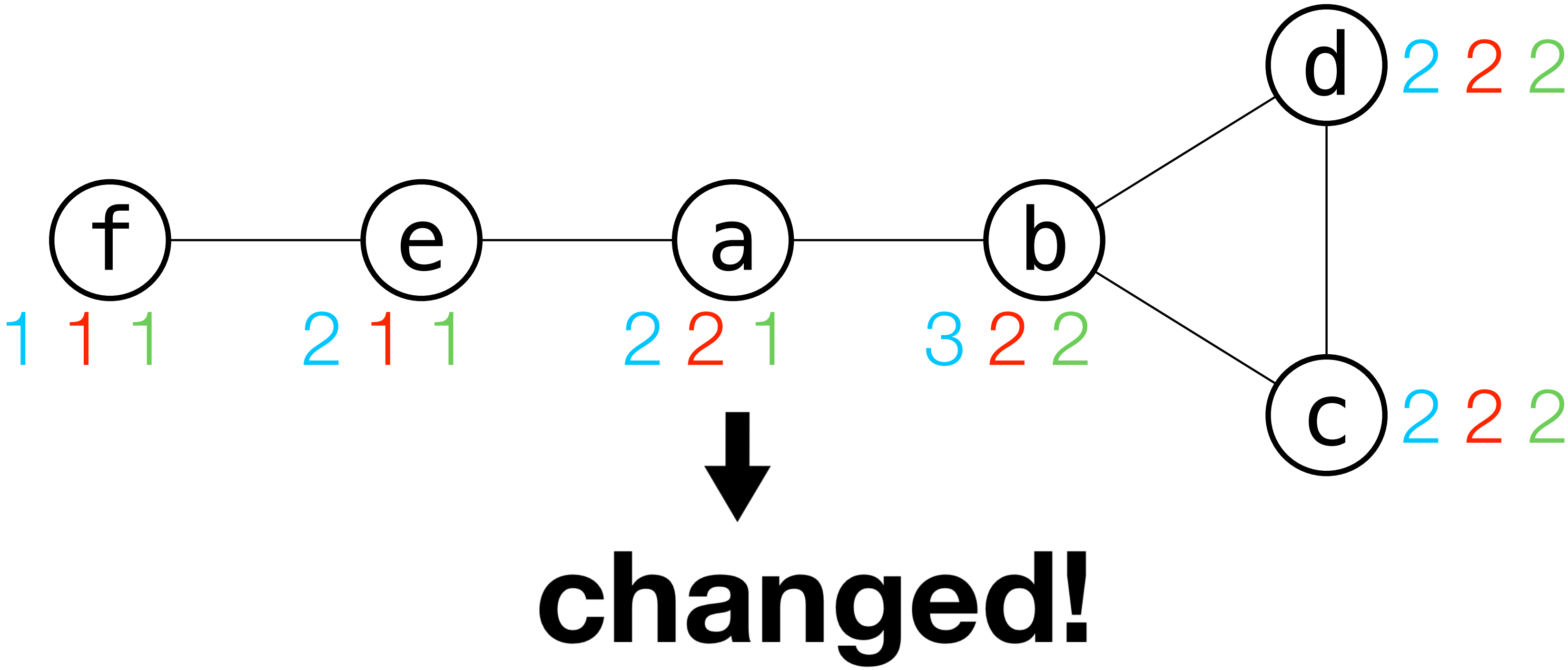
Example: Core numbers



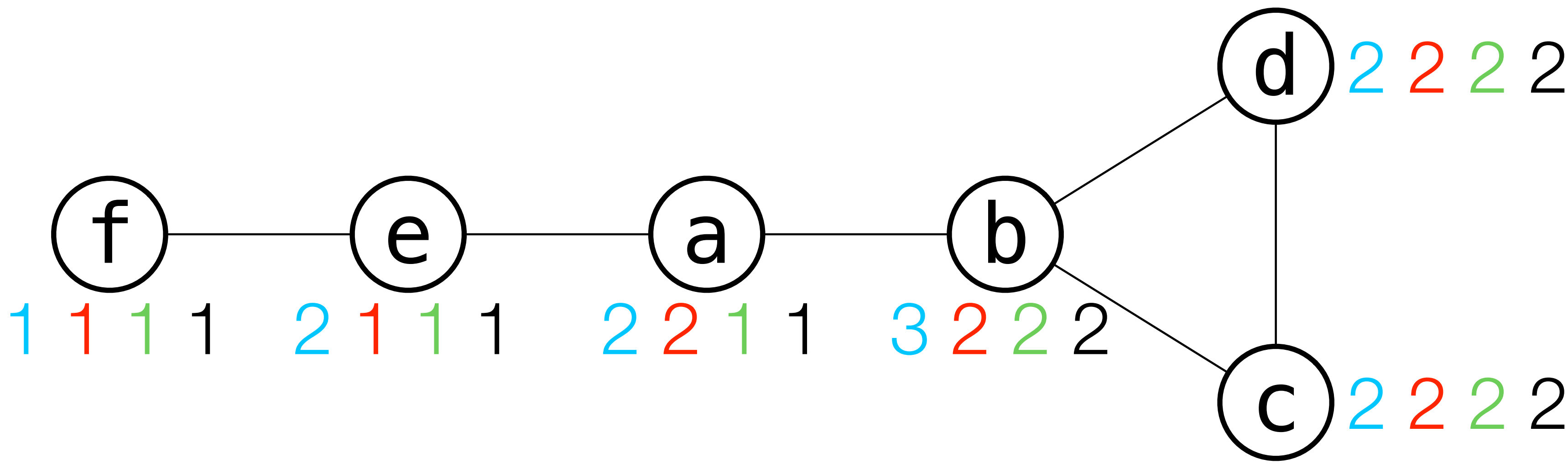
Example: Core numbers



Example: Core numbers

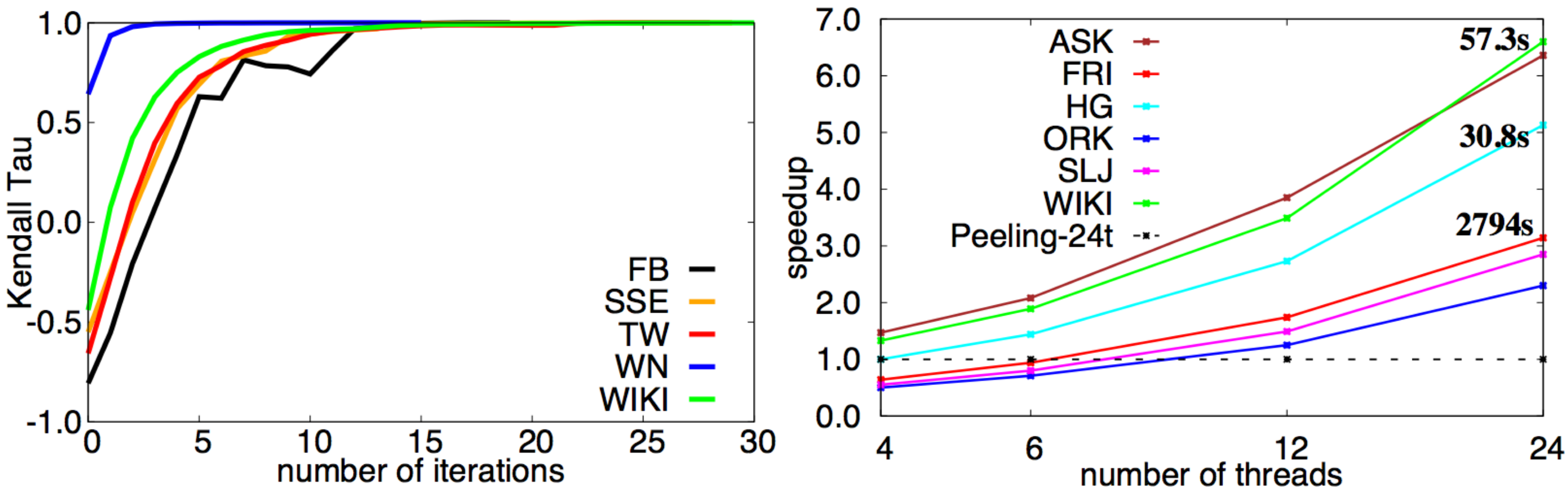


Example: Core numbers



converged!

Quick convergence, scalable computation



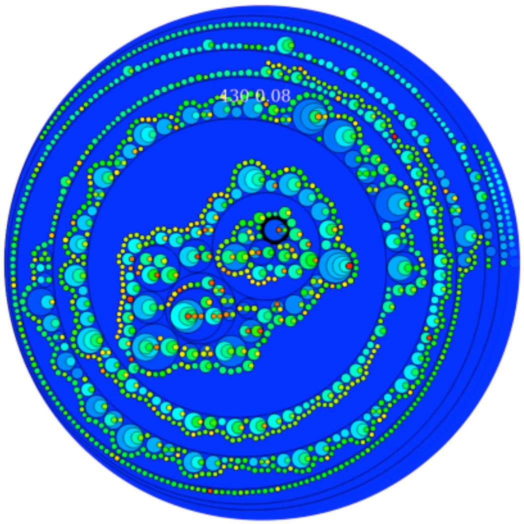
(a) Convergence rates (b) Scalability performance

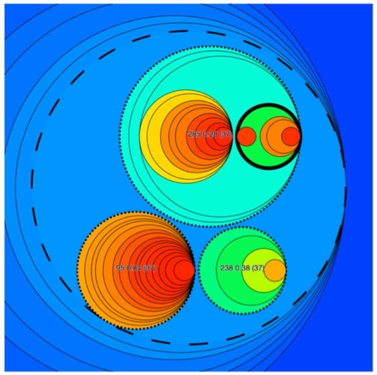
- Graphs with >100M edges
- 99% similarity in first few iterations
  - Approximation!

Sariyuce, Seshadhri, Pinar, VLDB’19

Conclusion

- Models & algorithms for hierarchical dense subgraph discovery
  - Nucleus decomposition
    - Generalizes *k*-core and *k*-truss; and extend
  - Wide application space
  - Challenging problems
- Exploring the bipartite realm
  - Many opportunities
- Local computations
  - Suitable for shared-nothing systems
- Network analysis by the nucleus hierarchy
  - Fast tools
  - Visualization





erdem@buffalo.edu

Papers and codes: http://sariyuce.com

Thanks!