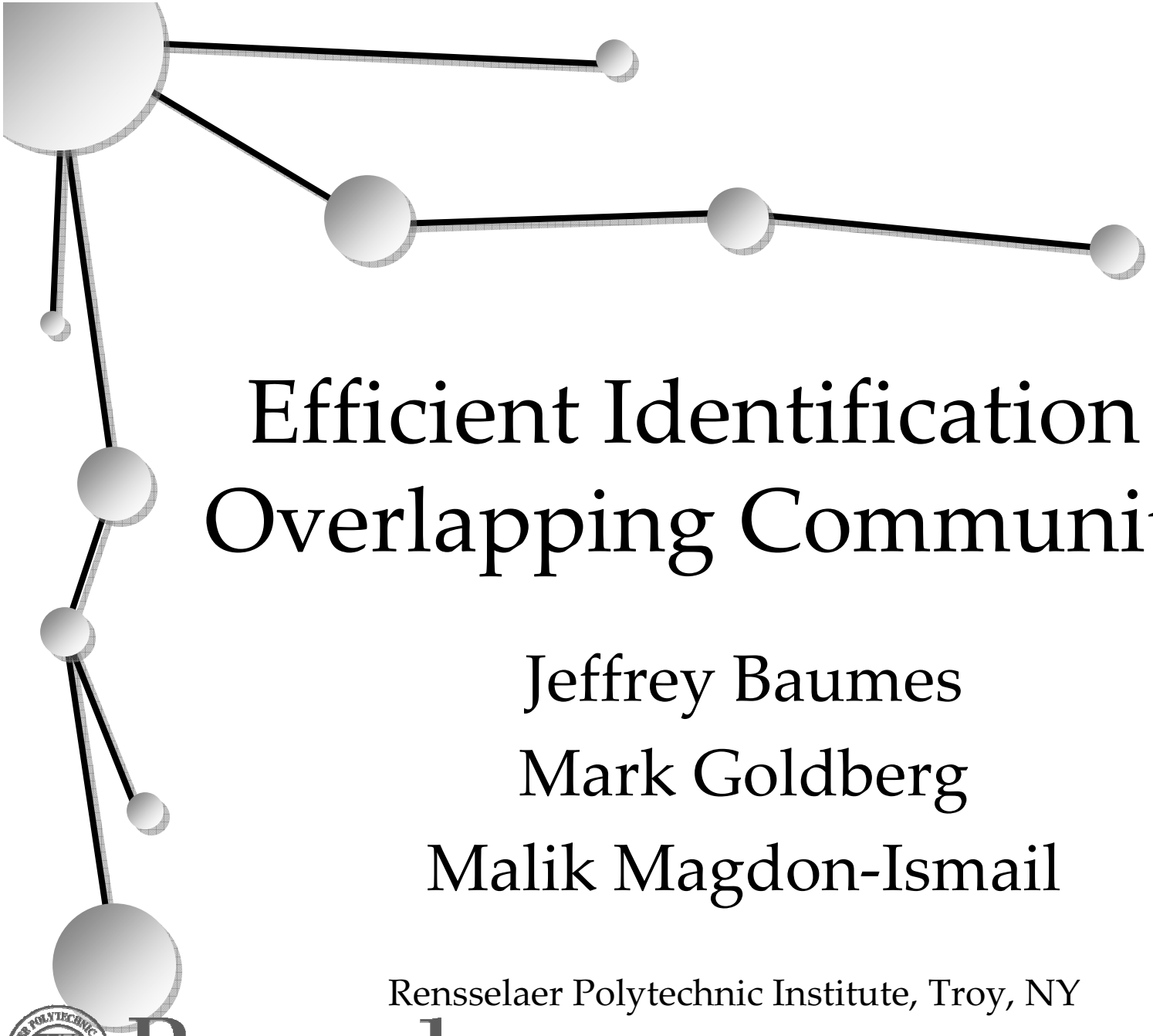


Efficient Identification of Overlapping Communities

Jeffrey Baumes

Mark Goldberg

Malik Magdon-Ismail

Rensselaer Polytechnic Institute, Troy, NY

Rensselaer

Outline

- Communities as clusters
- What is a cluster?
- Cluster seed procedure (LA)
- Cluster refinement procedure (IS^2)
- Experimental results
- Conclusions and future work

Rensselaer

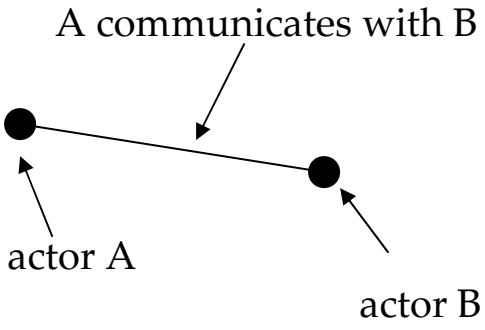
Communities as clusters

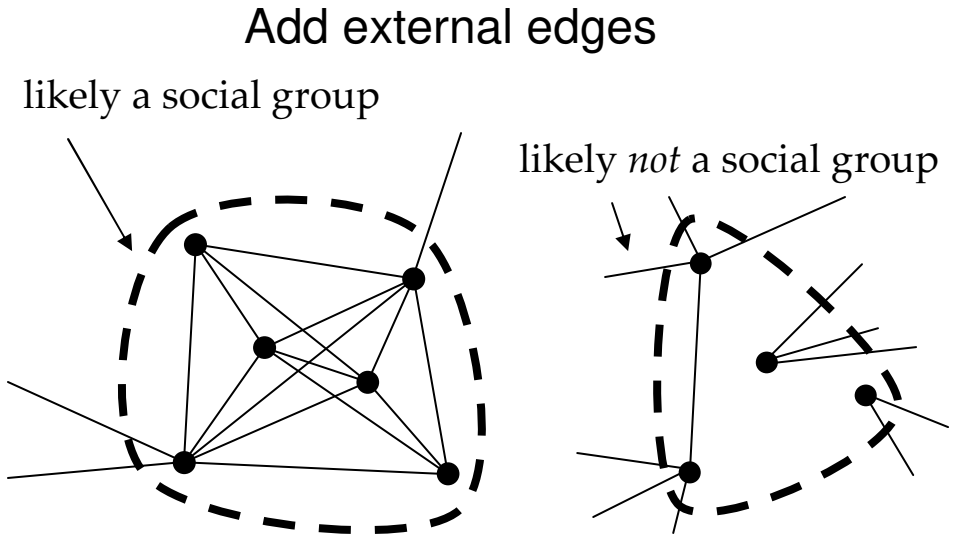
- Malicious groups use large communication networks for planning and coordination
- Their goal: remain undetected
- Our goal: sift through communications for suspicious patterns, using *structure only*, not content

Rensselaer

Communities as clusters

- Detecting all social groups (malicious or not) will aide in searching for "hidden" groups
- Social groups tend to communicate densely
- Approach: Find social groups by finding *clusters* in the graph of the communication network



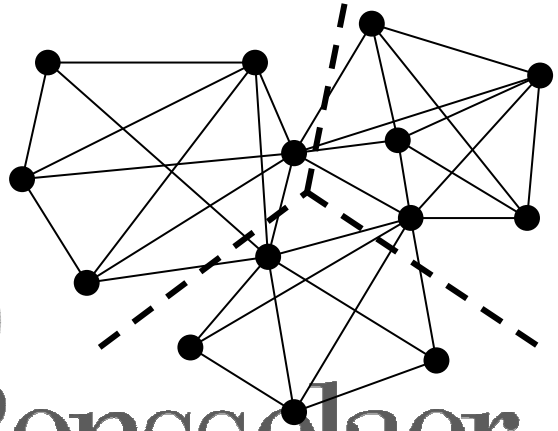


Rensselaer

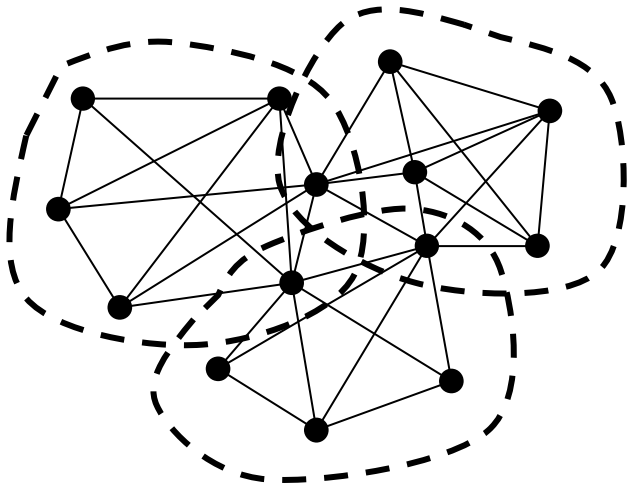
What is a cluster?

- Many partitioning algorithms exist
- Social groups often overlap
- Instead define clusters as locally optimal with respect to density

partitioning

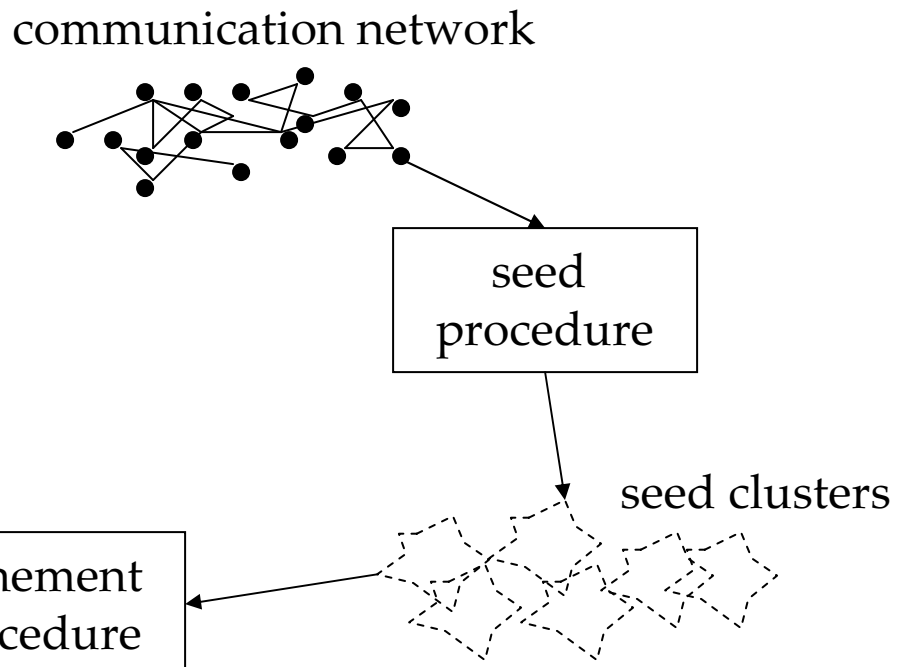


overlapping clustering



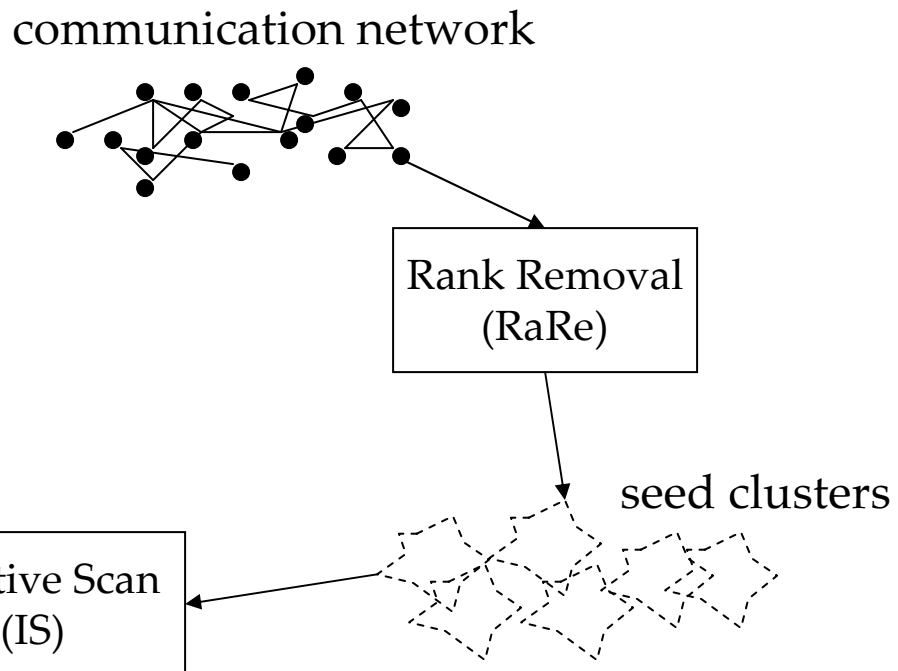
Rensselaer

Two-stage process



Rensselaer

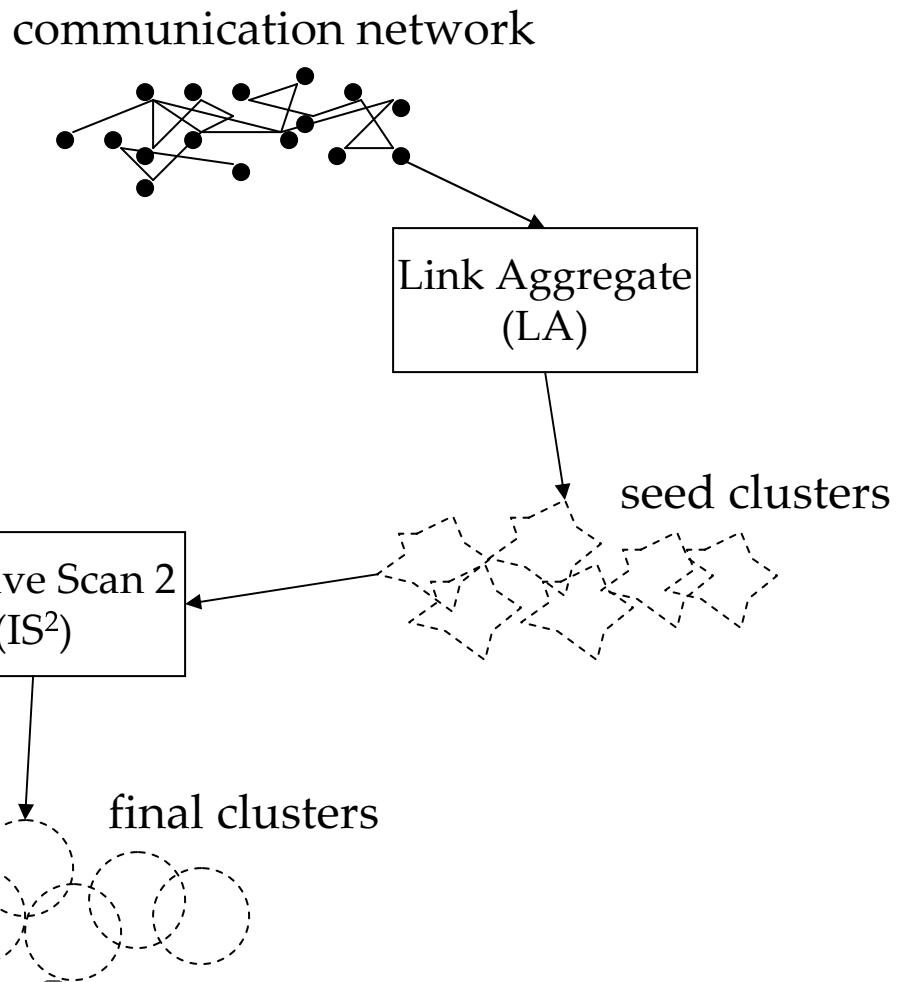
Original procedures



Jeffrey Baumes, Mark Goldberg, Mukkai Krishnamoorthy, Malik Magdon-Ismail, Nathan Preston. "**Finding Communities by Clustering a Graph into Overlapping Subgraphs**", International Conference on Applied Computing (IADIS 2005), Feb 22-25, Algarve, Portugal.

Rensselaer

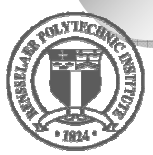
Proposed new procedures



Rensselaer

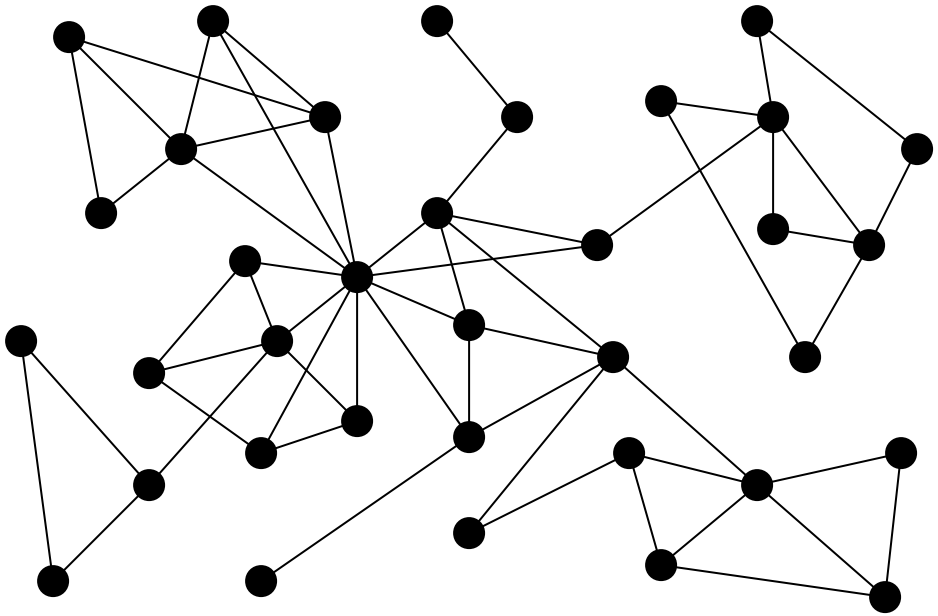
Link Aggregate (LA)

- Order the nodes *(two routines are used)*
- Pass through the nodes
  - For each node, add it to the clusters it improves, or start a new cluster



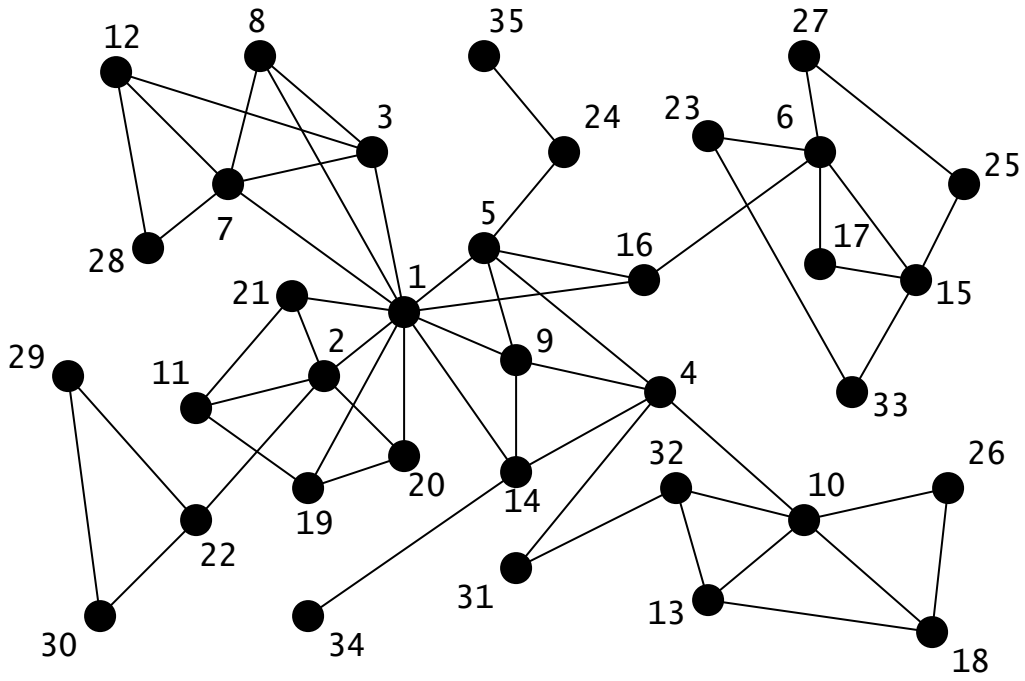
Rensselaer

LA procedure



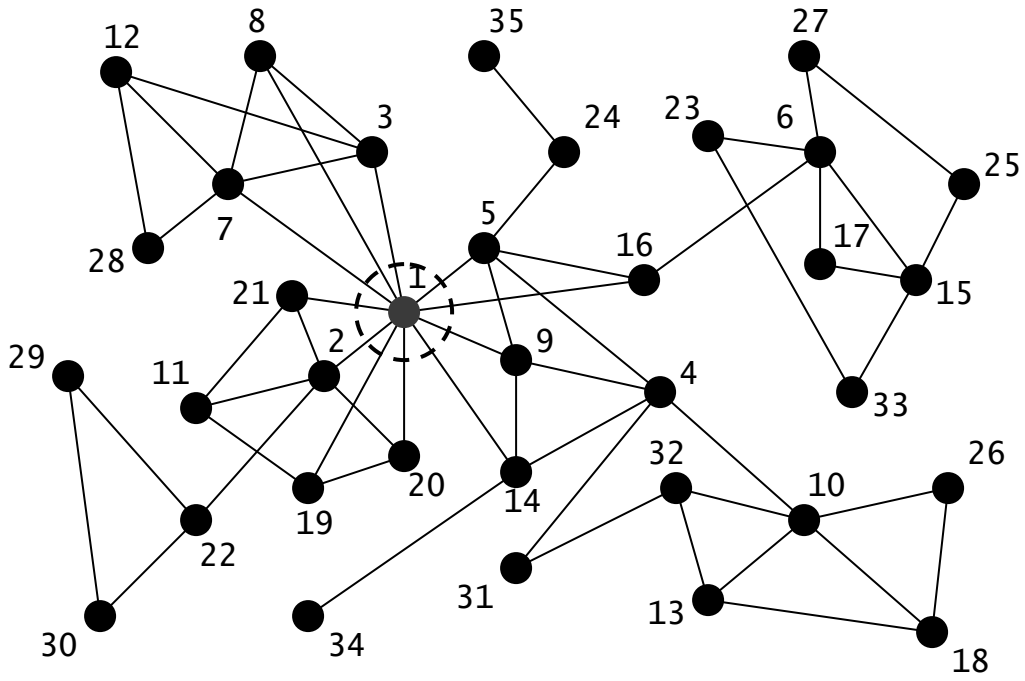
Rensselaer

LA procedure



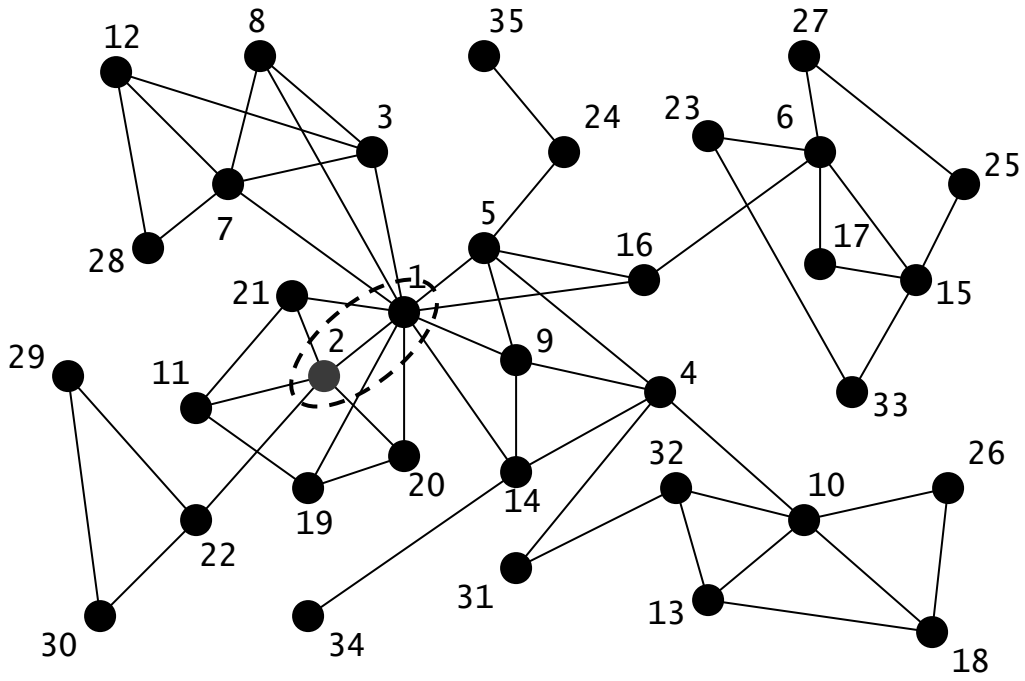
Rensselaer

LA procedure



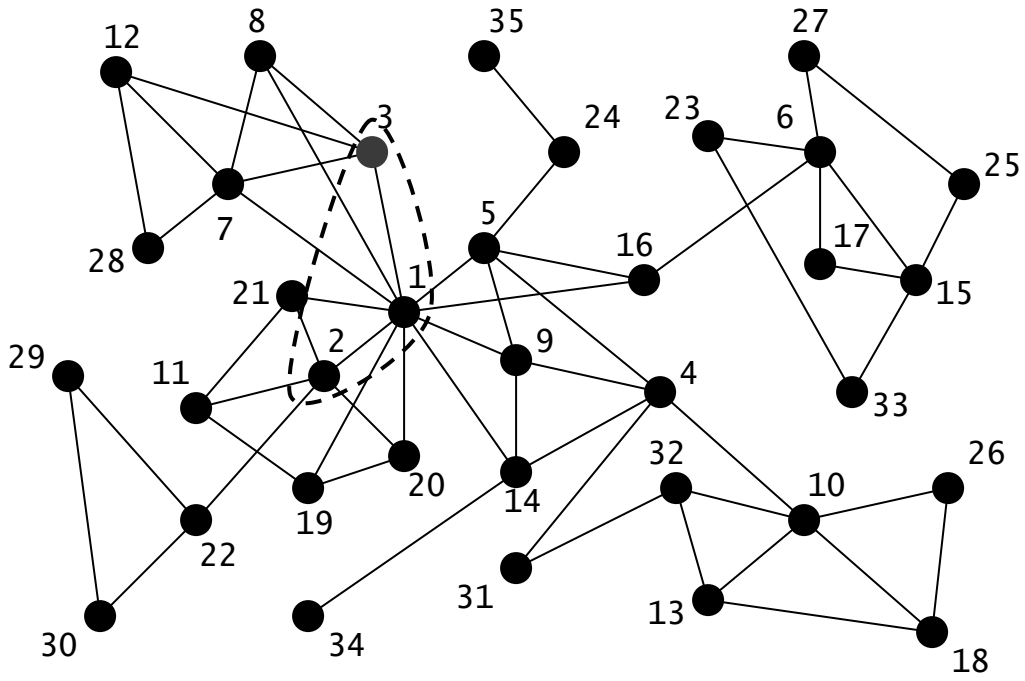
Rensselaer

LA procedure



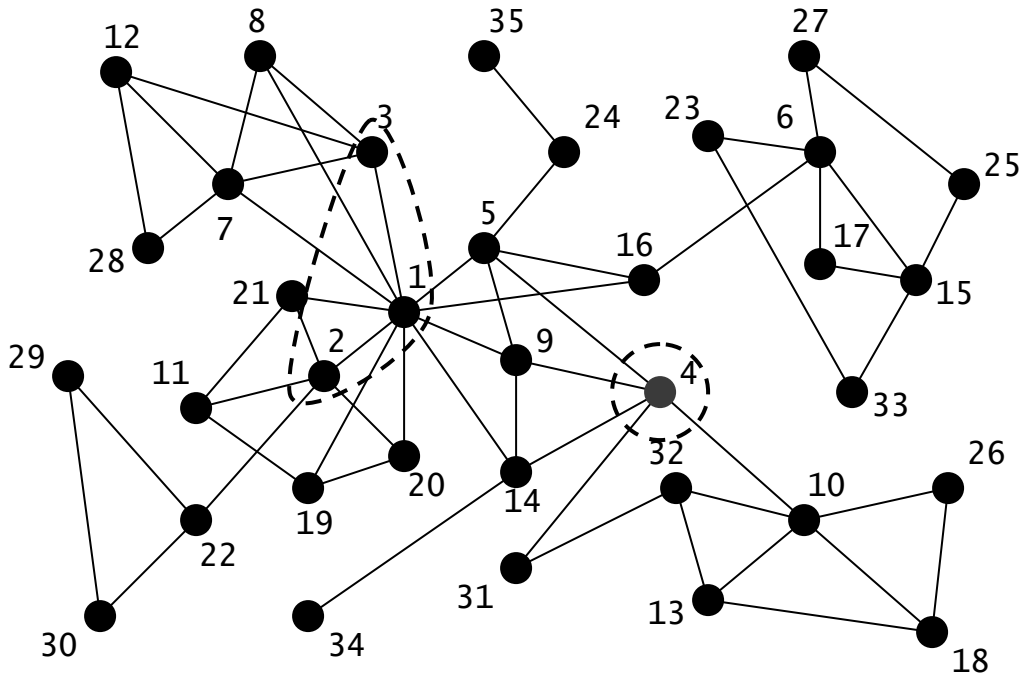
Rensselaer

LA procedure



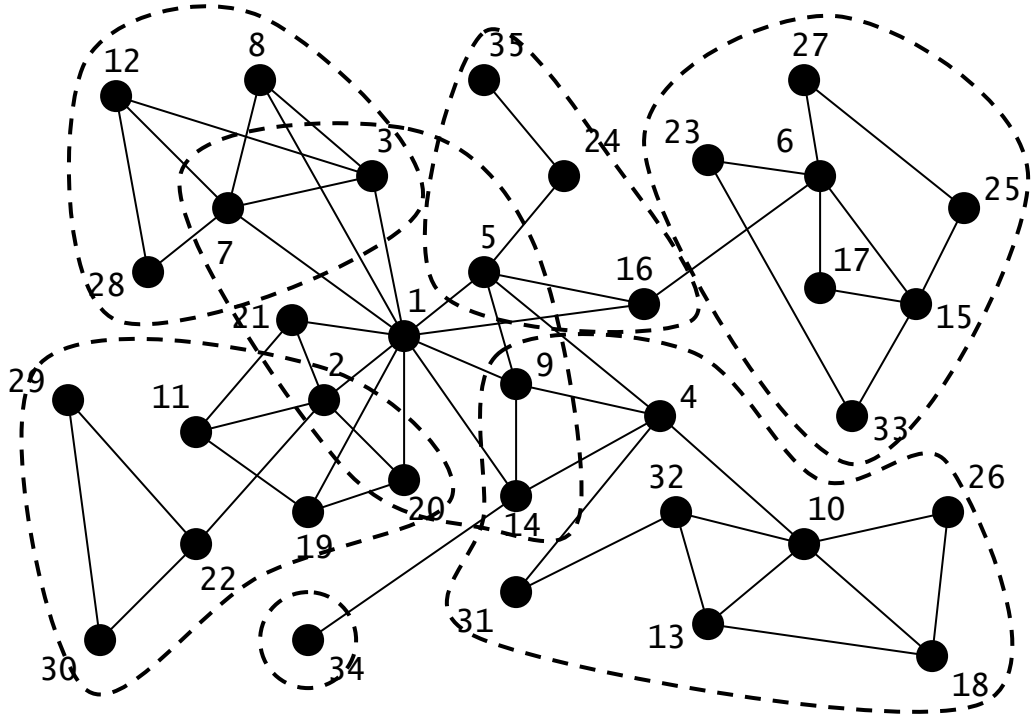
Rensselaer

LA procedure



Rensselaer

LA procedure



Rensselaer

Iterative Scan (IS)

- Old refinement procedure
  - Traverses *entire* node list, adding / removing nodes which increase the density
  - Repeats the process until no improvements are possible
- May be inefficient in sparse networks\
- Guaranteed to be locally optimal

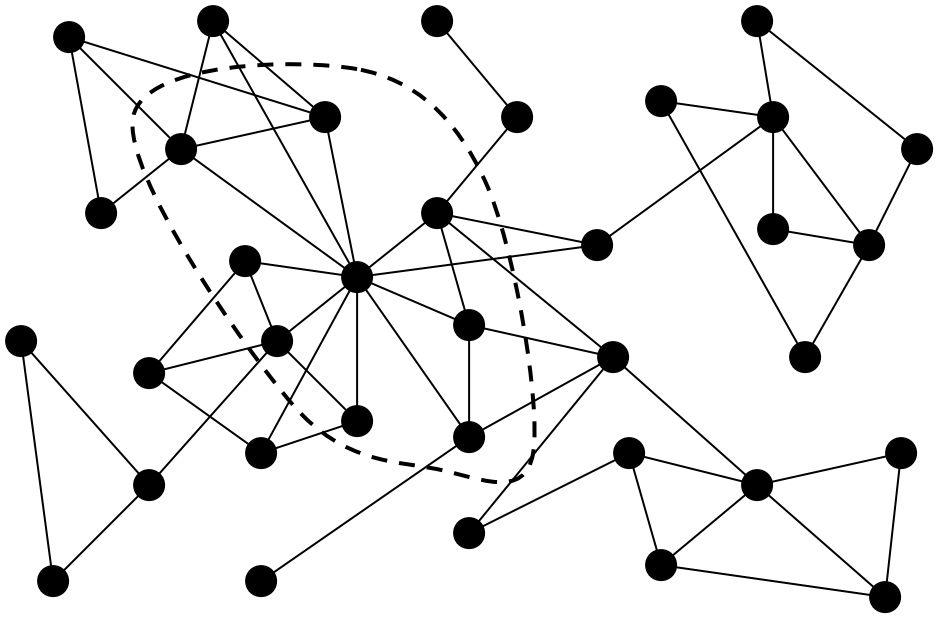
Rensselaer

Iterative Scan 2 (IS^2)

- New refinement procedure
  - Traverses *neighborhood of cluster only*, adding / removing nodes which increase the density
  - Repeats the process until no improvements are possible
- More efficient in sparse networks in spite of overhead, less efficient in dense networks

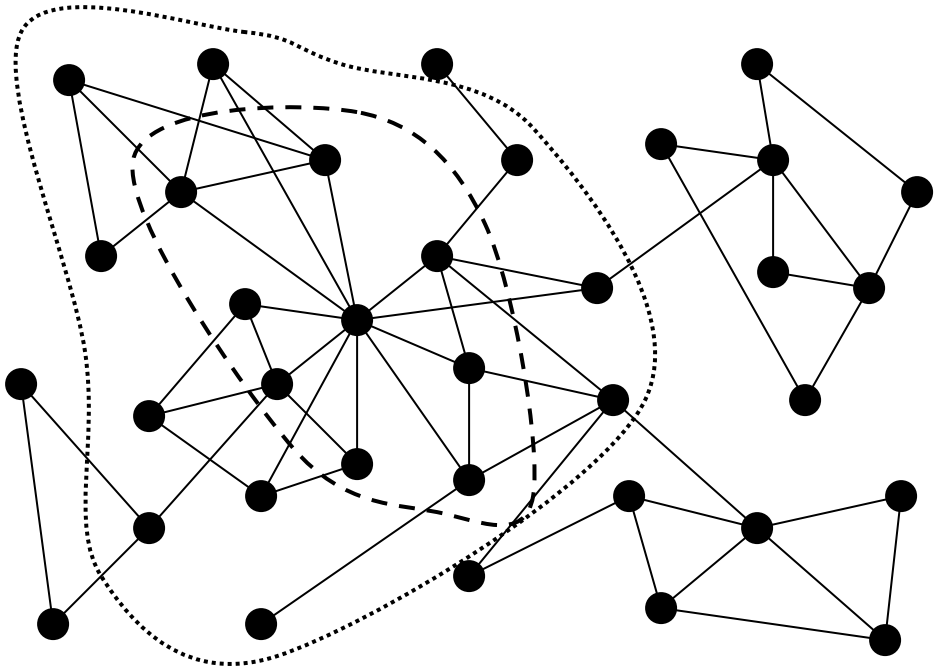
Rensselaer

IS² procedure



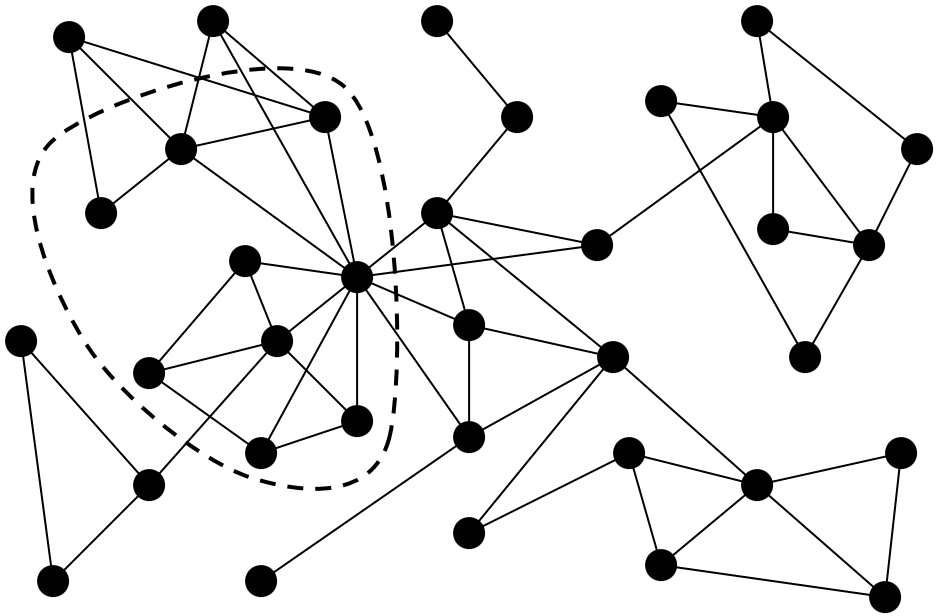
Rensselaer

IS² procedure



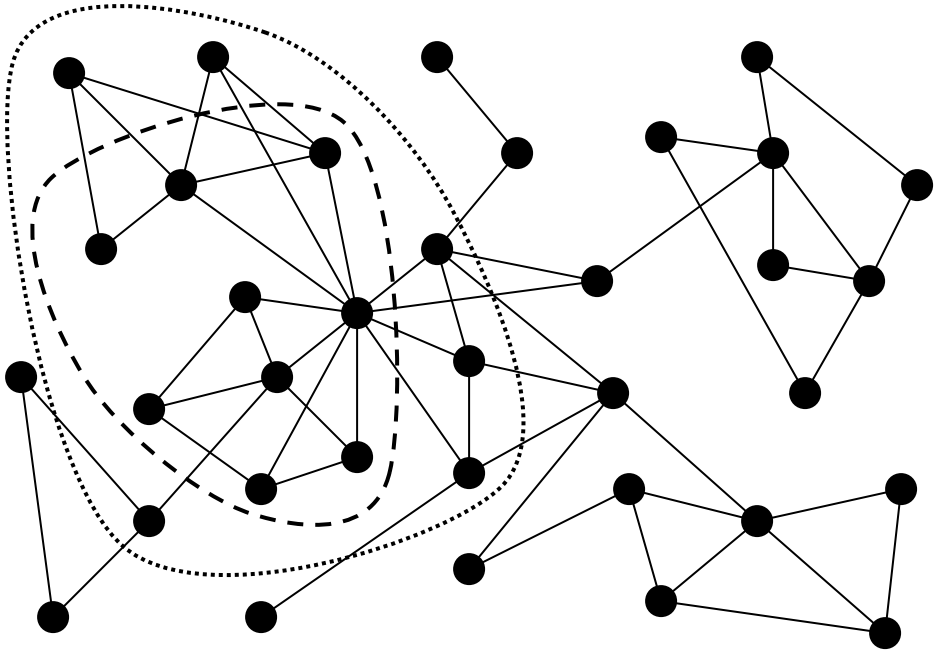
Rensselaer

IS² procedure



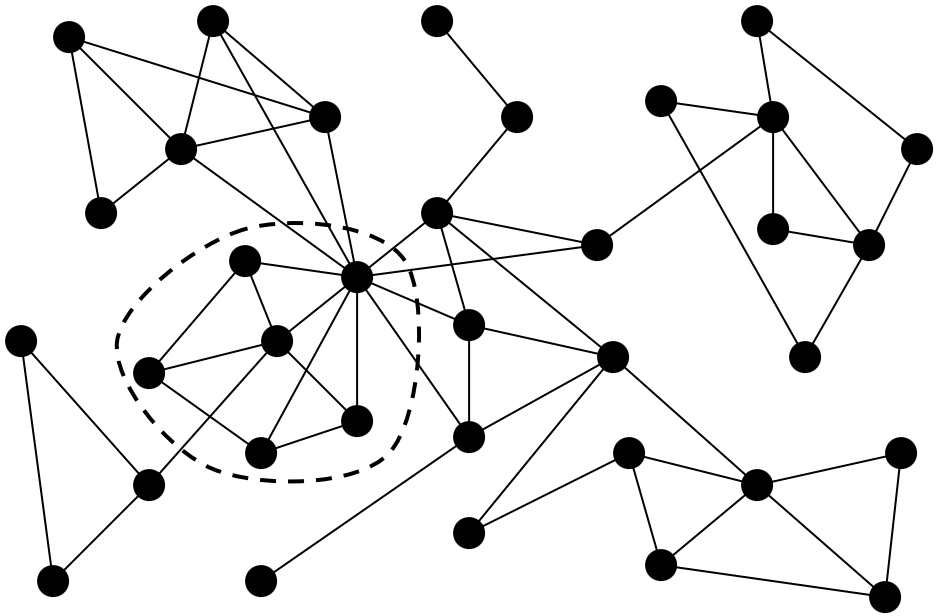
Rensselaer

IS² procedure



Rensselaer

IS² procedure



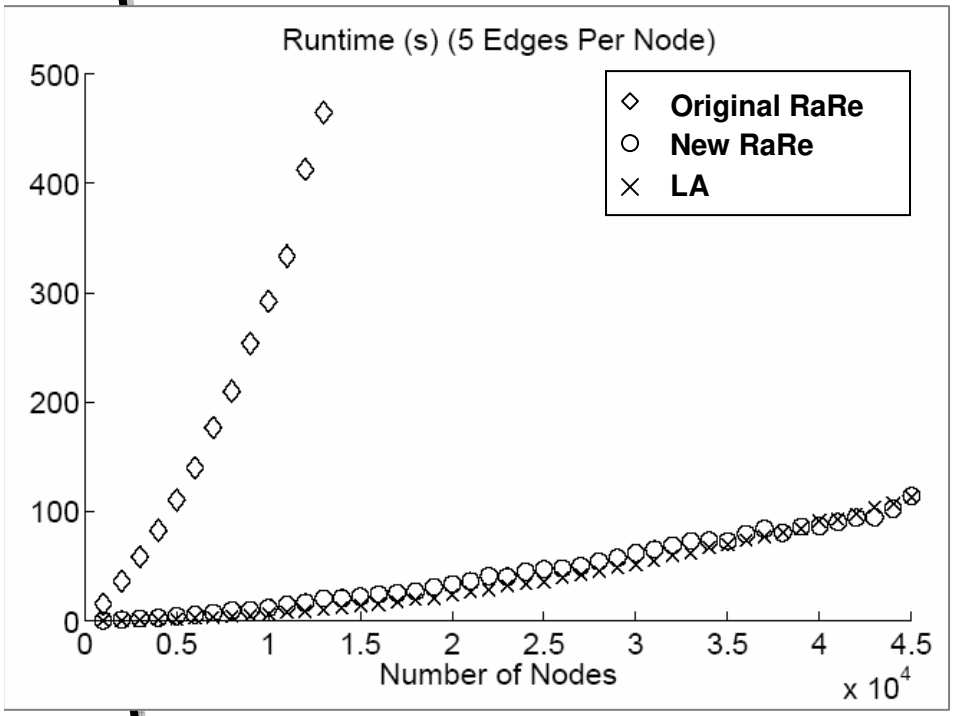
Rensselaer

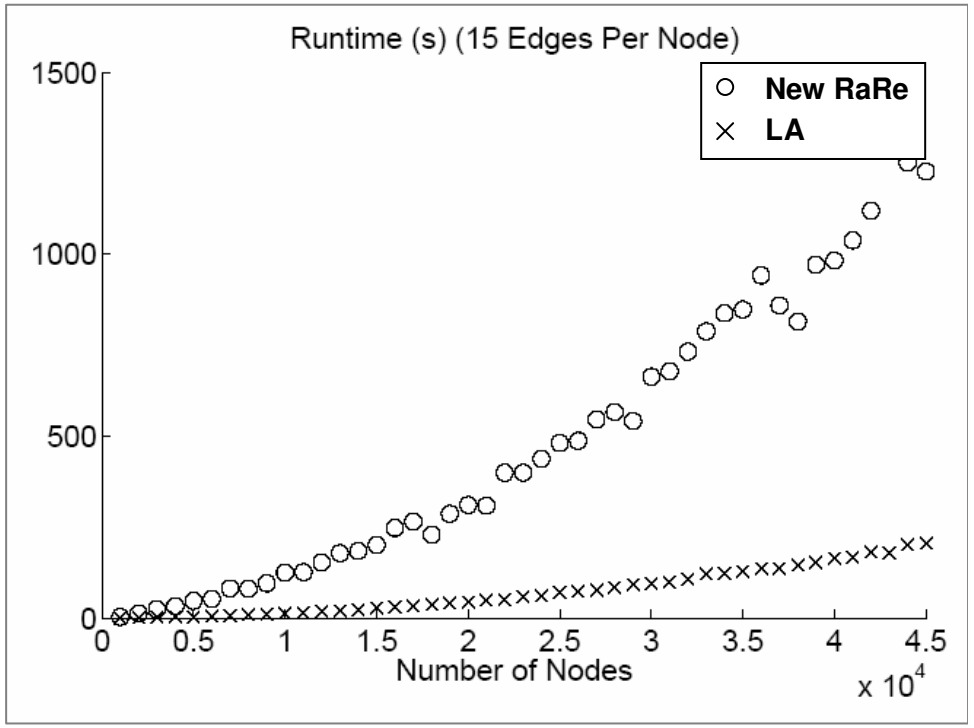
Experimental results

- Compare run time of new vs. old
- Compare cluster quality of new vs. old
- Compare on different network types
  - Random
  - Preferential attachment
  - Real-world
- Compare possible actor orderings for LA

Rensselaer

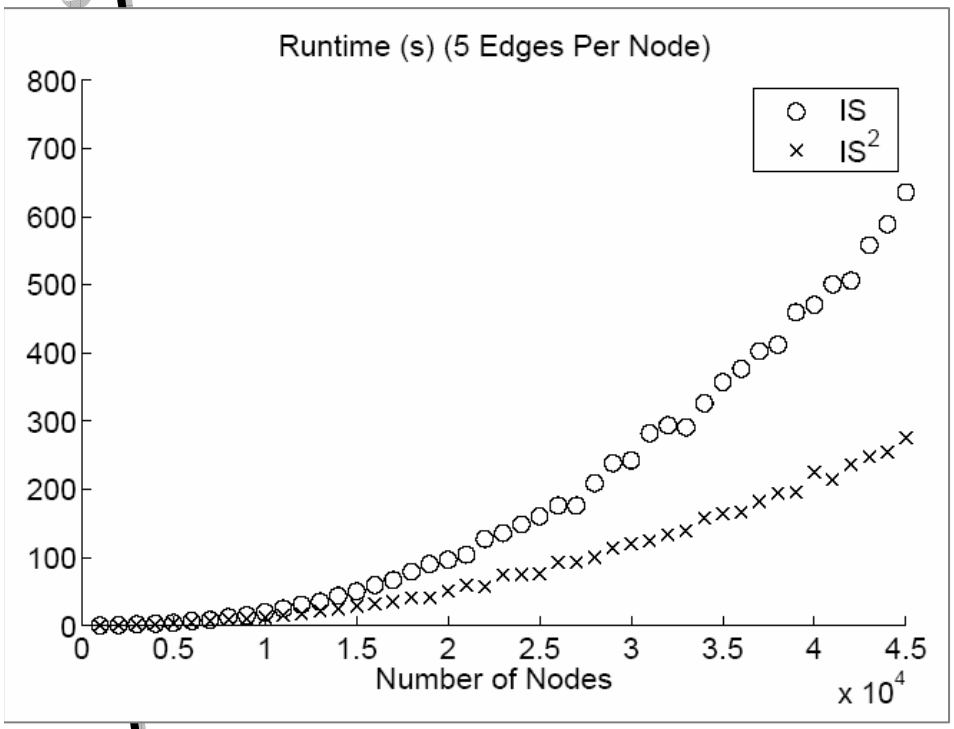
RaRe vs. LA run time

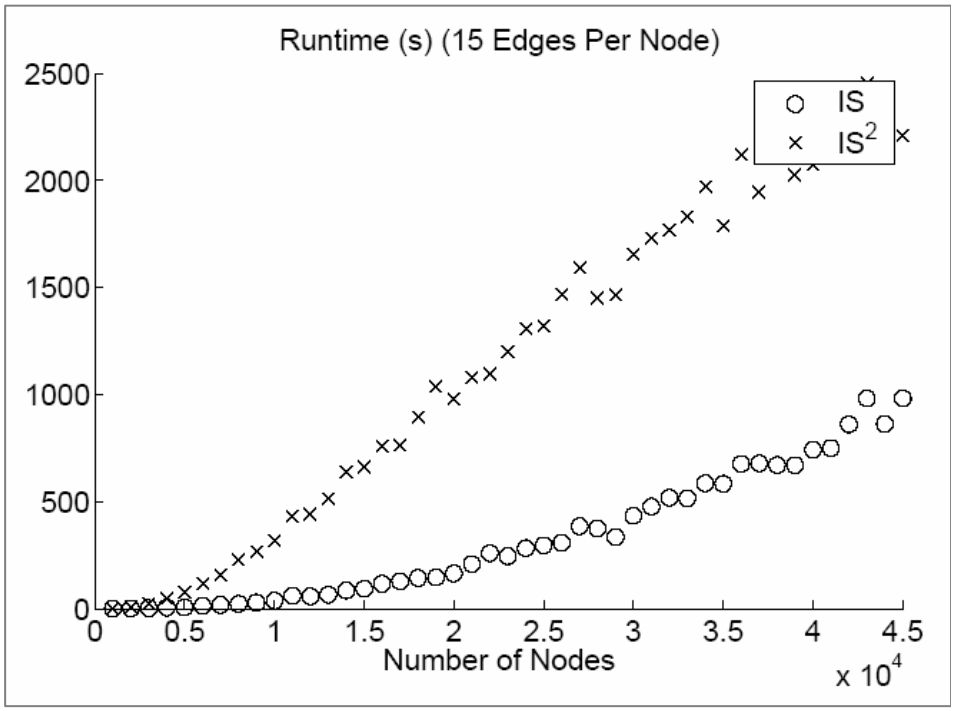




Rensselaer

IS vs. IS^2 run time

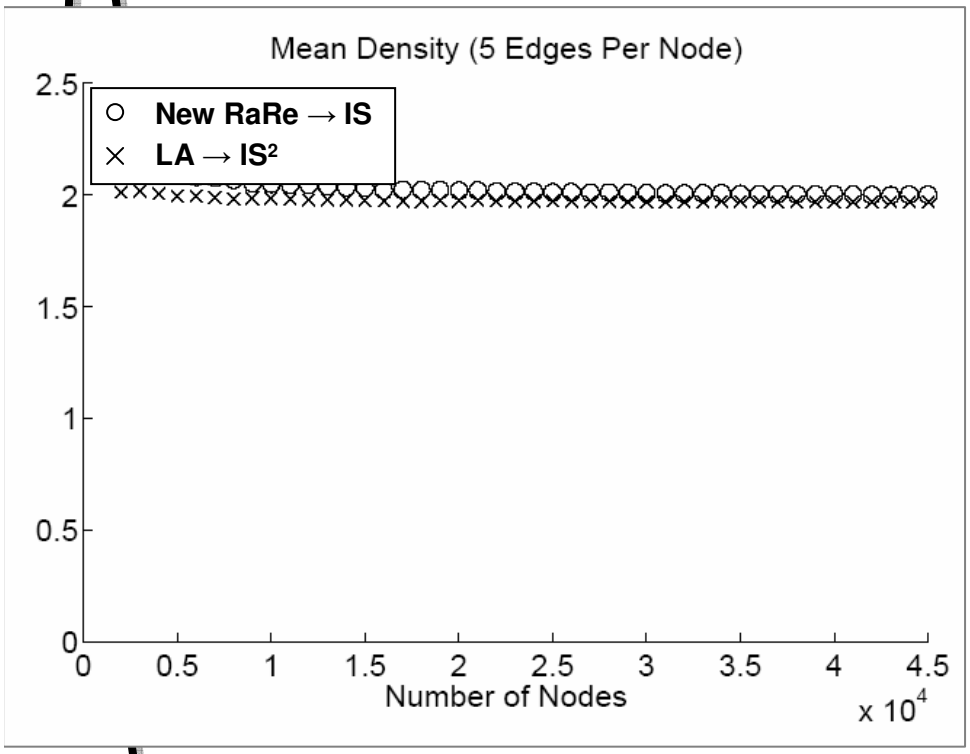


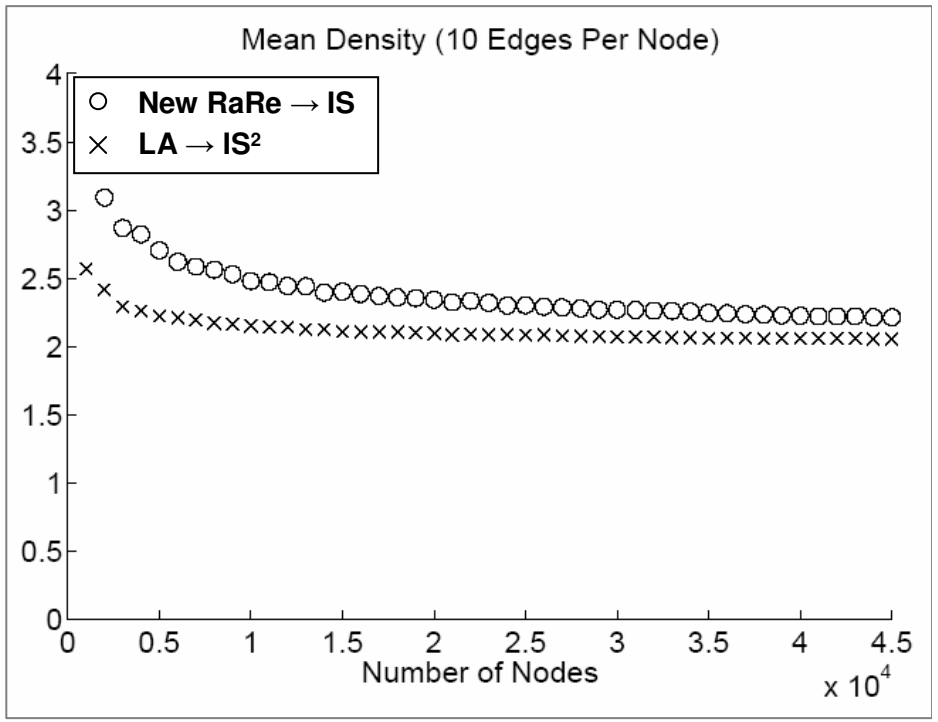


Define IS* = IS for dense graphs, IS^2 for sparse graphs

Rensselaer

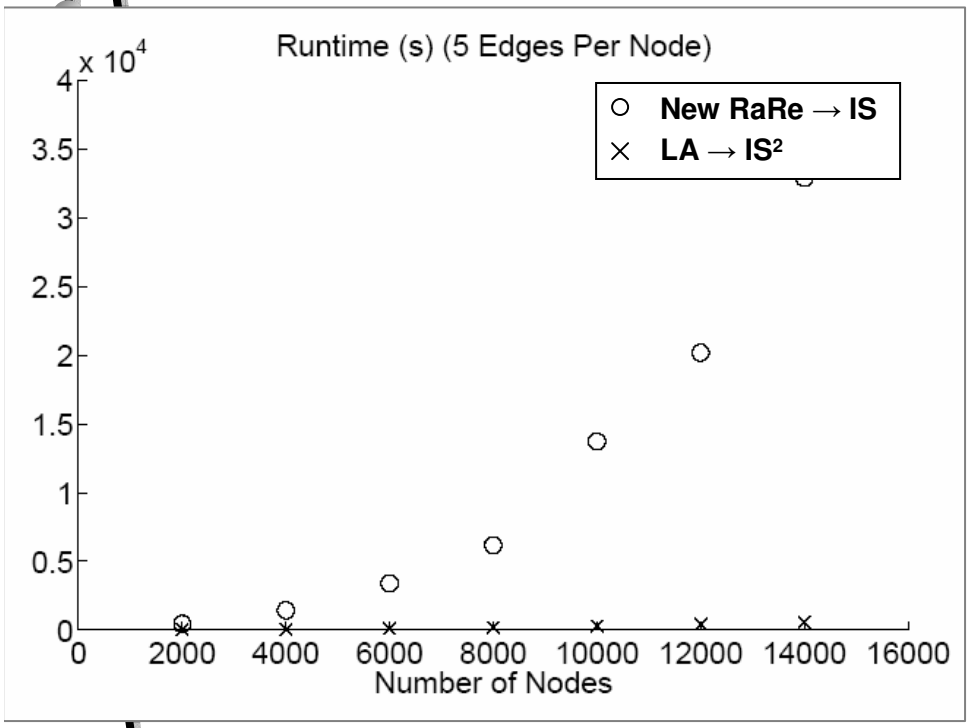
Old vs. new quality

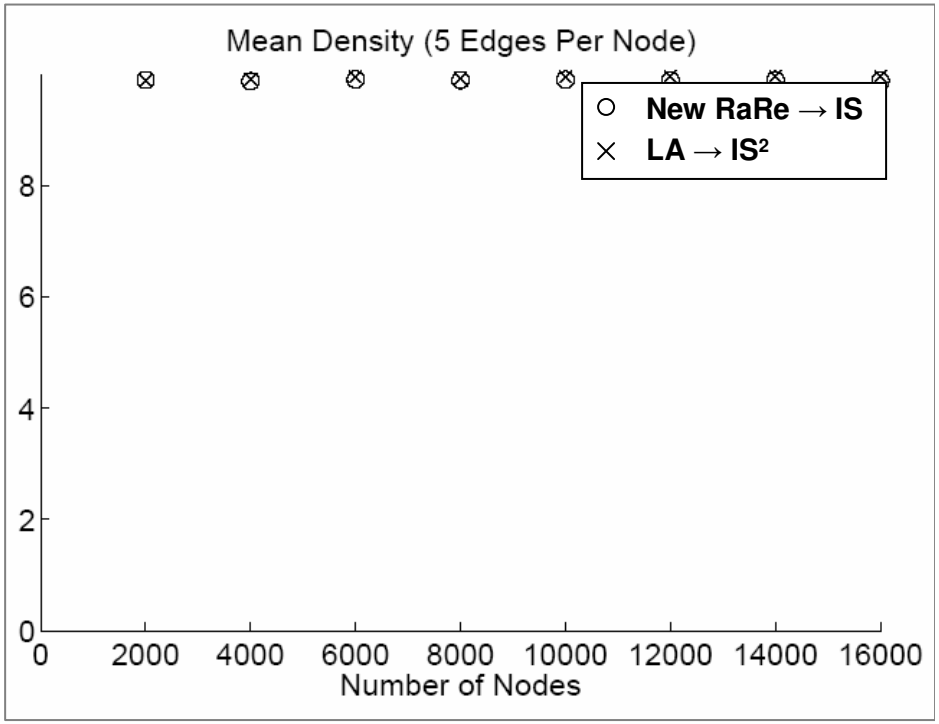




Rensselaer

Preferential attachment



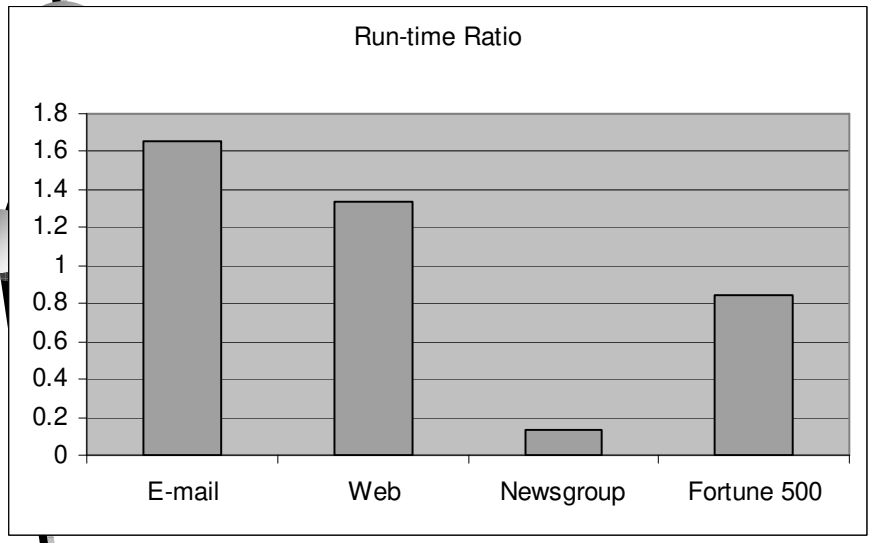


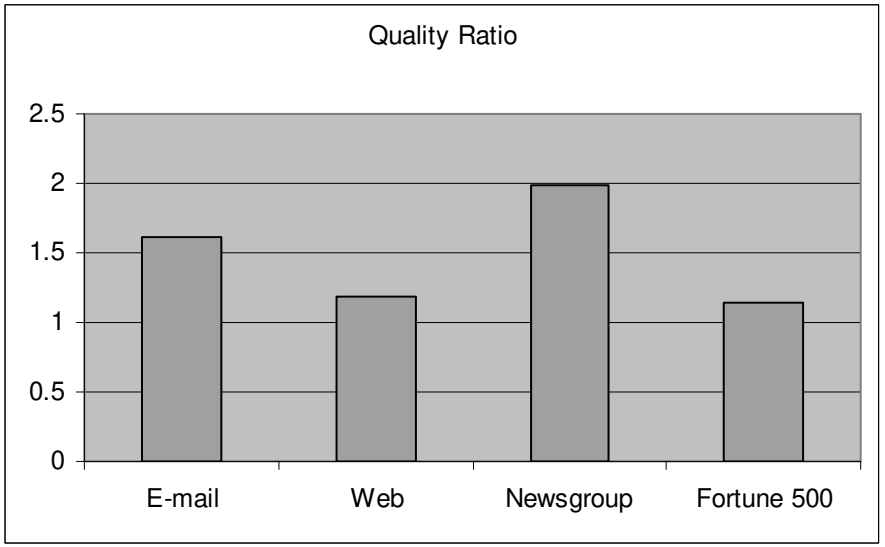
Rensselaer

Real-World Networks

Ratio = new/old

= (LA→IS*)/(RaRe→IS)

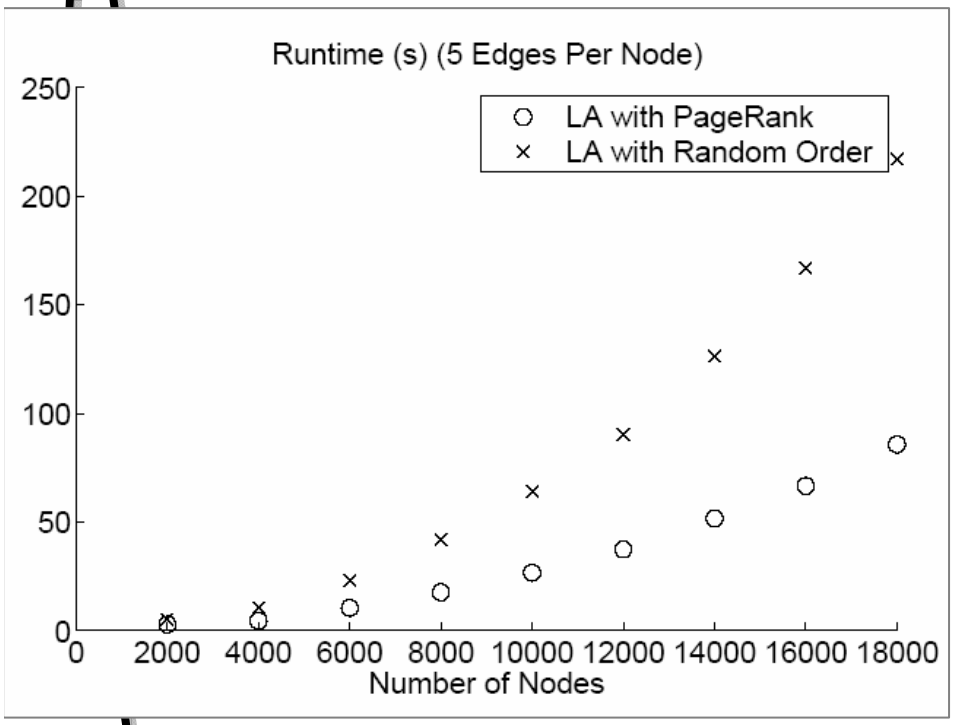


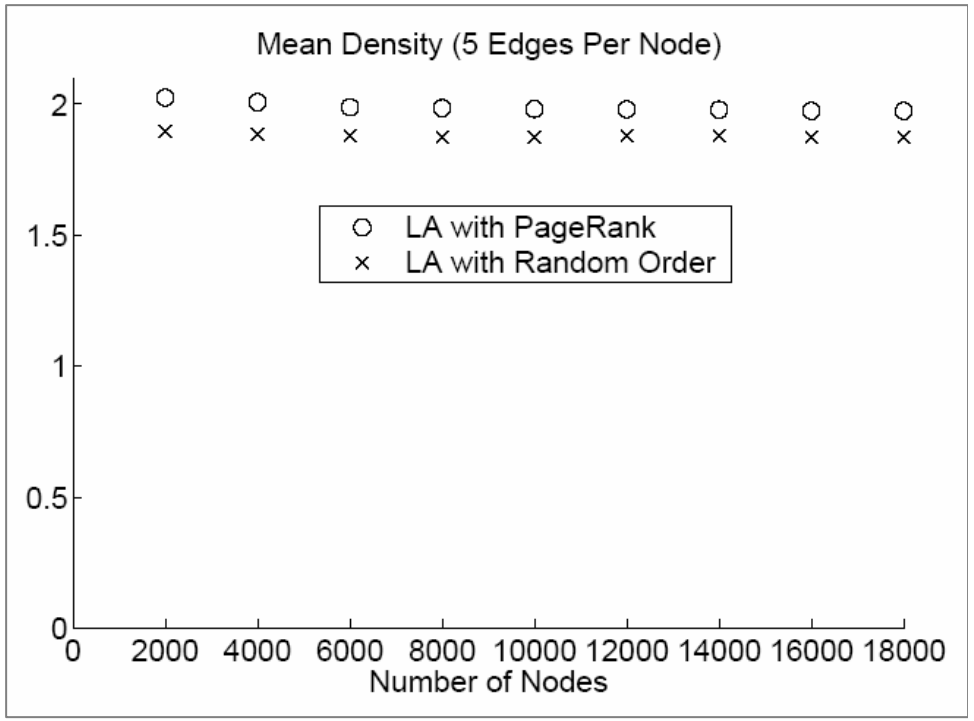


IS* = IS^2 IS IS^2 IS^2

Rensselaer

LA ordering





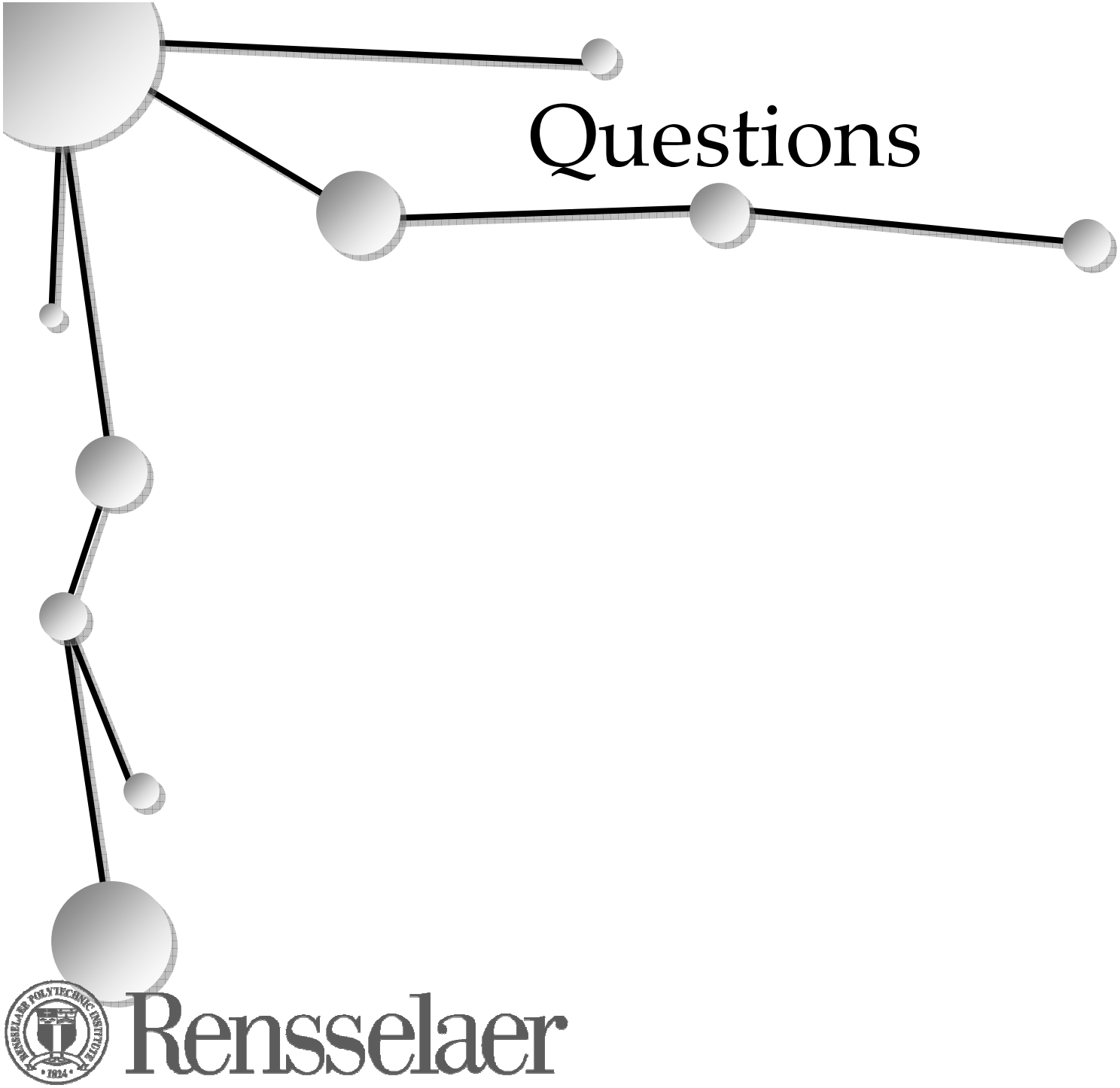
Rensselaer

Conclusions and future work

- Overlapping clustering may be used to discover social groups in communication networks
- The new algorithm is more efficient in many cases, while keeping the same or better quality
- A unified algorithm should choose strategies and parameters based on network properties

Rensselaer

Questions



Rensselaer

Rank Removal

- Existing seed procedure
  - Removes highly connected nodes until network is broken into small clusters
  - Adds removed nodes back into clusters it is well-connected to
- Two main inefficiencies
  - Computed Page Rank at each iteration
  - Computed connected components at each iteration
- Page Rank could be computed once, but reprocessing connected components is crucial

Rensselaer

LA procedure detail

procedure LA($G = (V, E), W$)
$C \leftarrow \emptyset$;
Order the vertices $v_1, v_2, \ldots, v_{|V|}$;
for $i = 1$ to $|V|$ **do**
  $added \leftarrow$ false;
  for all $D_j \in C$ **do**
    if $W(D_j \cup v_i) > W(D_j)$ **then**
      $D_j \leftarrow D_j \cup v_i$; $added \leftarrow$ true;
  if $added =$ false **then**
    $C \leftarrow C \cup \{\{v_i\}\}$;
return C;

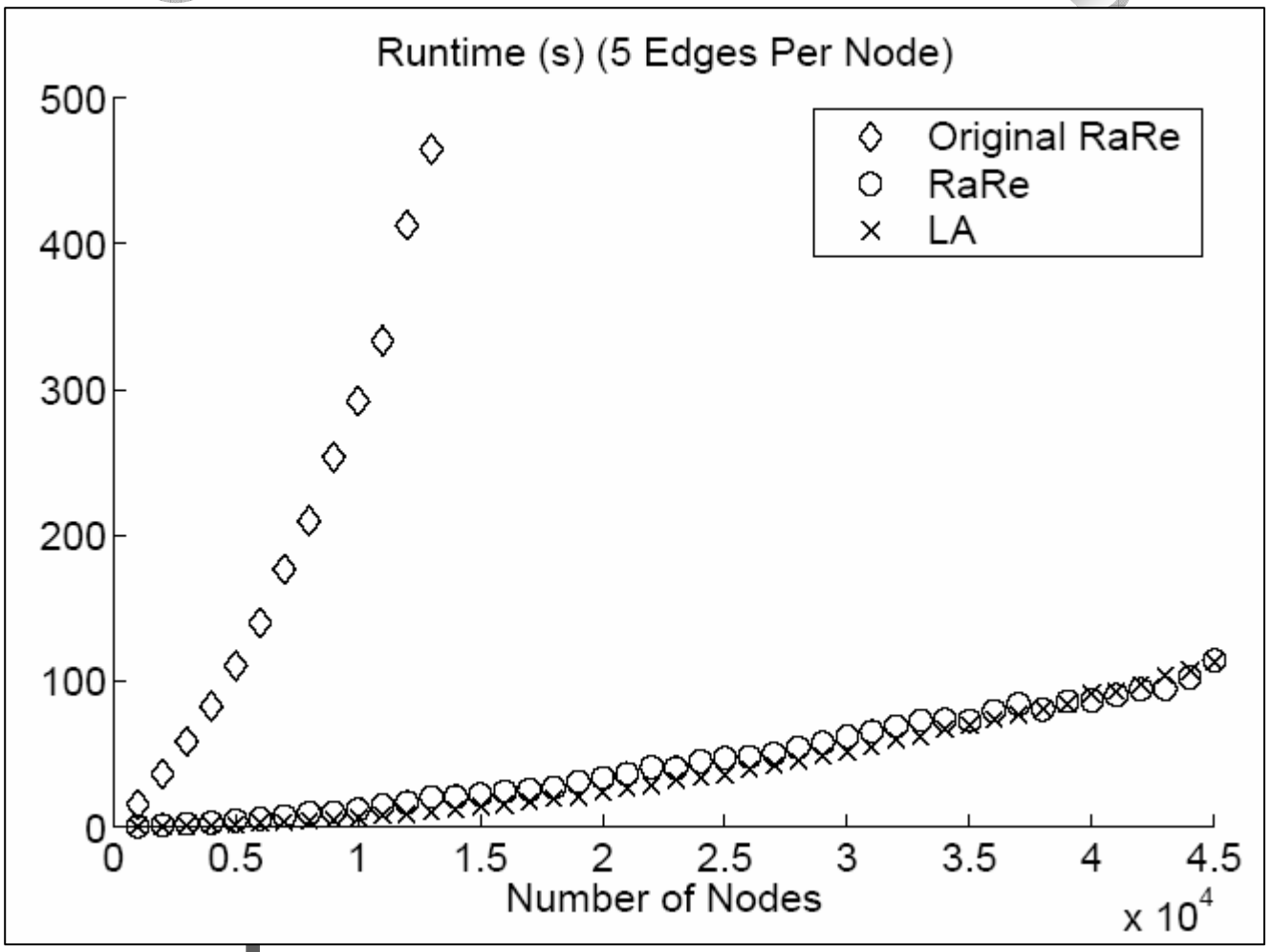
Rensselaer

IS^2 procedure detail

```
procedure IS²(seed, G, W)
C ← seed; w ← W(C);
increased ← true;
while increased do
  N ← C;
  for all v ∈ C do
    N ← N ∪ adj(v);
  for all v ∈ N do
    if v ∈ C then
      C′ ← C \ {v};
    else
      C′ ← C ∪ {v};
    if W(C′) > W(C) then
      C ← C′;
  if W(C) = w then
    increased ← false;
  else
    w ← W(C);
return C;
```

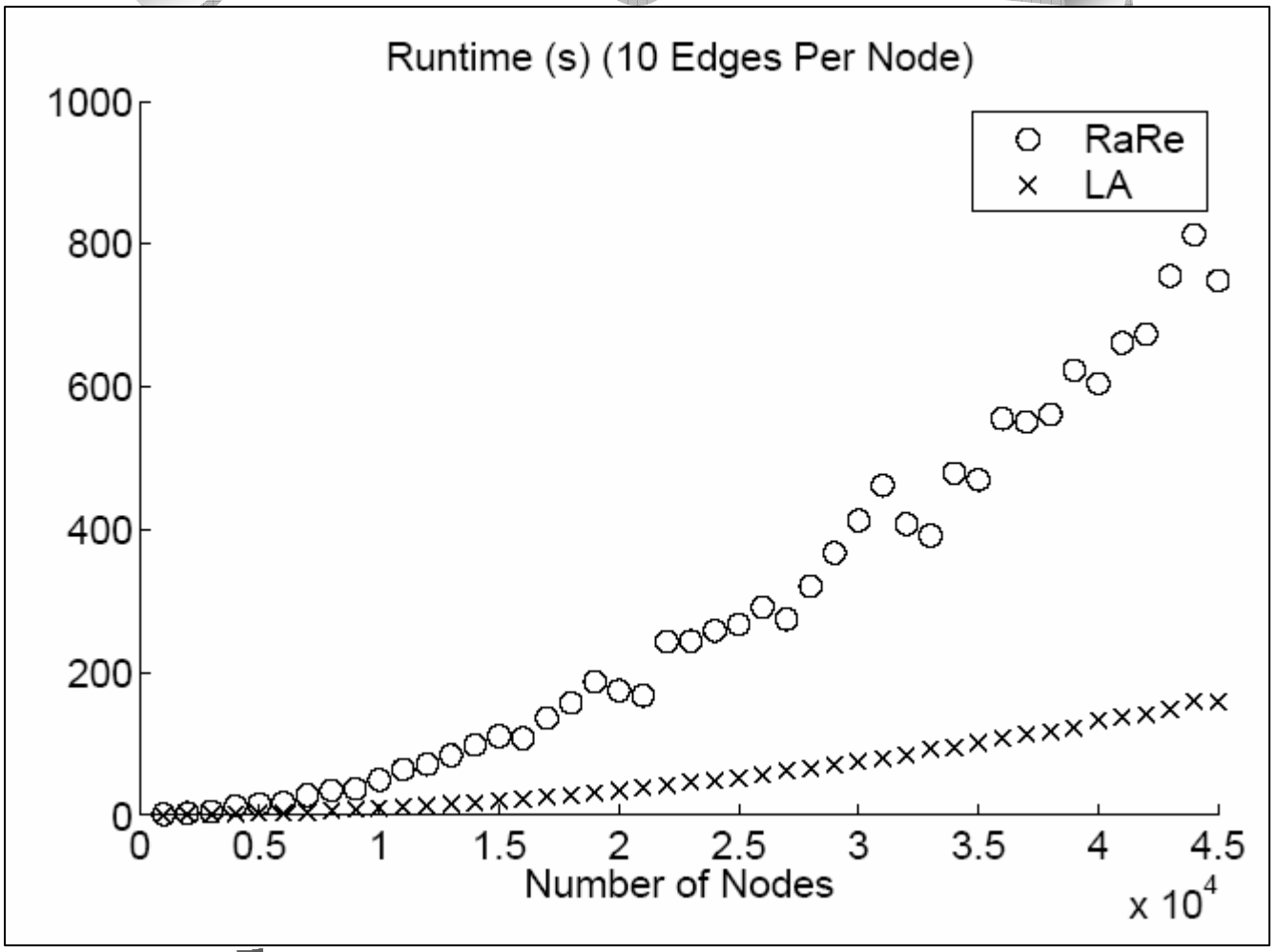
Rensselaer

RaRe vs. LA



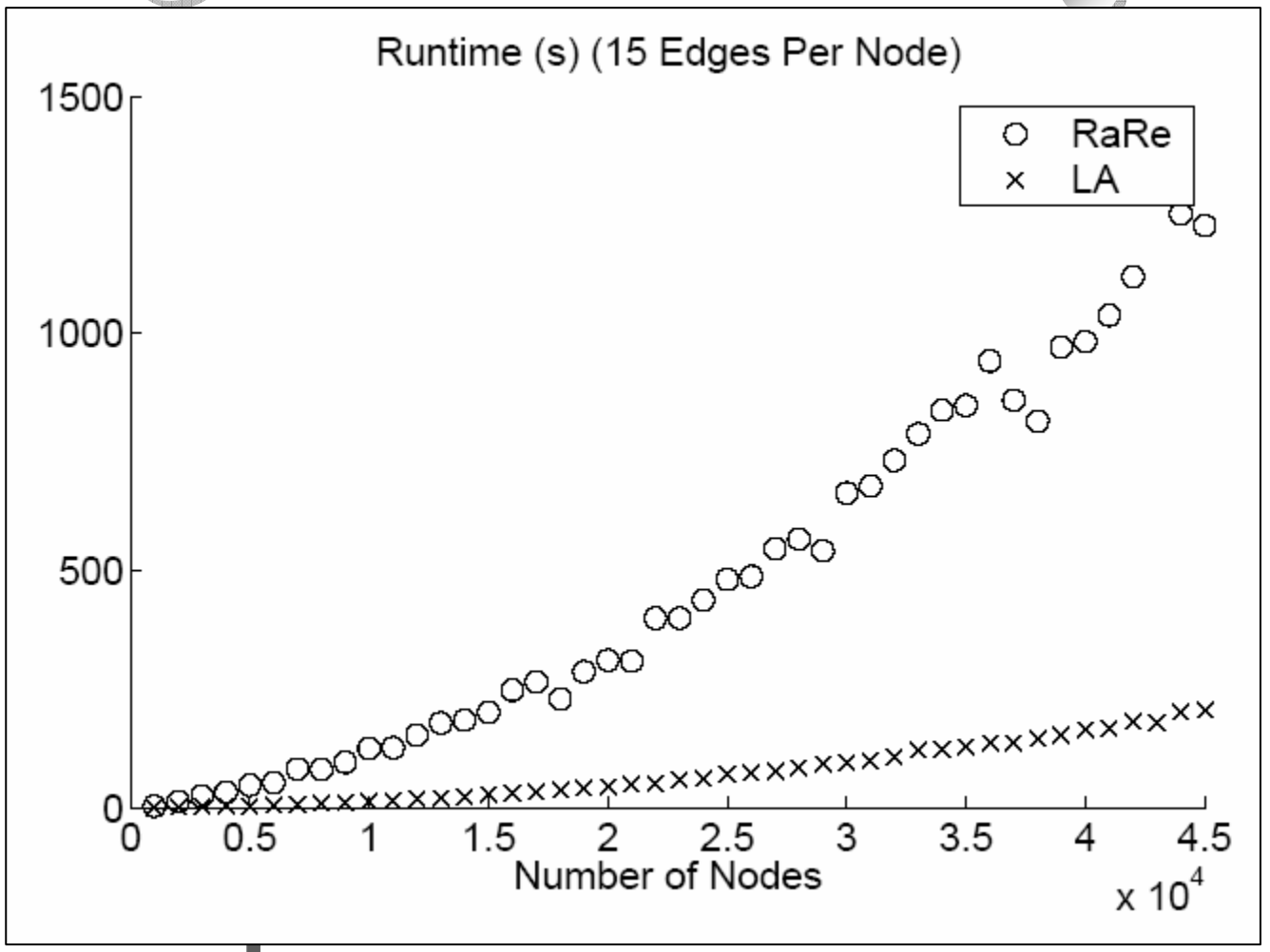
Rensselaer

RaRe vs. LA



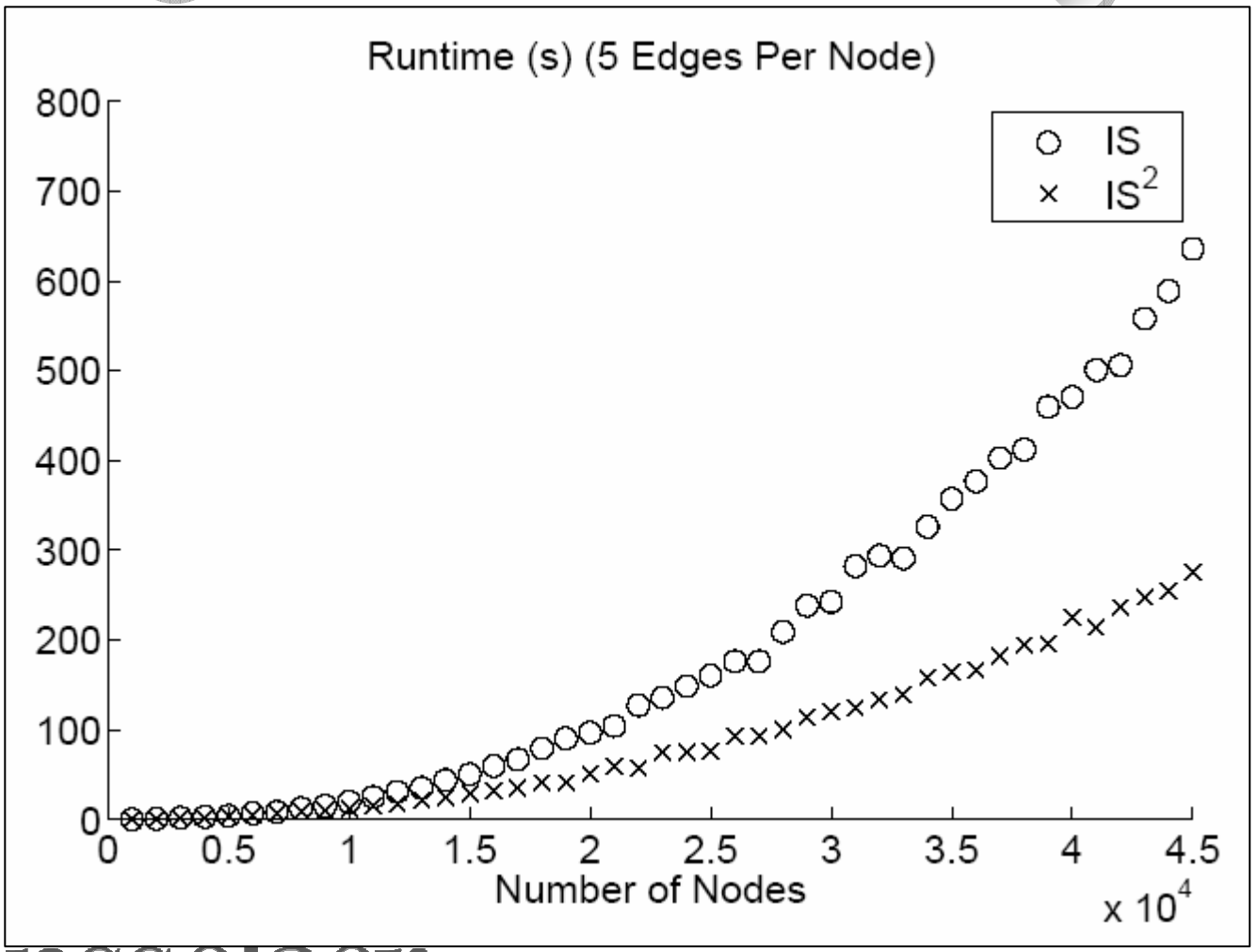
Rensselaer

RaRe vs. LA



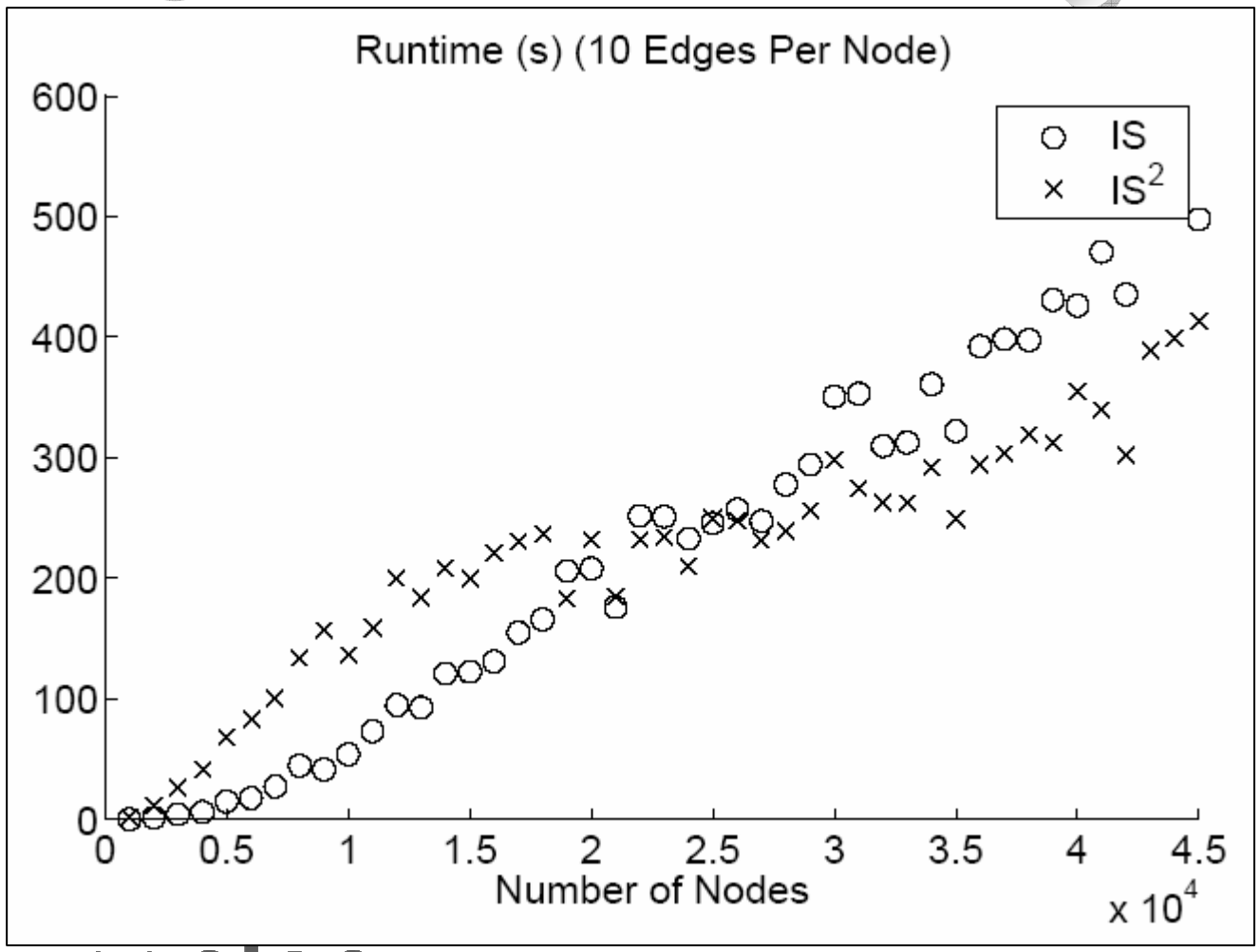
Rensselaer

IS vs. IS^2



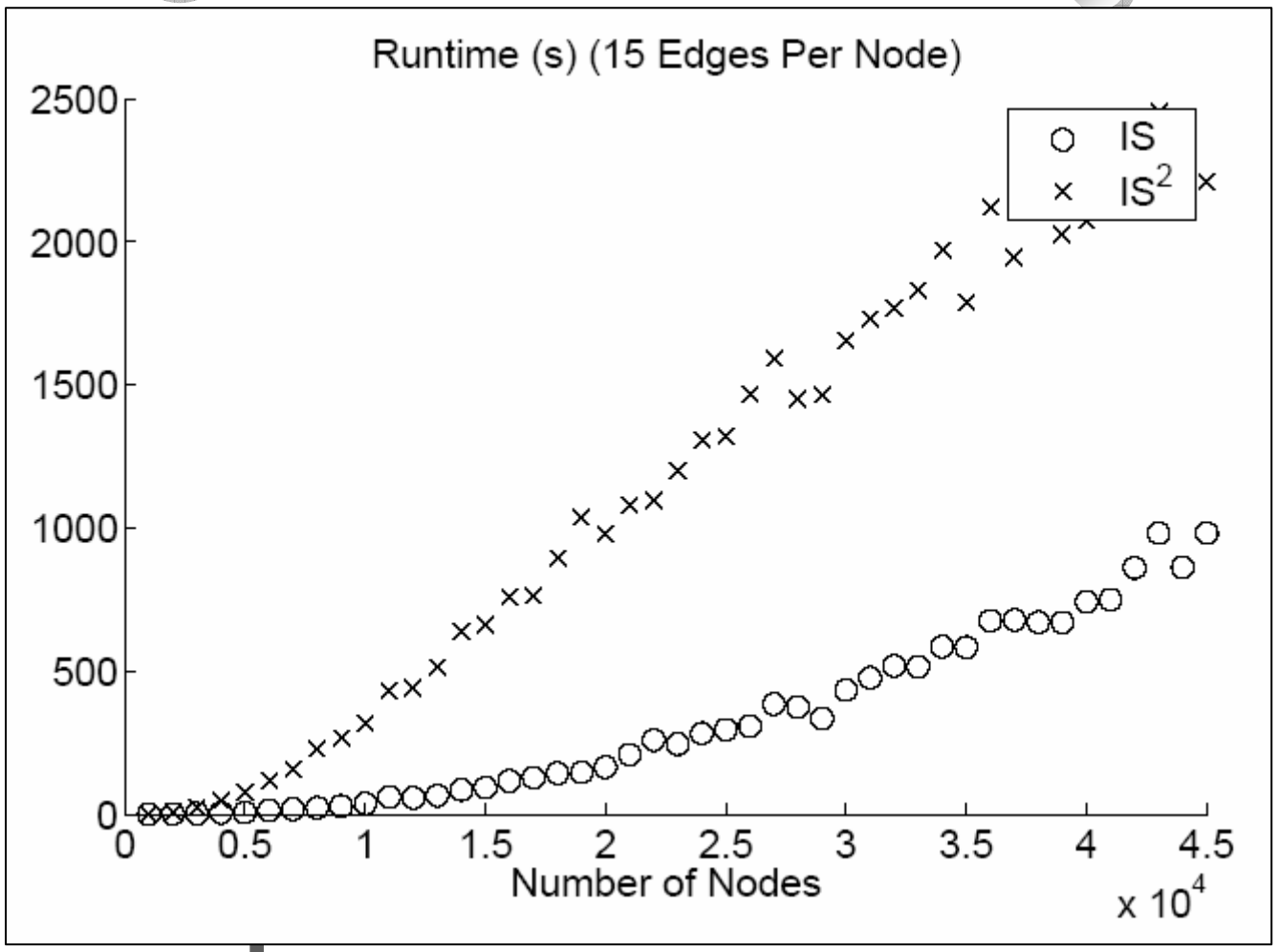
Rensselaer

IS vs. IS^2



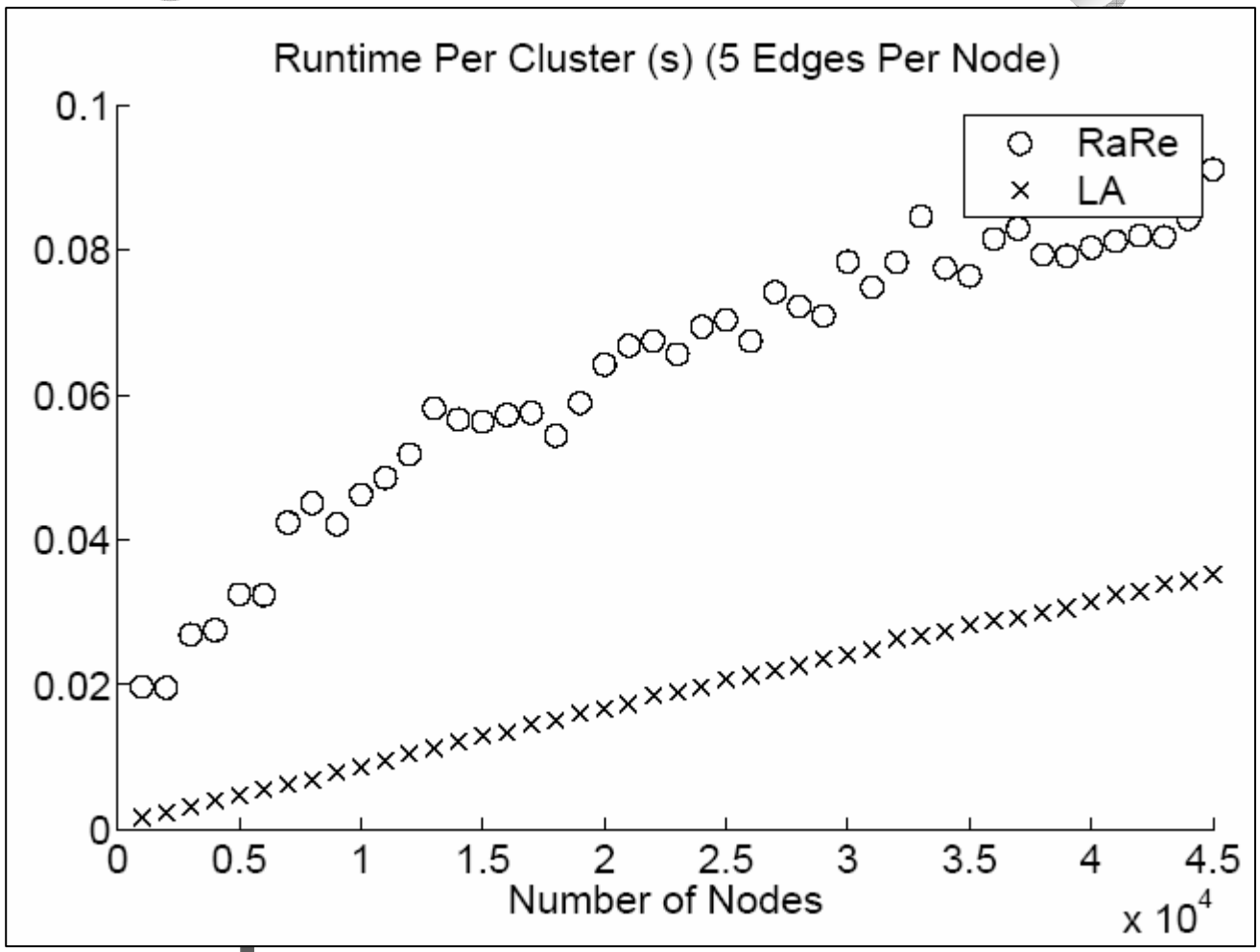
Rensselaer

IS vs. IS^2



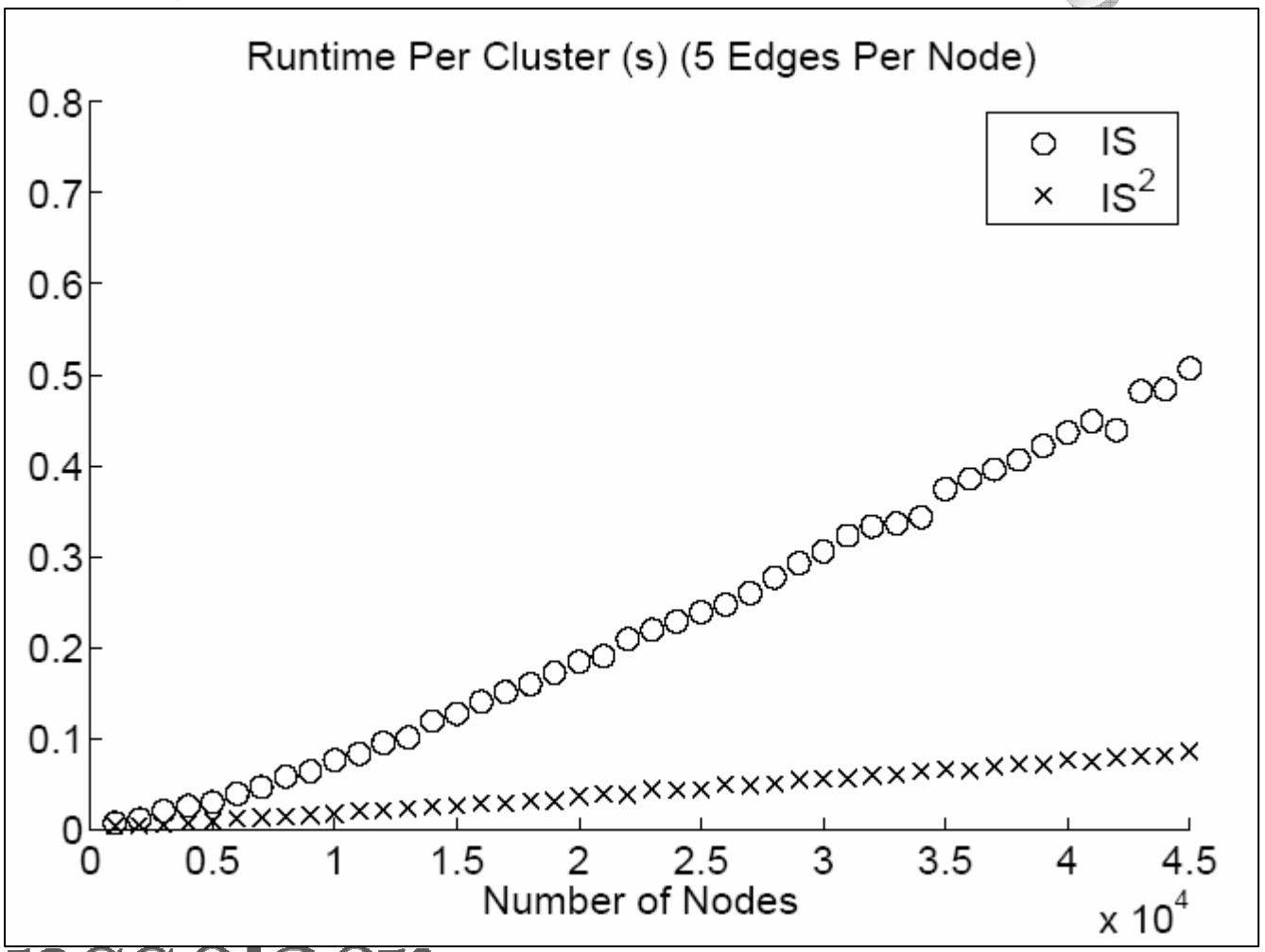
Rensselaer

Run time RaRe vs. LA



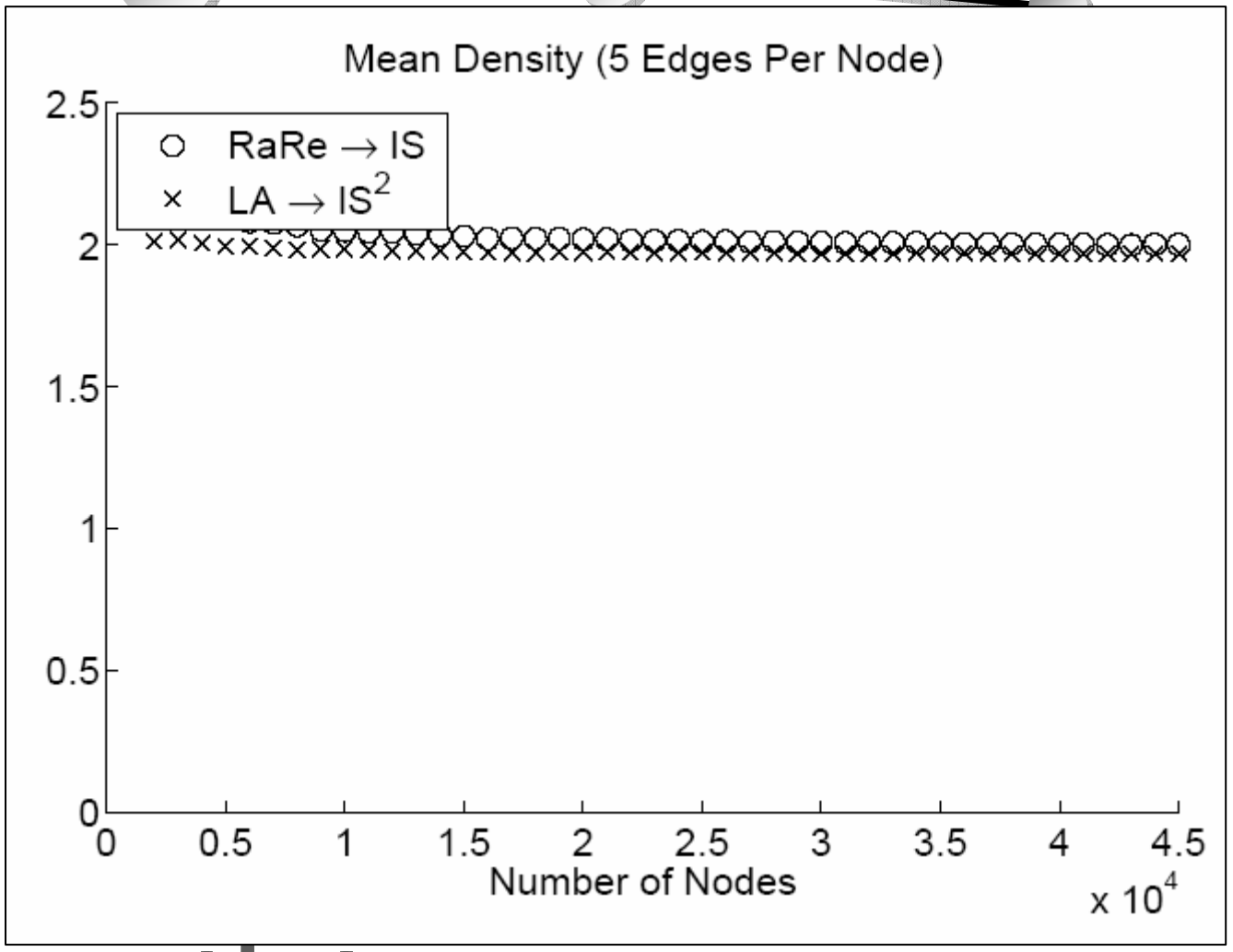
Rensselaer

Run time IS vs. IS^2



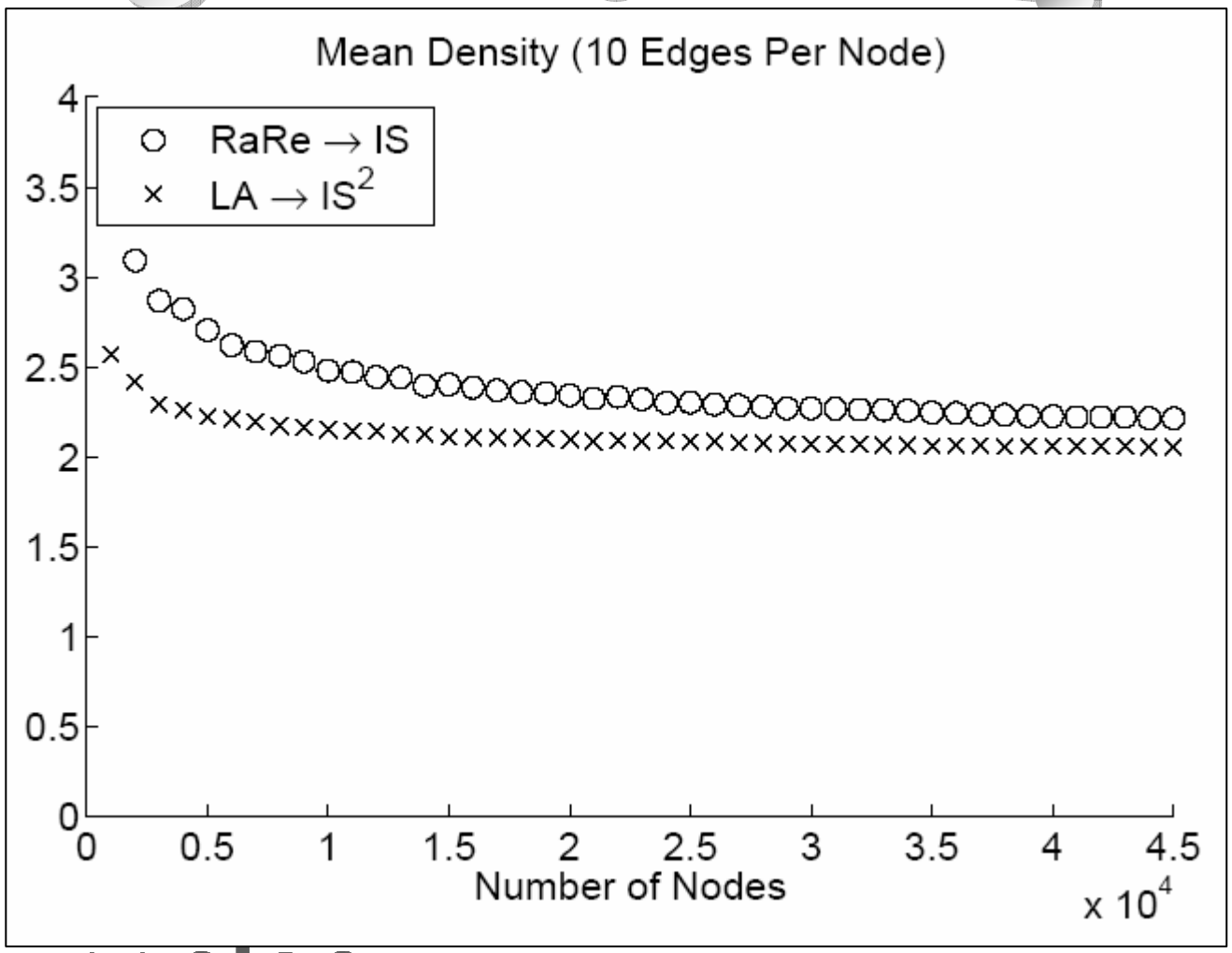
Rensselaer

Cluster quality



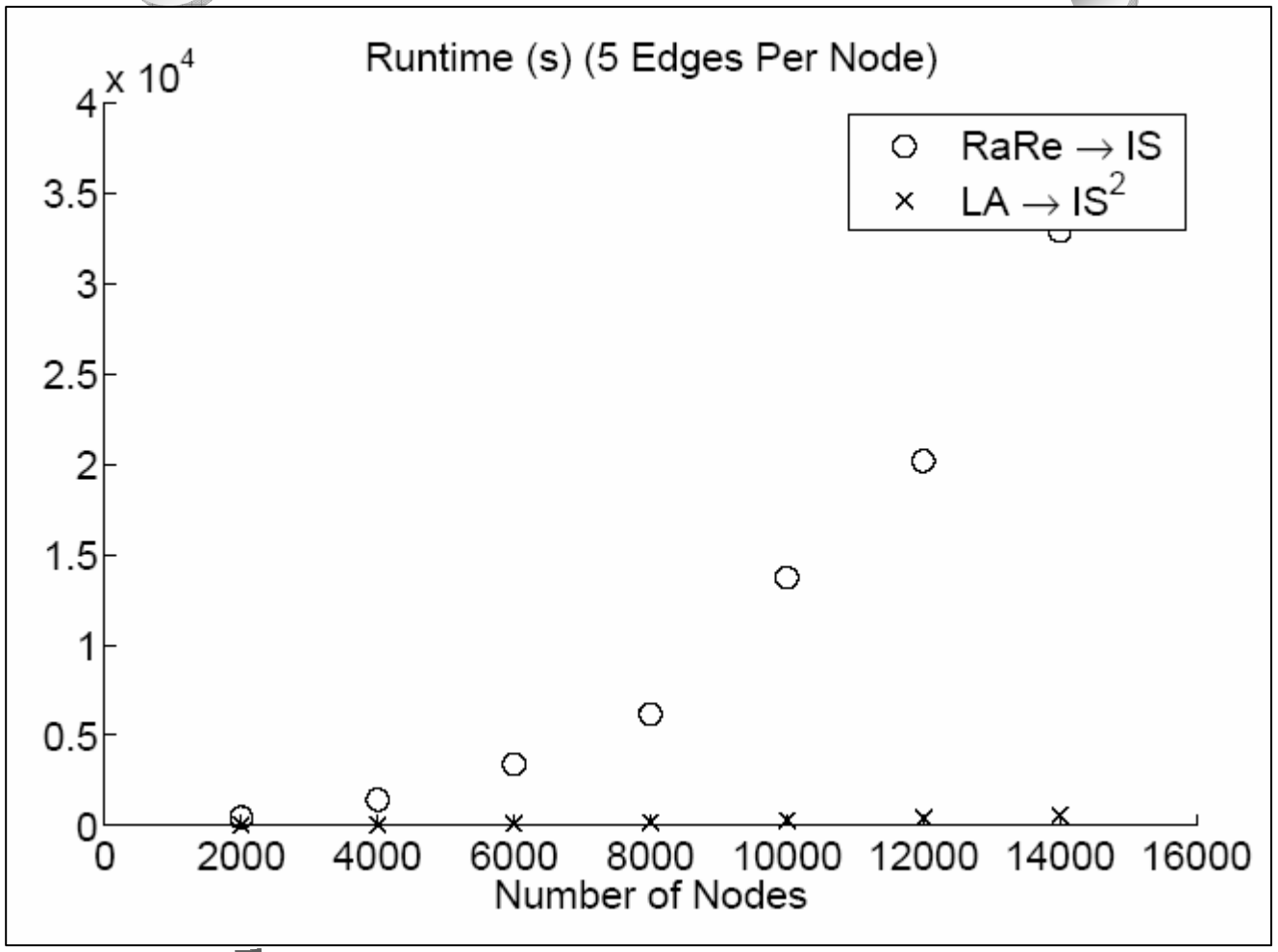
Rensselaer

Cluster quality



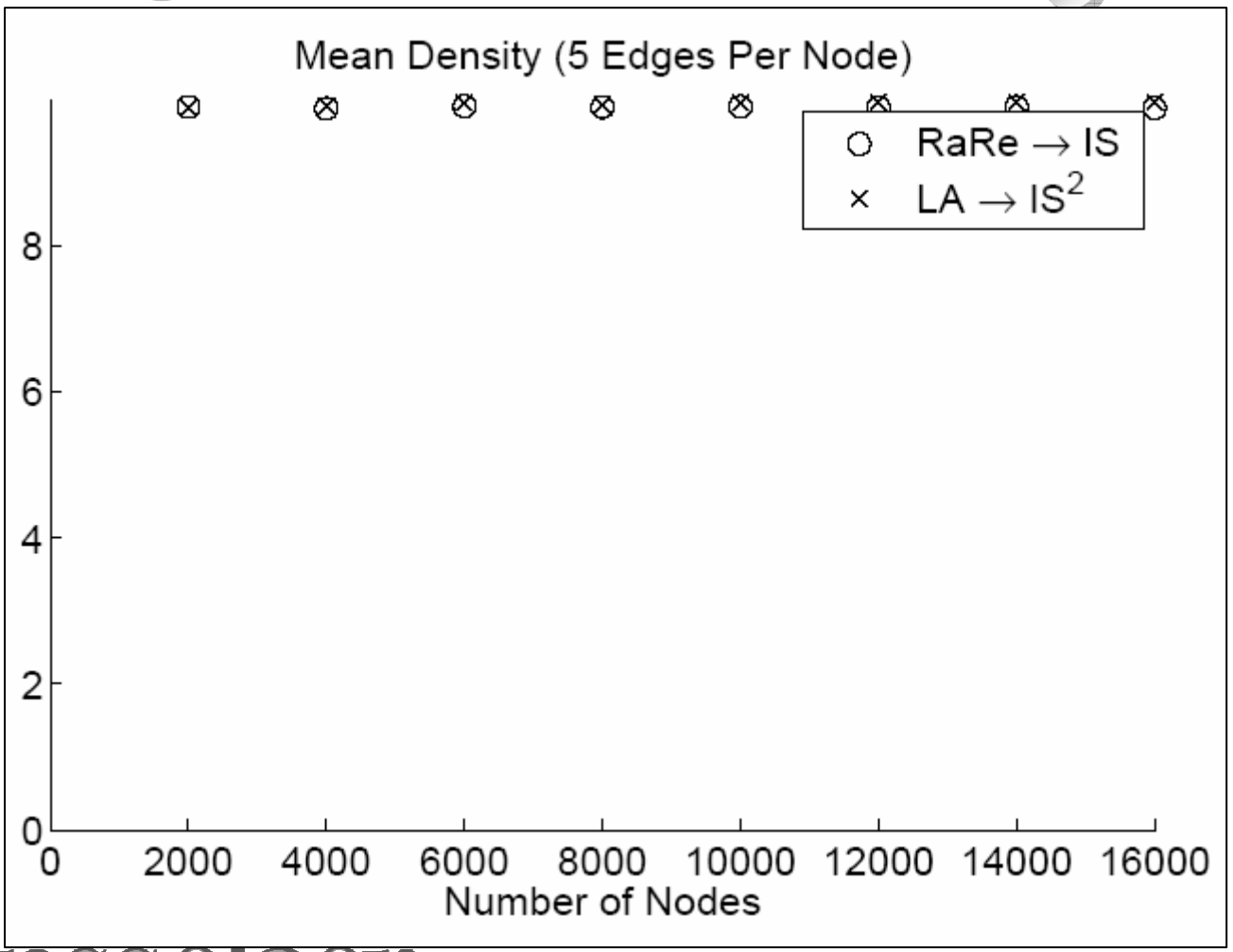
Rensselaer

Preferential attachment run time



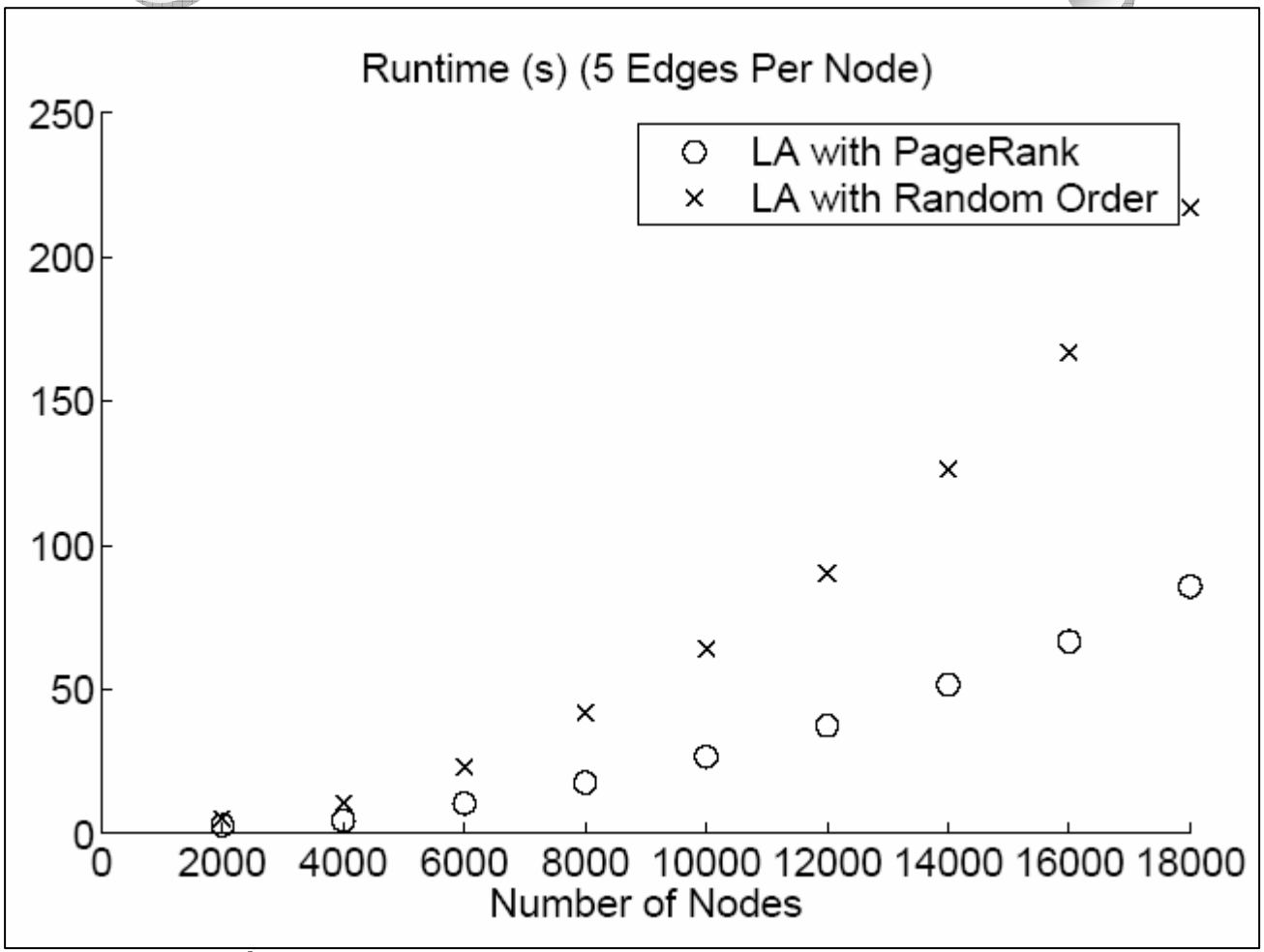
Rensselaer

Preferential attachment quality



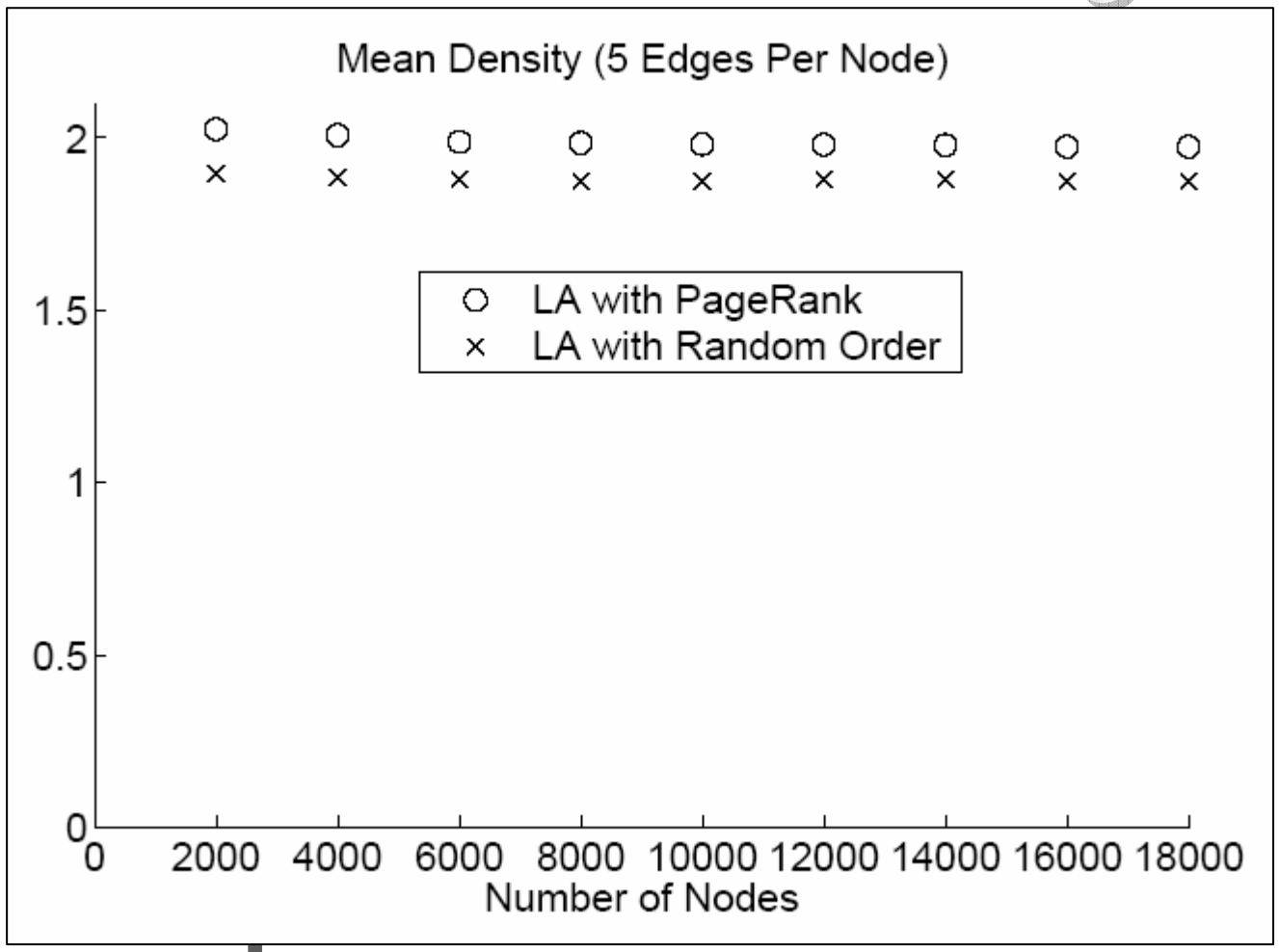
Rensselaer

LA ordering run time



Rensselaer

LA ordering quality



Rensselaer