SAND2009-5241P

High Performance Community Detection in Networks

Jonathan Berry (Sandia Labs)

Cynthia Phillips (Sandia Labs)

SIAM Annual Meeting

July 10, 2009

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Outline

- Community detection in networks
- HPC and community detection
- A parallel algorithm with accuracy and resolution tolerance
- Preliminary results

Community Detection in Networks

- **Break a network (graph with attributes) into modules**
  - Different from graph partitioning. E.g. O(n) communities, overlaps may be allowed
  - More than 400 recent papers in physics and computer science, **BUT**
  - **There's no canonical theory saying *what's best to do* (let alone the best way to do it)**

- **The state of the art: two diverging approaches**
  1. Modularity maximization and variants (Girvan, Newman 2004)
  2. Generative Models (Clauset, Moore, Newman 200x)

Sandia National Laboratories

Modularity

$Q = \sum_s e_{ss} - a_s^2$

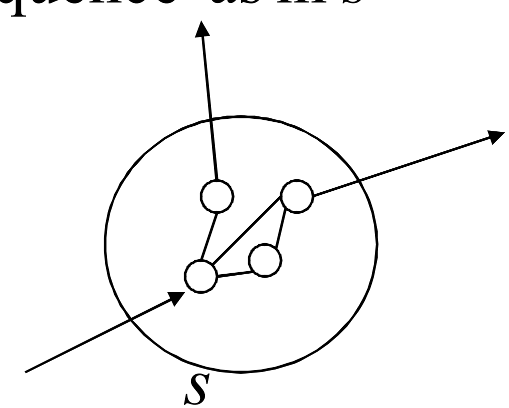
e_{ss} : (directed edges within s)/(all directed edges)

a_s : (directed edges originating in s)/(all directed edges)

in other words :

e_{ss} quantifies the actual edges within s

a_s^2 quantifies the expected number of edges in a set of vertices with the same degree sequence as in s (assuming random wiring)



Modularity Maximization

- **Dozens.. Hundreds(?).. of papers propose ways to maximize modularity or one of its variants**
- **Here are some things we (the community detection community) know about modularity maximization**
  - *It's NP-hard* (but greedy heuristics do ok)
  - *Succeeding doesn't necessarily mean that you've found something useful*
    - Many real-life networks have nodes that belong to many communities (e.g., Leskovec, et al. 2008)
    - The "resolution limit" says that a global maximum solution for large networks will have communities that have too many nodes (Fortunato, Barthelemy 2007)
  - But it's the most familiar way to evaluate the quality of a community assignment

Generative Models

- **What if being a "community" doesn't mean being tightly connected?**
  - Newman, Leicht (2007) example: classify English words by part of speech
  - Determine what a community "is" as the search for communities progresses
- **Clauset, Moore, Newman (Nature, 2008) give a "generative model" that**
  - Infers hierarchical community structure from data
  - Does this by searching a space of tree models of hierarchical community structure via a likelihood function
  - Benefits: accuracy
  - Challenges: scalability

HPC and Community Detection

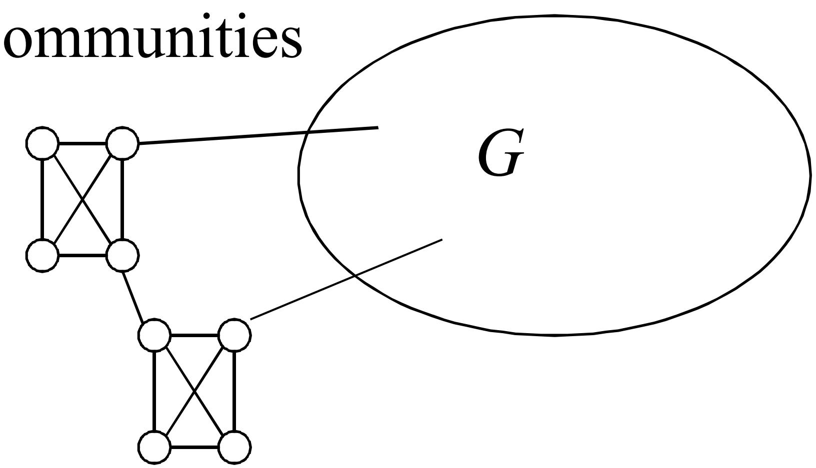
- **What algorithms could be made to scale on HPC?**
  - Subquadratic algorithms (e.g.) for modularity maximization
    - *ClausetNewmanMoore (CNM)* is a bottom-up approach that scales as O(m log^2 n), with priority queues as the kernel data struct.
    - *HQcut* (Ruan and Zhang, 2008) is a top-down approach that combines spectral methods with modularity-specific heuristics (also O(m log^2 n))
    - *Facility Location* via the "Volume algorithm" (Barahona, Anbil 2000) (empirical runtime similar to CNM)
  - For generative models: scalability work is in its infancy

Sandia National Laboratories

CNM

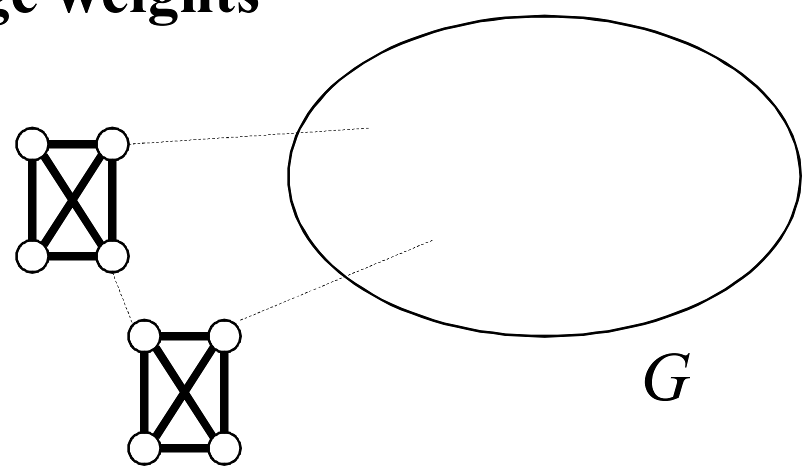
1. **Start with every vertex in its own community**
2. **Find the merger of two communities with best delta Q**
3. **Do the merge if this is a positive change, then goto 2 (else quit)**

- **Since this maximizes modularity, it suffers from a *resolution limit*:**
  - If G is large enough, these two cliques are not resolved as separate communities



Our Idea

1. **Observe that the mathematics behind the resolution limit argument permits a relaxation of the limit if the edges are *weighted* appropriately**
2. **Define such a weighting by finding *neighborhood coherence***
3. **Adapt CNM (with no change to its runtime complexity) to leverage weights**



Berry, Hendrickson, LaViolette, Phillips

"*Tolerating the Community Detection Resolution Limit with Edge Weighting*" (arXiv 2008)

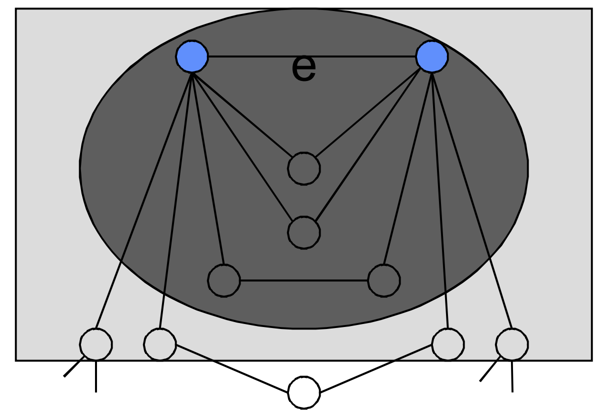
Sandia National Laboratories

Neighborhood Coherence

- **Assumption: communities are more tightly connected than non-communities**

- **We'd really like to compute the 2-neighborhood of each node (in parallel), but that's too expensive**

- **Solution: "edge 1-neighborhood" of an edge** ***(the light gray rectangle)***



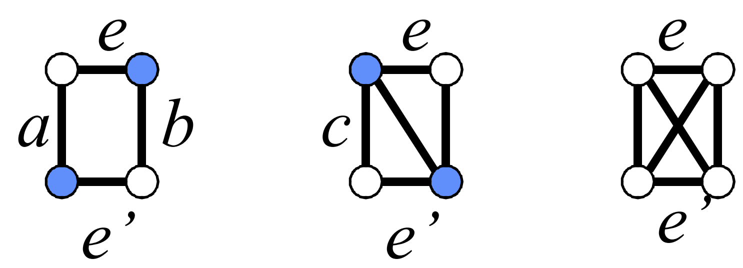
- **From that we distinguish "good" edges** ***(the dark gray oval)***

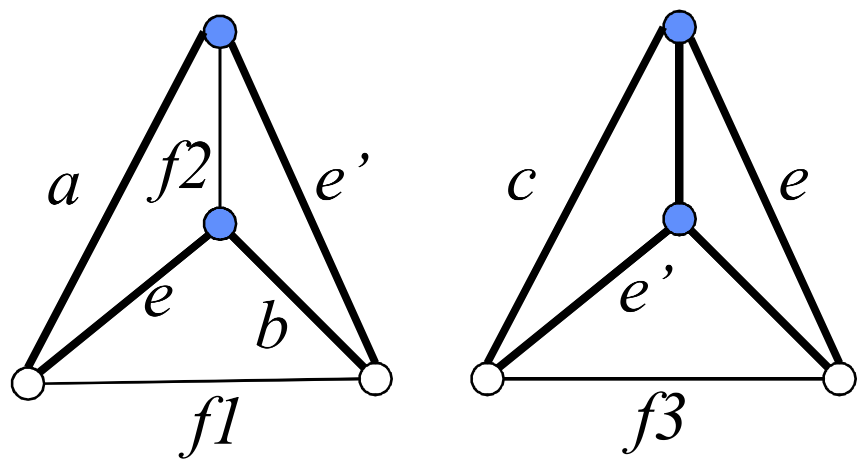
Parallel Algorithms

- **For this talk, we'll limit ourselves to this neighborhood computation**
  – *We can compute an approximation to the gray rectangle and oval for each edge in parallel*
  – We can use the weights in several ways
    - E.g. Helping CNM overcome resolution limit
    - E.g. assigning opening costs for facility location approach
- **The computation reduces to:**
  – Finding triangles (3-cyles)
  – Finding rectangles (4-cycles)
  – We'll describe a parallel algorithm for this and give some results for the Cray XMT

Some Terminology

- **Each of these concepts is defined with respect to an edge (*e*)**
  - e’ is an *opposing rectangle edge*
  - *a*,*b*, and *c* are *spokes*
  - The unlabeled edges are *tent poles*
  - Any missing chord in a rectangle is a *fake edge*

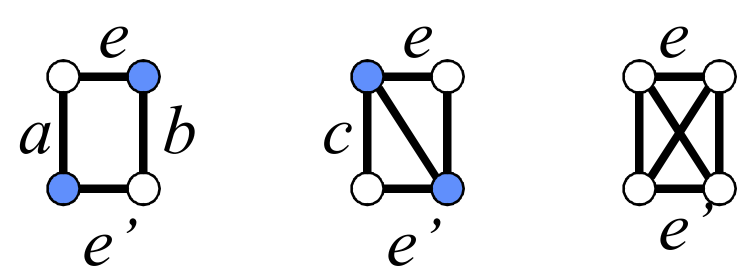


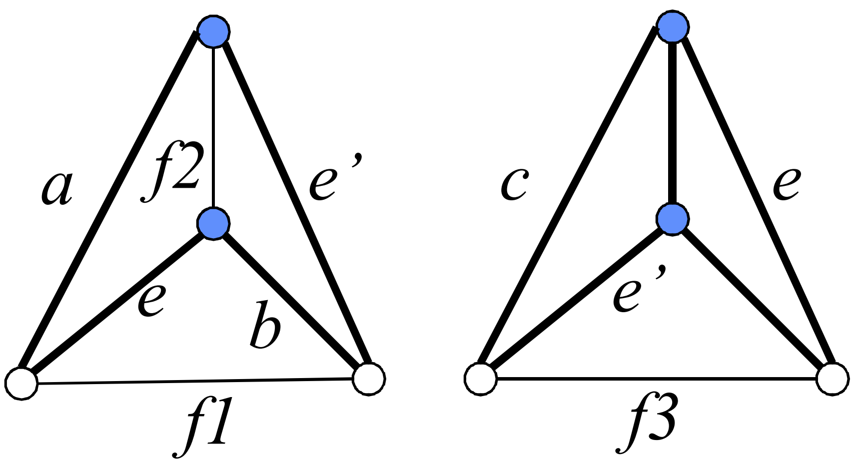


Shown: tent poles with respect to f1 and f3

The Algorithm in Abstract

- By simply iterating over all edges (real and fake) in parallel, we can give each edge *e* credit for its spokes and opposing rectangle edges
- This counting is correct if there are one or two chords, else we give ½ credit (details omitted)
- By iterating over triangles*, we account for tent poles
- We apply a degree threshold to limit the number fake edges
- We expect relatively small numbers of triangles for many real-world graphs
- This works with weighted edges too

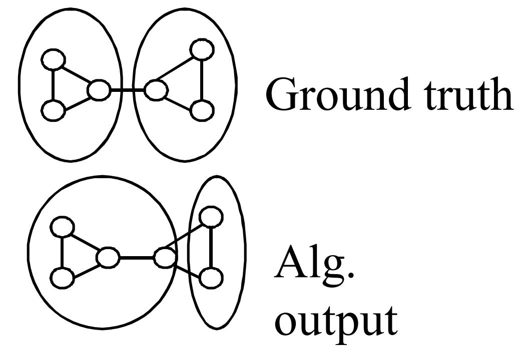


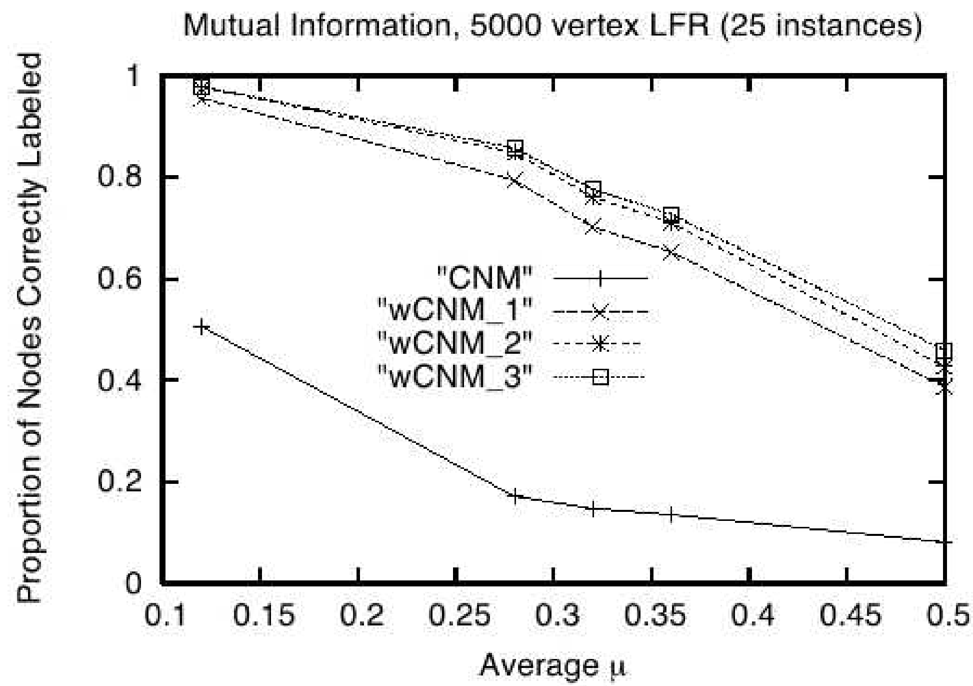


*e.g. triangle algorithm from *J. Cohen, "Graph Twiddling in a MapReduce World," CiSE, Vol. 11, No. 4*

Community Detection Results: Mutual Information

- **Given a ground truth assignment and an algorithmic assignment, the “mutual information” is a measure of how well we matched**
  - In the example at left,5 of 6 vertices in the alg. output are “good”
- **LFR is a benchmark for generating graphs with ground truth communities of differing sizes (arXiv search “Fortunato Benchmark”)**
- **wCNM_k is CNM weighed with our neighborhood coherence scores (k iter.)**
- **Summary: major improvement in accuracy**



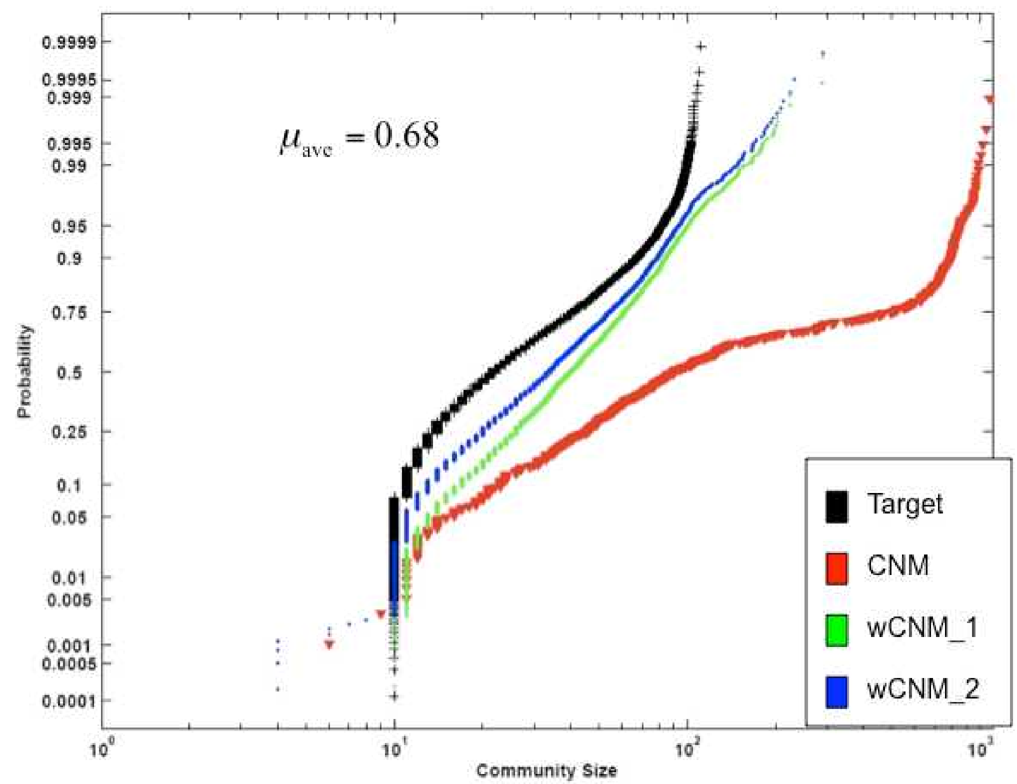


μ is a community coherence measure

Sandia National Laboratories

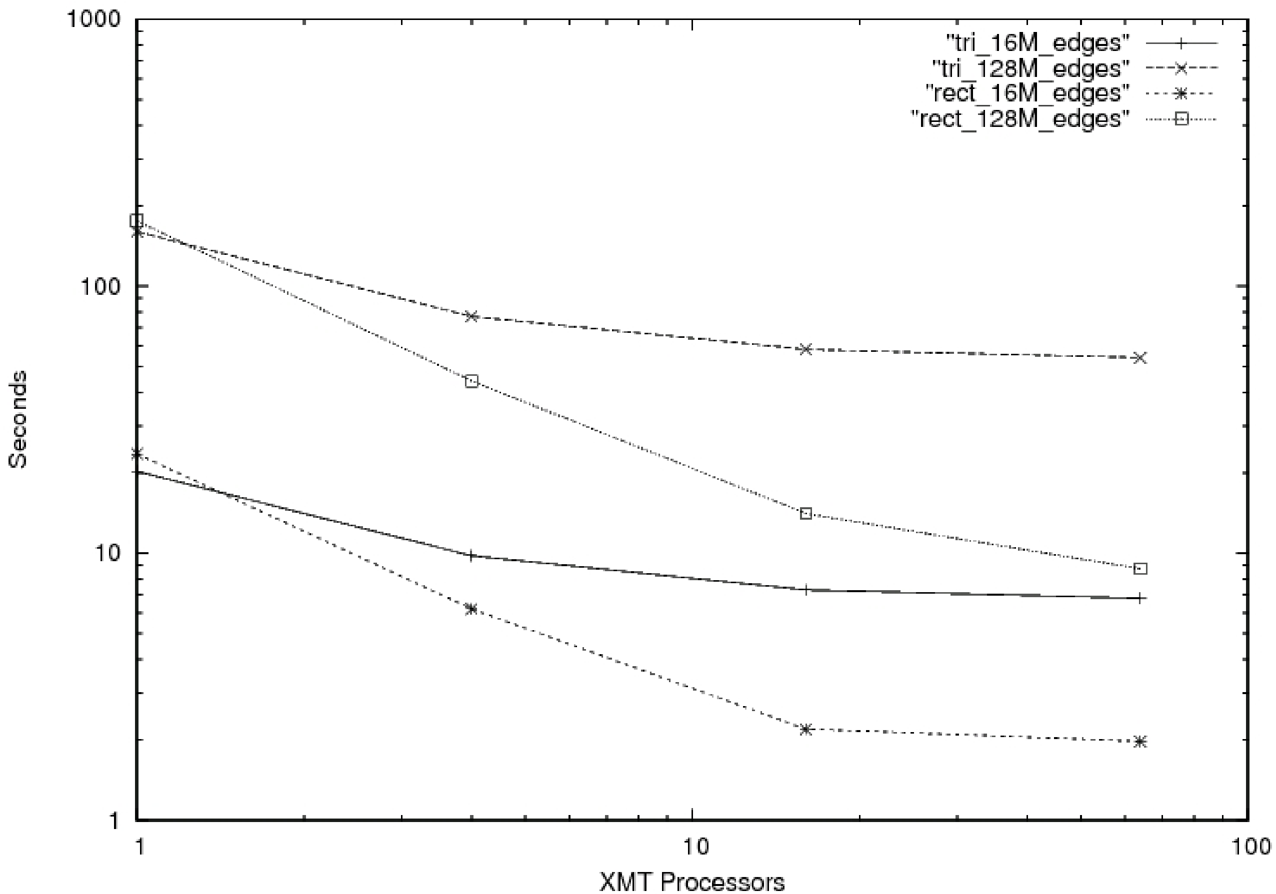
Community Detection Results: Resolution Limits

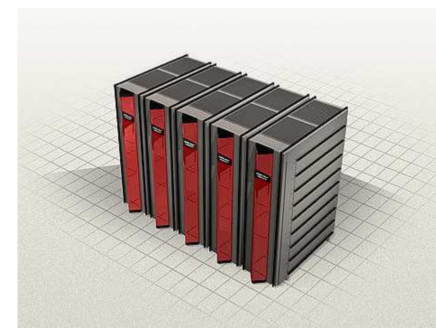
- We used LFR to generate ground truth communities with exponentially distributed sizes between 10 and 100 vertices
- The figure below shows empirical cumulative distribution functions over community sizes
- 95% of the communities found by wCNM are appropriately sized
- Less the 50% of the communities found by CNM are appropriately sized



Preliminary Computational Results

- On "R-MAT" graphs with inverse power law degree distribution
- A couple of minutes to compute neighborhood coherence in graphs with O(100M) edges on the Cray XMT (massively multithreaded supercomputer)





Sandia National Laboratories

Conclusions

- Our weighting improves CNM, which has become notorious for poor accuracy despite its speed

- This better tolerates the resolution limit, and matches the accuracy of more complicated methods

- Our initial parallelization results of neighborhood coherence suggest that large instances will be tractable

- CNM itself can be parallelized, but we haven't done that
- Differing approaches may parallelize better and still benefit from our neighborhood coherence computations

Sandia National Laboratories