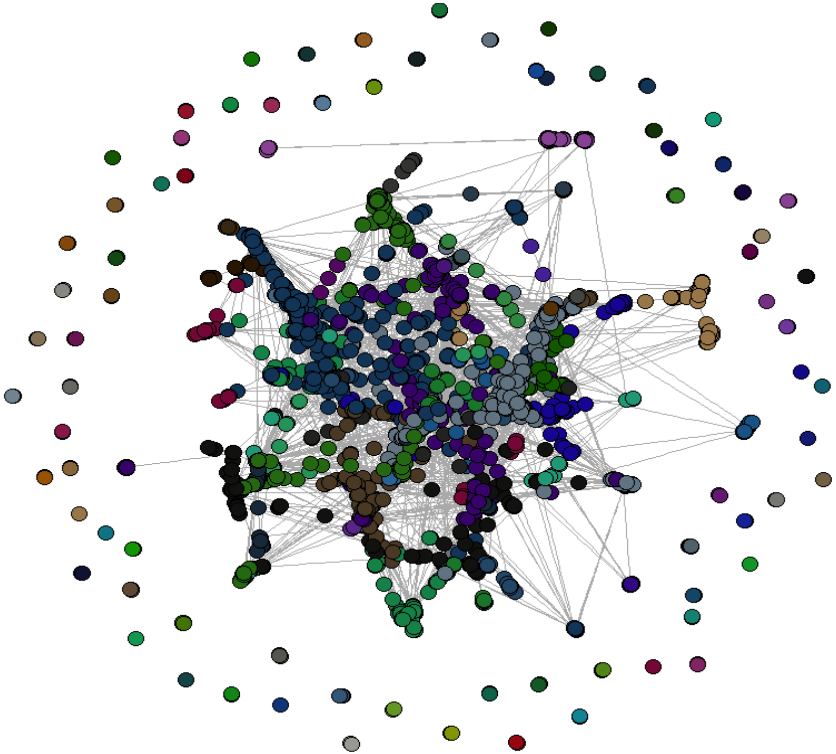
# Graph Clustering Algorithms

Alexandr Salo, Rovshen Nazarov

# Background



Undirected, unweighted, disconnected graph with:

- 2526 Vertices
- 11450 Edges
- 0.00179 Densite

Graph build after clustering with Greedy Modularity Optimization Algorithm

Newman, M. E. J. “Modularity and Community Structure in Networks.” *Proceedings of the National Academy of Sciences* 103, no. 23 (June 6, 2006): 8577–82. doi:10.1073/pnas.0601602103.

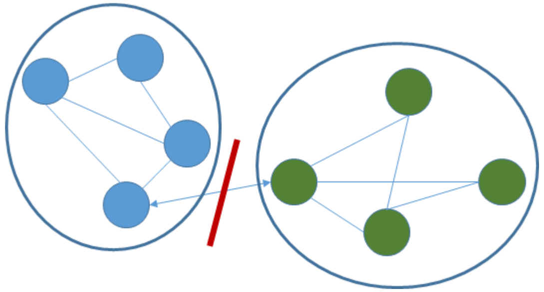
# Existing Clustering Algorithms

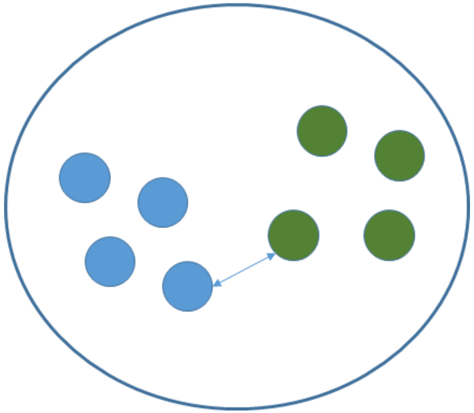
### Common Neighbors Cut

Iteratively eliminate the edge between the most dissimilar vertices until:

-the graph is separated

-all subgraphs reach a density threshold



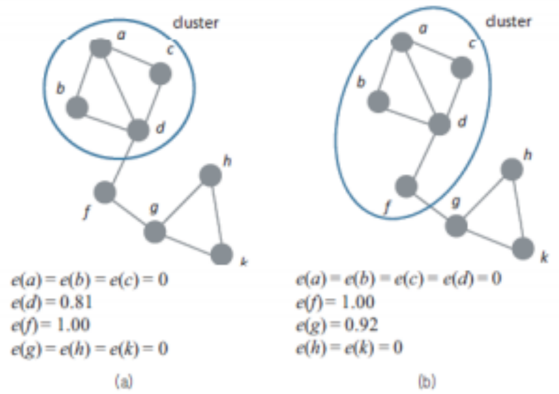


### Common Neighbors Merge

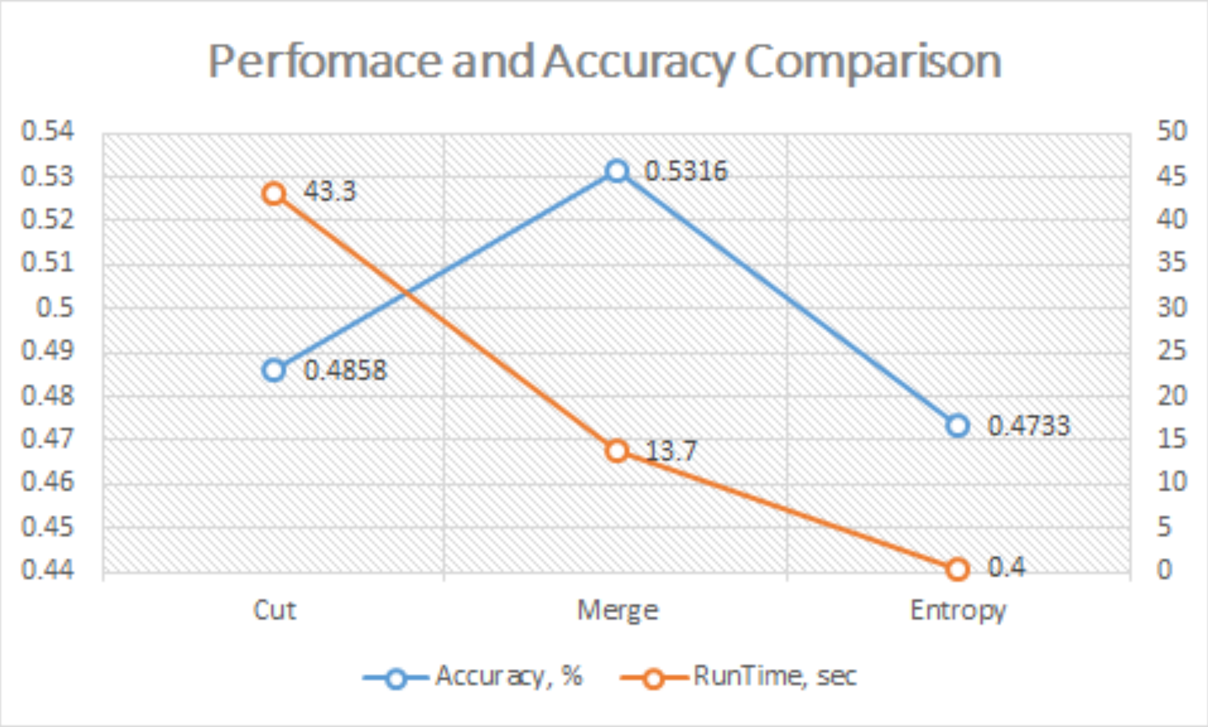
Merge most clusters with most similar vertices until violate density threshold

### Seed Growth: Graph Entropy

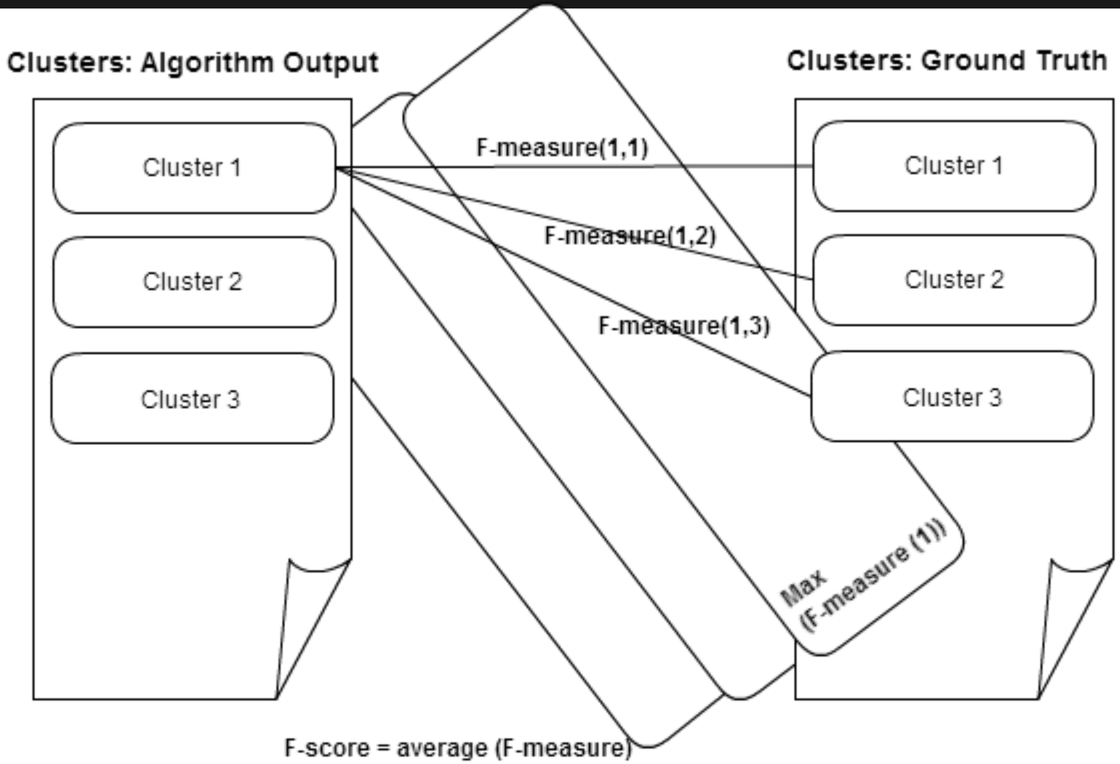
Grow cluster around high-deg seed using entropy property



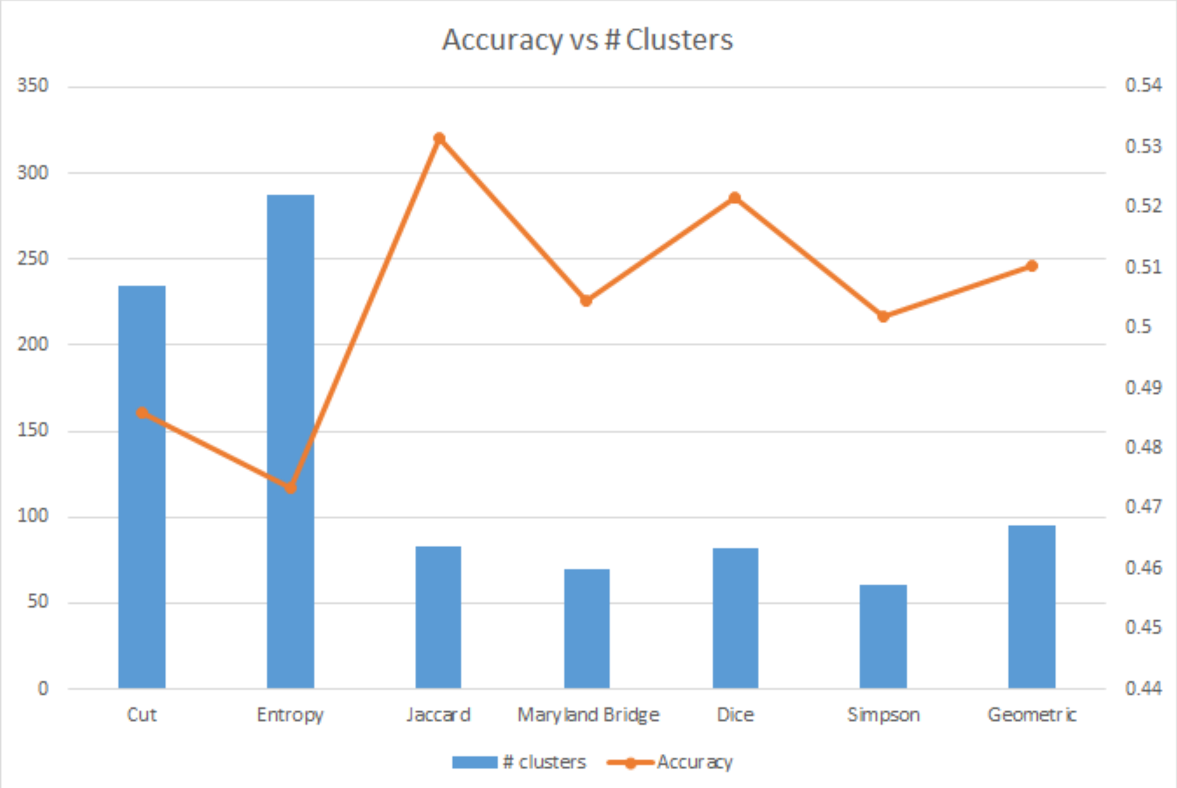
# Performance comparison



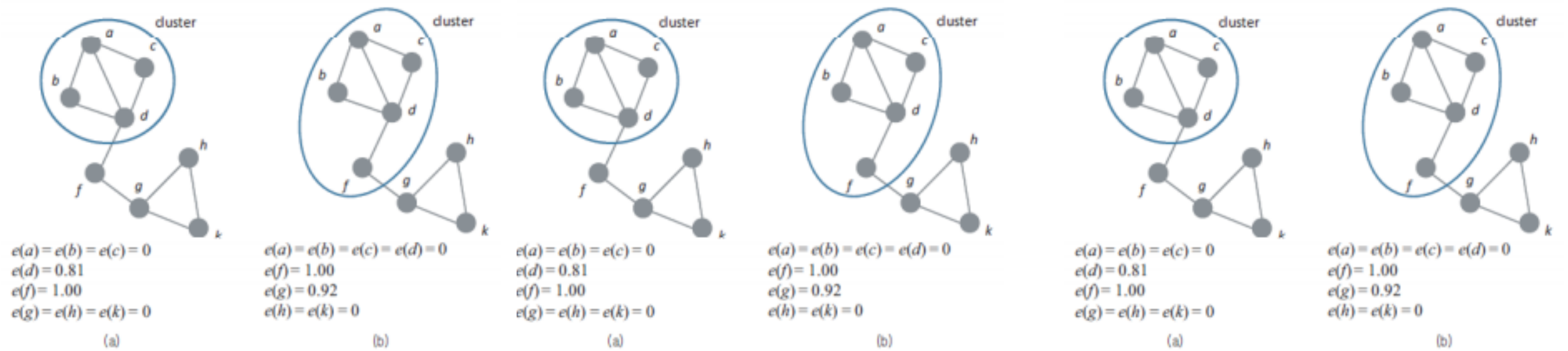
# Accuracy Measure



# Accuracy vs # Clusters



# Proposed Novel Algorithm



# Proposed Novel Algorithm

Input: An undirected graph $G = (V, E)$

Output: Graph Clusters (Overlapping)

# Proposed Novel Algorithm

- Density based
- Seed growth
- Entropy
- Localization

# Proposed Novel Algorithm

## Entropy computation optimization

for the vertex entropy

$$e(v) = -p_i(v)log_2p_i(v) - p_0(v)log_2p_0(v)$$

and a formula for the graph entropy

$$e(G(V,E)) = \sum_{v \in V} e(v)$$

# Novel Algorithm Preprocessing

Set of vertex degrees

Graph represented as adjacency list

Efficient way to get vertex adjacency list

# Novel Algorithm Difficulties

Possible duplicate clusters

# Novel Algorithm Difficulties

Shared state for N seeds

N parallelization factor

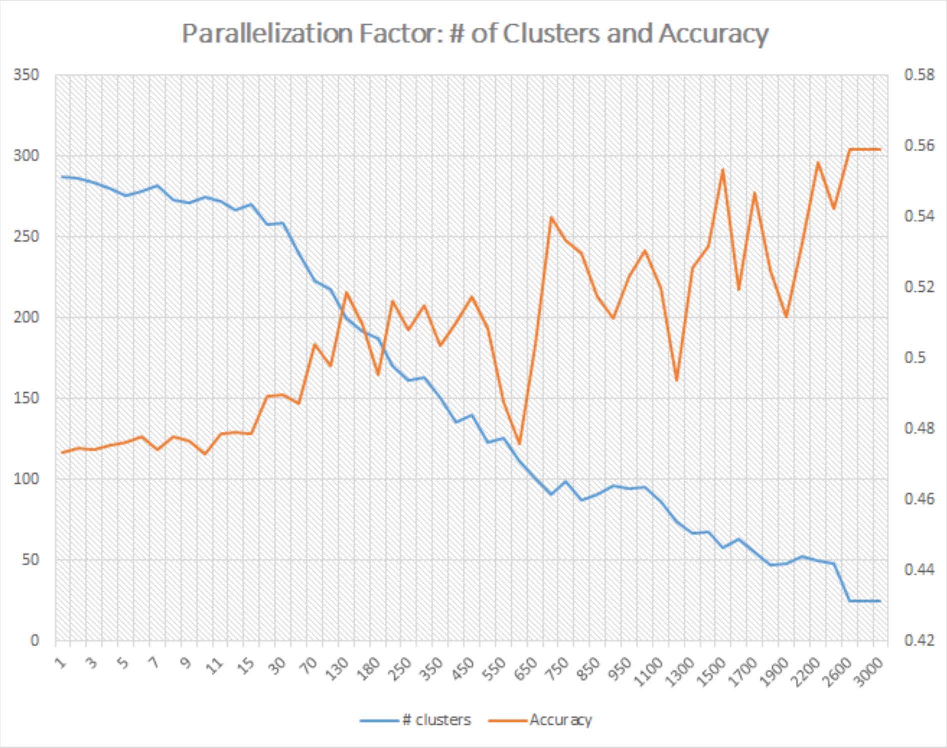
Update of the shared state

# Novel Algorithm Pseudo Code

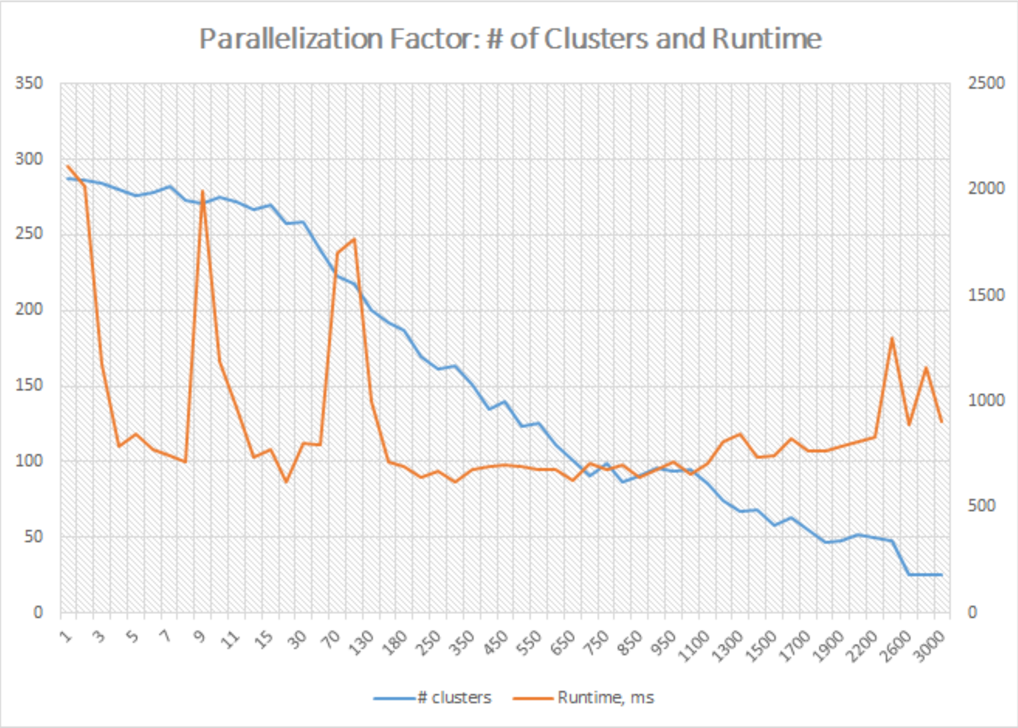
Create Seed Candidates set, which includes all unique vertices.

1. Select N seeds based on the highest degree first and store them as a set Seeds.
2. For each selected seed do in parallel:
    a. Add current seed’s immediate neighbor if it is NOT in the Seeds set.
        i. If the neighbor is in the Seeds set, stop the growth of that seed and remove it from the Seeds set.
    b. Compute seeds inner and outer links.
    c. Compute entropy for adding each outer link
        i. If entropy for the current cluster decreases
            1. Check if the cluster member candidate is in the Seeds set
            2. If in the Seeds set perform the same steps as for 2.a.i above
            3. If not in the set add the new member and proceed with the seed growth.
    d. Repeat b, c for N seeds in parallel until the N seed growth complete
3. Remove members of the computed N clusters from the Seed Candidates set.
4. Repeat 1 - 3 until Seed Candidates set is empty.

# Novel Algorithm Performance



# Novel Algorithm Performance



# Comparison to the original algorithm

| Algorithm Name | F-score | Runtime, milliseconds | Number of clusters, size > 2 |
|---|---|---|---|
| Seed Growth | 0.47338 | 753 | 272 |
| Parallel Seed Growth, N- 130 | 0.50634 | 551 | 202 |

# Comparison with Other Algorithms

Flexibility

- Adopt to for sparse or dense graph’s by modifying N

# Comparison with Other Algorithms

Optimization

- Make trade-offs speed vs accuracy by tweaking N

# Comparison with Other Algorithms

Extensibility

- Could be used on large scale parallel processing systems

- Benefit from localization and synchronization

  (shared state, localized growth)

# Benefits of the Novel Algorithm

- Parallelized processing of large volume graphs

# Drawbacks of the Novel Algorithm

- Single cluster graph
- Clustering quality depends on entropy computation
- Cluster collisions for large N

# Conclusion

- Novel contribution, introduction of N
- Localization of the seed growth
  - Seed selection
  - Entropy computation

# Future Work

Different way to select seed to avoid looking at degrees of all vertices in the graph.

Different method to compute graph density

Extend proposed algorithm for cluster computing