IMPROVED LCD-SN Algorithm

# 1. Node Importance Calculation

## Old LCD-SN

Formula:

Where:

•Importance score of node i

• Set of first-degree neighbors of node i

• Set of second-degree neighbors of node i (excluding N\_1(i))

•: Importance of neighbor j

• |N(j)|: Degree of neighbor j

• α, β: Weights for first and second-degree neighbors

## Improved LCD-SN

Formula:

Where:

• N(i): Set of all neighbors of node i

# 2. Similarity Calculation

Formula:

Where:

• N(i): Set of neighbors of node i

• C: Community

• |N(i) ∩ C|: Number of common neighbors between node i and community C

• |N(i)|: Total number of neighbors of node i

# 3. Community Merging

Weak Community Merging Condition:

Where:

•: Number of edges within community C

•: Number of edges between community C and the rest of the graph

• V: Set of all nodes in the graph

• I(u,v): Indicator function that is 1 if there is an edge between nodes u and v, and 0 otherwise

• mc: Merging criterion threshold

# 4. Modularity Calculation

Formula:

Where:

• Q: Modularity

•: Adjacency matrix element (1 if there is an edge between nodes i and j, otherwise 0)

•: Degrees of nodes i and j

• m: Total number of edges

• δ(c\_i, c\_j): Delta function that is 1 if nodes i and j are in the same community, and 0 otherwise

# 5. Validation

Formula:

Where:

• V: Set of all nodes in the graph

• : Union of all communities

ALGORITHM FOR IMPROVED LCD-SN

**Algorithm LCD-SN(G, α, β, γ, mc)**

Input:

G: Social Network Graph

α: Influence of first-degree neighbors

β: Influence of second-degree neighbors

γ: Maximum number of iterations for calculating node importance

mc: Merging criterion threshold

Output: Community structure of G

// Phase 1: Formation of Initial Communities

Initialize importance for each node to 1.0

Repeat γ times:

For each node in G:

neighbors ← Get neighbors of node

imp\_score ← 0

For each neighbor in neighbors:

neighbor\_imp ← importance[neighbor]

neighbor\_degree ← Degree of neighbor

imp\_score ← imp\_score + α \* (neighbor\_imp / neighbor\_degree)

Update importance[node] to imp\_score

Initialize empty list of communities

Initialize empty set of visited nodes

For each node in G sorted by importance (descending):

If node not in visited nodes:

Create new community with node and its neighbors

Add new community to list of communities

Add all nodes in new community to visited nodes

// Phase 2: Determining Status of Overlapping Nodes

For each overlapping node v in G:

Initialize max\_similarity to 0

Initialize best\_community to null

For each community C that contains v:

Calculate similarity between v and C

If similarity > max\_similarity:

max\_similarity ← similarity

best\_community ← C

Assign v to best\_community

// Phase 3: Integration of Communities

// Merge small communities with large ones

Initialize list L of small communities (size < 3)

While L is not empty:

Select a community C from L

Find the most similar neighboring community to C

Merge C into the most similar neighboring community

Remove C from L

// Merge weak communities with strong ones

Initialize final\_communities as empty list

For each community C\_i in list of communities:

Calculate E\_in and E\_out for C\_i

If E\_in ≤ mc \* E\_out:

Find the most similar neighboring community to C\_i

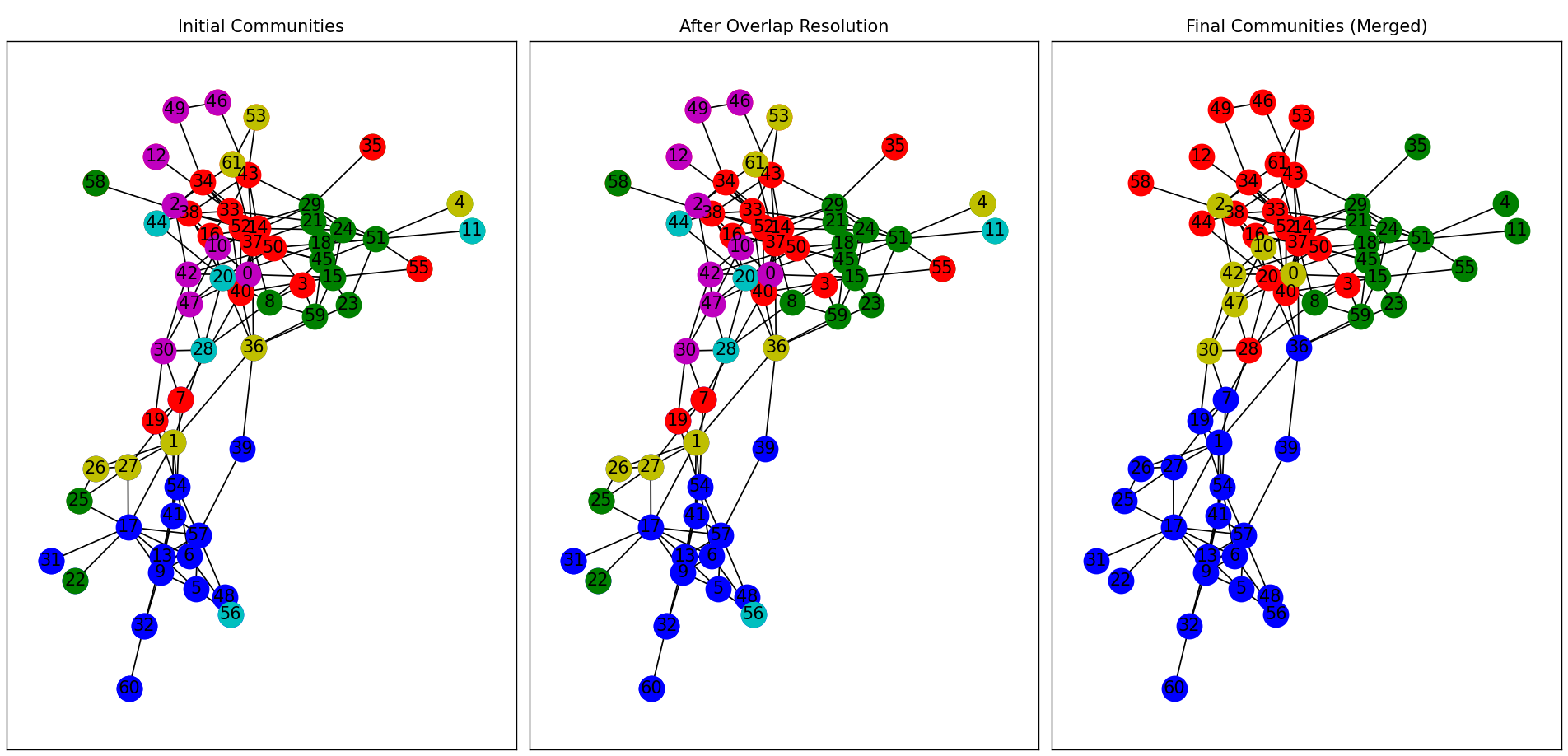
Merge nodes of C\_i into the most similar neighboring community

Else:

Add C\_i to final\_communities

Output final\_communities

Phases of the Community Formation Graph



Comparison of modularity with old LCS-SN AND IMPROVED LCD-SN

|  |  |  |
| --- | --- | --- |
| **NETWORKS** | **OLD LCD-SN** | **IMPROVED LCD-SN** |
| KARATE | 0.36146614069691 | 0.37146614069691 |
| DOLPHINS | 0.448577983465843 | 0.4979826747359677 |
| FOOTBALL | 0.01687446136167773 | 0.013576558460212527 |
| POLBOOKS | 0.3747743995557406 | 0.4816357381955049 |
| NETSCIENCE | 0.8162313222889432 | 0.8794230695666407 |
|  |  |  |