**On Improving the Cohesiveness of Graphs by Merging Nodes: Formulation, Analysis, and Algorithms**

Fanchen Bu and Kijung Shin

**Goal**: To study the problem of improving the cohesiveness of real-world systems using graph analysis and graph algorithms

**Formulation**: How can we mathematically formulate such a problem?

**Analysis**: To analyze the problem from both theoretical and empirical aspects

**Experiments**: To show the empirical performance of the proposed algorithm based on our analysis

**Fast**: The proposed algorithm time-efficiently finds good node pairs by pruning unpromising node pairs and identifying a small number of promising candidate node pairs

**Effective**: Merging the node pairs selected by the proposed algorithm effectively improves the optimization objective, and also effectively improves the cohesiveness of graphs

**Many real-world systems can be abstracted and represented as graphs**

**Cohesiveness**: We love cohesive systems in the real world

---

**Summary**

- **Goal**: To study the problem of improving the cohesiveness of real-world systems using graph analysis and graph algorithms
- **Formulation**: How can we mathematically formulate such a problem?
- **Analysis**: To analyze the problem from both theoretical and empirical aspects
- **Experiments**: To show the empirical performance of the proposed algorithm based on our analysis
- **Fast**: The proposed algorithm time-efficiently finds good node pairs by pruning unpromising node pairs and identifying a small number of promising candidate node pairs
- **Effective**: Merging the node pairs selected by the proposed algorithm effectively improves the optimization objective, and also effectively improves the cohesiveness of graphs

**Motivation**

- **Many real-world systems can be abstracted and represented as graphs**
- **Cohesiveness**: We love cohesive systems in the real world

---

**Problem Formulation**

- **Merging nodes**: An overlooked operation, which is realistic and interesting
- **Metric**: We use the size of a k-truss as the cohesiveness metric
- **Definition**: Given a graph G and k, the k-truss of G is the maximal subgraph such that each edge in the k-truss is at least k-2 triangles
- **Node engagements (neighbors) and edge interrelatedness (triangles)**
- **Final problem statement**: Given: graph G = (V, E), k ∈ N, and budget b ∈ N
  - Find: b pairs of nodes to be merged
  - To maximize: # edges in the k-truss after merging those b node pairs

**Hardness Analysis & A Naive Algorithm**

- **Theorems**: The considered problem is NP-hard and not submodular
- **A naive algorithm**: Until the budget is exhausted, we repeatedly compute # edges in the k-truss after all possible mergers and operate the best one
- **But it takes prohibitive time complexity**: \(O(b|E|^3)\)

---

**Experimental Results**

- **Extensive experiments on fourteen real-world graphs show the superiority of the proposed algorithm (BATMAN) w.r.t both speed and effectiveness**
- Since we are the first to study this problem, we use several other heuristic-based methods as our baselines
  - **NT**: To form most new triangles consisting of the nodes in the (k−1)-truss
  - **NE**: To form most new edges between the nodes in the (k−1)-truss
  - **RD**: Randomly sampling node pairs

---

**Proposed Method**: BATMAN

- **Best-merger seArcher for Truss MAximizatioN**

---

**Table**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>0</th>
<th>1.3</th>
<th>2.13</th>
<th>3.05</th>
<th>Avg. truss-size increase over all k values (×10^3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YT</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>4.91×</td>
</tr>
<tr>
<td>WT</td>
<td>0</td>
<td>25</td>
<td>50</td>
<td>0</td>
<td>10.08×</td>
</tr>
<tr>
<td>SF</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2.73×</td>
</tr>
<tr>
<td>BK</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1.42×</td>
</tr>
<tr>
<td>HP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1.60×</td>
</tr>
<tr>
<td>RL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1.38×</td>
</tr>
<tr>
<td>ER</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1.54×</td>
</tr>
</tbody>
</table>

---

**Figure**

The proposed method shows consistent superiority on different datasets