Enhancing EquiTree Index Via the Union-Find Optimized Node Merging

Michael Abiona, Bin Guo

Department of Computing & Information Systems, Trent University, Ontario, Canada
mabiona@trentu.ca; binguo@trentu.ca

Abstract

Community detection in large-scale graphs enables the discovery of cohesive structures such as k-truss and triangle-connected k-truss communities (k-TTC). Among the most efficient solutions, EquiTree has emerged as a compact index for organizing truss equivalence classes, significantly improving query performance in dynamic graph scenarios. However, EquiTree's construction is computationally expensive due to its *Nodemerge procedure*, which repeatedly merges overlapping truss partial classes to maintain structural consistency. This step becomes the dominant bottleneck in large or dense networks. In this work, we introduce a **Union-Find based optimization** of the Nodemerge procedure. By representing partial classes as disjoint-set forests, our optimized method preserves the correctness and space complexity of EquiTree while substantially reducing the time complexity of node merging.

1 Introduction

The analysis of large and dynamic graphs is central to many real-world applications, including social media analysis, biological network mining, and recommendation systems. Community detection, particularly through triangle-based measures such as k-truss and triangle-connected k-truss communities (k-TTC), provides an effective means of uncovering cohesive and meaningful structures. However, as graphs continue to grow in size, the scalability of community search algorithms becomes a significant challenge.

Recent work has proposed several indexing frameworks to accelerate k-TTC queries. TCP-Index [2] precomputes triangle connectivity, EquiTruss [1] improves space efficiency, and EquiTree [3] achieves high query performance by maintaining a hierarchical organization of truss equivalence classes. Despite these advances, EquiTree's construction process remains costly, primarily due to its iterative *Nodemerge procedure*, which performs numerous redundant structural updates. This inefficiency limits EquiTree's scalability on dense or streaming graphs. In this work, we propose a Union-Find based optimization of the *Nodemerge procedure* to enhance scalability and reduce construction costs. By representing partial classes as disjoint-set forests, our goal is to leverage path compression and union-by-rank heuristics to perform merging operations in nearly constant amortized time, thereby eliminating redundant structural adjustments inherent in the original method.

2 Contributions

This work presents a Union-Find based optimization of the EquiTree construction algorithm. The key contributions are:

- We identify and analyze the *Nodemerge procedure* as a computational bottleneck in EquiTree construction and propose a Union-Find based optimization to handle equivalence class merging in nearly constant amortized time.
- We will implement the optimized EquiTree and conduct extensive experiments on real-world graph datasets, demonstrating significant improvements in construction time and memory usage, with competitive query performance.
- We will adapt the Union-Find structure for incremental updates to handle edge insertions and deletions efficiently. This allows maintaining the EquiTree index dynamically rather than reconstructing it from scratch.

References

[1] Esra Akbas and Peixiang Zhao. Truss-based community search: a truss-equivalence based indexing approach. *Proceedings of the VLDB Endowment*, 10(11):1298–1309, 2017.

[2] Xin Huang, Hong Cheng, Lu Qin, Wentao Tian, and Jeffrey Xu Yu. Querying k-truss community in large and dynamic graphs. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1311–1322, 2014.

[3] Tianyang Xu, Zhao Lu, and Yuanyuan Zhu. Efficient triangle-connected truss community search in dynamic graphs. *Proceedings of the VLDB Endowment*, 16(3):519–531, 2022.