Scalable Algorithms for Analyzing Large Dynamic Networks using CANDY

ABSTRACT

As the dynamic network's topology undergoes temporal alterations, associated graph properties must be updated to ensure their accuracy. Addressing this requirement efficiently in large dynamic networks led to the proposal of a generic framework, CANDY (Cyberinfrastructure for Accelerating Innovation in Network Dynamics). This paper expounds on the development of algorithms and subsequent performance improvements facilitated by CANDY.

CCS CONCEPTS

• **Computing methodologies** → **Parallel algorithms**.

KEYWORDS

Large Dynamic Network, Parallel Graph Property update, Scalable Algorithms

ACM Reference Format:

. 2018. Scalable Algorithms for Analyzing Large Dynamic Networks using CANDY. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Complex systems comprising interacting entities are frequently represented as networks, where nodes represent the entities, and edges represent the interactions among them. Analyzing such networks by measuring different graph properties provides insights into the underlying system. However, in a dynamic network, where the network topology changes with time, analysis of the system requires continuous update of the graph properties. As the real-world networks are mostly dynamic and large, analysis of them involves additional challenges. To analyze and manage large dynamic networks, we previously proposed a software platform CANDY (Cyberinfrastructure for Accelerating Innovation in Network Dynamics). Here, we present the parallel algorithms developed using CANDY to deal with large dynamic networks.

1.1 CANDY Framework

Here we introduce CANDY, a framework designed for dynamic network computations, with the following key objectives:

- Establish a novel hierarchical taxonomy of dynamic network analysis algorithms
- Incorporate templates for developing new scalable, parallel algorithms dedicated to dynamic network analysis.
- Provide a way to generate and store synthetic graphs which preserve the real dynamic network properties.

CANDY introduces a template for creating parallel algorithms that efficiently update a graph property P in a dynamic network. The template takes the set of changed edges ΔE, which includes both deleted and inserted edges, as batch input. By utilizing the graph property from the previous time instance, the template computes the updated property without the need for a full recomputation from scratch. This approach enables significant time and resource savings when updating graph properties in a dynamic network. The template follows mainly two steps: **step 1:** Graph sparsification and identification of affected vertices by parallel processing of each changed edge in ΔE; **step 2:** Property update of affected vertices and maintaining correctness. Step 2 is an iterative process that uses parallel threads to operate on different affected vertices.

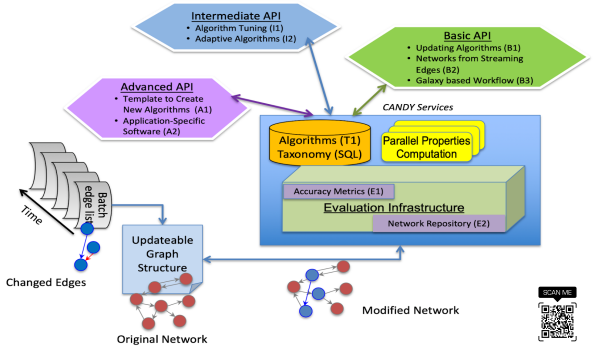

Figure 1: Overview of CANDY

2 DEVELOPED ALGORITHMS

Various algorithms developed using CANDY are discussed here.

2.1 Single Source Shortest Path (SSSP) update

The single source shortest path update algorithm [2] that computes the shortest route from a source vertex to all other vertices within a dynamic network is designed using the CANDY-suggested template. It efficiently detects any alterations in the shortest distances caused by changes in the network's structure and identifies the vertices affected by these changes. Subsequently, the algorithm iteratively updates these affected vertices by examining their neighboring vertices and determining the new shortest distances through them.

2.2 Multi-objective Shortest Path (MOSP)

When dealing with a dynamic network where each edge has multiple weights associated with different objective functions, the task of updating the shortest path that minimizes multiple objectives can be seen as a multi-objective shortest path update problem. Let there be objectives $O_1, \ldots, O_k$ and associated SSSP solution $S_1, \ldots, S_k$ (known as the SSSP tree S_1 associated to O_1). Let all these SSSP trees be combined to form a single network S'. Then the shortest path in the combined graph provides a heuristic solution for a single MOSP. Leveraging this insight, we apply SSSP update algorithm

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

to update each SSSP tree (associated with different objectives) in dynamic networks and develop a MOSP update algorithm that can efficiently update a single MOSP in dynamic networks. Fig. 2 shows the scalability of shared memory parallel implementation of our MOSP update algorithm.

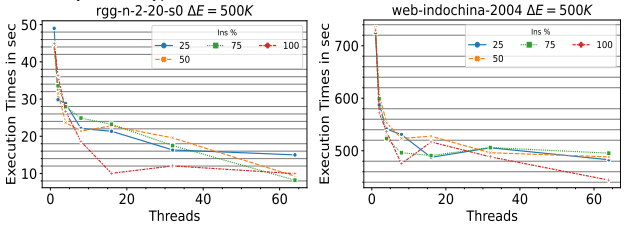

Figure 2: MOSP update (Shared-mem): Scalability plot

2.3 Vertex Color Update

We use the template provided by CANDY to develop a vertex color update algorithm. It first finds the color conflicts and probable color alteration in a network due to edge insertion and deletion and then recolors the affected vertices using a parallel heuristic algorithm in a time- and space-efficient manner [1].

2.4 PageRank update

We developed a shared memory, single/multi GPU implementation on the CANDY architecture to update Page Rank (PR) on streaming graphs. The algorithm uses the multistep approach to process the change edges; mark the affected vertices; and update the vertices whose PR values change significantly. We also empirically and mathematically have seen that when updating the PR of affected vertices whose initial PR values are high rather than updating all the affected vertices gives a significant boost in performance while keeping the accuracy intact.

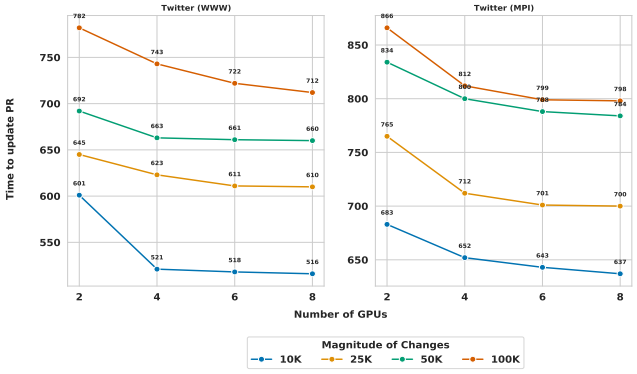

Figure 3: Multi GPU PageRank Update

2.5 Strongly Connected Component (SCC) update

Using CANDY, we developed a distributed algorithm for solving SCCs on dynamic graphs. We use higher-order graph representations called meta-graphs that capture existing structural information in a reduced-size DAG. The algorithm overlays dynamic edge updates and uses asynchronous color propagation to detect further SCCs. By the end of the algorithm, all the vertices in an SCC would be the same color. Fig 4 shows the distributed scalability and memory utilization of our algorithm.

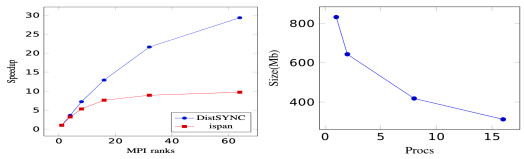

Figure 4: Scalability(left) and Memory Utilization(right) of SCC update algorithm (RMAT 27)

2.6 Dynamic Graph Generation

CANDY aims to enhance dynamic network algorithms by providing accessible and realistic datasets for testing algorithm performance. To achieve this, we are developing a generator that learns the timing and types of interactions observed during the graph's evolution. Exploiting the four interaction types defined by EASEE[3], our generator produces multiple synthetic graph datasets that closely resemble the original graph in terms of size, order, and timing of interactions. These graph datasets can be used to test the performance of different parallel dynamic graph algorithms. The workflow of dynamic graph generation is illustrated in Fig 5.

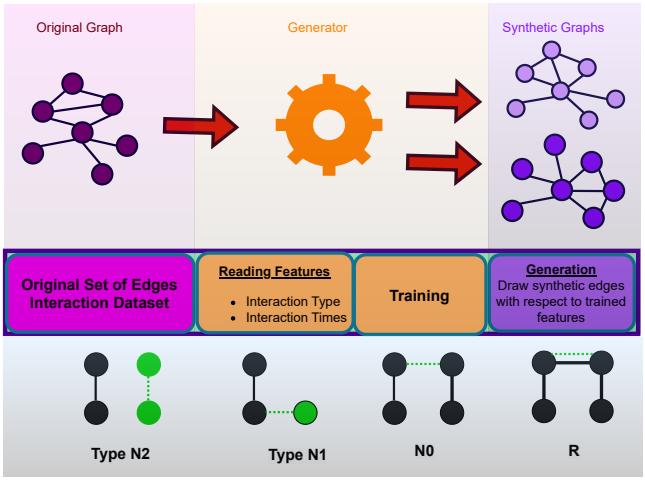

Figure 5: Dynamic Graph Generator

3 ACKNOWLEDGMENTS

This work was partially supported by the NSF projects SANDY (Award # OAC-1725755) and CANDY (Award # OAC-2104115, 2104076, 2104078). We extend our gratitude to Jeremy Wendt, Richard Field, Cynthia Philips, Arvind Prasadan and Sucheta Soundarajan for their guidance on generating synthetic dynamic graphs.

REFERENCES

[1] Arindam Khanda, Sanjukta Bhowmick, Xin Liang, and Sajal K Das. 2022. Parallel Vertex Color Update on Large Dynamic Networks. In *2022 IEEE 29th International Conference on High Performance Computing, Data, and Analytics (HiPC)*. IEEE, 115–124.

[2] Arindam Khanda, Sriram Srinivasan, Sanjukta Bhowmick, Boyana Norris, and Sajal K. Das. 2022. A Parallel Algorithm Template for Updating Single-Source Shortest Paths in Large-Scale Dynamic Networks. *IEEE Transactions on Parallel and Distributed Systems* 33, 4 (2022), 929–940. https://doi.org/10.1109/TPDS.2021.3084096

[3] Jeremy D. Wendt, Richard Field, Cynthia Phillips, Arvind Prasadan, Tegan Wilson, Sucheta Soundarajan, and Sanjukta Bhowmick. 2023. Partitioning Communication Streams Into Graph Snapshots. *IEEE Transactions on Network Science and Engineering* 10, 2 (2023), 809–826. https://doi.org/10.1109/TNSE.2022.3223614