Realistic Computationally Stressing Graph Benchmarks

Peter M. Kogge, Nitesh V. Chawla, Douglas Thain, Brian A. Page, Neil A. Butcher
Dept. of Computer Science and Engineering
Univ. of Notre Dame, Notre Dame, IN 46556
Email: (kogge,nchawla,dthain,bpage1,neil.a.butcher.7)@nd.edu

***Abstract*—There are presently several graph benchmarks in the literature, some with hundreds of published processing reports. They all, however, have several characteristics that make them of academic, but not necessarily real-world interest. This paper suggests three additional benchmark kernels that are more realistic in both graph structure, computational complexity, and support incremental "streaming" versions. For each benchmark, a reference implementation is reported, along with some initial scaling data.**

***Keywords*-Scalability, Hybrid SpMV, Communication Overhead, HPC;**

I. Introduction

A graph is of a set of objects (vertices), and links (edges) between pairs of objects that represent some sort of relationships. Computing over such graphs is of increasing importance to a wide spectrum of application areas ranging from "conventional" communication and power networks, transport, and scheduling, to emerging applications such as social networks, medical informatics, genomics, and cybersecurity.

While there are several current graph benchmarks, some with hundreds of reported implementations, most of them are based on "academic" graph problems, and often have little direct value to real-world applications, especially when we want to understand the relative efficiency of different hardware architectures and configurations. Further, given that many of these graphs are growing in size, it is critical that we understand how to do such processing in parallel in an efficient manner.

This paper discusses three additional benchmarks that have several relevant characteristics:

- There is some archetypal real-world problem that could clearly benefit from efficient solutions.
- The basic computational complexity is often greater than linear in problem size, thus raising the importance of both alternative algorithms and/or heuristics that can significantly reduce computation.
- Streaming, versus today's "batch", versions are of growing importance, where computation is performed "incrementally" as changes to the graph are provided.
- Unlike academic benchmarks, graphs of real-world interest often have are heterogeneous in nature, namely there are multiple classes of vertices, with important edges being between vertices of two different classes.

These benchmarks include computation of Jaccard coefficients between two vertices, determination of a set of edges that form a matching between vertices of two different classes, and learning of paths in heterogeneous networks.

The rest of this paper is organized as follows. Section **??** discusses some current graph benchmarks. Sections **??** through II discuss each of the proposed benchmarks. Section **??** discusses possible approaches for generating synthetic data sets for such benchmarks. Section **??** concludes.

For each of the benchmarks there is a discussion of an archetypal problem that justifies the problem, a formal description of the problem, variations that may make the problem more relevant, an estimate of the computationally complexity of a sequential implementation, a description of streaming variants, and considerations for parallel implementations. It is expected that follow on papers for each benchmark will discuss reference parallel implementations.

II. Stateful Random Walks

A *network* (also called a *flow network* or a *transportation network*) is a graph where the edges are weighted with something related to the "capacity" that it may carry, or has carried. Classical computations over such graphs include determining the "max flow" between two vertices that nowhere exceeds any edge capacity. However, newer problems involve analyzing and predicting traffic patterns within such networks, especially when they are updated dynamically. When the edges in the network are weighted by simply "summing" all flow between two vertices, analysis may miss some more fundamental behaviors that would have surfaced if the prior "paths" of flows had been considered. This benchmark is based on keeping track of such paths, and using path information to answer graph questions.

A. Archetypal Problem

Consider a graph where the vertices are port cities in the world, and edges are shipping lanes. Now consider problems akin to predicting the spread of invasive species that are carried in the bilge water of ships. To use an example from [1], assume the amount of shipping from Singapore to LA is roughly the same as to Seattle. In a "1st order network," weights on an edge represent the total number of ships that have taken that lane between two ports, regardless of where the ships were before. Thus we might assume (wrongly) that invasive species

on ships that went through Singapore were equally likely to spread to Seattle and LA. In reality, however, a ship that came first from Shanghai might be more likely to go to Seattle, whereas a ship that was in Tokyo before Singapore may be more likely to go to LA. Thus the probability of spreading invasive species may be more conditional on how it got to Singapore rather than just coming from Singapore.

A similar problem is predicting web traffic simply on the basis of where users go after any one web site. Studies have shown again that simple 1st order networks that don't account for how users got to a web site are poor indicators of where they go next.

Consider a graph where the vertices are port cities in the world, and edges are shipping lanes. Now consider problems akin to predicting the spread of invasive species that are carried in the bilge water of ships. To use an example from [1], assume the amount of shipping from Singapore to LA is roughly the same as to Seattle. In a "1st order network," weights on an edge represent the total number of ships that have taken that lane between two ports, regardless of where the ships were before. Thus we might assume (wrongly) that invasive species on ships that went through Singapore were equally likely to spread to Seattle and LA. In reality, however, a ship that came first from Shanghai might be more likely to go to Seattle, whereas a ship that was in Tokyo before Singapore may be more likely to go to LA. Thus the probability of spreading invasive species may be more conditional on how it got to Singapore rather than just coming from Singapore.

A similar problem is predicting web traffic simply on the basis of where users go after any one web site. Studies have shown again that simple 1st order networks that don't account for how users got to a web site are poor indicators of where they go next.

B. Formal Description

[2] [1]

C. Sequential Complexity

D. Streaming Variants

E. Parallelization Considerations

REFERENCES

[1] J. Xu, T. L. Wickramarathne, and N. V. Chawla, "Representing higher-order dependencies in networks," *Science Advances*, vol. 2, no. 5, 2016. [Online]. Available: http://advances.sciencemag.org/content/2/5/e1600028

[2] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *KDD '17*. ACM, 2017, pp. 135–144.