GraphMineSuite: Enabling High-Performanc andProgrammable Graph Mining Algorithms with Set Algebra

Author(s): Maciej Besta, Zur Vonarburg-Shmaria, Yannick Schaffner, Leonardo Schwarz, Grzegorz Kwasniewski, Lukas Gianinazzi, Jakub Beranek, Kacper Janda, Tobias Holenstein, Sebastian Leisinger, Peter Tatkowski, Esref Ozdemir, Adrian Balla, Marcin Copik, Philipp Lindenberger, Marek Konieczny, Onur Mutlu, Torsten Hoefler

16.1 General Notes

This paper presents GraphMineSuite (GMS), a comprehensive benchmarking suite for high-performance graph algorithms for graph mining (categories of graph mining problems that the authors identify pattern matching, learning, reordering, and optimization). The motivation differs somewhat from previous papers we've encountered in this class. The authors not only acknowledge the computational challenges that exist in a world with ever-increasing graph sizes and ever-greater importance of fast graph-mining algorithms, therefore requiring the development of better and faster versions of those algorithms, but astutely point out that, in such a landscape, identifying relevant baselines to which to compare such novel algorithms is tremendously challenging. To make meaningful advancements in the field, the very context of research into graph mining algorithms needs to be cleaned up.

The core contributions of GMS include: (a) a benchmark specification, (b) novel performance metric, (c) theoretical concurrency analysis (based on work-depth), (d) a software platform (highly modular), (e) reference implementations. As a result of this consolidation of expertise, GMS provides an environment where state-of-the-art algorithms can be identified, rapidly experimented upon, and dramatically improved.

Over the course of the paper, the authors walk through a number of the graph algorithms they were interested in collecting for GMS, with particular emphasis on a breadth of approaches and implementations. These algorithms were collected over the course of an intensive literature review. A similar overview is provided for the graph datasets, to which the authors append an interesting revelation – that typical graph metrics across which algorithm performances are measured (sparsities, diameters, amounts of locality etc.) are not actually related to the higher-order structure that actually govern the performance of graph mining algorithms, which is often determined by the *origin* of the graph. This makes the importance of a dataset that emphasizes diversity in higher-order structure critical for a project like GMS. The metrics which the authors incorporate into GMS include a range of those we have discussed in class, such as running time, scalability, and memory efficiency. In the spirit of their project, they also introduce a metric that is graph-mining specific, namely "algorithmic efficiency", which they define as the number of graph mining patterns processed per time unit (the exact "pattern" depends on the exact objective of the algorithm, but I understand it to be the smallest unit of useful work done).

Given this setup (a library of algorithms, a dataset of graphs, and useful metrics for evaluating them), the authors proceed to describe how one can experiment using GSM. The six approaches include providing a new graph representation (as competition for CSR), new preprocessing routines, implementing a new graph algorithm, improving upon an existing graph algorithm, and working on new subroutines within graph mining (particularly those which work with set algebra, which the authors dwell on as it is a critical way to represent graphs in the context of mining). This entire section is somewhat striking as it seems to also convey the authors' philosophy around what work is fundamentally worth pursuing in their field (maybe spawning a project idea in case the current thing doesn't work out? We do have this week.)

What is ultimately provided to the user is a suite of tools, algorithms, enhancements and implementations that can be combined to make extremely fast and efficient algorithms (the authors themselves show that a greater than 9x improvement can be achieved over known baselines for the Bron-Kerbosch algorithm). It is closely linked to implementation and experimentation pipelines, offering a true platform-based approach to developing new approaches to graph mining. It is conceptually incredibly impressive, and I would be curious to know if this approach has been replicated in other fields of algorithm engineering. A limitation the authors themselves acknowledge include limited support for multi-core machines, which are critical for real-world graph mining applications. Additionally, the framework does not yet incorporate mechanisms for dynamic graphs, nor the costs of modularity (if you design a new mode of representation, how does that interface with the existing algorithms?) For the purposes of scientific rigor, I would also hope that users can save subsets of the graph dataset that they want to run their tests on so that their results can be replicated and verified by third parties. All in all, however, this is a very cool re-imagining of what it means to be a researcher in a hot, highly relevant, yet dispersed field of research.

(On a bit of an aside, this platform reminds me of a conversation I recently had with a quant who claimed that their firm does things "best" because they use a "modularized, platform" approach. In that firm's case, this meant that they maintain a dataset of all the approaches and signals they have attempted trading on as well as metrics of their relative performance, the data that was used, etc. Because these resources remain available, future work can draw from previous learnings without being bogged down by miscommunication between teams and the risk of re-implementation of old ideas. GSM reminds me of this approach in the sense that there's a lot of extant work and data on graph algorithms that needs to be organized so that developers/researchers interested in a particular facet of the field can consolidate knowledge and resources.)

(Second aside, this is the most heavily annotated paper I have ever seen. Little blue circles with letters in them? Blue highlighting? Pop off.)