Combinatorial Mathematics and Algorithms at Exascale: Challenges and Promising Directions

Assefaw Gebremedhin* and Alex Pothen
Purdue University

Developing numerical algorithms and implementations that fully harness the potential of many-core and accelerated architectures brings on daunting challenges at different levels, but these challenges are so much more exasperated in the case of combinatorial algorithms. Yet combinatorial models and algorithms play a crucial and growing role both as "embedded" and "forefront" enablers in advanced computing. We discuss a selection of contexts in which combinatorial models arise, outline the challenges, discuss a sample of approaches we found promising in recent activities, and point out potentially fruitful avenues for future exploration.

Combinatorial algorithms embedded in scientific computing

At the core of algorithms for solving systems of linear equations, least-squares data fitting, eigenvector and singular vector computations, etc, lie *matrix factorizations*. Efficient factorization methods for *sparse* matrices heavily rely on and are empowered by combinatorial algorithms and data structures. Different types of *matchings* in graphs are used for numerical preprocessing and block triangular decomposition purposes in the solution of sparse linear systems. Computational work and storage reduction in direct solvers for large sparse systems of equations are fruitfully modeled as *vertex ordering* problems in graphs. When computational dependency among subtasks is modeled using a graph, a *vertex coloring* of the graph is used to determine a scheduling of the tasks for concurrent execution. Various types of specialized vertex coloring abstractions are used to meet matrix partitioning needs that arise in the efficient computation of sparse Jacobian and Hessian matrices in numerical optimization.

Combinatorial algorithms at the forefront of discovery

We are in an era when massive digital data continues to be collected at extraordinarily rapid rate and with high and growing complexity and uncertainty, and when the data and the actors behind are increasingly interconnected. There is thus an acute need for the design and implementation of fast and robust algorithms for analyzing massive datasets and extracting knowledge and insight. Exploring the interplay between combinatorial and numerical algorithms will be crucial for developing data analysis methods that perform at scale on contemporary platforms. For example, spectral graph theory can be used to gain insight into, discover important substructures in, and study various properties of networks. Graph traversals are key components of network analysis. Machine learning, data mining and text mining applications heavily involve combinatorial algorithms.

*Corresponding author. Email: agebreme@purdue.edu

Challenges on manycore computing

The design of scalable algorithms and software on manycore and accelerated platforms demands that careful attention be paid to, among others, programming models, algorithm and data structure design, memory management, and energy consumption. These performance-driven and resource-constrained prerequisites are significantly more stringent on combinatorial algorithms, because combinatorial algorithms are characteristically different from floating-point intensive scientific computing algorithms. In particular, they are highly memory-intensive, involve irregular memory access patterns, are highly unstructured, and possess low (immediately available) concurrency. Yet exascale computing demands: orders of 10^6 threads of computation or more per node with only a small amount of memory available, dynamic scheduling, and increased use of task parallelism.

Promising paradigms

In recent studies on various high-performance computing platforms, we have found algorithmic paradigms that have proved promising in effective parallelization of combinatorial algorithms [1–5]. These include:

- *speculation-and-iteration*: the idea is to maximize concurrency by tentatively tolerating potential inconsistencies and then detecting and resolving eventual inconsistencies later, iteratively. This approach has been found effective for parallelizing greedy algorithms.
- *approximate update*: the idea is to minimize synchronization cost by opting for concurrent data structure update with approximate data instead of serialized data structure update with exact data. This approach has been found effective for parallelizing ordering algorithms.
- *parallelized search tree*: the idea is to immediately exchange bounds among processors in a branch-and-bound algorithm so that super linear speedup is possible. This approach has been found effective for parallelizing a maximum clique algorithm.

We will present results on applications of these paradigms on a variety of graph problems. We will also make connections to two broader recent themes in theoretical computer science and parallel computing: *local computation algorithms* and *concurrent data structures*.

References

[1] Umit V. Catalyurek, Florin Dobrian, Assefaw H. Gebremedhin, Mahantesh Halappanavar, and Alex Pothen. Distributed memory parallel algorithms for matching and coloring. In *IPDPS Workshop on Parallel Computing and Optimization (PCO)*, 2011.

[2] Umit V. Catalyurek, John Feo, Assefaw H. Gebremedhin, Mahantesh Halappanavar, and Alex Pothen. Graph coloring algorithms for multicore and massively multithreaded architectures. *Parallel Computing*, 38:576–594, 2012.

[3] Md. Mostofa A. Patwary, Assefaw H. Gebremedhin, Fredrik Manne, and Alex Pothen. Paradigms for Effective Parallelization of Inherently Sequential Graph Algorithms on Multicore Architectures. Submitted, 2013.

[4] Md. Mostofa A. Patwary, Assefaw H. Gebremedhin, and Alex Pothen. New Multithreaded Ordering and Coloring Algorithms for Multicore Architectures. In E. Jeannot, R. Namyst, and J. Roman, editors, *Euro-Par 2011*, volume 6853 of *Lecture Notes in Computer Science*, pages 250–262. Springer, 2011.

[5] Ryan Rossi, David F. Gleich, Assefaw H. Gebremedhin, and Md. Mostofa A. Patwary. Parallel Maximum Clique Algorithms with Applications to Network Analysis and Storage. In *Poster Proceedings of WWW 2014*, 2013. Longer version at arXiv, cs.SI:1302.6256.