Brief Announcement: Streaming and Massively Parallel Algorithms for Edge Coloring

Soheil Behnezhad
Department of Computer Science, University of Maryland, College Park, MD, USA

Mahsa Derakhshan
Department of Computer Science, University of Maryland, College Park, MD, USA

MohammadTaghi Hajiaghayi
Department of Computer Science, University of Maryland, College Park, MD, USA

Marina Knittel
Department of Computer Science, University of Maryland, College Park, MD, USA

Hamed Saleh
Department of Computer Science, University of Maryland, College Park, MD, USA

Abstract

A valid *edge-coloring* of a graph is an assignment of "colors" to its edges such that no two incident edges receive the same color. The goal is to find a proper coloring that uses few colors. In this paper, we revisit this problem in two models of computation specific to massive graphs, the *Massively Parallel Computations* (MPC) model and the *Graph Streaming* model:

Massively Parallel Computation. We give a randomized MPC algorithm that w.h.p., returns a $(1 + o(1))\Delta$ edge coloring in $O(1)$ rounds using $\widetilde{O}(n)$ space per machine and $O(m)$ total space. The space per machine can also be further improved to $n^{1-\Omega(1)}$ if $\Delta = n^{\Omega(1)}$. This is, to our knowledge, the first constant round algorithm for a natural graph problem in the strongly sublinear regime of MPC. Our algorithm improves a previous result of Harvey et al. [SPAA 2018] which required $n^{1+\Omega(1)}$ space to achieve the same result.

Graph Streaming. Since the output of edge-coloring is as large as its input, we consider a standard variant of the streaming model where the output is also reported in a streaming fashion. The main challenge is that the algorithm cannot "remember" all the reported edge colors, yet has to output a proper edge coloring using few colors.

We give a one-pass $\widetilde{O}(n)$-space streaming algorithm that always returns a valid coloring and uses 5.44Δ colors w.h.p., if the edges arrive in a random order. For adversarial order streams, we give another one-pass $\widetilde{O}(n)$-space algorithm that requires $O(\Delta^2)$ colors.

2012 ACM Subject Classification Theory of computation → Massively parallel algorithms; Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Massively Parallel Computation, Streaming, Edge Coloring

Digital Object Identifier 10.4230/LIPIcs.DISC.2019.36

Acknowledgements Supported in part by Guggenheim Fellowship, NSF grants CCF:SPX 1822738, IIS:BIGDATA 1546108, DARPA grant SI3CMD, UMD Year of Data Science Program Grant, and Northrop Grumman Faculty Award.

1 Introduction & Results

Given a graph $G(V, E)$, an edge coloring of G is an assignment of "colors" to the edges in E such that no two incident edges receive the same color. The goal is to find an edge coloring that uses few colors. Edge coloring is among the most fundamental graph problems and has been studied in various models of computation, especially in distributed and parallel settings. In this paper, we study edge coloring in models that target massive graphs. Specifically, we focus on the *Massively Parallel Computations* (MPC) model and the *Graph Streaming* model.

The MPC Model & the Related Work. The MPC model is a popular abstraction of modern parallel frameworks such as MapReduce, Hadoop, Spark, etc. We have seen a plethora of results on graph problems ever since the formalization of MPC. The studied problems include matching and vertex cover [3, 5, 7, 9], maximal independent set [9, 10], vertex coloring [4, 6, 10, 12, 13], as well as graph connectivity and related problems [1, 2, 11].

Not much work has been done on the edge coloring problem in the MPC model. The only exception is the algorithm of Harvey et al. [10] which roughly works by random partitioning the *edges*, and then coloring each partition in a different machine using a sequential $(\Delta + 1)$ edge coloring algorithm. The choice of the number of partitions leads to a trade-off between the number of colors used and the space per machine required. The main shortcoming of this idea, however, is that if one desires a $\Delta + \widetilde{O}(\Delta^{1-\Omega(1)})$ edge coloring, then a strongly super linear local space of $n\Delta^{\Omega(1)}$ is required. In comparison, for the related $(\Delta + 1)$ *vertex* coloring problem, Assadi et al. [4] recently presented an algorithm that takes $O(1)$ rounds and requires a near linear space of $\widetilde{O}(n)$. Unfortunately, this progress on vertex coloring does not imply a better edge coloring MPC algorithm even if we consider the more relaxed $(2\Delta - 1)$ edge coloring problem. The reason is that the well-known reduction, which yields a $(2\Delta - 1)$ edge coloring via a $(\Delta + 1)$ vertex coloring on the line-graph, is not applicable in the MPC model as the line-graph may be significantly larger than the original graph.

Our main result is the following algorithm which achieves a near-optimal edge coloring within a constant number of rounds using a near-linear in n space.

▶ **Theorem 1.** *There exists an MPC algorithm that using $O(n)$ space per machine and $O(m)$ total space, returns a $\Delta + \widetilde{O}(\Delta^{3/4})$ edge coloring in $O(1)$ rounds.*

The Streaming Model. In the standard graph streaming model, the edges of a graph arrive one by one and the algorithm has a space that is much smaller than the total number of edges. A particularly important choice of space is $\widetilde{O}(n)$ – which is also known as the *semi-streaming* model [8] – so that the algorithm has enough space to store the vertices but not the edges. For edge coloring, the output is as large as the input, thus, we cannot hope to be able to store the output in bulk at the end. For this, we consider a standard twist on the streaming model where the output is also reported in a streaming fashion. This model is referred to in the literature as the "W-streaming" model. We particularly focus on one-pass algorithms.

Note that designing one-pass W-streaming algorithms is particularly challenging since the algorithm cannot "remember" all the choices made so far (e.g., the reported edge colors). Therefore, even the sequential greedy algorithm for $(2\Delta - 1)$ edge coloring, which iterates over the edges in an arbitrary order an assigns an available to each color upon visiting it, cannot be implemented since we are not aware of the colors used incident to an edge.

Our first result presented in Theorem 2 is to show that a natural algorithm w.h.p. provides an $O(\Delta)$ edge coloring if the edges arrive in a random-order. Further, we show that for any arbitrary arrival of edges, there is a one-pass $\widetilde{O}(n)$ space W-streaming edge coloring algorithm that succeeds w.h.p. and uses $O(\Delta^2)$ colors.

▶ **Theorem 2.** *If the edges arrive in a random-order, there is a one-pass $\widetilde{O}(n)$ space W-streaming edge coloring algorithm that always returns a valid edge coloring and w.h.p. uses $(2e + o(1))\Delta \approx 5.44\Delta$ colors.*

References

1 Alexandr Andoni, Aleksandar Nikolov, Krzysztof Onak, and Grigory Yaroslavtsev. Parallel algorithms for geometric graph problems. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 574–583, 2014. doi:10.1145/2591796.2591805.

2 Alexandr Andoni, Zhao Song, Clifford Stein, Zhengyu Wang, and Peilin Zhong. Parallel Graph Connectivity in Log Diameter Rounds. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 674–685, 2018. doi:10.1109/FOCS.2018.00070.

3 Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab S. Mirrokni, and Cliff Stein. Coresets Meet EDCS: Algorithms for Matching and Vertex Cover on Massive Graphs. *Proceedings of the 30th annual ACM-SIAM Symposium on Discrete Algorithms (SODA), to appear*, 2019.

4 Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear Algorithms for $(\Delta+1)$ Vertex Coloring. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 767–786, 2019. doi:10.1137/1.9781611975482.48.

5 Soheil Behnezhad, MohammadTaghi Hajiaghayi, and David G. Harris. Exponentially Faster Massively Parallel Maximal Matching. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, to appear*, 2019.

6 Yi-Jun Chang, Manuela Fischer, Mohsen Ghaffari, Jara Uitto, and Yufan Zheng. The Complexity of $(\Delta + 1)$ Coloring in Congested Clique, Massively Parallel Computation, and Centralized Local Computation. *CoRR*, abs/1808.08419, 2018. arXiv:1808.08419.

7 Artur Czumaj, Jakub Lacki, Aleksander Madry, Slobodan Mitrovic, Krzysztof Onak, and Piotr Sankowski. Round Compression for Parallel Matching Algorithms. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 471–484, 2018. doi:10.1145/3188745.3188764.

8 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On Graph Problems in a Semi-streaming Model. In *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings*, pages 531–543, 2004. doi:10.1007/978-3-540-27836-8_46.

9 Mohsen Ghaffari, Themis Gouleakis, Christian Konrad, Slobodan Mitrovic, and Ronitt Rubinfeld. Improved Massively Parallel Computation Algorithms for MIS, Matching, and Vertex Cover. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, pages 129–138, 2018. doi:10.1145/3212734.3212743.

10 Nicholas J. A. Harvey, Christopher Liaw, and Paul Liu. Greedy and Local Ratio Algorithms in the MapReduce Model. In *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures*, SPAA '18, pages 43–52, New York, NY, USA, 2018. ACM. doi:10.1145/3210377.3210386.

11 Tomasz Jurdzinski and Krzysztof Nowicki. MST in $O(1)$ rounds of congested clique. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2620–2632, 2018. doi:10.1137/1.9781611975031.167.

12 Merav Parter. $(\Delta + 1)$ Coloring in the Congested Clique Model. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 160:1–160:14, 2018. doi:10.4230/LIPIcs.ICALP.2018.160.

13 Merav Parter and Hsin-Hao Su. Randomized $(\Delta + 1)$-Coloring in $O(\log^* \Delta)$ Congested Clique Rounds. In *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, pages 39:1–39:18, 2018. doi:10.4230/LIPIcs.DISC.2018.39.