Brief Announcement: Memory Efficient Massively Parallel Algorithms for LCL Problems on Trees

Sebastian Brandt
ETH Zürich, Switzerland

Rustam Latypov
Aalto University, Finland

Jara Uitto
Aalto University, Finland

Abstract

We establish scalable Massively Parallel Computation (MPC) algorithms for a family of fundamental graph problems on trees. We give a general method that, for a wide range of LCL problems, turns their message passing counterparts into exponentially faster algorithms in the sublinear MPC model. In particular, we show that any LCL on trees that has a deterministic complexity of $O(n)$ in the LOCAL model can be sped up to $O(\log n)$ (high-complexity regime) in the sublinear MPC model and similarly $n^{o(1)}$ to $O(\log \log n)$ (intermediate-complexity regime). We emphasize, that we work on bounded degree trees and all of our algorithms work in the sublinear MPC model, where local memory is $O(n^{\delta})$ for $\delta < 1$ and global memory is $O(m)$.

For the high-complexity regime, one key ingredient is a novel *pointer-chain* technique and analysis that allows us to solve any solvable LCL on trees with a sublinear MPC algorithm with complexity $O(\log n)$. For the intermediate-complexity regime, we adapt the approach by Chang and Pettie [FOCS'17], who gave a canonical algorithm for solving LCL problems on trees in the LOCAL model. For the special case of 3-coloring trees, which is a natural LCL problem, we provide a conditional $\Omega(\log \log n)$ lower bound, implying that solving LCL problems on trees with deterministic LOCAL complexity $n^{o(1)}$ requires $\Theta(\log \log n)$ deterministic time in the sublinear MPC model when using a natural family of component-stable algorithms.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Distributed computing, Locally checkable labeling problems, Trees, Massively Parallel Computation, Sublinear memory, 3-coloring

Digital Object Identifier 10.4230/LIPIcs.DISC.2021.50

Funding *Rustam Latypov*: Supported in part by the Academy of Finland, Grant 334238.

1 Introduction

Many fundamental graph problems in the study of distributed and parallel graph algorithms fall under the umbrella of locally checkable labeling (LCL) problems, where each node can locally check whether the output is locally correct. Classic examples include graph colorings, MIS, maximal matching, distributed Lovász Local Lemma (LLL), and many more. These problems serve as abstractions to fundamental primitives in large-scale graph processing and have recently gained a lot of attention [6, 7, 8, 9].

In this work, we study LCL problems in the sublinear Massively Parallel Computation (MPC) model introduced by Karloff et al. [11]. The MPC model is a mathematical abstraction of modern data processing platforms such as MapReduce, Hadoop, Spark, and Dryad. In this model, we have M machines who communicate in an all-to-all fashion. We focus on problems where the input is modeled as a graph of n vertices and m edges. Initially, the graph is divided among the M machines. Each machine has n^{δ} *local memory*, where the units of memory are *words* of $O(\log n)$ bits. When $\delta < 1$, the memory regime is often referred

35th International Symposium on Distributed Computing (DISC 2021).
Editor: Seth Gilbert; Article No. 50; pp. 50:1–50:4
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to as sublinear (or strongly sublinear [3, 4]). The number of machines is chosen such that $n^\delta \cdot M = \Omega(m)$. Ideally, the total memory $M \cdot n^\delta$ is linear in or only slightly higher than the input size m. In this work, we restrict ourselves to linear global memory, and note that, for the $O(\log n)$ regime, speeding up everything exponentially in bounded degree trees with superlinear $O(m^{1+\delta})$ global memory would be straightforward using the well-known graph exponentiation technique [12]. For simplicity, we assume that each vertex is hosted on an independent virtual machine and the memory restriction is that no virtual machine should use more than $O(n^\delta)$ memory. A crucial challenge that comes with the linear global memory restriction is that only a small fraction of $n^{1-\delta}$ of these virtual machines can simultaneously use up their maximum local memory.

1.1 Related Work

In the last decade, there has been tremendous progress in understanding the complexities of locally checkable problems in various models of distributed and parallel computing.

As the most relevant examples to our work, we want to highlight two recent papers. In the randomized/deterministic LOCAL and CONGEST model, Balliu et al. [1] showed that the possible complexity classes of locally checkable problems in rooted regular trees are fully understood. In the CONGEST model, Balliu et al. [2] showed that on trees, the complexity of an LCL problem is asymptotically equal to its complexity in the LOCAL model. They also showed that the same does not hold in general graphs. It is worth noting that in order to prove the asymptotic equality between LOCAL and CONGEST for LCL problems on trees, Balliu et al. [2] use very similar, but independently developed, techniques to what we use in our intermediate-complexity regime.

Regarding our coloring result, it is worth mentioning that in the low-memory setting an $O(\log \log n)$-round *randomized* MPC algorithm for 4-coloring trees was given by Ghaffari, Grunau, and Jin [9]. For the classical $(\Delta + 1)$-coloring problem the deterministic state of the art is given by a very recent result by Czumaj, Davies, and Parter providing an $O(\log \log \log n)$-round algorithm [7].

2 Results

Our main results are twofold: we show that any LCL on trees that has a complexity of $O(n)$ in the LOCAL model can be sped up to $O(\log n)$ (high-complexity regime) in the sublinear MPC model and similarly $n^{o(1)}$ to $O(\log \log n)$ (intermediate-complexity regime). For the LCL problem of 3-coloring trees, we provide a conditional $\Omega(\log \log n)$ lower bound for component-stable algorithms. This implies that solving LCL problems on trees with LOCAL complexity $n^{o(1)}$ requires $\Theta(\log \log n)$ time in the sublinear MPC model by component-stable algorithms. The lower bound is conditioned on a widely believed conjecture and currently, the family of component-stable algorithms rules out the known techniques for coloring trees with few colors. Next, we introduce the formal statements and a high level idea of the techniques that we used to obtain our results; the proof details and a more thorough discussion of related work will appear in the full version of the paper.

▶ **Theorem 1** (High-complexity regime). *Every LCL on constant degree trees that admits a correct solution can be solved deterministically in $O(\log n)$ rounds with $O(n^\delta)$ words of local memory for any constant $\delta > 0$ and $O(m)$ words of global memory in the MPC model.*

The proof of Theorem 1 is constructive: we explicitly provide, for any solvable LCL, an algorithm $\mathcal{A}$ that has a runtime of $O(\log n)$. On a high level, algorithm $\mathcal{A}$ proceeds in 3 phases. The first phase consists in rooting the input tree by using a Rake & Compress style process that also orients the removed paths. In the second phase, roughly speaking, the goal is to compute, for a substantial number of nodes v, the set of output labels that can be output at v such that the label choice can be extended to a (locally) correct solution in the subtree hanging from v. This is done in an iterative manner, proceeding from the leaves towards the root. The last phase consists in using the computed information to solve the given LCL from the root downwards.

While this outline sounds simple, there are a number of intricate challenges that require the development of novel techniques, both in the design of the algorithm and its analysis: for instance, the depth of the input tree can be much larger than $\Theta(\log n)$ (which prevents us from performing the above ideas in a sequential manner even when using a Rake & Compress process, and introduces a new challenge in the form of interleaving Rake & Compress steps), and the storage of the required completability information in a standard implementation exceeds the available memory even when using graph exponentiation. One of our key technical contributions is the design of a fine-tuned potential function for the analysis of the complex algorithm resulting from addressing these issues.

▶ **Theorem 2** (Intermediate-complexity regime). *Consider an LCL problem Π with a deterministic LOCAL complexity $n^{o(1)}$. Then, there is a deterministic MPC algorithm that solves Π in time $O(\log \log n)$ with $O(n^\delta)$ words of local memory for any constant $\delta > 0$ and $O(m)$ words of global memory.*

The proof is similar to the proof of Lemma 14 by Chang and Pettie [5]. Essentially, we simulate their LOCAL algorithm but pay special attention to the memory usage. Let us outline the modifications required to reach our goal.

First, we adopt their Rake & Compress decomposition for trees by synthesizing an exponentially faster algorithm that computes the decomposition in the MPC model. Using this decomposition we divide the nodes into *batches* according to which layer (or partition) they belong to such that removing one batch from the decomposition reduces the number of nodes by a factor of Δ (maximum degree of the graph). We process the graph one batch at a time for $O(\log \log n)$ phases until we have enough global memory to directly simulate the LOCAL algorithm in a constant number of rounds. During each phase we simulate the LOCAL algorithm for the lowest batch, after which we perform graph exponentiation.

▶ **Theorem 3** (Conditional hardness). *The problem of 3-coloring constant degree trees in the sublinear MPC model has deterministic complexity $O(\log \log n)$. Under the connectivity conjecture, there is no $o(\log \log n)$ round component-stable MPC algorithm for 3-coloring of constant degree trees.*

Our conditional lower bound is based on the existence of high-girth graphs that are not 3-colorable by Marshal [13], and on the work of Ghaffari et al. [10], which assumes component-stability and conditions on the widely believed connectivity conjecture in MPC. The family of component-stability algorithms captures all the currently known methods to color trees with 3 colors. In a very recent work, Czumaj et al. [6] show that some problems can be solved much faster with component-unstable algorithms than with stable ones. While their results cast uncertainty on the strength of our lower bound, we note that it is not clear at all how to extend the separation result to problems such as the 3-coloring that relies on "global" graph properties such as the arboricity. Furthermore, their algorithms require much more global memory than what we allow in our paper.

References

1 Alkida Balliu, Sebastian Brandt, Dennis Olivetti, Jan Studeny, Jukka Suomela, and Aleksandr Tereshchenko. Locally checkable problems in rooted trees. In *PODC*, 2021. arXiv:2102.09277.

2 Alkida Balliu, Keren Censor-Hillel, Yannic Maus, Dennis Olivetti, and Jukka Suomela. Locally checkable labelings with small messages. In *DISC*, 2021. arXiv:2105.05574.

3 Sebastian Brandt, Manuela Fischer, and Jara Uitto. Matching and MIS for uniformly sparse graphs in the low-memory MPC model. *Theorerical Computer Science*, 2018. arXiv:1807.05374.

4 Sebastian Brandt, Manuela Fischer, and Jara Uitto. Breaking the linear-memory barrier in MPC: Fast MIS on trees with strongly sublinear memory. In *SIROCCO*, 2019. doi:10.1007/978-3-030-24922-9_9.

5 Yi-Jun Chang and Seth Pettie. A time hierarchy theorem for the LOCAL model. In *FOCS*, 2017. doi:10.1109/FOCS.2017.23.

6 Artur Czumaj, Peter Davies, and Merav Parter. Component stability in low-space massively parallel computation. In *PODC*, 2021. arXiv:2106.01880.

7 Artur Czumaj, Peter Davies, and Merav Parter. Improved deterministic $(\Delta + 1)$ coloring in low-space MPC. In *PODC*, 2021. doi:10.1145/3465084.3467937.

8 Michal Dory, Orr Fischer, Seri Khoury, and Dean Leitersdorf. Constant-round spanners and shortest paths in Congested Clique and MPC. In *PODC*, 2021. doi:10.1145/3465084.3467928.

9 Mohsen Ghaffari, Christoph Grunau, and Ce Jin. Improved MPC algorithms for MIS, matching, and coloring on trees and beyond. In *DISC*, 2020. arXiv:2002.09610.

10 Mohsen Ghaffari, Fabian Kuhn, and Jara Uitto. Conditional hardness results for massively parallel computation from distributed lower bounds. In *FOCS*, 2019. doi:10.1109/FOCS.2019.00097.

11 Howard J. Karloff, Siddharth Suri, and Sergei Vassilvitskii. A model of computation for MapReduce. In *SODA*, 2010. doi:10.1137/1.9781611973075.76.

12 Christoph Lenzen and Roger Wattenhofer. Brief announcement: Exponential speed-up of local algorithms using non-local communication. In *PODC*, 2010. doi:10.1145/1835698.1835772.

13 Simon Marshall. Another simple proof of the high girth, high chromatic number theorem. *The American Mathematical Monthly*, 2008. doi:10.1080/00029890.2008.11920498.