LITERATURE REVIEW: — Distributed Graph Colouring —

David Worley
School of Computer Science
University of Ottawa
Ottawa, Canada
dworley@uottawa.ca

November 1, 2021

1 Introduction

Graph colouring, the problem of assigning a colour to each vertex of a graph such that no two neighbouring vertices share the same colour, is one of the most important problems in graph theory. From the proposal of the Four Colour Theorem in the mid-1850s until today, graph colouring has been a fundamental problem in graph theory, and the focus of an incredible amount of research. The optimization aspect of this problem is identifying what the minimum amount of colours is needed to create such a proper colouring on a graph, and of course implementing fast algorithms to find such colourings.

Furthermore, graph colouring sees many applications in scheduling and identifying groups in social graphs. Unfortunately, the problem of deciding if a graph has a k-colouring (can be coloured with k colours) is NP-complete, and thus no sequential, polynomial algorithms for finding graph colourings exist for general cases. This makes finding ways to improve current state-of-the-art algorithms incredibly important, as these algorithms need to run efficiently on graphs with massive scale.

Distributed Graph Colouring is an area looking to solve this problem by parallelizing algorithms for graph colouring. While this introduces additional considerations regarding the parallel model under consideration, the speed-ups obtained as a result are strongly beneficial. Thus, the development of simple, efficient distributed graph colouring algorithms is a large research area, with entire books dedicated solely to the topic [2].

2 Literature Review

2.1 Bounding Distributed Graph Colouring

Currently, the majority of work in distributed graph colouring focuses on finding a k-colouring such that $\Delta + 1 \leq k \leq O(\Delta^2)$, where Δ is the maximum valency of a vertex. The lower bound results from the common fact that any graph where the maximum degree is Δ can be coloured with $\Delta + 1$ colours, with simple greedy algorithms able to find such a colouring. The upper bound is a result of Linial, who proposed a model for distributed graph algorithms called the LOCAL model [5]. This model is round-based, allowing for each vertex to transmit information to each of its neighbours at the end of each round. Using this model, Linial shows that an $O(\Delta^2)$ colouring can be generated in $O(\log^* n)$ rounds, where $\log^*$ is the iterative logarithm.

This model proves to be the simplest to work with, though other common models are also important in the literature. These include the SET-LOCAL model, sometimes called the weak LOCAL model, in which vertices do not have IDs, and so cannot distinguish between the messages of its neighbours. The main assumption algorithms work off if within the SET-LOCAL model is that the algorithm *begins* with a proper colouring [4]. Another common model is the more restrictive CONGEST model, in which the vertices can only transmit $O(\log n)$ data to each neighbour per round. This is particularly restrictive as vertices are typically identified using bit IDs of size $O(\log n)$.

2.2 One Round Colour Reductions

While the $O(\Delta^2)$ bound proposed above is large, this is acceptable due to the presence of colour-reduction algorithms that can be used to reduce the amount of colours in the colouring. Working under the LOCAL model Linial proposed, many researchers focus on improving the efficiency of the colour-reduction algorithms so that, in combination with the colouring algorithms, more optimal graph colourings can be obtained without large trade-offs in speed. Linial himself once again sets strong foundations for this area, presenting an algorithm to reduce a k-colouring of a graph to a $O(\Delta^2 \log m)$-colouring in a single round under his model [5].

From there, much research focuses on one-round colour reduction algorithms that can reduce the colours significantly for certain classes of graphs. For example, one-round colour reduction algorithms were developed for directed paths that can reduce k-colourings to 3-colourings in $\frac{1}{2}\log^* n + O(1)$ rounds [3]. This was later reduced into a tight bound of $\frac{1}{2}\log^* n$ [7]. This result is an important due to its usefulness as a subroutine of other distributed graph colouring algorithms. Other improvements look towards specific classes of graphs or reductions from k-colourings under some set of assumptions about k. For example, Maus proposes a one-round colouring algorithm that reduces an k-colouring to a $m(\Delta - m + 2)$-colouring, given that $k \geq m(\Delta - m + 3)$, removing m colours from the colouring, where $1 \leq m \leq \Delta/2 + 3/2$ [6].

2.3 Improving Colouring Algorithms

With strong colour reduction algorithms, efficient colourings can be obtained through the use of a colouring algorithm followed by repeated colour reductions to said graph. Considering our lower bound of $O(\Delta + 1)$, its natural to ask how many rounds a colouring algorithm requires to result in a $(\Delta + 1)$-colouring. In 1993, Szegedy and Vishwanathan showed that for algorithms that were locally iterative, a $(\Delta + 1)$-colouring algorithm requires $\Omega(\Delta \log \Delta + \log^* n)$ rounds, barring the existence of a special type of colouring whose reduction could be done very efficiently. A locally-iterative algorithm is one where each vertex chooses its next colour based solely on the colours of its local neighbourhood, so this bound applied to a large class of the algorithms within the field. For a while, no algorithm could make use of such a special colouring nor could it beat the so-called SV Barrier lower bound. It wasn't until 2021 that this barrier was broken, with an iterative algorithm that could compute a $(\Delta + 1)$-colouring with runtime $O(\Delta + \log^* n)$ [1].

2.4 Maus's Algorithm[6]

The algorithm proposed by Maus in [6] is a round based colour reduction algorithm scaling between the two bounds presented above as follows. For a given integer $1 \leq k \leq O(\Delta)$, the algorithm generates an $O(\Delta k)$-colouring in $O(\Delta/k)$ rounds through a trial based reduction of an $O(\Delta^2)$ colouring, like one given by Linial's algorithm. Each vertex of the graph will compute a sequence of colours and attempt to colour it self with the first k colours in its sequence, stopping on the first one that is found to be without conflict. If no colours in the first k are without conflict, then the next k are tested in the subsequent round. This process repeats until all vertices are coloured.

Since Δ is known and k can be chosen before the algorithm is run, k can always be selected proportionally to Δ to make the algorithm into a $O(1)$-round reduction algorithm, beating many well known heuristic based reduction algorithms that ran in poly-Δ rounds. Maus also shows that this algorithm can be used, with minor modifications, as a subroutine for other types of graph colouring problems such as d-defective colourings, where vertices are allowed to have the same colour as at most d of their neighbours.

References

[1] L. Barenboim, M. Elkin, and U. Goldenberg. Locally-iterative distributed $(\delta + 1)$-colouring below szegedy-vishwanathan barrier, and applications to self-stabilization and to restricted-bandwith models. *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 437–446, 2018.

[2] Leonid Barenboim and Michael Elkin. Distributed graph coloring: Fundamentals and recent developments. *Synthesis Lectures on Distributed Computing Theory*, 4, 07 2013.

[3] R. Cole and U. Vishkin. Deterministic coin tossing with applications to optimal parallel list ranking. *Information and Control*, 70(1):32–53, 1986.

[4] D. Hefetz, F. Kuhn, Y. Maus, and A. Steger. Polynomial lower bound for distributed graph colouring in a weak local model. *Proc. of the 30th International Symp. on Distributed Computing*, pages 99–113, 2016.

[5] N. Linial. Locality in distributed graph algorithms. *Society for Industrial and Applied Mathematics*, 21(1):193–201, 1992.

[6] Y. Maus. Distributed graph colouring made easy. *SPAA '21*, pages 362–372, 2021.

[7] J. Rybicki and J. Suomela. Exact bounds for distributed graph colouring. *SIROCCO 2015*, 9439:46–60, 2015.