Proof of Work Algorithms for Solving NP-Complete Problems

Pericles Philippopoulos[1], Alessandro Ricottone[1], Carlos Oliver[2,*]

[1] Department of Physics, McGill University, Montreal, Canada

[2] School of Computer Science, McGill University, Montreal, Canada

July 5, 2017

Abstract

Chill abstract

*To whom correspondence should be addressed. Tel: +1 514-555-5018; Email: carlos.gonzalezoliverl@mcgill.ca

1 Introduction

The large scale success of cryptocurrency platforms such as Bitcoin[1], Ethereum[2] and many others stands to change the nature of computational networks as we know them. Bitcoin has proven to be a reliable way to transfer value in a secure and efficient manner, and Ethereum is showing great promise in the de-centralization not only of transaction management but also of software execution in general. However, in order for these technologies to function in a secure and reliable manner, most cryptocurrency networks allocate large amounts of computational effort to executing proof of work algorithms . Proof of work (PoW) is used to reach consensus and the immutability of the ledger by allowing the network to easily verify that a considerable amount of computational work went towards validating a given version of the ledger. Therefore, anyone wishing to alter or to create their own version of the ledger would have to go perform a substantial amount of computation in order for their version of the ledger to be accepted. Given the large scale use of popular cryptocurrencies, such an attack is impossible unless the unlikely scenario of attacker controlling the majority of the network's computing power takes place. The most accepted versions of proof of work at the moment are based on producing outputs of cryptographic hash functions with a desired form which can be tuned to adjust difficulty. While such algorithms have shown to be very reliable and scalable, the output of the computations is only informative in so far as it is used directly to validate the state of the ledger. In this work we present a proof of work algorithm that retains the important properties of a reliable consensus and immutability algorithm while at the same time harnessing the available computational time to provide information on mathematically interesting problems. More specifically, we present `graff`, a proof of work algorithm based on solving the important problem of graph colouring.

how much comp power is bitcoin using

1.1 Proof of Work

1.2 Graph Colouring and NP-completeness

1.3 `graff`

Topics in intro

- crypto background

- review of proof of work algorithms
- motivate graph coloring as proof of work (present applications, etc)
- summarize our contribution

Ideas for algorithm.

- We proposed to generate graph from the hash of the previous block.
- Proof of work comes from providing a valid colouring of the resulting graph.
- The validity of a colouring can be verified in linear time.
- We can tune difficulty by generating graphs with varying connectivities and connectivity patterns.
- Problem: asking for a k colouring of a graph is not guaranteed to yield a solution.
- We proposes to ask for a k colouring that is bounded by the max degree (Brooks theorem: in a connected graph in which every vertex has at most Δ neighbors, the vertices can be colored with only Δ colors, except for two cases, complete graphs and cycle graphs of odd length, which require $\Delta + 1$ colors) and number of edges of the graph. There is the issue of producing graphs with a sufficiently large gap between the two bounds. Alessandro is working on guiding the design of the graphs, potentially inducing cliques.
- Alternate proposition: miners work on the branch with the maximum difficulty. In this case, miners would look at the total number of colors used in the branch and mine on the one with the smallest number. If a miner decides to provide trivial colorings, then they would be contributing less difficulty to the branch and run the chance of their branch not being mined on. This solves the problem of not a k coloring not being guaranteed to exist.
- What kind of hardware could solve this problem? Is it ASIC resistant?

2 Methods

3 Results

4 Discussion

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[2] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151, 2014.