Proposal for a fully decentralized blockchain and proof-of-work algorithm for solving NP-complete problems

Carlos G. Oliver†
School of Computer Science, McGill University, Montreal, Canada

Alessandro Ricottone and Pericles Philippopoulos
Department of Physics, McGill University, Montreal, Canada
(Dated: September 2, 2017)

Abstract We propose a proof-of-work algorithm that rewards blockchain miners for using computational resources to solve NP-complete puzzles. The resulting blockchain will publicly store and improve solutions to problems with real world applications while maintaining a secure and fully functional transaction ledger.

I. INTRODUCTION

The widespread success of cryptocurrency platforms such as Bitcoin [1], Ethereum [2] has attracted a substantial amount of computational resources [3]. However, the majority of this computing power is destined for executing proof-of-work algorithms (for example, the Bitcoin hashrate is over four exahash per second [4]). While proof-of-work algorithms are highly reliable, the information generated by mining does not extend beyond guaranteeing the validity of the information on the network. In this work, we aim to design a new mining paradigm that diverts some of the computational resources from mining to solving problems with real world applications while simultaneously maintaining a secure blockchain.

We focus on the class of problems known as NP-complete problems. Such problems are most readily applied to blockchain systems because they possess the important property that the solution to the problem can be verified in polynomial time, while identifying the solution in the first place has no known polynomial algorithm. One of the first systems to attempt such a paradigm was Primecoin [5], which in 2013 proposed that users devote their computational power towards finding specific chains of prime numbers instead of cryptographic mining. While the identification of prime numbers is of interest generally, technical limitations force the coin to solve for a specific type of prime number whose scientific impact is not clear at the moment. Other blockchains such as the CureCoin [6] (previously known as Foldcoin) and Coinami [7] have attempted to solve bioinformatics problems which are of more direct impact. However, both systems depend on a central authority to delegate the problems and validate the identity of users who must register. We therefore aim at developing a cryptocurrency that addresses these shortcomings while still reliably acting as a trusted distributed ledger. The key components we aim to implement are:

- Full decentralization
- Anonymity
- Generation of solutions to practical problems

By proposing a novel mining and incentive protocol to the established Bitcoin framework, we believe we have achieved a system that satisfies all of the above features.

II. BLOCKCHAIN

The proposed blockchain is based closely on the well established Bitcoin blockchain protocol, with one key difference: miners are rewarded for solving an NP-complete problem via a mining difficulty reduction. The idea is that miners, after building a block are allowed to choose between submitting a block mined using the standard Bitcoin protocol (see [1]) by finding a valid nonce, or if they have computed an improved solution to an NP-complete problem can publish it along with their block which would be verified and accepted by the network at a reduced difficulty.

For the sake of simplicity, we assume that the blockchain will deal with a single instance of a given NP-complete problem, we call this problem $P$. For example, if the NP-complete problem were finding a graph coloring (GC) of a graph $g$ of size less than some current best $k$, the blockchain would only work on that specific $g$ and $P := (g, k)$. In Section III we discuss ways to incorporate a greater number of instances of the problem. The only requirement for a problem to be used in the proof-of-work is that verifying a solution to the problem can be accomplished in polynomial time with a clearly defined scoring scheme.

A. Block structure

Blocks are structured identically to Bitcoin blocks (i.e. Transaction Merkle Root, miner address, nonce, etc.), with the addition of a new field for storing the problem,
The block normally. on a null value that indicates that the miner produced
this would be a vector with the index of visited nodes. However, not all blocks are forced to contribute a solu-
representation of a solution. In the case of TSP,
this be a vector with the index of visited nodes. However, not all blocks are forced to contribute a solu-
to the given problem, and this field is allowed to take
on a null value that indicates that the miner produced
the block normally.

B. Incentivization

The main task a Bitcoin miner has to accomplish in
order to produce a valid block once he has obtained a
set of valid transactions is to find a valid nonce, $n$. More
specifically, miners need to find an integer $n$ such that

$$H(B, n) < \epsilon_d,$$  \hfill (1)

where $H(B, n)$ is the hash function applied to block $B$
and $n$, and $\epsilon_d$ is the target associated with difficulty $d$.
Because $H$ is a cryptographic hash function, the output
of $H$ is completely unpredictable, therefore mining con-
ists of brute forcing values of $n$ until the condition is
satisfied. Clearly an $\epsilon_d$ with smaller $d$ is easier to satisfy
than the inverse. Once a miner finds a valid nonce they
can publish the block along with the $n$ and the network
can easily verify the validity of the block according to the
network’s current difficulty $d$. Because mining is a com-
petitive endeavor, we reward miners that compute solutions
to $P$ by reducing mining difficulty. In our protocol,
iminers will accept blocks that satisfy a reduced difficulty
$d_r$ constraint if the block contains a solution that is bet-
ter than the current best solution on the blockchain. If
the average time needed to mine a block with a solution
is shorter than the average time needed to mine a block
without, we expect miners to spend their computational
power in improving the solution to the problem.

C. Mining & Difficulty Scaling

In Bitcoin, the hashing difficulty is retargetted every
$N (= 2016)$ blocks, by comparing the time it took to
mine these blocks, $T^*$, to a target time $T$. In the pro-
posed blockchain, we have two different difficulties which
must each be retargetted. These difficulties are $d_b$, for
blocks mined without a solution to problem $P$, and $d_r$, a
reduced difficulty for blocks mined with a solution to problem $P$. For the retargetting of both difficulties to
be accomplished, a value for $T$ must be fixed by the
blockchain, as is the case for Bitcoin. However, we must
also establish the constant $0 < \eta < 1$, which represents
the desired ratio of time it takes to mine a block with a
solution to problem $P$, $t_s$, to the time it takes to mine
a block without a solution to problem $P$, $t_b$. A smaller $\eta$
would result in a greater incentive for puzzle solutions to
be found.

We define $t^*_s$ to be the average time it took to mine
a block with a solution to problem $P$ and $t^*_b$ to be the
average time it took to mine a block without a solution
to problem $P$. We can thus write these two quantities as

$$t^*_s = \frac{d_r}{p_H} + \frac{d_b}{p_H},$$ \hfill (2)
$$t^*_b = \frac{d_b}{p_H},$$  \hfill (3)

where $p_H$ is the rate at which computations can be per-
formed by the network, and $d_p$ is the difficulty associated
with solving the problem $P$. We expect $d_p$ to increase
with time as better and better solutions to problem $P$
are found. We also define the measured quantity, $\eta^* = t^*_s/t^*_b$
and $b$, the fraction of blocks mined with base difficulty $db$.
Using Eqs. (2) and (3),

$$\eta^* = \frac{d_r + d_p}{d_b}.$$ \hfill (4)

The goal, after $N$ blocks, is to set $d_r$ and $d_b$ to new values $d_r'$ and $d_b'$ so that the time it takes to mine the
$N$ blocks, $T^*$, under fixed $p_H$, $d_p$, and $b$, readjusts
towards the target $T$ and $\eta^*$ readjusts towards $\eta$. We can
therefore write

$$T^* = \frac{N}{p_H} d_b [b + (1-b)\eta^*],$$ \hfill (5)
$$T = \frac{N}{p_H} d_b' [b + (1-b)\eta].$$ \hfill (6)

Solving these equations for the retargetted base difficulty
yields,

$$d_b' = d_b \left[ \frac{b + (1-b)\eta^*}{b + (1-b)\eta} \right] \frac{T}{T^*}. \hfill (7)
$$

To retarget $d_r$, we use the fact that the difficulties
should be updated so that

$$\eta = \frac{d_r' + d_p}{d_b'}. \hfill (8)
$$

We note that $d_p$ is not retargetted to $d_r'$ as it is complete-
ly determined by the status of the problem $P$. Com-
bining Eq. (8) with Eqs. (2) and (3), we can write

$$d_r' - d_r = \eta d_b' - \eta^* d_b. \hfill (9)
$$

Thus, Eqs. (7) and (9) display the rules for the difficulty
retargetting of the proposed blockchain. As with
the Bitcoin blockchain, it would be advisable to intro-
duce a maximum retargetting factor (4 for the Bitcoin
chain) to avoid a change in difficulty that is too abrupt.
In other words, we would enforce
After \( P \) has become too difficult to improve, it is possible that \( d_p > d_b \). This implies that the updated difficulty \( d'_p \) becomes negative at which point it would be necessary to introduce a new problem \( P \) to work on. We discuss how this can be implemented in Section III. However, at the point when it is no longer worthwhile to solve the puzzle, the blockchain naturally functions with \( d_b \) and the standard Bitcoin proof-of-work.

\[
\frac{1}{4} \leq \frac{d'_p}{d_b} \leq 4. \tag{10}
\]

D. Possible Puzzles

There is a wealth of interesting NP-complete problems that could be used as puzzles for the blockchain. Here we suggest a list, which is by no means exhaustive, of possible applications:

- Multiple Sequence Alignment: many databases with DNA sequence information are widely available (e.g. [8]). The problem of improving aligned multiple DNA sequences is NP-complete and has many applications in biology and medicine.
- Protein/biomolecule folding and design: computing the 2D and 3D geometry of chains of DNA/RNA/Protein as well as designing sequences with desired geometries. Many databases for this problem are also available, e.g. Protein Data Bank [9]. Solutions so this problem can have direct medical applications.
- Ising-lattice: The decision form of the Ising model (to decide whether the ground state of an Ising Hamiltonian has energy \( E \leq 0 \)) is NP-complete [10] and it can be mapped to many other NP-complete problems. These models are widely studied in physics.

III. MULTIPLE-PUZZLE BLOCKCHAIN

Clearly a blockchain that only works on a single problem, \( P \) will quickly exhaust its usefulness. Ideally, we would want the blockchain to simultaneously solve a set \( \Omega \subseteq \Omega_j \) of instances of an NP-complete problem \( j \), where \( i \) is an index over all possible NP-complete problems, \( j \in \{ \text{GC, TSP, } \ldots \} \), and \( \Omega_j \) contains all possible NP-complete problems of type \( j \). While we leave the specifics of how this can be achieved to future work, here we discuss some of the challenges and potential solutions.

A. Puzzle Storage

Because we wish to have full decentralization, \( \Omega \) must be stored on the blockchain. This can be achieved in one of two manners. If the size of \( \Omega \) is small enough to be stored by all the miners, the genesis block could simply be used to store an indexed database of \( \Omega \). Subsequent blocks that contain solutions to a problem \( P \in \Omega \) would simply include with their solution a pointer to the corresponding problem in the genesis block. If \( \Omega \) is too large for all the miners to store, the network could instead allow some nodes to participate as ‘storage’ nodes and collect a reward for doing so in a manner similar to file storage coins [11]. The main blockchain would then simply contain pointers to the relevant problem for each block (Fig. 1, lower half).

B. Puzzle Selection

In the single-puzzle blockchain, miners always work on the same problem. In the multiple-puzzle setting there must be a protocol for selecting the problem for the current block. The most natural approach is to allow miners to submit a solution to any \( P \in \Omega \) they choose for a reduced difficulty. This has the advantage of potentially maximizing the network’s efficiency in solving problems if miners work on non-overlapping regions of \( \Omega \). However, if one wishes to force some distribution on the frequency that each puzzle gets included in a block, one could use the hash of the previous block to determine the index \( j \) of the current admissible puzzle, \( P \in \Omega \). (Fig. 1, upper half) Such a function could be augmented for example to favor problems not yet been included and suppress problems that have been worked on too much.

C. New Puzzles

Eventually (unless \( \Omega \) is very large), all puzzles will reach some plateau of optimality and further computation will not produce significant gains. The obvious solution to this situation is to allow the blockchain to incorporate new problems into \( \Omega \). The challenge is to manage new puzzle incorporation without a central authority that ensures problems are: 1) valid instances of \( j \) 2) of interest 3) not already solved. New puzzles can be included as a special transaction. This transaction can store the problem in the current block, or add it to the storage nodes. To encourage puzzles that do not satisfy the three criteria, new problems should be submitted with a fee which is then distributed as a reward to miners that solve the puzzle. Alternatively, the network can agree on a new problem set off-chain and in a manner similar to Bitcoin, induce a fork or upgrade of the chain that includes a new agreed upon problem set.

IV. CONCLUSION

We have outlined the major components of a cryptocurrency system that incentivizes the identification of
solutions to scientifically interesting problems without relying on any central authorities or servers. The benefit of using this proposed blockchain instead of other proof-of-work blockchains is twofold. A portion of the power expended on hashing is redirected to solving problems that are scientifically relevant. And second, solutions to these problems are naturally stored and updated in the blockchain for public access. To our knowledge, this is the first blockchain solution that achieves these features in a fully decentralized manner. The implementation of this system is left for future work.