LEDGER
ledgerjournal.org

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2020.194

RESEARCH ARTICLE

Difficulty Scaling in Proof of Work for Decentralized Problem Solving

Pericles Philippopoulos,[*†] Alessandro Ricottone,[‡] Carlos G. Oliver[§]

Abstract. We propose DIPS (**D**ifficulty-based **I**ncentives for **P**roblem **S**olving), a simple modification of the Bitcoin proof-of-work algorithm that rewards blockchain miners for solving optimization problems of scientific interest. The result is a blockchain which redirects some of the computational resources invested in hash-based mining towards scientific computation, effectively reducing the amount of energy 'wasted' on mining. DIPS builds the solving incentive directly in the proof-of-work by providing a reduction in block hashing difficulty when optimization improvements are found. A key advantage of this scheme is that decentralization is not greatly compromised while maintaining a simple blockchain design. We study two incentivization schemes and provide simulation results showing that DIPS is able to reduce the amount of hash-power used in the network while generating solutions to optimization problems.

1. Introduction

Energy spent hashing in Proof of Work (PoW) is critical for guaranteeing the integrity of transactions on a blockchain. However, the information that results from the computations in itself is of low value. More specifically, the outcome of mining a block is simply a proof that there exists a nonce for which the hash of the given block is lower than some threshold. Naturally, this knowledge is not applicable outside of the blockchain itself. Mining protocols which secure the network and are themselves informative have therefore been an attractive goal since the early days of blockchain.[1–3] Meanwhile, crowd sourcing efforts have been successful in showing that solutions to difficult scientific problems can be discovered by a large community.[4–6] However, current crowd-sourcing models offer few incentives for participation which limits the size of the user-base. On the other hand, cryptocurrency mining has been shown to offer a very strong incentive scheme and currently draws a much larger pool of contributors. Including scientific computing in the mining protocol of a blockchain (more specifically a cryptocurrency) would therefore introduce and incentivize a larger community to solving scientific problems.

1.1. Related Work—We borrow the term 'useful work' to describe protocols which aim at incentivizing computations with real-world applications other than securing blockchain integrity, while acknowledging that this is not to say that standard PoW is not 'useful.'[7] An obvious

[*] 16GW5qUFZ2uCL8WizS8qrMMVvZDoQu4LcU

[†] P. Philippopoulos (pericles@ozeki.io) is co-founder of Ōzeki Inc. a blockchain consultancy firm in Montreal, Canada.

[‡] A. Ricottone is a Doctoral Student in Quantum Computing at McGill University, Montreal, Canada.

[§] C. G. Oliver (carlos@ozeki.io) is co-founder of Ōzeki Inc.

application of computational resources to 'useful work' is finding solutions to problems of scientific interest such DNA alignment, protein folding, machine learning parameter searches.[8]

'Useful work' protocols can be grouped in two major types: *economy-based* and *mining-based*. The former has been proposed in works such as CureCoin and Coinami which use the interest in cryptocurrencies to reward users who complete certain tasks such as DNA alignments or protein folding directly with tokens.[2,3] However, for the most part the blockchain protocol remains untouched and PoW is still required in its entirety, typically with some additional centralization. On the other hand, *mining-based* approaches attempt to replace PoW with alternative forms of work. Proposed methods can vary widely in this category. Protocols such as PrimeCoin and Conquering Generals attempt to fully replace the classical PoW problem of block hashing to another NP-complete problem.[1,9] These problems are picked such that they preserve the following properties:

(1) Solving is difficult (hash functions are non-invertible)
(2) Difficulty can be easily tuned (target value of hash digest determines difficulty)
(3) Fast solution verification (computing and comparing hashes is constant time)
(4) Easy to generate new problems (block data defines a problem)

Very few problems of scientific interest can satisfy all of these criteria. However, in the present authors' 2017 "Proposal for a Fully Decentralized Blockchain and Proof-of-Work Algorithm for Solving NP-Complete Problems" (hereafter Oliver *et al.*) we observed that many scientifically relevant problems, such as protein folding, machine learning parameter searches, and DNA alignments can be phrased an NP-complete problems.[8] Since NP-complete problems preserve the property (3) there exist scientifically relevant problems that can be partially integrated into the standard PoW. Miners optionally submit a block at a reduced *hashing* difficulty if the block includes a valid solution to a problem selected by the network. If a solution to the problem is not submitted, the protocol behaves identically to Bitcoin. The difficulty of the problem is estimated by the frequency of blocks mined with solutions and difficulty is adjusted accordingly to maintain a desired block time. Recently, in their "Hybrid Mining: Exploiting Blockchain's Computational Power for Distributed Problem Solving" (2019) (hereafter Chatterjee *et al.*), K. Chatterjee, A. Goharshady, and A. Pourdamghani proposed a special case of the protocol described in Oliver *et al.*, which allows problem solvers to submit blocks without hashing (reduced difficulty of zero) if they provide a solution.[8,10] Attacks where malicious miners pre-solve many problems to win a large portion of consecutive blocks are prevented by enforcing that at least 50% of blocks be mined classically (effective reduced difficulty half of classical difficulty). Finally, in "Incentive-Based Integration of Useful Work into Blockchains" (2019) (hereafter Amar *et al.*), D. Amar and L. Zilpa reward solutions to 'useful' problems with votes in a hybrid Proof of Work/Proof of Stake model which, by definition, reduces the required amount of hashing.[11] However, this also comes with a complex additional layer of governance and potential centralization. The protocols discussed here have the additional advantage of reducing the benefit of ASIC hardware in mining, thus reducing the barrier of entry to mining.

1.2. Contribution—Here we build on Oliver *et al.* to provide stronger problem-solving incentives without greatly compromising security and decentralization.[8] We describe, in Section 2, a novel difficulty-adjustment scheme which provides these problem-solving incentives. The result of this difficulty-adjustment scheme is a simple soft-fork compatible modification to the core Bitcoin mining protocol. This altered mining protocol would allow the incentivization of many scientifically-relevant problems such as DNA alignment, protein folding, Ising-lattice minimization, and machine-learning optimization. In Section 3, we present simulations of the resulting network behaviour. In Section 4, we address potential attacks and emphasize how DIPS differs from the Bitcoin proof-of-work algorithm. Finally, conclusions are given in Section 5.

2. Protocol

Here we describe the DIPS protocol to incentivize the mining network to solve an agreed-upon optimization problem. Solving incentives come in the form of mining difficulty reductions. For simplicity, we phrase an optimization problem $\mathscr{P}$ as a sequence of NP-complete decision problems $\{p^1, .., p^k, ...\}$ of the form, *does there exist a solution with objective value greater than some fixed target?* As an example, $\mathscr{P}$ can be a specific graph for which we want to find the maximal clique. Since verifying that a clique is globally maximal is intractable, we let the network solve a series of decision problems. For example, at a given point of the blockchain the network would be trying to solve a p^k in the form, *does there exist a clique in this graph of size greater than k?* Solutions to such a decision problem can be easily checked since a miner would provide the nodes in the clique and the network does a lookup in an adjacency table. Once a solution of score k is found, the network would accept blocks only if they satisfy the regular bitcoin difficulty, or a reduced difficulty along with a clique of size $> k$. As k increases the network arrives at solutions that are closer to the global solution to $\mathscr{P}$.

2.1. Single update—In the first version of the protocol (Oliver *et al.*), miners are given the option of mining blocks 'classically' (as in the Bitcoin protocol) with a given difficulty d_b or by including solutions to a given $\mathscr{P}$ in a block and mining that block with a reduced difficulty d_r.[8, 12] Since there are two difficulties in Oliver *et al.*, two conditions are required to be satisfied to update the difficulties. This situation is in contrast to the Bitcoin protocol, where a single difficulty is updated using a single condition: the average time required to mine each block T is fixed (taken to be 10 minutes in Bitcoin). The first condition used to update the difficulty in Oliver *et al.* is the same as in the Bitcoin protocol. In addition to this condition, the difficulties are updated so that the average ratio between d_r and d_b, η is fixed,

$$\left\langle \frac{d_r}{d_b} \right\rangle = \eta, \tag{1}$$

where $\langle Q \rangle$ represents the long-time average of the quantity Q. Using Eq. (1) and the equation determining the update of d_b, an equation for the update of d_r can be derived (see Section II.C of Oliver *et al.*).[8] As is the case in Bitcoin, the difficulties are updated after N_1 blocks are mined ($N_1 = 2016$ in Bitcoin). In Oliver *et al.* both difficulties are updated independent of how many blocks are mined including a solution to $\mathscr{P}$ (see Figure 1). Moreover, as is the case with Bitcoin, measures are taken to ensure that difficulty does not change too quickly. In particular, the protocol enforces a maximal update factor for the difficulties so that

$$\frac{1}{x} \leq \frac{d_i^{j+1}}{d_i^j} \leq x, \tag{2}$$

where $i \in \{b, r\}$, d_i^j is the value for d_i after jN_1 blocks and x is the maximum factor by which the difficulties can be updated. After jN_1 blocks, if the difficulty d_i^{j+1} is calculated so that $x < d_i^{j+1}/d_i^j$ ($d_i^{j+1}/d_i^j < 1/x$) then we take $d_i^{j+1} = x d_i^j$ ($d_i^{j+1} = d_i^j/x$) instead.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2020.194

Although this version of the protocol incentivizes solving NP-complete problems by providing a reduced mining difficulty (assuming $\eta < 1$), d_r increases with d_b since the average value of d_r/d_b is fixed. Therefore, even in the case where no solutions are submitted, d_r can increase. This means that even as $\mathscr{P}$ becomes more difficult to solve (which is expected to happen over time), d_r can continue to increase, limiting the incentive provided by the network to solve problems and therefore the number of solutions that the network will find. Consequently, the amount of resources redirected to solving useful problems will also be limited.

2.2. Independent Updates—We propose a second version of the protocol (DIPS) where miners are again free to submit 'classical' blocks with difficulty d_b or blocks containing a solution to an NP-complete problem, with difficulty d_r. However, in DIPS the difficulties d_b and d_r are updated independently (see Figure 1). That is, after N_2^b classical blocks have been mined, d_b is updated so as to keep the average time spent by the network to mine a classical block fixed (to a predetermined value t_2^b). Similarly, after N_2^r blocks have been mined containing a solution, d_r is updated so as to keep the average time spent by the network to mine a block with a solution fixed (to a predetermined value t_2^r). Since it becomes increasingly difficult to find solutions to NP-complete problems, eventually the network might not be able to mine new blocks with the current difficulty d_r. Therefore in the DIPS protocol if N_2^b consecutive classical blocks are mined, d_r is decreased by the maximum factor, x ($d_r \to d_r/x$). In this way, even if the problem becomes increasingly difficult to solve, miners that attempt to solve the problem are incentivized with a proportionally decreased d_r. Miners are naturally discouraged from holding on to their solution until the difficulty is lowered by a large factor since this comes at the risk of other miners finding and publishing a better solution first.

2.3. Problem Submission—In the type of modified proof-of-work protocol described in the above subsections, some of the hashing power reserved in traditional blockchains (such as Bitcoin) to ensure the security of the network is diverted to solving an NP-complete problem, $\mathscr{P}$. The difficulty in finding better and better solutions provides the network with security and replaces the repeated hashing of proposed blocks. Because the best score of $\mathscr{P}$ will eventually saturate (either when the network no longer wishes to find a better score or when the optimal score has been found) this modified proof-of-work system will revert back to the traditional proof-of-work protocol. Therefore, to create a blockchain where NP-complete problems keep being solved, it might be interesting to consider a system where new problems can be submitted to the network. We note that Chatterjee *et al.* propose a convenient formulation of any NP-complete problem as a boolean satisfiability problem which can be used for convenient solution checking.[10]

The inclusion of new problems in the network can be done in a variety of ways. We leave determining a specific implementation for a future work and discuss some possible implementations here instead. One possible implementation could have a committee of users or a group of special nodes chosen (through an election, or otherwise) to submit new problems once the best score of the current problem has been saturated. Alternatively, but in the same spirit, individuals (participating in the network, or not) can propose new problems off-chain and the community can vote on which new problem to replace the current problem with. Once the new problem has been chosen, the blockchain can be hard forked to ignore solutions to the old (saturated problem) and accept solutions to the new agreed-upon problem. Specific strategies for selecting new problems are explored in depth in the work of Amar *et al.*, and forking as a software evolution system is studied in Andersen *et al.*[11,13]

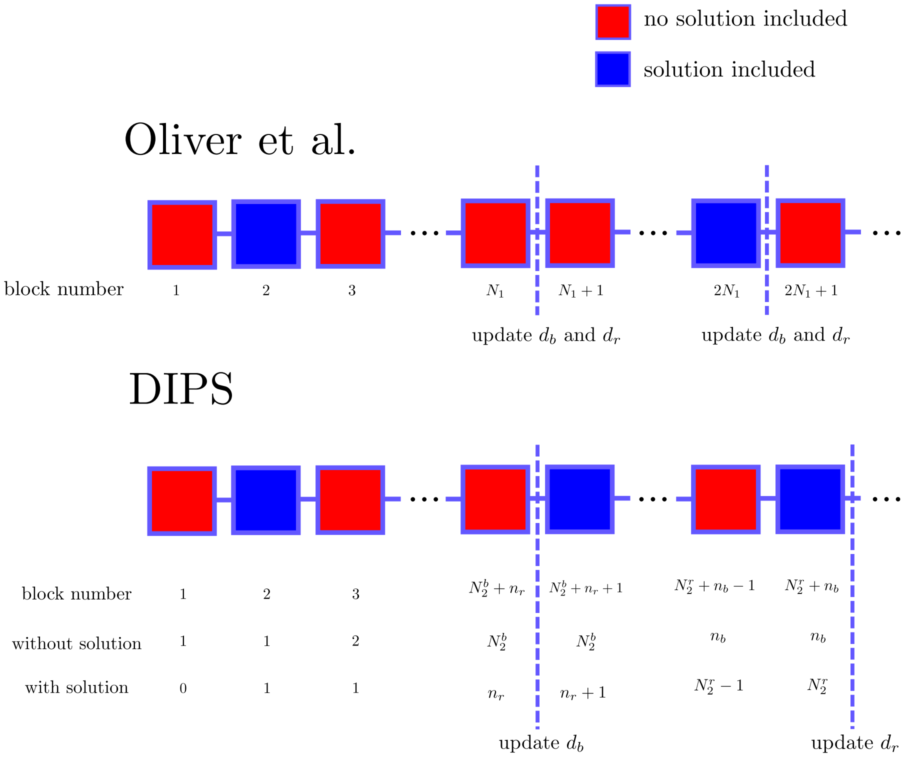


Fig. 1. Schematic depicting the difference between a blockchain with proof-of-work protocol Oliver *et al.* and DIPS. The red blocks are standard blocks that do not contain a solution to a problem (classical blocks), while the blue blocks contain a solution that results in a score that is better than the current best score of $\mathscr{P}$ (solution blocks). In Oliver *et al.* both d_b and d_r are updated every N_1 blocks. In DIPS, d_b is updated every N_2^b blocks that do not contain a solution to a problem and d_r is updated every N_2^r blocks that do contain a solution. n_r and n_b are arbitrary numbers.

3. Simulation Results

Here we perform simple experiments to visualize the impact of protocol and parameter choice on the behaviour of the network. We implement a simulated blockchain with 10 miners performing classical mining and 10 miners problem solving.

As a sample problem, we let the network solve the well-known maximum clique problem on randomly-generated graphs.[14] Given a graph $G = (V, E)$ which is a tuple of vertices (V) and edges $(E \in V \times V)$, and a clique size $k \in \mathbb{N}$, the NP-complete formulation of the problem is, *does there exist a clique of size at least k in G*? If V^k is the set of all sets of k vertices of G, a clique $Q^k \in V^k$ is a set of nodes such that $(u, v) \in E \quad \forall \quad (u, v) \in Q^k \times Q^k$, that is, Q^k is a set of k fully connected nodes. A brute force approach to solving this problem is to enumerate all possible k-subsets of V, and check that the condition is satisfied for each of the $2^{|V|}$ subsets. In fact, no other known algorithms performs substantially better than the brute force approach: the maximum clique problem has been shown to be NP-complete.[15] We can also see that checking a solution once a clique is proposed is fast, since it reduces to ensuring that all off-diagonal entries in the clique's adjacency matrix are 1. This problem is chosen for simplicity of implementation, but DNA alignment, protein folding and many other optimization problems can admit this formulation.[8] However, we note that this problem is already of great scientific interest as it has been used in many real-world applications such as DNA sequencing mapping and motif finding.[16,17] At the start of the chain, we generate a random graph and miners apply the Bronn-Kerbosch algorithm which enumerates cliques until a large enough solution is found.[18] The current largest clique size is stored in each block. Since mining hardware is typically used only for hashing, we assume miners are concurrently mining at d_b while attempting to solve the problem on separate hardware. If a miner finds a clique which beats the current best, he includes the solution in his next block and begins to mine at a reduced difficulty d_r. If he fails to win the next block, the miner keeps his current best solution to try again in the next block.

In Figure 2 we plot the number of classical and solution blocks as a function of the blockchain height. In a standard proof-of-work blockchain (such as Bitcoin), there is only one type of block—a classical block. The dashed line in Figure 2 represents how the number of classical blocks grows with blockchain height in a standard blockchain (they grow together since they represent the same quantity). In contrast, for the DIPS proof-of-work protocol, some of the hashing power is diverted to solving NP-complete problems. Therefore there are fewer fully classical blocks mined at a given blockchain height and therefore, less energy invested by the network to hash blocks. The energy saved is used to solve NP-complete problems as indicated by the blue curve, which represents blocks being mined by the network that contain solutions to NP-complete problems - solution blocks. At the time of writing, (April 5th 2020), the Bitcoin network hashrate is roughly $100\,\mathrm{EH/s}$.[19] If we assume a network energy consumption per hash rate of $\sim 0.1\,\mathrm{GW/(EH/s)}$,[20] then the power consumption of the Bitcoin network can be estimated at $10\,\mathrm{GW}$. The associated energy cost per block is on average $\sim 1.7\,\mathrm{GWh}$ (the average time to mine a Bitcoin block is $10\,\mathrm{min}$). The energy diverted to solving a scientifically relevant problem can therefore be estimated at $\sim (1 - d_r/d_b)1.7\,\mathrm{GWh}$ for every block mined with a solution.

A feature of optimization problems is that eventually the optimal solution to the problem will be found. At this point (or before) the number of blocks with solutions will saturate because no new solutions can be found. In this case, the system reverts to a standard proof-of-work

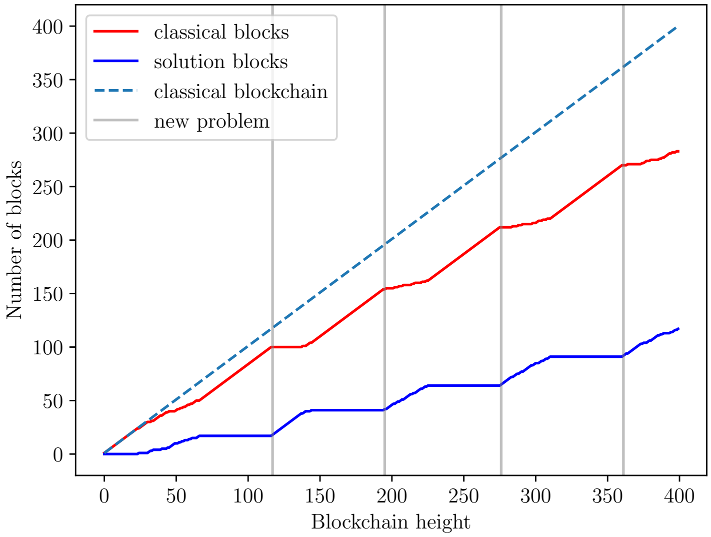
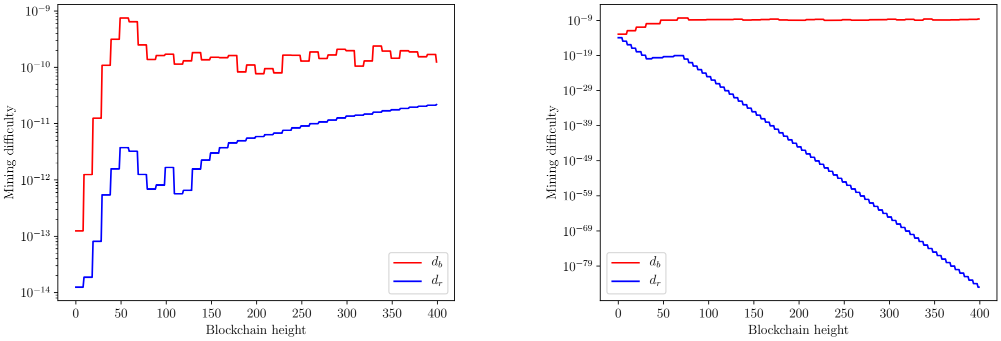


Fig. 2. Number of classical (red) and solution blocks (blue) mined as a function of the blockchain height using the simulations described in Section 3. The dashed line represents the number of classical blocks mined on a classical blockchain (such as Bitcoin) as a function of the blockchain height. We have also replaced the problem in the network by a new one once the best score saturates. The introduction of a new problem is represented by the gray vertical lines. We use protocol DIPS with $t_2^b = t_2^r = 0.1s$. The update frequency for solution blocks is set to $N_2^b = 10$ and $N_2^r = 5$ since at early stages of the problem, solutions are found quickly.

blockchain. One way to prevent the system from becoming a standard blockchain is to introduce new problems when the solution saturates (see Section 2.3 for a discussion on possible ways to include new problems). We have simulated the inclusion of a new problem (randomly-generated graph) when the best score of the previous problem saturates (gray vertical lines in Figure 2). In this way, instead of reaching a point where no more energy is diverted to solving NP-complete problems, as the blockchain grows, more energy is diverted to solving these types of problems. This energy is loosely represented by the difference in the dashed line and the red curve in Figure 2, or equivalently, the blue curve.

We have also studied how this energy is affected by the parameter η. For the Oliver *et al.* protocol, η is a parameter that is enforced by the network as the difficulties get updated (see Section 2.1). In the case of the DIPS protocol, η indicates the initial value of d_r/d_b. As expected, since η is not enforced by the network, the fraction of blocks that are solution blocks is independent of η for DIPS (see Figure 4). In contrast, in Oliver *et al.*, η determines how much incentive the community is given to solve the blockchain's optimization problem. As blockchains are initialized with smaller values of η (as you move to the right along the horizontal axis of Figure 4) we expect the fraction of solution blocks to increase as is shown in Figure 4. Furthermore, because the network enforces that the ratio d_r/d_b (long-time) averages to η, d_b and d_r vary together (see Figure 3a).

After a certain time, as the problem becomes increasingly more difficult to solve, d_r will not



(a) Oliver *et al.*

(b) DIPS

Fig. 3. Difficulties d_r (blue) and d_b (red) as a function of block height for a typical run of the simulation described in Section 3. For Oliver *et al.* we set $\eta = 200^{-1}$, and remaining parameters are kept from Figure 2

decrease (and might even increase if the hashrate of the network increases) if the ratio d_r/d_b has reached its equilibrium value η. Therefore in Oliver *et al.* a situation is eventually reached where the problem difficulty increases and d_r remains fixed, *i.e.* as the problem becomes more difficult, the hashing difficulty remains constant, increasing the total difficulty of mining a solution block. This type of situation does not encourage nodes to solve problems. Alternatively, in DIPS, if no solutions are found, d_r decreases (see Section 2.2 and Figure 3b). Therefore, as the problem becomes more difficult and new solutions become harder to find, the incentive for finding new solutions increases. These aspects of the protocol are reflected in Figure 4. We find that for DIPS the best score always saturates, while for Oliver *et al.* the saturation occurs only if η is small enough. As simulations are run with smaller values for η, the fraction of blocks that are solution blocks in Oliver *et al.* tends to the saturated value (average DIPS value, given by the dashed blue line).

4. Potential Attacks

Besides the standard attacks possible with traditional proof-of-work blockchains (*e.g.* 51% attacks performed by controlling a majority of the network's mining power), the DIPS proof of work introduces additional attack avenues, namely stemming from the ability to reuse solutions. We name the main (to our knowledge) attack for this version of proof of work the *Bubka attack* after Ukrainian pole vaulter Sergey Bubka who obtained many consecutive world records by incrementally improving his score instead of posting one world record by a large margin. This attack strategy involves finding multiple successive solutions to $\mathscr{P}$ and using these to get an advantage in hashing for multiple blocks in a row. Worse still, one could copy the solutions to problems from solution blocks and use them to fork the chain at a different block height. Although the solutions cannot be used to mine blocks for free (the attacker would still have to perform classical mining, albeit at a reduced difficulty d_r), this strategy would allow the attacker to double spend by forking the network at some past block and creating the longest chain with less than 51% of the network's hashing power. Estimating the exact fraction of hashing power

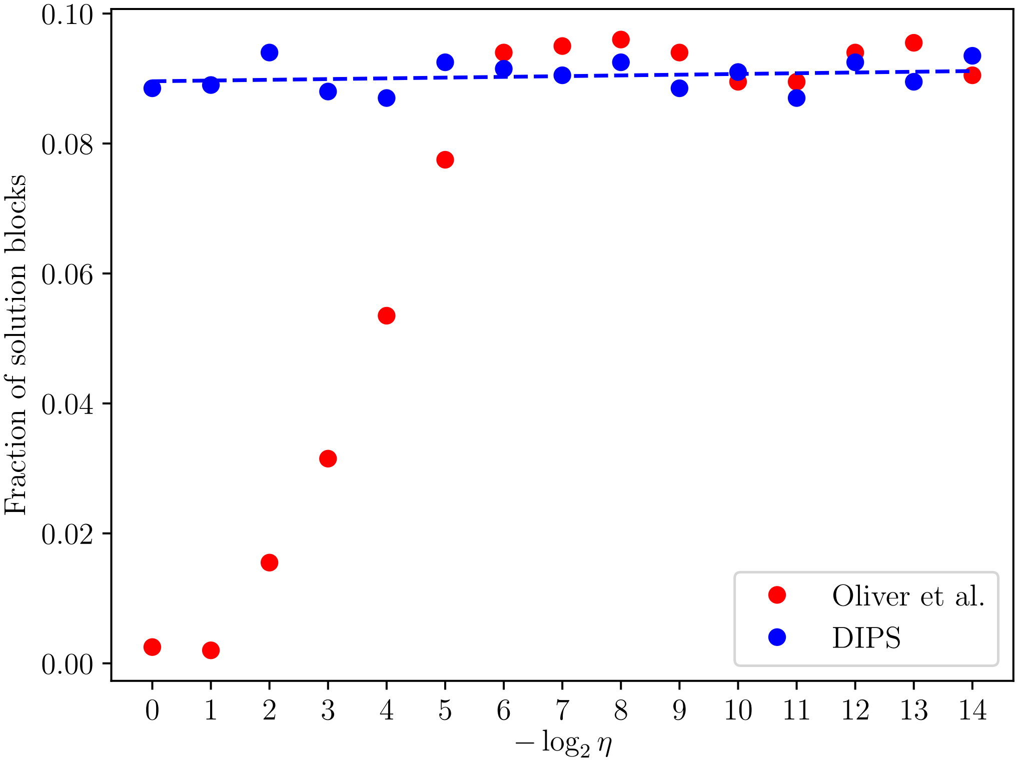


Fig. 4. Fraction of blocks mined with a solution after 200 total blocks for the proof-of-work protocol Oliver *et al.* (red) and DIPS (blue) as a function of η. In Oliver *et al.*, η is enforced by the network, while in DIPS it is the initial ratio d_r/d_b. The fraction of blocks with solutions does not depend on the value of η for DIPS. The blue dashed line is the average value of the fraction of blocks with solutions over values of η. For Oliver *et al.* the fraction of solution blocks converges to the DIPS value (dashed blue line) in the limit where $\eta \to 0$. Each point represents an average of 10 independent blockchain instances of height 200. Parameters are chosen as in Figure 1.

required for a double-spend attack in this case is not straightforward and depends on, among other factors, the ratio d_r/d_b at the time of the attack, the rate at which the network finds new solutions to the problem, the hash rate of the network, and the values of the parameters N_2^r and N_2^b. We leave the exploration of this question for future work.

Other attacks such as well-connected miners claiming other miners solutions as their own are addressed in Amar *et al.* with protocols that are fully compatible with DIPS.[11]

There are several options for addressing this threat on the social layer of the network. Unlike in other proposed methods, mining a solution to a block still requires a hashing step, the network could therefore choose a large enough update frequency (small enough value of N_2^r) to rapidly increase mining difficulty if solutions are posted in rapid succession. Additionally, users could follow higher transaction 'confirmation times' or not allow long consecutive chains of blocks with solutions to the problem. Such heuristic solutions are currently common usage in blockchains such as Bitcoin, where a 6 block confirmation time is socially enforced to safeguard against 51% attacks. We also note that the nature of NP-complete problems is that there is strong evidence that no algorithms better than exponential time exist, leveling the playing field across the network. This assumption is the same one that is made by Bitcoin whereby the hashing problem is assumed to be unsolvable in sub-exponential time by anyone. Interestingly, recent integrations of zk-SNARKs in mainstream blockchains raise potential methods for temporarily hiding solutions.[21]

5. Conclusion

In this article we have compared two modified proof-of-work protocols that divert energy from mining by hashing blocks to solving NP-complete problems (useful work). We have studied, through simulations, the dependence on the total diverted energy on different parameters.

Although this article provides a protocol that can convert the energy used by miners to a form of useful work (solving NP-complete problems), creating a fully functioning and useful blockchain using any one of the two protocols (Oliver *et al.* or DIPS) would require the understanding of other aspects of system. For example, a concrete protocol to add new problems to the network must be developed. Moreover, a standard way of storing and solving submitted problems must be established. Other works have made some effort to solve these problems,[10,11] however more work is needed to adapt these solutions for use in the proof-of-work protocols Oliver *et al.* and DIPS. We believe protocols such as DIPS are an important step towards combining the potential of crowd-sourcing initiatives (such as Folding@Home and Phylo which have resulted in new solutions for important problems)[4,5] with the strong incentive structures native to blockchains.

Author Contributions

P.P and C.G.O conceived the research, prepared the manuscript and performed the simulations. A.R. helped conceive the simulation protocol.

Notes and References

[1] King, S. "Primecoin: Cryptocurrency with prime number proof-of-work." *primecoin.io* (accessed 18 July 2020) `https://primecoin.io/bin/primecoin-paper.pdf`.

[2] No Author. “CureCoin Whitepaper.” *FoldingCoin* (2015) (accessed 18 July 2020) `http://foldingcoin.net/the-coin/white-paper/`.

[3] Ileri, A. M., Ozercan, H. I., Gundogdu, A., Senol, A. K., Ozkaya, M. Y., Alkan, C. “Coinami: A Cryptocurrency with DNA Sequence Alignment as Proof-of-work.” *arXiv* (2016) (accessed 18 July 2020) `https://arxiv.org/abs/1602.03031`.

[4] Kawrykow, A., *et al.* “Phylo: A Citizen Science Approach for Improving Multiple Sequence Alignment.” *PloS one* **7.3** e31362 (2012) `https://doi.org/10.1371/journal.pone.0031362`.

[5] Larson, S. M., Snow, C. D., Shirts, M., Pande, V. S. “Folding@Home and Genome@Home: Using Distributed Computing to Tackle Previously Intractable Problems in Computational Biology.” *arXiv* (2009) (18 July 2020) `https://arxiv.org/abs/0901.0866`.

[6] Anderson, D. P., Cobb, J., Korpela, E., Lebofsky, M., Werthimer, D. “SETI@home: An Experiment in Public-Resource Computing.” *Communications of the ACM* **45.11** 56–61 (2002) `https://doi.org/10.1145/581571.581573`.

[7] Ball, M., Rosen, A., Sabin, M., Vasudevan, P. N. “Proofs of Useful Work.” *IACR Cryptology ePrint Archive* **2017** 203 (2017) `https://eprint.iacr.org/2017/203.pdf`.

[8] Oliver, C. G., Ricottone, A., Philippopoulos, P. “Proposal for a Fully Decentralized Blockchain and Proof-of-Work Algorithm for Solving NP-Complete Problems.” *arXiv* (2017) (accessed 18 July 2020) `https://arxiv.org/abs/1708.09419`.

[9] Loe, A. F., Quaglia, E. A. “Conquering Generals: an NP-Hard Proof of Useful Work.” In *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems* ACM 54–59 (2018) `https://doi.org/10.1145/3211933.3211943`.

[10] Chatterjee, K., Goharshady, A. K., Pourdamghani, A. “Hybrid Mining: Exploiting Blockchain’s Computational Power for Distributed Problem Solving.” In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* ACM 374–381 (2019) `https://doi.org/10.1145/3297280.3297319`.

[11] Amar, D., Zilpa, L. “Incentive-Based Integration of Useful Work into Blockchains.” *arXiv* (2019) (accessed 18 July 2020) `https://arxiv.org/abs/1901.03375`.

[12] Nakamoto, S. “Bitcoin: A Peer-to-Peer Electronic Cash System.” (2008) (accessed 18 July 2020) `https://bitcoin.org/bitcoin.pdf`.

[13] Andersen, J. V., Bogusz, C. I. “Patterns of Self-Organising in the Bitcoin Online Community: Code Forking as Organising in Digital Infrastructure.” In *Thirty Eighth International Conference on Information Systems, Seoul 2017* (2017) `https://aisel.aisnet.org/icis2017/DigitalPlatforms/Presentations/4/`.

[14] Tomita, E., Tanaka, A., Takahashi, H. “The Worst-Case Time Complexity for Generating All Maximal Cliques and Computational Experiments.” *Theoretical Computer Science* **363.1** 28–42 (2006) `https://doi.org/10.1016/j.tcs.2006.06.015`.

[15] Karp, R. M. “Reducibility Among Combinatorial Problems.” In R. E. Miller, J. W. Thatcher, J. D. Bohlinger (Eds.), *Complexity of Computer Computations* Springer 85–103 (1972) `https://doi.org/10.1007/978-1-4684-2001-2_9`.

[16] Li, J., Zimmerman, L. J., Park, B.-H., Tabb, D. L., Liebler, D. C., Zhang, B. “Network-Assisted Protein Identification and Data Interpretation in Shotgun Proteomics.” *Molecular Systems Biology* **5.1** `https://doi.org/10.1038/msb.2009.54`.

[17] Huang, C.-W., Lee, W.-S., Hsieh, S.-Y. “An Improved Heuristic Algorithm for Finding Motif Signals in DNA Sequences.” *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **8.4** 959–975 (2010) `https://doi.org/10.1109/TCBB.2010.92`.

[18] Bron, C., Kerbosch, J. “Algorithm 457: Finding All Cliques of an Undirected Graph.” *Communications of the ACM* **16.9** 575–577 (1973) `https://doi.org/10.1145/362342.362367`.

[19] No Author. “Total Hashrate.” *Blockchain.com* (accessed 18 July 2020) `https://www.blockchain.com/charts/hash-rate`.

[20] No Author. "Bitcoin Energy Consumption Index." *Digiconomist* (accessed 18 July 2020) https://digiconomist.net/bitcoin-energy-consumption.

[21] Xu, L., *et al.* "Enabling the Sharing Economy: Privacy Respecting Contract Based on Public Blockchain." In S. Lokam, S. Ruj, K. Sakurai (Eds.), *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts* 15–21 (2017) https://doi.org/10.1145/3055518.3055527.

Ledger is published by the University Library System of the University of Pittsburgh as part of its D-Scribe Digital Publishing Program and is cosponsored by the University of Pittsburgh Press.