LEDGER
ledgerjournal.org

ISSN 2379-5980 (online)
associated article DOI 10.5915/LEDGER.2020.194

Difficulty Scaling in Proof of Work for Decentralized Problem Solving: Open Review

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

Reviewers: Reviewer A, Reviewer B, Reviewer C

Abstract. The final version of the paper "Difficulty Scaling in Proof of Work for Decentralized Problem Solving" can be found in Ledger Vol. 5 (2020) 62-73, DOI 10.5915/LEDGER.2020.194. There were three reviewers involved in the review process, none of whom have requested to waive their anonymity at present, and are thus listed as Reviewers A through C. After initial review by Reviewers A, B, and C, the submission was returned to the authors with feedback for revision (1A). The authors responded (1B) and resubmitted their work. After further review, Reviewer C had further reservations (2A) which the authors responded to (2B). Reviewer C agreed that their concerns had been adequately resolved, thus ending the peer review process. Author responses have been lightly edited for reader clarity.

1A. Review

Reviewer A

This paper describes two protocol modifications of the Bitcoin Proof-of-Work mechanism aiming to divert some energy spent on hash computation to finding solutions to computational problems that are useful outside of the blockchain network.

This is a solid paper presenting good research in an important field. Below are some comments and ideas for improvement.

In Section 4, as a countermeasure to potential attacks higher confirmation time is recommended. Choosing the right confirmation time can be already problematic and finding a

* 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.

good balance between wait time and security is not always straightforward. Complicating this issue further (especially when it is not known that some miners may have better algorithms for doing useful work) may be problematic. The authors may want to expand on this point.

As the aim of this work is to find ways to divert energy spent on "useless" hashing to solving useful problems, it would be interesting to discuss how much energy could be saved and how much useful computations could be performed with the energy, based on the simulation results.

The main issue with this line of work is the fact that few useful computational problems have the properties necessary for Proof-of-Work. Therefore, while the arguments and simulations presented in the paper are of academic value, it is hard to imagine it being practical. Of course one paper cannot solve all issues and this limitation should not reduce the value of the contribution of this paper.

Reviewer B

The paper presents a dual-system proof of work proposal which incorporates useful problems-to-be-solved into the proof of work algorithm of a blockchain.

To the Authors: please see attached file for reference to placement of the following comments throughout your paper. I would also suggest having the paper review by other peers in order to gain the most value for your contribution. Thank you for the opportunity to review.

Comments:

(1) Since mining converges to be only marginally profitable for people who can find favorable costs to purchase resources necessary to min, it will never be profitable for people who hash something else at the same time. However if you’re going to do the work anyway, it might be a nice kickback.

(2) How is the proof verified? Is it able to be done in a decentralized way?

(3) How is the problem determined? Voting? Etc? Also, this comment is unclear. Does it converge back to the traditional PoW function that Bitcoin uses?

(4) I suspect there is still some gamble attack vector here. Should be included in future follow-up research. (also, not a complete sentence)

(5) dr must have a known difficulty function for this to work. ie. Bitcoin’s proof of work is linear in difficulty adjustments. dr must have some known linear (or otherwise) function as well.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

(6) Interesting to look at how this plays into the price function. If NP problems would have been done anyway and don't need to sell what they mine to account for mining costs, what does that do to price? Potential future research.

(7) How do you address network governance? Who is determining this?…

(8) This fully dilutes any security gained by hash power accumulation and introduces centralized security holes.

(9) Hard forks should be avoided at almost all costs

(10) Suggestion to bring Fig. 2 forward to be included close to first reference

(11) I am concerned that the nature of NP complete problems is such that they necessarily get harder over time. I apologize for my unfamiliarity with these types of computational problems but if this is the case and there's no way to reduce their difficulty then the difficulty adjustment algorithm won't work. Am I thinking about this incorrectly?

(12) How will the network definitively know when this point is reached? It seems that for some bit of time after a problem reaches optimization that the efficiency of the difficulty function is off balance…

(13) The references to figures and their positioning in the paper is very difficult to follow

(14) Clever. I much like this naming convention

(15) Another attack vector would be for a traditional miner to use an NP problem to gain multiple blocks in a row, forking the network. NP blocks are not as secure as traditional.

Reviewer C

It presents a new difficult adjustment algorithm that retargets the traditional hash-based proof-of-work independently from the scientifically-interesting proof-of-work.

This paper is about modifying traditional hash-based proof-of-work so that miners can optional perform work on problems of scientific interest. I am skeptical about the value of such schemes for a few reasons: (1) the problems of interest need to change from time to time and fairly introducing new problems is a significant problem in itself, (2) the class of problems that have a form suitable for introduction into PoW seems quite narrow to me and this paper does not convince me otherwise, (3) since the solutions to the scientifically interesting problems do NOT depend on the previous block header, the solutions can be recycled to 51% attack the coin.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

Overall, I think the paper is interesting and of publishable quality if the authors tone down their claims (more below) and are more honest with the significant problems that remain unsolved.

1. INTRODUCTION

The authors' write:

"Here we build on Oliver et al., [9] to provide stronger problem-solving incentives while maintaining simplicity, full decentralization, and blockchain security. We describe in Section 2 a novel difficulty adjustment scheme which provides stronger problem solving incentives. In Section 3 we present simulations of the resulting network behaviour. Finally, in Section 4 we address potential attacks."

This significantly overstates the authors' contribution. They have proposed a novel difficulty adjustment algorithm that allows the difficulty of the scientifically-interesting problem to be adjusted independently of the traditional hash-based proof-of-work problem and they performed some simulations to explore their proposal.

The claim that their method provides "full decentralization" is unsupported at best, and false at worse. Such a PoW scheme is only useful if new problems can be introduced when the last problem has been exhausted. Fairly introducing new problems will be an attack vector. Who will pick the next problem? How will we know ahead of time that it is a scientifically-important problem and not a problem that some group just so happens to be really good at solving? The authors leave _this_ problem for future study, but this is a BIG problem. The claim of "full decentralization" must be removed.

Further, they also haven't demonstrated "blockchain security" as they claim in the introduction. In fact, in Section 4 they write:

"Worse still, one could copy the solutions to problems from solution blocks and use them to fork the chain at a different block height. 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."

Exactly! This is my point about the solutions not depending on the info in the previous block. This is RADICALLY different than bitcoin PoW where a solution is ONLY valid for the block upon which it was built. The authors recognise this potential problem, yet defer it to future work (which is OK) but then you cannot ALSO make the claim that your solution is secure!

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

Additional comments:

2. PROTOCOL

I'd like to see a simple example of how the solution to the decision problems can be checked in constant time, perhaps as an appendix. Maybe give more background on the maximal clique problem too, since this is what you use as an example. How are the larger cliques found and how can one quickly verify that a clique is valid and of a given size?

More generally, how many scientifically important problems can be expressed in a form suitable for inclusion in PoW. It's not at all obvious to me as a reader than many problems that are actually useful will have the necessary form, like the maximum clique problem does.

2.1. SINGLE UPDATE

It wasn't clear to me at first that "v1" of the protocol is from reference [9], and that the authors' contribution is "v2" described in Section 2.2. What about calling v1 the "Oliver et al" method and calling v2 the "method proposed in this paper"?

Also, I find the mathematical formalism here distracting. What you're saying is that the ratio of the two difficulty targets increases in lock step, right? And further that for each difficulty retargetting, there is a */ limit of, e.g. 4, like there is for bitcoin.

Lastly, I don't understand the average-value notation in Eq. 1. If both difficulty targets change in lock step, then the ratio between the two targets at EVERY point in time is constant.

2.2 INDEPENDENT UPDATES

This is the author's proposed difficult retargeting method. It adjusts the difficulty of solving the hash-based PoW problem independently from solving the scientifically-interesting problem.

It's not clear to me precisely how it works. Are you looking at the average solve time for the scientifically-interesting problems ONLY when retargeting d_r? And then only looking at the average solve time for the traditional blocks when retargeting d_b? More formalism here would help. What about include illustrating the solve times on your diagram to make this more clear?

2.3. PROBLEM SUBMISSION

I appreciate the authors addressing the obvious challenge. I find the last paragraph speculative though (and the second-to-last somewhat speculative too). Consider removing this content

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

rather than discussing half backed ideas. Refer also to my comments about how leaving problem submission unsolved means that the entire idea is still unsolved.

3.0 SIMULATION RESULTS

As eta decreases —> confusing because eta doesn't get decreased by the retargeting algorithm. I think you mean as you re-run the simulation with smaller and smaller values of eta, right?

// end of comments

1B. Author Responses

I. MAJOR COMMENTS

Here we briefly synthesize and paraphrase what we understood as the major concerns raised by the reviewers and indicate the relevant changes we hope will be satisfactory. We hope this will better contextualize our responses to the individual reviewers' comments.

A. Blockchain Security

The reviewers raise concerns with the fact that solutions to NP-complete problems can be re-used to fork the chain and launch potential double spend attacks. We agree that this is a vulnerability and it is discussed briefly in the manuscript, but we agree that it can be expanded upon. We have included a deeper discussion of the impact our protocol has on the 51% attack.

B. Blockchain Governance & Problem Selection

The second major point is the question of deciding which problems will be solved by the network. We believe this issue is the focus of other recent publications (e.g. Amar et al., Ref. [11]) and our work is compatible with any choice of network consensus protocol.

II. INDIVIDUAL COMMENTS

Reviewer A, comment 1:

"In Section 4, as a countermeasure to potential attacks higher confirmation time is recommended. Choosing the right confirmation time can be already problematic and finding a good balance between wait time and security is not always straightforward. Complicating this issue further (especially when it is not known that some miners may have better algorithms for doing useful work) may be problematic. The authors may want to expand on this point."

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

Reply:

We agree that choosing a specific update frequency that is best suited for the proposed mining scheme is a highly non-trivial problem. Different block confirmation times may be adequate in specific contexts. Relating the context the optimal confirmation time is already a difficult problem for Bitcoin. The suggested six-block confirmation time is, to the best of our knowledge, enforced as general social convention and is subject to change. For this reason, we are unable to delve further into this issue here.

We believe it is safe to assume that no miner will have a significantly better algorithm for solving a problem than the rest of the network. We base this assumption on the conjecture that NP-complete problems will always admit at best a non-polynomial running time. This assumption is equivalent to the assumption made about the Bitcoin hash function being non-invertible in polynomial time. Therefore NP-complete problems are a natural option to partially replace hashing because they are difficult to solve (hash cannot be inverted), but the solutions are easily verified (hash of any input is easily computed). Furthermore, a disparity in the solving rate between miners is fully within the usual Proof-of-Work scheme as one could imagine having a more efficient solving algorithm than the network is equivalent to using more CPUs with a less efficient algorithm.

See: P.8 end of last paragraph

Reviewer A, comment 2:

"As the aim of this work is to find ways to divert energy spent on "useless" hashing to solving useful problems, it would be interesting to discuss how much energy could be saved and how much useful computations could be performed with the energy, based on the simulation results."

Reply:

Indeed, this is an important feature of our work. Of course, obtaining a concrete value of the energy re-purposed would be highly speculative. However, to get a rough idea we can expand on Figure 2 and calculate the potential savings on a blockchain of the scale of Bitcoin.

Reviewer A, comment 3:

"The main issue with this line of work is the fact that few useful computational problems have the properties necessary for Proof-of-Work. Therefore, while the arguments and simulations presented in the paper are of academic value, it is hard to imagine it being practical. Of course one paper cannot solve all issues and this limitation should not reduce the value of the contribution of this paper."

Reply:

We agree that it was not clear to what extent our protocol is amenable to a diverse set of problems. The reviewer is correct in stating that not all NP-complete problems can be used as

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

the core Proof of Work (see list in introduction). The main criterion that is difficult to meet is that the problem instance be dependent on the current state of the chain. However, by introducing a second difficulty term, we are able to accept a large class of problems (as long as they still follow the NP-complete property of fast solution checking) as partial proof of work. This opens the door to many other useful problems such as protein folding, prime finding, machine learning model optimization, etc. We have clarified this point in the revised version.

See section 1.2

Reviewer B, comment 1:

"Since mining converges to be only marginally profitable for people who can find favorable costs to purchase resources necessary to mine, it will never be profitable for people who hash something else at the same time. However if you're going to do the work anyway, it might be a nice kickback."

Reply:

Indeed, there is typically a strong financial barrier to mining profitably. However, we believe that introducing a larger set of potential mining problems would reduce this barrier since it makes the use of hard-coded ASIC-style miners less beneficial.

See 1.1 paragraph 3

Reviewer B, comment 2:

"How is the proof verified? Is it able to be done in a decentralized way?"

Reply:

Since the problems used would be NP-Complete (e.g. clique finding) checking solutions is efficient and can be implemented in an analogous manner to Bitcoin block hash checking, i.e. it can be part of the block-validation protocol. Efficient encodings for problems on blockchains have been proposed in Chatterjee et al.

See Section 3, paragraph 2 for a more detailed explanation of the solution checking.

Reviewer B, comment 3:

"How is the problem determined? Voting? Etc? Also, this comment is unclear. Does it converge back to the traditional PoW function that Bitcoin uses?"

Reply:

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

The focus of this work is on the mining and difficulty scaling aspect. Selecting the problem the to work on is another question which has been addressed in more detail in Amar et al. Indeed, our protocol is strictly a generalization of the Bitcoin protocol thus if there are never any solutions, the difficulty is scaled in the same manner as Bitcoin. We have clarified this point.

See Final sentence of section 2.3

Reviewer B, comment 4:

"I suspect there is still some gamble attack vector here. Should be included in future follow-up research. (also, not a complete sentence)"

Reply:

We agree that there could be other attack vectors to the protocol proposed in Ref. 10, however we are unable to speculate further on their work.

Reviewer B, comment 5:

"dr must have a known difficulty function for this to work. ie. Bitcoin's proof of work is linear in difficulty adjustments. dr must have some known linear (or otherwise) function as well."

Reply:

dr does indeed have a known difficulty function. This function is written explicitly in Ref. 9. See Paragraph 2 in Section 2.1

Reviewer B, comment 6:

"Interesting to look at how this plays into the price function. If NP problems would have been done anyway and don't need to sell what they mine to account for mining costs, what does that do to price? Potential future research."

Reply:

Ultimately price will be determined by supply and demand of the coins. Both supply and demand are determined at a social level and are influenced by many factors.

Reviewer B, comment 7:

"How do you address network governance? Who is determining this?. . . "

Reply:

Section 2.3 is meant to give some ideas about including new problems to be solved into the blockchain and is not meant to give the impression that we have determined the best way for

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

this to be done. In fact, we state: `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.' We agree that some proposed implementations will affect network governance, but a discussion of this topic is beyond the scope of this work.

See End of section 2.3

Reviewer B, comment 8:

"This fully dilutes any security gained by hash power accumulation and introduces centralized security holes."

Reply:

Indeed, such a scenario would imply a degree of centralization which we have emphasized in the text. However, this remark was intended as a discussion point, hence the exact implications of such a system are left for future work.

Reviewer B, comment 9:

"Hard forks should be avoided at almost all costs."

Reply:

While hard-forking is one of many potential problem-selection strategies, it has the benefit of not requiring a complex blockchain protocol. This point also lives in a social layer of the network but it has been seen in many networks that hard forks are a necessary component of the evolution of a blockchain (bitcoin scalability, etc.). We include a reference which studies the self-organizing behaviour induced by the Bitcoin community and forking.

See p.5 paragraph 1

Reviewer B, comment 10:

"Suggestion to bring Fig. 2 forward to be included close to first reference"

Reply:

We agree that Fig. 2 should be closer to the position where it is first referenced.

Reviewer B, comment 11:

"I am concerned that the nature of NP complete problems is such that they necessarily get harder over time. I apologize for my unfamiliarity with these types of computational problems but if this is the case and there's no way to reduce their difficulty then the difficulty adjustment algorithm won't work. Am I thinking about this incorrectly?"

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

Reply:

Generally, every time a solution to the problem P is found, it becomes harder to find a better solution. Although there is in fact no way to reduce this difficulty in this protocol, we have proposed a system where the difficulty related to mining can be readjusted. In this system, as the difficulty to solve the problem increases, we expect the difficulty related to mining, dr to decrease. If the sum of the difficulty of the problem and dr is less than the difficulty to mine a block classically, db, then the optimal mining strategy will involve providing a better solution to problem P.

Reviewer B, comment 12:

"How will the network definitively know when this point is reached? It seems that for some bit of time after a problem reaches optimization that the efficiency of the difficulty function is off balance. . . "

Reply:

This is correct, knowing whether a certain solution to an NP-complete problem is globally optimal is often intractable and hence there is no efficient way of knowing. The difficulty scaling itself can handle this since in the absence of solution blocks, the system behaves identically to Bitcoin. To improve useful work, there would need to be a social layer for accepting novel problems into the network which brings in the standard governance procedures other blockchains deal with. While this is an interesting point, it is not the main focus of the work, and there are many options for potential solutions. We have included a related citation to Section 2.3

Reviewer B, comment 13:

"The references to figures and their positioning in the paper is very difficult to follow "

Reply:

We have attempted to reposition the figures to make it easier for the reader.

Reviewer B, comment 14:

Clever. I much like this naming convention

Reply:

Thank you.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

Reviewer B, comment 15:

"Another attack vector would be for a traditional miner to use an NP problem to gain multiple blocks in a row, forking the network. NP blocks are not as secure as traditional."

Reply:

This is indeed a potential attack that we mentioned as the `Bubka' vulnerability. We will clarify that link in the text. Indeed, the network would have to adopt some external social practices, such as the one used in Bitcoin of waiting 6 confirmations for each block. We have added a paragraph at the end of section 4 emphasizing this point.

Reviewer C, comment 1:

"This significantly overstates the authors' contribution. They have proposed a novel difficulty adjustment algorithm that allows the difficulty of the scientifically-interesting problem to be adjusted independently of the traditional hash-based proof-of-work problem and they performed some simulations to explore their proposal.

The claim that their method provides "full decentralization" is unsupported at best, and false at worse. Such a PoW scheme is only useful if new problems can be introduced when the last problem has been exhausted. Fairly introducing new problems will be an attack vector. Who will pick the next problem? How will we know ahead of time that it is a scientifically-important problem and not a problem that some group just so happens to be really good at solving? The authors leave this problem for future study, but this is a BIG problem. The claim of "full decentralization" must be removed.

"Worse still, one could copy the solutions to problems from solution blocks and use them to fork the chain at a different block height. 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."

Exactly! This is my point about the solutions not depending on the info in the previous block. This is RADICALLY different than bitcoin PoW where a solution is ONLY valid for the block upon which it was built. The authors recognise this potential problem, yet defer it to future work (which is OK) but then you cannot ALSO make the claim that your solution is secure!"

Reply:

While our protocol does not grant special permissions to any users, we agree that, as in Bitcoin, it is vulnerable to centralization as a byproduct of mining. We agree that the introduction of new problems to be solved into the blockchain is a delicate issue. Although we have provided potential mechanisms for addressing this issue, we agree that it is by no means `solved'. However, just like the decision on the size of the Bitcoin blocks is decided on a social layer, we believe that including new problems (and checking that they are valid) can be done in a similar manner. In particular, in the case where the network decides on which problems to include, the only requirements is that the newly included problem be NP-

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

complete. If this requirement is fulfilled (and we assume the complexity class of P 6= NP) then there is no algorithm that offers an exponential speedup in solving the newly introduced problem. Therefore, if the network can verify whether a proposed problem is NP-complete, then no adversary can have an exponential advantage in generating new solutions over the rest of the network.

Nevertheless, we agree with the referee that our language is slightly `strong'.

We have relaxed the wording in Section 1.2, and expanded the discussion of attack vectors in Section 4.

Reviewer C, comment 2:

"I'd like to see a simple example of how the solution to the decision problems can be checked in constant time, perhaps as an appendix. Maybe give more background on the maximal clique problem too, since this is what you use as an example. How are the larger cliques found and how can one quickly verify that a clique is valid and of a given size?

More generally, how many scientifically important problems can be expressed in a form suitable for inclusion in PoW. It's not at all obvious to me as a reader than many problems that are actually useful will have the necessary form, like the maximum clique problem does."

Reply:

We have included a formal description of the maximum clique problem with intuitions on how to solve and check it. It is also worth noting that all instances of NP-Complete problems are reducible to each other, and are typically encoded as boolean satisfiability problems, Chatterjee et.al 2019 propose a convenient encoding and checking of the problem.

See Section 1.1 Paragraph 2, Section 2.3 Paragraph 1, Section 3 Paragraph2

Reviewer C, comment 3:

“It wasn't clear to me at first that "v1" of the protocol is from reference [9], and that the authors' contribution is "v2" described in Section 2.2. What about calling v1 the "Oliver et al" method and calling v2 the "method proposed in this paper"?”

Reply:

We agree with this observation and have amended the manuscript accordingly. We refer to `v1' as the "Oliver et al" method and `v2' as the `DIPS' method.

Reviewer C, comment 4:

*“Also, I find the mathematical formalism here distracting. What you're saying is that the ratio of the two difficulty targets increases in lock step, right? And further that for each difficulty retargetting, there is a */ limit of, e.g. 4, like there is for bitcoin.”*

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

Reply:

We apologize for any confusion. While the figures may suggest that the two difficulties are scaled in lock step, this is not necessarily what is implied by the update formulas. The difficulties are meant to scale in proportion to the ratio between the number of solutions and classical blocks observed over the update window. This was explained in more detail in Ref. [9]. We have made some clarification after Eq. (1).

Indeed, there is an upper an a lower bound to the change factor as with bitcoin. See sentence after Eq. (2).

Reviewer C, comment 5:

"Lastly, I don't understand the average-value notation in Eq. 1. If both difficulty targets change in lock step, then the ratio between the two targets at EVERY point in time is constant."

Reply:

The difficulties do not change in lock step (see previous point) since the difficulties are a function of quantities observed during the update window (number of solution blocks, etc.). In fact, dr has a known update function that depends on the chosen value of □. We have added a sentence after Eq. (1) to clarify. The specifics are discussed in Ref. [9].

Reviewer C, comment 6:

"This is the author's proposed difficult retargeting method. It adjusts the difficulty of solving the hash-based PoW problem independently from solving the scientifically-interesting problem.

It's not clear to me precisely how it works. Are you looking at the average solve time for the scientifically-interesting problems ONLY when retargeting dr? And then only looking at the average solve time for the traditional blocks when retargeting db? More formalism here would help. What about include illustrating the solve times on your diagram to make this more clear?"

Reply:

The proof-of-work difficulties db and dr are updated independently. After Nb 2 blocks have been mined without a solution, db is updated according to the Bitcoin update scheme: by comparing the actual average time to mine each of the Nb 2 blocks to the desired average time per block (10 minutes in Bitcoin). Similarly, dr is updated after Nr 2 blocks containing a solution have been mined. Again, dr is updated according to the Bitcoin scheme. This update scheme is illustrated in Fig. 1. We have attempted to clarify this issue by referencing Fig. 1 in sections 2.1 and 2.2.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

Reviewer C, comment 7:

"I appreciate the authors addressing the obvious challenge. I find the last paragraph speculative though (and the second-to-last somewhat speculative too). Consider removing this content rather than discussing half backed ideas. Refer also to my comments about how leaving problem submission unsolved means that the entire idea is still unsolved."

Reply:

We agree with the referee that the last paragraph of Sec. II C is speculative. In accordance with the referee's suggestion, we agree to remove this paragraph.

Reviewer C, comment 8:

"As eta decreases ! confusing because eta doesn't get decreased by the retargeting algorithm. I think you mean as you re-run the simulation with smaller and smaller values of eta, right?"

Reply:

The referee is correct in this interpretation of the text. We have changed the phrasing in the last 2 paragraphs of section 3 to make the meaning clearer.

2A. Further Review

Reviewer C

The authors addressed most the points I brought up and spent reasonable efforts revising their manuscript.

But I still believe they do not appreciate how different their PoW proposal is to Bitcoin's PoW:

In bitcoin, the PoW solution for block N+1 depends on the solution for block N. If block N differed by a single transaction, that solution for N+1 would not be valid. It is because of this property that forking the chain is so costly and bitcoin is secure. You cannot recycle PoW solutions on forks. But in the authors' proposal, PoW solutions are ***not*** *tied like this to their ancestor blocks. Those PoW solutions _are_ recyclable.*

My suggestion is for the authors to either explain how what I said is not true (I'm misunderstanding their proposal), or in plain words describe this very important difference to the reader.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194

2B. Author Response

We agree with the reviewer in the way our protocol differs from standard PoW. We had attempted to explain this difference in terms of an attack avenue in Section 4.
The property of solution reuse, we understand was possibly not emphasized enough, for this reason we are attaching an updated manuscript which explicitly states this as a key difference.

As stated in the original manuscript, better understanding such attack vectors is not trivial and is left for future work. However, we still included some suggestions for mitigating these attacks. We also emphasize that this property is not unique to our work. For example, in reference 11 (Amar et al.) propose additional ways to address this threat.

We hope that this addresses the reviewer's concerns and appreciate the feedback.

ISSN 2379-5980 (online)
associated article DOI
10.5915/LEDGER.2020.194