# Proof-of-Work as a Stigmergic Consensus Algorithm*

Extended Abstract

Önder Gürcan
Université Paris-Saclay, CEA, List
F-91120, Palaiseau, France
onder.gurcan@cea.fr

## ABSTRACT

In this paper, we make a theoretical analysis from a coordination perspective and conclude that the Proof-of-Work (PoW) algorithm is a *stigmergic consensus algorithm* where the trace left by an action in the blockchain through indirect coordination of agents stimulates subsequent actions and eventually creates a single chain of blocks.

## KEYWORDS

Coordination and Control, Self-Organizing Systems, Swarm and Collective Behaviour

**ACM Reference Format:**
Önder Gürcan. 2022. Proof-of-Work as a Stigmergic Consensus Algorithm: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

## 1 INTRODUCTION

Bitcoin [25] employs the Proof-of-Work algorithm (PoW) [2] for maintaining an agreement in a trustless network, by including mechanisms that ensure that the effort of block creation is represented within the block submitted by its creator.

Technically speaking, in PoW blockchain systems there are two main kinds of participants: users and miners [15, 16]. All participants store unconfirmed transactions in their memory pools and confirmed transactions in their local blockchains. Users create transactions by carefully choosing a fee and then diffuse them for being confirmed in a block in the blockchain. Miners continuously attempt to order and approve a selection of the transactions they received as a block (that includes the hash value of a previous block in the blockchain) by executing a compute-intensive algorithm of a predetermined difficulty to generate a valid hash value for this block as a proof of their work (PoW). Upon finding such hash values, the corresponding miners transmit their blocks to the network to be appended to the blockchain. The result is a tree of blocks where the longest chain of that tree is considered as the valid blockchain.

While some researchers classify PoW as a Nakamoto consensus algorithm [7, 8, 14, 24, 34, 42] or as a proof based (also called Proof-of-X) consensus algorithm [4, 6, 10, 26, 37, 38] or as a compute-intensive based consensus algorithm [21], some other researchers do not even consider it as a consensus algorithm[1]. Hence, our objective is to shed light on this issue.

## 2 POW AS A MAS CONSENSUS ALGORITHM

Put into the blockchain context, consensus algorithms aim to agree on the new blocks among the participants. In multi-agent systems (MAS), consensus algorithms aim to make agents to decide *asymptotically* (i.e. the finality is not necessarily immediate) on a *common value* which is *based on the values* that the agents propose through *local interactions* and *computations* (definition based on [5]). There are basically two types of agreements in MAS consensus: (1) by agreeing on one of the proposed values (e.g., bees deciding on a food source [32], ants deciding on a trail for a food source [9]) or (2) by agreeing on a fusion of all proposed values (e.g., time synchronization in a wireless sensor network [19, 41], collective behaviour of flocks and swarms [27, 31]). Besides, in MAS consensus algorithms agents can coordinate by *indirectly* interacting with each other through *persistent* and *perceivable* changes to a common *environment* where recipients are all agents who will perceive these changes [22]. Environment is a first-class abstraction in the MAS paradigm [39] and, is a participant and a shared memory, not just a medium for interaction.

Considering a blockchain system as a MAS (as proposed in [1, 11, 17, 18, 23, 29]), the shared data structure blockchain is a *persistent*, *dynamic* and *virtual* environment allowing agents to interact indirectly with each other and aggregating the information.

## 3 POW AS A STIGMERGIC ALGORITHM

Concretely, PoW resembles the stigmergic MAS of *ants deciding on a trail for a food source* [35]. The objective of an ant society is to bring back as much food as possible from a food source that will satisfy its members. When worker ants leave their nest to find food (i.e. foraging), they leave behind a trail of pheromones along the way. When they find food, they fill up their *social stomach* with as much food as it can. Then they scurry back to their nest following the trail leaving again pheromones along the way. The ants will begin sharing food with other members of its colony through the social stomach and more ants will begin to follow the traced scent back to the food source. Every time an ant visits the food source, it adds to the effectiveness of the scent trail. Meanwhile, other ants may take various ways to the food source and back to the nest, leaving again traces of scent. This eventually leads to an optimization of

*This work is supported by ECSEL Joint Undertaking (JU) through the Project ADACORSA under grant agreement No 876019.

*Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.* 

[1] As put forward by [14]: "It is not easy to relate the probabilistic guarantees offered by PoW consensus protocols to the consensus definition used in the distributed systems literature (i.e. Byzantine Fault-Tolerant [BFT] consensus protocols).". Thus the common approach of the distributed systems community is either trying to define PoW as a randomized consensus algorithm with deterministic termination or not considering it as a consensus algorithm at all.

the path: *since pheromones are evaporative, the shorter the trail is the stronger the scent is – so more ants take the strongest trail.*

Now, consider a PoW-based blockchain system like an ant society. The objective of such a society is to confirm as much transactions as possible to satisfy its members. The food source resembles the state in which all (including future) transactions are confirmed and bringing a part of the food to the nest resembles confirming some unconfirmed transactions. When a miner finds unconfirmed transactions, it fills up its new block with as much transaction as it can. Then it appends its block to the blockchain and creates a trail of blocks linked by cryptographic links. This way, it shares confirmed transactions with other members of its society and more miners begin to follow the traced scent to reach the state where all transactions are confirmed. Every time a miner confirms transactions as a block, it adds to the effectiveness of the branch. Meanwhile, some other miners may take different branches, leaving again trails of blocks. This eventually leads to an optimization of the path: *since cryptographic links are very strong, the longer the branch is the stronger it is – so more miners follow the strongest branch.*

In conclusion, just like in an ant society [36], miners work as if they were alone while their collective activities appear to be coordinated. This is called *stigmergy*: an indirect, mediated mechanism of coordination between actions, in which the trace of an action left on a medium stimulates the performance of a subsequent action [20]. Stigmergy fits well to PoW since the blocks adopted by agents in the system guide which next blocks will be created and which next transaction will be issued. Consequently, we can conclude that: *PoW is a stigmergic consensus algorithm.*

## 4 THE FUNDAMENTAL PROPERTIES OF POW

Stigmergy enables complex, coordinated activity *without* any need for planning, memory, communication, mutual awareness, simultaneous presence, imposed sequence, commitment or supervision [20]. Below, we show how well PoW holds these properties.

**Planning:** Agents should only be conscious of the current state of the operation. There is no strategy or plan defining which block (or transaction) needs to be appended to the blockchain or when. **Memory:** Agents do not need to recall their previous activities; there is no need to store information about the state of the work anywhere but in the blockchain. Hence, agents can easily exit a blockchain system and then come back again. **Communication:** No information for negotiating over the actions to be taken needs to be exchanged between the agents, except through the work performed in the blockchain. **Mutual awareness:** Every agent operates independently; they do not even need to know that others are present in the system to decide which actions to be taken. **Simultaneous presence:** The agents need not to be present simultaneously; they can work whenever and wherever they are available, thanks to the fact that the traces (i.e. blocks and transactions in them) are persistent and can guide agents at any later time. **Imposed sequence:** Actions are carried out inherently in the right order, since an action would not be commenced until the right condition is in place; the work-flow *emerges* as the finishing of one block creation, creation of the next or as the confirmation of a transaction triggers the creation of the successive transactions. **Commitment:** Agents do not need to commit to creation of a specific block; they choose instantly what actions they should perform, depending on incentive and other contingent conditions; agents that quit or otherwise become unavailable is spontaneously replaced by other agents. **Supervision:** Disruptions (e.g., branches) are automatically fixed, without any centralized control directing the activity.

PoW also holds some self-* properties [33] such as self-healing and self-resilience. *Self-healing* property represents the ability to recover under failures. Recently, it has been reported that PoW is a self-healing algorithm [3]. *Self-resilience* property represents the ability to reliably provide a service while failures. In PoW, no agent is committed to create a future block and thus in case of a failure or an attack another agent can maintain the service.

## 5 DISCUSSION AND CONCLUSION

To the best of our knowledge, there is no other study categorizing PoW as a stigmergic consensus algorithm. Considering that the blockchain is a shared memory structure, albeit with interesting safety guarantees, the only study we are aware of are [28] that proposes the idea that shared memory structures can be used as *virtual stigmergy* for swarms of robots and [12] that proposes a language for describing *virtual stigmergy*. However, both studies do not take into account a system like blockchain that can be deployed to the Internet and that can be subject to severe failures like selfishness [13, 30] and Byzantine failures [8].

A PoW blockchain is a sort of *collective intelligent system* [33] in the sense that it relies on *feedback*: action elicits action, through the intermediary of the blockchain (i.e. the trace). Typically this feedback is positive with actions intensifying and elaborating the blockchain, thus eliciting more intense and diverse further actions. The resulting cycle enables blockchains (the common good) to be increasingly built up. Any agent may then profit from this common good without putting in any effort in return without reducing its value. An agent without leaving a stigmergic trace (i.e. blocks or transactions) does not, by that action, make the blockchain less useful to the other agents[2]. This conclusion enables us to identify what type of system PoW is creating, how it can be improved and where it can be deployed (e.g., to the Internet, to robot swarms [40] or to wireless sensor networks, where, for instance, *mutual awareness* and *simultaneous presence* may not necessarily be mandatory).

Nevertheless, PoW was not initially engineered as a stigmergic algorithm. Thus, its entire potential strength remains *unused* (e.g., frugality). However, re-engineering such a system is not trivial. Unlike other stigmergic systems, PoW blockchains are deployed to the Internet and they should be tolerant to any kind of fault. To this end, we identified the following key principles: (1) the agents cannot inherently rely on (i.e. take into account without questioning) the information coming from others, (2) the agents can inherently rely on its own local information (e.g., the ban score, its own local clock), (3) the agents can take into account the information found in its local blockchain directly (e.g., confirmed transactions and blocks) and (4) the agents can take into account the information validatable on its local blockchain (e.g., an unconfirmed transaction that is using the other transactions on the blockchain, an unconfirmed block whose hashcode is successively linked other blocks).

[2] In fact, by validating all the data it receives and diffusing only the ones that are valid, it already increases the quality of the stigmergic trace.

## REFERENCES


[1] Morteza Alaeddini, Julie Dugdale, Paul Reaidy, Philippe Madiès, and Önder Gürcan. 2021. An Agent-Oriented, Blockchain-Based Design of the Interbank Money Market Trading System. In *Agents and Multi-Agent Systems: Technologies and Applications 2021*, G. Jezic, J. Chen-Burger, M. Kusek, R. Sperka, R. J. Howlett, and Lakhmi C. Jain (Eds.). Springer Singapore, Singapore, 3–16.

[2] Adam Back. 2002. *Hashcash - A Denial of Service Counter-Measure.* Technical Report.

[3] Christian Badertscher, Peter Gaži, Aggelos Kiayias, Alexander Russell, and Vassilis Zikas. 2020. Consensus Redux: Distributed Ledgers in the Face of Adversarial Supremacy. Cryptology ePrint Archive, Report 2020/1021.

[4] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. 2019. SoK: Consensus in the Age of Blockchains. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies* (Zurich, Switzerland) *(AFT '19)*. ACM, New York, NY, USA, 183–198.

[5] Dimitri P. Bertsekas and John N. Tsitsiklis. 1989. *Parallel and Distributed Computation: Numerical Methods.* Prentice-Hall, Inc., USA.

[6] U. Bodkhe, D. Mehta, S. Tanwar, P. Bhattacharya, P. K. Singh, and W. Hong. 2020. A Survey on Decentralized Consensus Mechanisms for Cyber Physical Systems. *IEEE Access* 8 (2020), 54371–54401.

[7] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. 2015. SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*. 104–121.

[8] C. Cachin and M. Vukolic. 2017. Blockchain consensus protocols in the wild. In *31 International Symposium on Distributed Computing (DISC)*.

[9] Scott Camazine, Nigel R. Franks, James Sneyd, Eric Bonabeau, Jean-Louis Deneubourg, and Guy Theraula. 2001. *Self-Organization in Biological Systems.* Princeton University Press, USA.

[10] N. Chaudhry and M. M. Yousaf. 2018. Consensus Algorithms in Blockchain: Comparative Analysis, Challenges and Opportunities. In *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*. 54–63.

[11] Giovanni Ciatto, Stefano Mariani, Andrea Omicini, and Franco Zambonelli. 2020. From Agents to Blockchain: Stairway to Integration. *Applied Sciences* 10, 21 (2020). https://doi.org/10.3390/app10217460

[12] Rocco De Nicola, Luca Di Stefano, and Omar Inverso. 2020. Multi-agent systems with virtual stigmergy. *Science of Computer Programming* 187 (2020), 102345. https://doi.org/10.1016/j.scico.2019.102345

[13] Ittay Eyal and Emin Gün Sirer. 2014. Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security*. Springer, 436–454.

[14] Vincent Gramoli. 2020. From blockchain consensus back to Byzantine consensus. *Future Generation Computer Systems* 107 (2020), 760 – 769.

[15] Önder Gürcan, Antonella Del Pozzo, and Sara Tucci-Piergiovanni. 2017. On the Bitcoin Limitations to Deliver Fairness to Users. In *On the Move to Meaningful Internet Systems. OTM 2017 Conferences*, H. Panetto, C. Debruyne, W. Gaaloul, M. Papazoglou, A. Paschke, C-A. Ardagna, and R. Meersman (Eds.). Springer International Publishing, Cham, 589–606.

[16] Önder Gürcan, Alejandro Ranchal Pedrosa, and Sara Tucci-Piergiovanni. 2018. On Cancellation of Transactions in Bitcoin-Like Blockchains. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11229 LNCS (2018), 516–533.

[17] Önder Gürcan. 2019. Multi-Agent Modelling of Fairness for Users and Miners in Blockchains. In *Highlights of Practical Applications of Survivable Agents and Multi-Agent Systems. The PAAMS Collection*, Fernando De La Prieta, Alfonso González-Briones, Pawel Pawleski, Davide Calvaresi, Elena Del Val, Fernando Lopes, Vicente Julian, Eneko Osaba, and Ramón Sánchez-Iborra (Eds.). Springer International Publishing, Cham, 92–99. https://doi.org/10.1007/978-3-030-24299-2_8

[18] Önder Gürcan. 2020. On Using Agent-based Modeling and Simulation for Studying Blockchain Systems. In *JFMS 2020 - Les Journées Francophones de la Modélisation et de la Simulatio - Convergences entre la Théorie de la Modélisation et la Simulation et les Systèmes Multi-Agents*. Cargèse, France. https://www.cepadues.com/livres/jfms-2020-les-journees-francophones-modelisation-simulation-convergences-entre-theorie-modelisation-simulation-les-systemes-multi-agents-9782364937574.html

[19] Önder Gürcan and Kasim Sinan Yildirim. 2013. Self-Organizing Time Synchronization of Wireless Sensor Networks with Adaptive Value Trackers. In *2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems*. 91–100. https://doi.org/10.1109/SASO.2013.22

[20] Francis Heylighen. 2016. Stigmergy as a Universal Coordination Mechanism I. *Cogn. Syst. Res.* 38, C (June 2016), 4–13.

[21] Leila Ismail and Huned Materwala. 2019. A Review of Blockchain Architecture and Consensus Protocols: Use Cases, Challenges, and Solutions. *Symmetry* 11, 10 (2019).

[22] David Keil and Dina Goldin. 2006. Indirect Interaction in Environments for Multi-agent Systems. In *Environments for Multi-Agent Systems II*, Danny Weyns, H. Van Dyke Parunak, and Fabien Michel (Eds.). Springer, 68–87.

[23] Nicolas Lagaillardie, Mohamed Aimen Djari, and Önder Gürcan. 2019. A Computational Study on Fairness of the Tendermint Blockchain Protocol. *Information* 10, 12 (2019). https://doi.org/10.3390/info10120378

[24] Loi Luu, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. 2015. Demystifying Incentives in the Consensus Computer. In *Proc. of the 22nd ACM SIGSAC Conf. on Computer and Comm. Security (CCS '15)*. New York, NY, USA, 706–719.

[25] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf.

[26] Giang-Truong Nguyen and Kyungbaek Kim. 2018. A Survey about Consensus Algorithms Used in Blockchain. *Journal of Information Processing Systems* 14, 1 (2018), 101–128.

[27] R. Olfati-Saber. 2006. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Trans. Automat. Control* 51, 3 (2006), 401–420.

[28] Carlo Pinciroli, Adam Lee-Brown, and Giovanni Beltrame. 2016. A Tuple Space for Data Sharing in Robot Swarms. *EAI Endorsed Transactions on Collaborative Computing* 2, 9 (5 2016).

[29] Hector Roussille, Önder Gürcan, and Fabien Michel. 2022. AGR4BS: A Generic Multi-Agent Organizational Model for Blockchain Systems. *Big Data and Cognitive Computing* 6, 1 (2022). https://doi.org/10.3390/bdcc6010001

[30] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. 2016. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*. Springer, 515–532.

[31] A. V. Savkin. 2004. Coordinated collective motion of Groups of autonomous mobile robots: analysis of Vicsek's model. *IEEE Trans. Automat. Control* 49, 6 (2004), 981–982.

[32] Thomas D. Seeley. 1995. *The Wisdom of the Hive.* Harvard University Press.

[33] G. Serugendo, M.-P. Gleizes, and A. Karageorgos (Eds.). 2011. . Springer. 347–377 pages.

[34] Nicholas Stifter, Aljosha Judmayer, Philipp Schindler, Alexei Zamyatin, and Edgar R. Weippl. 2018. Agreement with Satoshi - On the Formalization of Nakamoto Consensus. *IACR Cryptology ePrint Archive* 2018 (2018), 400.

[35] David J.T Sumpter and Madeleine Beekman. 2003. From nonlinearity to optimality: pheromone trail foraging by ants. *Animal Behaviour* 66, 2 (2003), 273 – 280.

[36] Guy Theraulaz and Eric Bonbeau. 1999. A Brief History of Stigmergy. *Artif. Life* 5, 2 (April 1999), 97–116.

[37] Eric Ke Wang, RuiPei Sun, Chien-Ming Chen, Zuodong Liang, Saru Kumari, and Muhammad Khurram Khan. 2020. Proof of X-repute blockchain consensus protocol for IoT systems. *Computers & Security* 95 (2020), 101871.

[38] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim. 2019. A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access* 7 (2019), 22328–22370.

[39] Danny Weyns, Andrea Omicini, and James Odell. 2007. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems* 14, 1 (01 Feb 2007), 5–30.

[40] Guang-Zhong Yang, Jim Bellingham, Pierre E. Dupont, Peer Fischer, Luciano Floridi, Robert Full, Neil Jacobstein, Vijay Kumar, Marcia McNutt, Robert Merrifield, Bradley J. Nelson, Brian Scassellati, Mariarosaria Taddeo, Russell Taylor, Manuela Veloso, Zhong Lin Wang, and Robert Wood. 2018. The grand challenges of Science Robotics. *Science Robotics* 3, 14 (2018).

[41] K. S. Yildirim and Ö. Gürcan. 2014. Efficient Time Synchronization in a Wireless Sensor Network by Adaptive Value Tracking. *IEEE Transactions on Wireless Communications* 13, 7 (July 2014), 3650–3664.

[42] R. Zhang and B. Preneel. 2019. Lay Down the Common Metrics: Evaluating Proof-of-Work Consensus Protocols' Security. In *2019 IEEE Symposium on Security and Privacy (SP)*. 175–192.