Information Processing and Management 59 (2022) 102749

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

A novel proof of useful work for a blockchain storing transportation transactions

Mohamed Haouari [a], Mariem Mhiri [a,*], Mazen El-Masri [b], Karim Al-Yafi [b]

[a] Department of Mechanical and Industrial Engineering, College of Engineering, Qatar University, Doha, Qatar
[b] Department of Accounting and Information Systems, College of Business and Economics, Qatar University, Doha, Qatar

ARTICLE INFO

Keywords:
Blockchain
Supply chain
Proof of Useful Work
NP-hard optimization problem

ABSTRACT

Proof-of-Work (PoW) is a common mechanism used to validate peer-to-peer transactions and maintain highly secured immutability of the blockchain. However, this mechanism has been criticized due to its inefficient use of computing resources and its limited usefulness. In this paper, we propose the Proof-of-Useful-Work (PoUW) as an alternative mechanism for transaction validation that puts the squandered computing resources to beneficial use. The main premise is to replace the mathematical puzzle, which constitutes a fundamental part of the Proof-of-Work mechanism, with NP-hard optimization problems whose solutions benefit the participants of the blockchain. We demonstrate its usefulness in the context of transportation. Accordingly, PoUW-based blockchain not only tracks, manages and validates transactions, but also optimizes transportation requests profiting its ecosystem. We describe the framework of the proposed PoUW along with the associated optimization model and the miner's reward mechanism.

1. Introduction

Long before Bitcoin's introduction, blockchain is a distributed ledger technology that was developed in 1991 by Haber and Stornetta (1991). Seventeen years later, Nakamoto (2008) proposed the first large-scale and applied use-case of blockchains with Bitcoin, the cryptocurrency working without a trusted intermediary (Zhang, 2019). The technology facilitates online transactions between parties without relying on a centralized financial institution (Nakamoto, 2008). This new concept has revolutionized the traditional ways of transacting tokenized value between parties (nodes) in a trustless and distributed environment. Depending on the purpose and capacity of blockchain employed, such transactions can be performed to exchange intangible assets such as cryptocurrencies, pay for music royalties, vote, or buy a house. Indeed, any digital asset can be exchanged such as the case of non-fungible tokens like GIFs, artwork, or essay.

Blockchain is defined as a chain of chronologically ordered and cryptographically linked blocks (Perboli, Musso, & Rosano, 2018). Each block is an immutable record of a set of transactions that were validated and approved by the participating nodes on that blockchain network. Immutability is ensured by linking the blocks sequentially so that each block includes the signature (known as the hash) of the previous block. Therefore, a change in a block will require modifying all subsequent blocks. The rest of the block records a collection of transactions between nodes along with a timestamp (Singhal, Dhameja, & Panda, 2018). When a block is to be appended to the blockchain, a certain validation mechanism is required by the nodes. This step ensures that the transactions included in the new block are valid and approved by the majority of the nodes. This validation process requires a protocol called "consensus mechanism" that must be followed by all participating nodes. Proof-of-Stake (PoS) and Proof-of-Work

* Corresponding author.
E-mail addresses: mohamed.haouari@qu.edu.qa (M. Haouari), meriam.mhiri@qu.edu.qa (M. Mhiri), mazen.elmasri@qu.edu.qa (M. El-Masri), karim.alyafi@qu.edu.qa (K. Al-Yafi).

https://doi.org/10.1016/j.ipm.2021.102749
Received 11 November 2020; Received in revised form 22 August 2021; Accepted 31 August 2021
Available online 14 September 2021

(PoW) are the most common mechanisms for blockchain consensus (Saad, Qin, Ren, Nyang, & Mohaisen, 2021). Albeit their proven effectiveness in validating blocks, they both have their shortcomings. On the one hand, PoS may lead to centralization, unfairness and less reliability (Nair & Dorai, 2021; Saad et al., 2021; Zhang & Chan, 2020). To update the blockchain, PoS protocol selects the validating node according to its share of stake estimated as the fraction of coins owned which defines its probability to be selected (Roşu & Saleh, 2021). The higher the stake, the more probable the node will be selected to mint the new block and get rewarded. Consequently, PoS may lead to wealth concentration and centralization like an auction process (Roşu & Saleh, 2021; Saad et al., 2021). On the other hand, PoW relies on having the nodes to computationally compete on solving a mathematical puzzle that is hard to solve but whose solution is easy to verify. PoW involves finding a string value called 'nonce' that, when hashed with the block contents and the hash of the previous block, will result in a hash value that has a certain number of leading zeros. The usefulness of finding a valid nonce is strictly limited to block validation purposes stripped from any further use (Baldominos & Saez, 2019). The fierce race in computational power caused by the PoW makes blockchain highly expensive in terms of computing resources and electrical energy (Drescher, 2017) resulting in huge waste of resources and severe impacts on the environment, especially that the problem difficulty increases proportionally with the employed computational resources (Dhillon, Metcalf, & Hooper, 2017). In fact, PoW protocols self-adjust the problem difficulty to maintain a steady rate of block creation. As per a study in 2014, the total power consumed by the whole Bitcoin mining industry is around 0.1–10 GW compared to Ireland's average electricity consumption estimated at 3 GW (Drescher, 2017; O'Dwyer & Malone, 2014). Nevertheless, the consensus mechanism is an indispensable process in blockchain and has been proven to be an effective validation method. Accordingly, the motivation of this paper is to funnel the computation power necessary to maintain the indispensable validation feature of blockchain to solve useful problems that could contribute beyond finding meaningless hash values. This would extend blockchain functionalities and turn one of its worst deficiencies into an advantage. In this context, the first work dealing with improving the PoW was (Ball, Rosen, Sabin, & Vasudevan, 2017) where a so-called Proof of Useful Work (PoUW) was developed. The idea consists in solving mathematical problems by joining hardness to usefulness while still preventing malicious blockchain use. Our anticipated PoUW mechanism not only contributes to reducing energy waste but also solves blockchain application related problems that are useful for its ecosystem. For our use-case, we focus on the PoW to retain network decentralization by introducing a new PoUW that provides useful outputs for a maritime transportation blockchain. Accordingly, we posit that the purpose of this paper is not only to provide a generic PoUW mechanism that is useful in terms of energy reduction for a maritime transportation blockchain but also to achieve saving for the participating nodes.

1.1. Paper's contributions

The purpose of the paper is to prove that a real-world NP-hard problem could be used in a PoUW. The considered approach is modeled as a mathematical optimization problem in the context of supply chain management and more specifically maritime transportation. The new PoUW proposed in this paper requires solving an NP-hard optimization problem minimizing the total cost of the transportation between same origin port and same destination port. The proposed PoUW enables: (1) clustering the transportation requests and thereby minimizing the total transportation cost and therefore the individual costs, and (2) validating the creation of new blocks in the blockchain. To the best of our knowledge, optimization problems with useful outcomes, both for the blockchain and the supply chain, have seldom been investigated in the literature. Indeed, the only relevant reference in this context is Manupati et al. (2019) where the authors carried out a live monitoring of the supply chain within the context of smart contracts. They developed a blockchain-based approach enabling in a synchronized way, the control of the supply chain performance and the optimization of emission levels and operational costs. The problem was formulated as a mixed integer non-linear programming model providing a significant outcome for the supply chain.

1.2. Paper's outline

This paper is organized as follows. In Section 2, we describe the concept of PoW and shed light on the importance of the PoUW mechanisms. In Section 3, a novel PoUW is proposed for the considered blockchain. Afterward, a mathematical study for cost allocation between the blockchain participants is elaborated. In Section 4, we discuss some implementation remarks. Finally, the conclusion is drawn in Section 5.

2. Literature review

Given its multiple advantages, blockchain is attracting an ever-increasing interest in the academic literature. In this context, many papers have been published to date on applications of blockchain technology for enhancing the security, management and development of information systems (Berdik, Otoum, Schmidt, Porter, & Jararweh, 2021; El-Masri & Hussain, 2021). In this regard, digital security has been explored in many application settings including, but not limited to, combating fake media (Chen, Srivastava, Parizi, Aloqaily, & Ridhawi, 2020), monitoring and managing vehicular networks (Campanile, Iacono, Marulli, & Mastroianni, 2021; Khalid et al., 2021; Oham, Michelin, Jurdak, Kanhere, & Jha, 2021), managing code copyright (Jing, Liu, & Sugumaran, 2021), checking data integrity (Li, Wu, Jiang, & Srikanthan, 2020; Zhao, Chen, Liu, Baker, & Zhang, 2020) and sharing data (Hardin & Kotz, 2021; Putz, Dietz, Empl, & Pernul, 2021). Additional relevant blockchain applications include predicting average transaction latency performance (Xu et al., 2021), enabling smart contracts distinction, anomaly detection and malicious contract identification (Hu et al., 2021) and ensuring blockchain scalability and flexibility (Yu et al., 2021), to quote just a few.

The trust problem in blockchain is addressed by the consensus mechanism which helps establish an agreement between the majority of network nodes (Yu, Liu, & Wang, 2018; Zhang, 2019) on the validity of blocks to be added to the ledger. It ensures secure updates on the blockchain, its integrity and consistency, orders the transactions and the blocks, increases trust between participants and prevents from attacks (Wan, Li, Liu, & Wang, 2020; Zhang, 2019). Although numerous consensus mechanisms have been introduced (Bamakan, Motavali, & Bondarti, 2020; Ferdous, Chowdhury, & Hoque, 2021; Lashkari & Musilek, 2021; Monrat, Schelén, & Andersson, 2019), the most common and widely adopted consensus protocols/algorithms are mainly Proof-of-Work and Proof-of-Stake. In order to respect the decentralization principle of blockchain, we opt for the PoW and propose to turn its biggest drawback (namely, its uselessness) into a benefit for blockchain participants.

The concept of *Proof-of-Work* (PoW) has been introduced in Dwork and Naor (1993) by presenting a computational technique to combat junk mail and control access to a shared resource. The basic idea is to compel a user to calculate a moderately hard yet feasible function in order to allow access to the resource and inhibit trivial use (Dwork & Naor, 1993).

The blockchain implementation strongly depends on the PoW to form a consensus where miners compete to solve a cryptographic challenge. The first to successfully provide a solution to the challenge adds a new block to the chain and wins a reward, such as portions of Bitcoins (Loe & Quaglia, 2018; Zhang, 2019). From a technical perspective, to mine in Bitcoin is to find a value so that when hashed (e.g., with SHA-256) with the given challenge, the hash starts with a certain number of zero bits (Nakamoto, 2008). The hashing function works unpredictably through trial and error thereby ensuring hardness. Still, the resulting value is completely useless (Ball et al., 2017; Drescher, 2017).

That said, PoW mechanisms lead to high energy consumed in the challenge resolution without providing useful outputs. Furthermore, the computational complexity problem remains unknown (Loe & Quaglia, 2018). Based on an online tool called Cambridge Bitcoin Electricity Consumption Index (CBECI, 2020), enabling real-time estimation of the Bitcoin network power, it estimated that Bitcoin mining operation consumed during a year more energy (64 Terawatt-hours per year) than Switzerland (58 Terawatt-hours per year) (Vincent, 2019).

To this end, the PoW is costly in terms of energy as the challenges are randomly generated and the corresponding solutions do not reflect any further information apart from validation. This drawback demanded the classic PoW be replaced by a Proof of Useful Work in Ball et al. (2017). Indeed, authors developed a PoUW whose hardness is related to computational problems, such as Orthogonal Vectors, 3SUM, All-Pairs Shortest Path (APSP). The corresponding results are no longer energy wasteful but useful as solutions to computational problems of practical interest (Baldominos & Saez, 2019) while keeping the basic properties of the classic PoW. This requires the concurrent assurance of Ball et al. (2017) (1) ***hardness*:** through the work required by the challenges and (2) ***usefulness*:** by allowing useful problem instances to be delegated as challenges and the resulting solutions are reconstructible quickly and verifiably. More specifically, the proposed PoUW is designed to generate challenges dependent on an instance x so that the resulting solutions depending on x serve to the reconstruction of some $g(x)$. Thus, while the classic PoW consists of three algorithms (Gen, Solve, Verify), a Recon algorithm is added to the PoUW to reconstruct $g(x)$. Four algorithms are then involved as follows (Ball et al., 2017):

- **Gen**(x): a randomized algorithm that generates from an instance x a challenge h_x.
- **Solve**(h_x): an algorithm that produces a solution l when solving the challenge h_x.
- **Verify**(h_x, l): a randomized algorithm verifying l to h_x.
- **Recon**(h_x, l): an algorithm that reconstructs $g(x)$ when given a valid l for h_x.

The following properties are required to the four algorithms: *efficiency, completeness, soundness, hardness* and *usefulness*. They are detailed in Ball et al. (2017).

In the literature, a few studies focused on developing PoUWs. Authors in Loe and Quaglia (2018) developed a PoUW based on the Traveling Salesman Problem (TSP) according to two rounds. In the first round, the Hashcash stage, ASIC hardware is used in a deterministic time interval. In the second round, an instance of the NP-Hard Traveling Salesman Problem is constructed. In Lihu, Du, Barjaktarević, Gerzanics, and Harvilla (2020), a novel PoUW concept using a distributed and decentralized machine learning system on blockchain is presented. It was shown that the proposed solution is more cost-efficient to a client than regular cloud machine learning training as well as more useful than Bitcoin mining. The main purpose of Merlina (2019) is to develop a new PoUW based on the training of machine learning models. Usefulness is achieved through the supervised training of neural networks. In Mittal and Aggarwal (2020), a new PoUW called Proof of Deep Learning with Hyperparameter Optimization (PoDLwHO) was developed using Bayesian Optimization. The deep learning model parameters are trained and tuned using the useless hash computations. Instead of competing to solve the puzzle in the PoW, the participating nodes in the blockchain compete against each other to produce models with better performances. Lastly, a PoUW relying on deep learning training is proposed in Chenli, Li, and Jung (2020) which provides a satisfying security level with weaker assumptions when comparing to the classic PoW.

3. A novel PoUW for blockchain optimizing transportation requests between ports

3.1. Blockchain description

We put forward a blockchain use-case in which customers request transport services from carriers. For example, a blockchain could be dedicated to retail companies that use trucking companies to transport pallets from distant warehouses (e.g., located in ports) to retail stores. As a second example, we consider a blockchain which is used by import companies to store container transport requests for shipping lines. In both cases, the blockchain is used for storing a list of transportation requests (TRs) where each TR includes the following information:

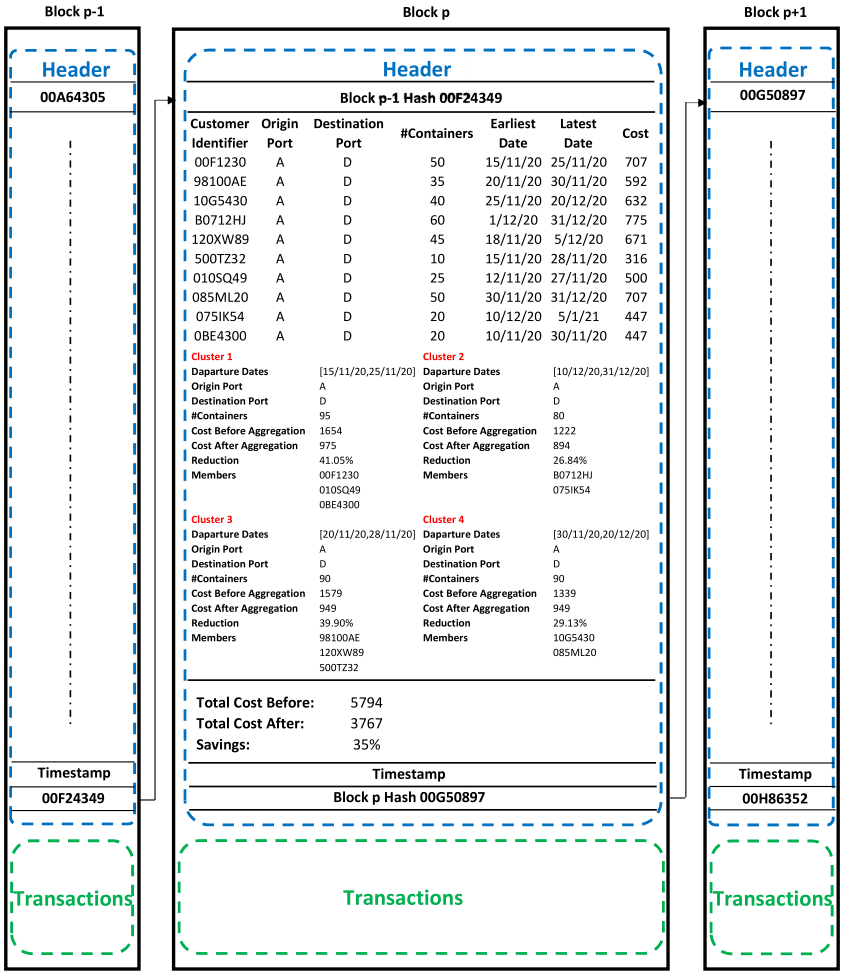

Fig. 1. Illustrative example of blocks in a blockchain storing transportation requests.

1. A hash of the username of the customer who is expressing demand. The hash is used to prevent disclosure of the customer's identity,
2. The origin point and the destination point of the trip,
3. The number of units (pallets or containers) to be transported,
4. The earliest and latest departure dates of the trip, and
5. An estimate of the transportation cost.

The transportation costs are often publicly available from logistics companies and shipping lines. A particular feature of these transportation costs is that they usually exhibit decreasing returns to scale. In other words, the unit transportation cost decreases as the total transported quantity increases. Accordingly, we henceforth assume that the transportation cost is a strictly concave function of the transported volume. Consequently, each group of users of the blockchain having TRs that share the same origin, destination, and departure dates, may form a coalition by consolidating their respective TRs, thereby taking advantage of economies of scale that are enabled by the structure of the cost function. Indeed, since the unit transportation is a decreasing function of the total volume then the total cost paid by the coalition (after consolidation) is always strictly less than the sum of individual costs. In so doing, the members of the coalition would achieve a reduction in their transportation costs. In Section 3.4, we describe the mechanism that iteratively considers all the TRs that appear in a given block, and clusters those TRs into virtual coalitions so that the total transportation cost is minimized. A key contribution of this paper, is that the resolution of the clustering problem will be used as a Proof-of-Work for validating the block. Hence, a block header of the proposed blockchain includes the following information:

1. The hash of the preceding block,
2. The list of TRs together with the sum of the corresponding costs, and

3. The list of clusters together with the sum of the total cost that is obtained upon clustering,
4. Other meta-data related to the block such as its own hash and timestamp.

An example is displayed in Fig. 1.

In this example, we assume that Bloc p contains 10 TRs, the vessel capacity Q is 100, and that the cost function is proportional to the squared root of the number of containers. The total number of clusters is four. We notice then that the total transportation cost, in Block p of the blockchain, was reduced by 35% upon clustering the TRs.

3.2. PoUW sequence

When a new TR is submitted to the system, it is first sent to a node. That recipient node will share it with other nodes in a distributed manner. Each node keeps the TRs it received in its own local pool (which is identical to all nodes) until a certain number of TRs having similar origin/destination ports pairs is reached (e.g., $k = 20$ or 50). This is to ensure a minimum size of the problem and that proper bundling resulting in decent saving can be generated. Moreover, it ensures that all nodes are working on the same instance of the problem at any given time. When that minimum number of TRs is reached, PoUW recovers from the pool all the TRs having an identical origin and the same destination port and starts the optimization process. When a node finds a solution, it shares it with the other nodes for verification. When a node verifies and approves the correctness of the received solution, it creates a block in its own ledger, locally. It is only at this stage that TRs are included into a block which is added to the blockchain. So do all other nodes.

The transportation requests flow is displayed in Fig. 2 (Salha, El-Hallaq, & Alastal, 2019; Zhang, 2019).

3.3. The clustering problem

For the sake of convenience, the problem shall be described in the context of a blockchain that is used in maritime transportation. We assume that all transportation requests (TRs) sharing the same origin and destination ports are in the same pool. The problem is formally defined as follows. We have n transportation requests/orders. Each $\mathrm{TR}_j (j = 1, \ldots, n)$ is characterized by a number of containers q_j and a time window $[a_j, b_j]$ during which the loading of the containers on the vessel should occur. The transportation of the containers from the origin port to the destination port can be achieved by a fleet of homogeneous vessels. The capacity of each vessel is Q. We assume that the transportation cost function $f(.)$ is: (1) increasing, and (2) strictly concave. Consequently, the unit transportation cost is a decreasing function of the total number of transported containers.

The problem requires finding a set of m TR clusters/coalitions that will be assigned to different vessels while accommodating the following constraints: (1) each TR is assigned to exactly one cluster, (2) each cluster is composed with time-compatible TRs (that is, having overlapping time windows), (3) the vessel capacity constraint is satisfied. The objective is to minimize the total transportation costs. The considered clustering problem is NP-hard problem in the strong sense.

Theorem 1. *The clustering problem is NP-hard in the strong sense.*

Proof. See Appendix A. □

As a consequence, finding a solution for the clustering problem that has a cost that is less or equal than a threshold value $\kappa > 0$ would require a very significant computational effort. In contrast, since the problem is in class NP, then it is easy to check for a given solution whether its cost is indeed less or equal than κ. This property makes the clustering problem an appealing alternative of useful Proof-of-Work mechanism.

3.4. Description of the blockchain mining process

Let us consider that we want to validate a block that includes a set of TRs sharing the same origin/destination ports. Correspondingly, we propose to validate this block by solving only one clustering problem. In this regard, miners will compete for solving the clustering problem using a process which is in the same vein as the one proposed by Baldominos and Saez (2019). The scheme is as follows. The platform sets an initial threshold value $\alpha \in (0, 0.5]$. Miners compete to find a solution that is less than $(1 - \alpha)C$. Towards this goal, they may use either exact or heuristic solution strategies. So long as no miner provides a feasible solution that costs less than the fixed threshold, the value of parameter α gets periodically reduced by a constant factor. The first miner to exhibit a feasible solution having a cost that is less than $(1 - \alpha)C$ is the designated winner.

The detailed step-by-step description of the mining process is provided below and depicted by the flowchart in Fig. 3:

Blockchain mining process

⋄ The optimization problem used to create a new block consists of a set of transportation requests that share the same origin and destination ports.

⋄ The miners compete to solve the corresponding clustering problem using the following steps:

• **Generate**

(1) Select all transportation requests having the same origin and destination from the TR pool and build the respective clustering problem instance as follows:

- TR_j is the j^{th} transportation request with $j = 1, .., n$;
- q_j is the number of containers assigned to TR_j;
- $[a_j, b_j]$ is the time window of TR_j;
- Q is the capacity of each vessel;
- C is the total cost before aggregation.

(2) Define the computational challenge that nodes will race each other to be the first to find a solution minimizing the total cost of transportation after clustering. This consists in determining a clustering scheme of the TRs which minimizes the total cost of transportation under the following constraints:

- each TR is assigned to exactly one cluster;
- each cluster is composed with time-compatible TRs;
- the capacity constraint is satisfied.

• **Solve**

(3) Initialize the value of α.

(4) While no miner succeeds in providing a feasible solution costing less than $(1-\alpha)C$ **Do**

$\alpha \leftarrow \alpha\beta$, where $\beta < 1$;

EndWhile

• **Verify**

(5) Select the first miner who successfully provides a solution having a cost less than the threshold value.

(6) Verify that the proposed solution contributes to :

- the resulting total cost (after clustering) is less than $(1-\alpha)C$;
- each TR is included in a cluster;
- the capacity constraint is satisfied;
- the time windows of the TRs are overlapping ($\neq \emptyset$).

• **Recon**

(7) Interpret the clusters/coalitions as the result of partitioning a set of integers (containers q_j) into subsets where the sum of each subset is $\leq Q$ and the time window of the subset is $\neq \emptyset$.

Remark. Solving the clustering problem with concave cost function is done in Step 4. This problem can be formulated as a mixed-integer nonlinear program and optimally solved using integer programming techniques (e.g., branch-and-cut, branch-and-price, etc.). Alternatively, it can be solved using stochastic search or other sophisticated metaheuristics.

3.5. Savings allocation

The blockchain mining process would be incomplete if it does not include an effective reward mechanism to motivate miners to invest effort in solving the mathematical problem. In this section, we describe the procedure for rewarding the winning miner. This procedure is essential to ensure the continuity of the blockchain. The idea is to reward the winning miner an amount of tokens. In so doing, each participant of the blockchain can accumulate a balance of tokens. Now, assume that a block requires solving a clustering problem that involves n TRs, where each TR_j has a cost c_j (before clustering). Thus, the total cost is $C = \sum_{j=1}^{n} c_j$. The new cost that is obtained after clustering is C^*. Thus, the reduction rate is $\Delta = 1 - \frac{C^*}{C}$. We assume that $\Delta > 0$. Hence, the virtual coalition achieves a total savings of ΔC and all coalition members are entitled to a reduction on their transportation costs. In this regard, a minimum discount factor is guaranteed to all coalition members. However, the total savings will be allocated to the different coalition members in a *biased* way that depends on their respective token balances. In other words, the savings allocation procedure favors those participants who have accumulated a large token balance. In the sequel, we shall provide a detailed description of the token balance management and the savings allocation procedures.

3.5.1. Token balance management

We assume that each user of the blockchain has a balance of tokens whose initial value is zero.

Crediting the balance: Each time a miner solves a clustering problem, his balance is credited an amount that is an increasing function with the achieved saving. Hence, if the initial cost is C and the reduction rate is Δ then the balance of the winning miner is increased by ΔC.

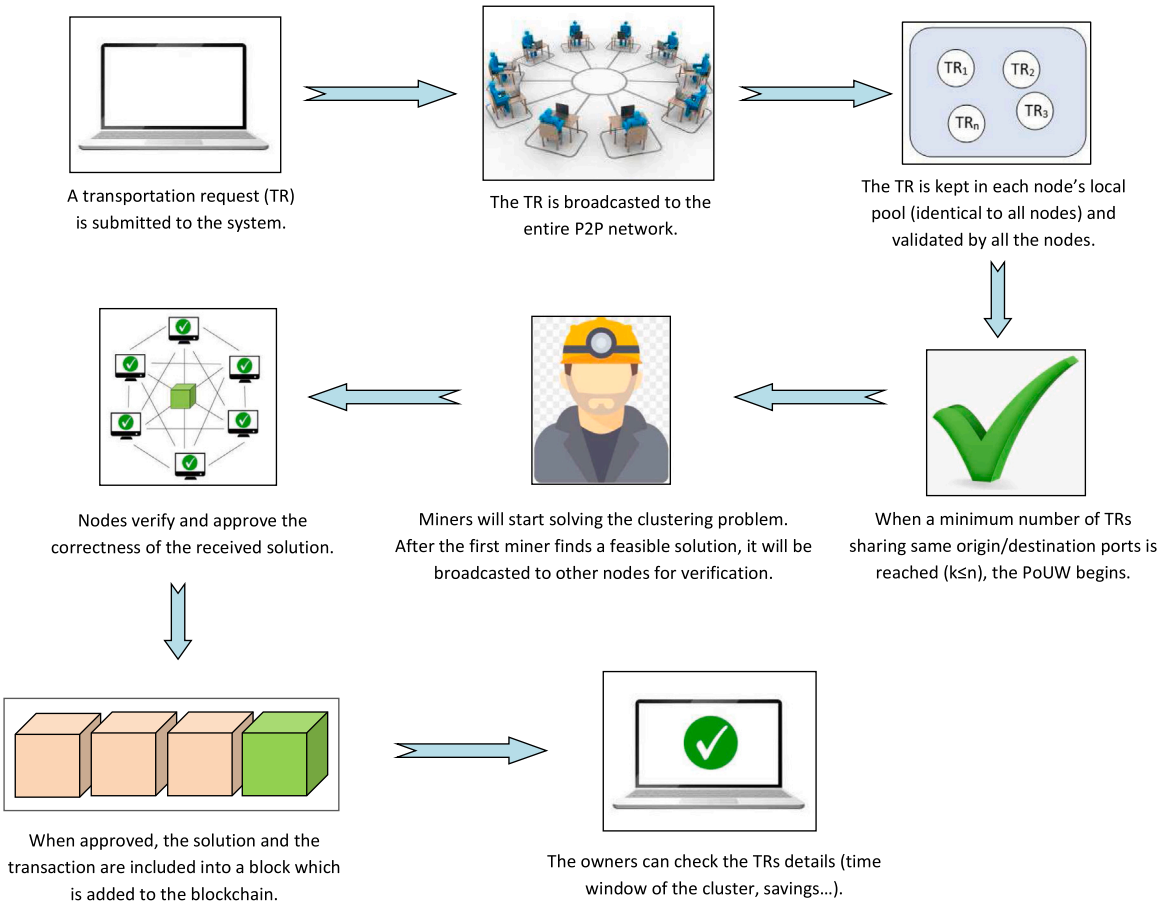

Fig. 2. Flow of bundling the transportation requests as part of PoUW.

Debiting the balance: Each time a participant is granted an *extra* savings (that is, the excess amount with respect to a minimum guaranteed amount), his balance is debited. The debited amount increases with the achieved extra savings. Hence, if the extra savings correspond to an amount p, then its balance is debited the same number of tokens.

3.5.2. Differential allocation of the savings

Assume that the clustering problem that corresponds to a block that includes n orders was solved, and that the token balance of user j is T_j, the problem requires finding a fair allocation of the total savings (i.e. ΔC) among the different orders. To deal with this, we denote by δ_j the reduction rate of order j. That is, the savings of user j is $s_j = \delta_j c_j$. We postulate that the allocation function should satisfy the following axioms.

A1 $\sum_{j=1}^{n} \delta_j c_j = \Delta C$, (obvious)

A2 $\delta_j \in [a_{\min}, 1]$ with $0 < a_{\min} \leq \Delta$. Each order should be granted a nonzero minimum saving rate $a \in [a_{\min}, \Delta]$.

A3 $T_i = T_j$ and $c_i = c_j \Rightarrow \delta_i = \delta_j$ (similar orders should be granted the same savings).

A4 $T_i > T_j$ and $c_i \leq c_j \Rightarrow (\delta_i > \delta_j)$ or $(\delta_i, \delta_j = 1)$. In order to favor those users that have accumulated a large balance, an order with higher balance and lower cost than another order, will get a reduction rate higher than the other order, or both orders will get 100% of reduction.

A5 $(\delta_j - a)c_j \leq T_j$, the amount of extra savings that are granted to j should not exceed its balance. This implies that the saving rate of j satisfies $\delta_j \leq a + \frac{T_j}{c_j}$.

We assume that the minimum rate $a_{\min}$ is given. We postulate that the target saving for j is an affine function of T_j. That is, the target saving rate for j is the sum of a minimum rate a and an extra discount rate bT_j.

Clearly, we have:

$$\delta_j \leq 1 \tag{1}$$

Since, the target value of δ_j is $a + bT_j$ then we define a deviational variable u_j that satisfies:

$$\delta_j + u_j = a + bT_j \tag{2}$$

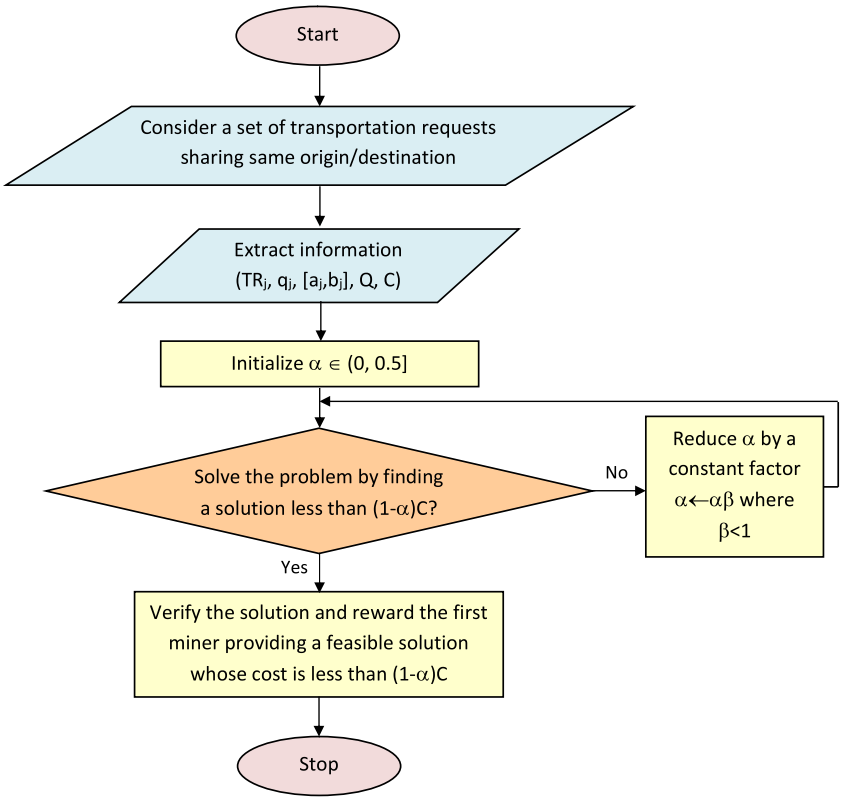

Fig. 3. Flowchart of the PoUW.

To compute the values of the δ-variables, we proceed in two steps:

1. We compute the minimum saving rate by solving the following linear program:

$$\text{(P): Minimize} \quad a \tag{3}$$

$$\delta_j + u_j = a + bT_j, \quad j = 1, \ldots, n, \tag{4}$$

$$\sum_{j=1}^{n} \delta_j c_j = \Delta C, \tag{5}$$

$$\delta_j \leq a + \frac{T_j}{c_j}, \quad j = 1, \ldots, n, \tag{6}$$

$$\delta_j \leq 1, \quad j = 1, \ldots, n, \tag{7}$$

$$a_{\min} \leq a, \tag{8}$$

$$b \geq 0, \tag{9}$$

$$\delta_j, u_j \geq 0, \quad j = 1, \ldots, n. \tag{10}$$

2. Let a^* denote the optimal value that is obtained upon solving (P). We compute the values of the saving rates by solving the following Quadratic Goal Programming problem:

$$\text{(QGP): Minimize} \quad \sum_{j=1}^{n} u_j^2 \tag{11}$$

$$\delta_j + u_j = a^* + bT_j, \quad j = 1, \ldots, n, \tag{12}$$

$$\sum_{j=1}^{n} \delta_j c_j = \Delta C, \tag{13}$$

$$\delta_j \leq a^* + \frac{T_j}{c_j}, \quad j = 1, \ldots, n, \tag{14}$$

Table 1
Data of example 1.

j	1	2	3	4	5	6	7	8	9	10
T_j	40	45	40	40	100	60	50	42	10	37
c_j	20	20	20	20	30	40	50	10	100	70

Table 2
Resulting saving rates for example 1.

j	1	2	3	4	5	6	7	8	9	10
$\delta_j(\%)$	21.04	23.05	21.04	21.04	45.11	29.06	25.05	21.84	9.01	19.84

Table 3
Data of example 2.

j	1	2	3	4	5	6	7	8	9	10
T_j	30	40	40	50	45	60	35	50	30	40
c_j	20	20	200	20	40	60	10	100	500	10

Table 4
Resulting saving rates for example 2.

j	1	2	3	4	5	6	7	8	9	10
$\delta_j(\%)$	100	100	42.5	100	100	100	100	72.5	28.50	100

$$\delta_j \leq 1, \qquad j = 1, \ldots, n, \tag{15}$$

$$b \geq 0, \tag{16}$$

$$\delta_j, u_j \geq 0, \qquad j = 1, \ldots, n. \tag{17}$$

Proposition 1. *The optimal solution of (QGP) satisfies (A1)–(A5).*

Proof. See Appendix B. □

3.6. Two illustrative examples

In this section, we provide two illustrative examples of the tokens allocation algorithm. We posit that this implementation is solely meant to clarify the proposed methodology. Current and future works could focus on implementing the proposed mechanism in a common blockchain infrastructure that is a variant of Ethereum.

For each participant j, we denote by T_j ($j = 1...10$) and c_j the token balance and the order cost before clustering respectively.

First example

In the first example, we consider 10 participants whose values of (T_j, c_j) are given in Table 1. The corresponding total costs before and after aggregation are $C = 380$ and $C^* = 300$ respectively (thus, the reduction rate Δ is 21.05%), and the minimum rate a_{min} is 0.05. After solving the linear program, we get the minimum saving rate $a = a_{min} = 0.05$. Also, after solving the quadratic program, we get $b = 0.004011$, and the resulted saving rates δ_j ($j = 1...10$) are displayed in Table 2. The sum of deviations is zero.

We observe that Participant 5 who has the largest tokens balance (100) gets the highest saving rate of 45.11% while participant 9 with the smallest tokens balance (10) gets the lowest saving rate of 9.01%.

Second example

We consider a different instance having 10 participants. The values of (T_j, c_j) corresponding to this second example are given in Table 3. The total cost C is 980, the total cost after clustering C^* is 500 (hence, $\Delta = 48.98\%$), and the minimum rate a_{min} is 0.05. The resulting values of the minimum saving rate a, and b are 0.225, and 0.025833, respectively. The corresponding saving rates δ_j ($j = 1...10$) are displayed in Table 4 and the objective function (sum of deviations) is equal to 3.

We highlight that most of the participants get 100% of saving and thus they will not pay any transportation cost. However, participant 9 with the highest order cost of 500 gets the smallest saving rate of 28.50%.

Remark. Tokens can be viewed as cryptocurrencies that are created during the block validation process, when a clustering problem is solved, and are spent whenever a participant receives an additional discount on their transportation costs. It is therefore conceivable that these tokens be the subject of purchase and sale transactions.

4. Implementation remarks

The approach developed in this paper provides a new concept of the PoUW while proving its suitability and efficiency. That said, some features of the intended approach were not covered here and shall be addressed in future work. Principally, the proposed PoUW depends on time when reducing α periodically by a constant factor β. This process requires the nodes to incorporate the notion of synchronization, which is absent in current blockchain systems. Moreover, the subject of transactions may not only be limited to transportation requests. Indeed, a blockchain using our proposed PoUW can be of general purpose and considers different types of transactions, such as managing token transactions. Therefore, transportation requests and the bundling problem are only used as means for the PoUW to operate on validating transactions that may not be related to transportation at all. Future work will take into consideration a TRs pool and a transactions pool (as in Fig. 1) which might include the tokens balance for example. Furthermore, we recommend investigating advanced optimization approaches to solve the bundling problem (heuristics/metaheuristics). Such approach might either entail a unified algorithm employed by all nodes, a set of determined algorithms, or each node having its own solving mechanism. Finally, a suitable protocol is required to ensure the best interaction between the nodes while accounting for the aforementioned points. Additionally, the protocol should ensure the necessary security and incentive mechanisms to discourage nodes from acting maliciously. For instance, employing a two-step solution verification process to avoid malicious nodes "stealing" a solution and proclaiming it.

5. Conclusion

In this paper, we propose a novel Proof of Useful Work that is relevant to a blockchain which stores transportation requests. The proposed PoUW requires solving an NP-hard problem that bundles virtual coalitions of users cooperating to optimize their respective transportation costs. To the best of our knowledge, this is the first contribution towards using a PoUW that not only validates new blocks but also generates cost saving for its ecosystem. The significance of this work is pivoted on overcoming PoW's disadvantage of useless power consumption. PoUW can prove to be an efficient mechanism for validating transactions on blockchains employed in operations and finance applications. Correspondingly, the mechanism enables distributed optimization that in turn leads to distributed decision-making relevant to the ecosystem the blockchain serves. For example, a blockchain used in finance with our optimization-based PoUW, would optimize investment portfolios thereby greatly increasing the usefulness of the blockchain beyond serving as a simple distributed ledger. It should be noted that this dual role of PoUW can hardly be fulfilled by any other alternative validation mechanism, such as Proof of Stake (PoS). Not all blockchains are intended to serve applications needing optimized decision making. Still, when it is employed in settings where such problems arise (such as optimizing operations, financial portfolios, etc..), PoUW can turn into the stone that kills two birds. Hence, turning the main disadvantage of PoW, PoUW might lead to extend the use cases of blockchains beyond serving as a ledger and a virtual environment for running smart contracts.

CRediT authorship contribution statement

Mohamed Haouari: Conceptualization, Methodology, Investigation, Validation, Formal analysis, Writing – review & editing, Visualization, Supervision. **Mariem Mhiri:** Methodology, Investigation, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Mazen El-Masri:** Writing – review & editing, Project administration, Funding acquisition. **Karim Al-Yafi:** Methodology, Investigation, Writing – review & editing, Visualization.

Acknowledgment

This research was made possible by NPRPC Grant No. NPRP11C-1229-170007 from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Appendix A. Proof of Theorem 1

By reduction from the 3-partition problem.

3-partition problem instance: A finite set J of $3m$ elements, a bound $Q \in \mathbb{N}$, and a size $q_j \in \mathbb{N}$ for each $j \in J$, such that q_j satisfies:

$$\frac{Q}{4} < q_j < \frac{Q}{2}, \tag{18}$$

and such that:

$$\sum_{j \in J} q_j = mQ. \tag{19}$$

Answer: "Yes" if J can be partitioned into m disjoint subsets $J_1, J_2, \ldots, J_m$ such that, for each $1 \leq i \leq m$, $\sum_{j \in J_i} q_j = Q$. Otherwise, "No".

Now, we build a clustering problem instance as follows:

- all TR_j's share the same time window bounds (that is all TR_j's are time-compatible);

- the q_j satisfy: (a) $\frac{Q}{4} < q_j < \frac{Q}{2}$, and (b) $\sum_{j\in J} q_j = mQ$.

If the optimal cost is $mf(Q)$ then the optimal solution includes m fully loaded vessels. This can be seen as follows. First, we observe that since $f(.)$ is strictly concave then the minimum unit cost is $f(Q)/Q$. This unit cost is only achieved if a vessel is fully loaded. If all items are assigned to fully loaded vessels then the corresponding cost is $(\sum_{j\in J} q_j)f(Q)/Q$. Since $\sum_{j\in J} q_j = mQ$, then the corresponding cost is $mf(Q)$. This cost corresponds to m fully loaded vessels. It follows that the answer for 3-partition problem is "Yes".

Conversely, if the answer to the partition problem is "Yes", then the corresponding clustering solution has a cost $mf(Q)$.

3-partition is NP-complete in the strong sense, thus, the concave clustering problem is NP-hard in the strong sense.

Appendix B. Proof of Proposition 1

We just need to prove that it satisfies (A3) and (A4). Let (δ, u) denote an optimal solution of (QGP). First, we prove that (A3) is satisfied.

Proof of (A3):

Assume that $c_i = c_j$, $T_i = T_j$, and $\delta_i < \delta_j$.

$\delta_i < \delta_j$ and $T_i = T_j \Rightarrow u_i > u_j$. Consider, a solution $(\bar{\delta}, \bar{u})$ that is defined as follows:

$$(\bar{\delta}, \bar{u}) = \begin{cases} \bar{u}_k = u_k & , \quad \bar{\delta}_k = \delta_k & \text{for } k \neq i, j \\ \bar{u}_i = u_i - \lambda & , \quad \bar{\delta}_i = \delta_i + \lambda & \\ \bar{u}_j = u_j + \lambda & , \quad \bar{\delta}_j = \delta_j - \lambda & \end{cases}$$

where $\lambda = \frac{u_i - u_j}{2}$. It is easy to check that $(\bar{\delta}, \bar{u})$ is feasible. Furthermore, $\bar{u}_i^2 + \bar{u}_j^2 = u_i^2 + u_j^2 - 2\lambda^2$. Hence $\bar{u}_i^2 + \bar{u}_j^2 < u_i^2 + u_j^2$ which contradicts the fact that (δ, u) is optimal.

Proof of (A4):

Now, assume that $T_i > T_j$, $c_i \leq c_j$ and $\delta_i \leq \delta_j < 1$. Consider a solution $(\bar{\delta}, \bar{u})$ that is defined as follows:

$$(\bar{\delta}, \bar{u}) = \begin{cases} \bar{u}_k = u_k & , \quad \bar{\delta}_k = \delta_k & \text{for } k \neq i, j \\ \bar{u}_i = u_i - \epsilon_i & , \quad \bar{\delta}_i = \delta_i + \epsilon_i & \text{where } \epsilon_i > 0 \\ \bar{u}_j = u_j + \epsilon_j & , \quad \bar{\delta}_j = \delta_j - \epsilon_j & \text{where } \epsilon_j > 0 \end{cases}$$

First, we prove that $(\bar{\delta}, \bar{u})$ is feasible. From (13), we get:

$$c_i\epsilon_i - c_j\epsilon_j = 0.$$

Also, assume that $\delta_i = a^* + \frac{T_i}{c_i}$. Since, $\frac{T_i}{c_i} > \frac{T_j}{c_j} \Rightarrow \delta_i = a^* + \frac{T_i}{c_i} > a^* + \frac{T_j}{c_j} \geq \delta_j$ which is impossible. Therefore, we necessarily have $\delta_i < a^* + \frac{T_i}{c_i}$. If ϵ_i is small enough then the feasibility of (14) is ensured.

Finally, since $\delta_i < 1$ then it is always possible to set at ϵ_i an appropriate small value so that the feasibility of (12) is ensured.

The difference of the objectives of solutions (δ, u) and $(\bar{\delta}, \bar{u})$ is $\Psi = (\bar{u}_i^2 + \bar{u}_j^2) - (u_i^2 + u_j^2)$. Hence,

$$\Psi = ((u_i - \epsilon_i)^2 + (u_j + \epsilon_j)^2) - (u_i^2 + u_j^2) \tag{20}$$

Thus,

$$\Psi \approx ((u_i^2 - 2u_i\epsilon_i) + (u_j^2 + 2u_j\epsilon_j)) - (u_i^2 + u_j^2) \tag{21}$$

$$\Psi \approx (-2u_i\epsilon_i + 2u_j\epsilon_j) \tag{22}$$

Since $\epsilon_j = \frac{c_i}{c_j}\epsilon_i$, we get

$$\Psi \approx -2\epsilon_i(u_i - \frac{c_i}{c_j}u_j) \tag{23}$$

$T_i > T_j$ and $\delta_i \leq \delta_j \Rightarrow u_i > u_j$.

$c_i \leq c_j$ and $u_i > u_j \Rightarrow \frac{u_i}{c_i} > \frac{u_j}{c_j} \Rightarrow u_i - \frac{c_i}{c_j}u_j > 0 \Rightarrow \Psi < 0$, which contradicts the fact that (δ, u) is optimal.

References

Baldominos, A., & Saez, Y. (2019). Coin.AI: A proof-of-useful-work scheme for blockchain-based distributed deep learning. *Entropy*, *21*(8), 1–17. http://dx.doi.org/10.3390/e21080723.

Ball, M., Rosen, A., Sabin, M., & Vasudevan, P. N. (2017). Proofs of useful work. *International Association for Cryptologic Research*, 1–28, URL https://eprint.iacr.org/2017/203.pdf.

Bamakan, S. M. H., Motavali, A., & Bondarti, A. B. (2020). A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications*, *154*, Article 113385. http://dx.doi.org/10.1016/j.eswa.2020.113385.

Berdik, D., Otoum, S., Schmidt, N., Porter, D., & Jararweh, Y. (2021). A survey on blockchain for information systems management and security. *Information Processing & Management*, *58*(1), Article 102397. http://dx.doi.org/10.1016/j.ipm.2020.102397.

Campanile, L., Iacono, M., Marulli, F., & Mastroianni, M. (2021). Designing a GDPR compliant blockchain-based IoV distributed information tracking system. *Information Processing & Management*, *58*(3), Article 102511. http://dx.doi.org/10.1016/j.ipm.2021.102511.

CBECI (2020). Cambridge bitcoin electricity consumption index, URL https://www.cbeci.org/.

Chen, Q., Srivastava, G., Parizi, R. M., Aloqaily, M., & Ridhawi, I. A. (2020). An incentive-aware blockchain-based solution for internet of fake media things. *Information Processing & Management*, *57*(6), Article 102370. http://dx.doi.org/10.1016/j.ipm.2020.102370.

Chenli, C., Li, B., & Jung, T. (2020). DLchain: blockchain with deep learning as proof-of-useful-work. In J. Ferreira, B. Palanisamy, K. Ye, S. Kantamneni, & L. J. Zhang (Eds.), *Lecture notes in computer science*: *Vol. 12411*, *Services – services 2020. services 2020* (pp. 43–60). Springer, Cham, http://dx.doi.org/10.1007/978-3-030-59595-1_4.

Dhillon, V., Metcalf, D., & Hooper, M. (2017). *Blockchain enabled applications: understand the blockchain ecosystem and how to make it work for you*. Berkeley, CA: Apress.

Drescher, D. (2017). *Blockchain basics: a non-technical introduction in 25 steps*. Berkeley, CA: Apress.

Dwork, C., & Naor, M. (1993). Pricing via processing or combatting junk mail. In *Advances in Cryptology* - CRYPTO' *92, 12th annual international cryptology conference santa barbara, california,* USA A*ugust 16-20, 1992 proceedings, crypto 1992, In: E.F. Brickell (eds.), Lecture notes in computer science, Vol. 740*, Springer, Berlin, Heidelberg, pp. 139–147, http://dx.doi.org/10.1007/3-540-48071-4_10.

El-Masri, M., & Hussain, E. M. A. (2021). Blockchain as a mean to secure Internet of Things ecosystems - a systematic literature review. *Journal of Enterprise Information Management*, http://dx.doi.org/10.1108/JEIM-12-2020-0533.

Ferdous, M. S., Chowdhury, M. J. M., & Hoque, M. A. (2021). A survey of consensus algorithms in public blockchain systems for crypto-currencies. *Journal of Network and Computer Applications*, *182*, Article 103035. http://dx.doi.org/10.1016/j.jnca.2021.103035.

Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology*, *3*, 99–111. http://dx.doi.org/10.1007/BF00196791.

Hardin, T., & Kotz, D. (2021). Amanuensis: Information provenance for health-data systems. *Information Processing & Management*, *58*(2), Article 102460. http://dx.doi.org/10.1016/j.ipm.2020.102460.

Hu, T., Liu, X., Chen, T., Zhang, X., Huang, X., Niu, W., et al. (2021). Transaction-based classification and detection approach for ethereum smart contract. *Information Processing & Management*, *58*(2), Article 102462. http://dx.doi.org/10.1016/j.ipm.2020.102462.

Jing, N., Liu, Q., & Sugumaran, V. (2021). A blockchain-based code copyright management system. *Information Processing & Management*, *58*(3), Article 102518. http://dx.doi.org/10.1016/j.ipm.2021.102518.

Khalid, A., Iftikhar, M. S., Almogren, A., Khalid, R., Afzal, M. K., & Javaid, N. (2021). A blockchain based incentive provisioning scheme for traffic event validation and information storage in VANETs. *Information Processing & Management*, *58*(2), Article 102464. http://dx.doi.org/10.1016/j.ipm.2020.102464.

Lashkari, B., & Musilek, P. (2021). A comprehensive review of blockchain consensus mechanisms. *IEEE Access*, *9*, 43620–43652. http://dx.doi.org/10.1109/ACCESS.2021.3065880.

Li, J., Wu, J., Jiang, G., & Srikanthan, T. (2020). Blockchain-based public auditing for big data in cloud storage. *Information Processing & Management*, *57*(6), Article 102382. http://dx.doi.org/10.1016/j.ipm.2020.102382.

Lihu, A., Du, J., Barjaktarević, I., Gerzanics, P., & Harvilla, M. (2020). A proof of useful work for artificial intelligence on the blockchain. *ArXiv*, 1–24, URL https://arxiv.org/abs/2001.09244.

Loe, A. F., & Quaglia, E. A. (2018). Conquering generals: an NP-hard proof of useful work. In *CryBlock'18: Proceedings of the 1st workshop on cryptocurrencies and blockchains for distributed systems* (pp. 54–59). http://dx.doi.org/10.1145/3211933.3211943.

Manupati, V. K., Schoenherr, T., Ramkumar, M., Wagner, S. M., Pabba, S. K., & Singh, R. I. R. (2019). A blockchain-based approach for a multi-echelon sustainable supply chain. *International Journal of Productions Research*, *58*(7), 2222–2241. http://dx.doi.org/10.1080/00207543.2019.1683248.

Merlina, A. (2019). BlockML: a useful proof of work system based on machine learning tasks. In *Proceedings of the 20th international middleware conference doctoral symposium* (pp. 6–8). http://dx.doi.org/10.1145/3366624.3368156.

Mittal, A., & Aggarwal, S. (2020). Hyperparameter optimization using sustainable proof of work in blockchain. *Frontiers in Blockchain*, *3*(23), 1–13. http://dx.doi.org/10.3389/fbloc.2020.00023.

Monrat, A. A., Schelén, O., & Andersson, K. (2019). A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access*, *7*, 117134–117151. http://dx.doi.org/10.1109/ACCESS.2019.2936094.

Nair, P. R., & Dorai, D. R. (2021). Evaluation of performance and security of proof of work and proof of stake using blockchain. In *IEEE 2021 third international conference on intelligent communication technologies and virtual mobile networks (icicv)* (pp. 279–283). http://dx.doi.org/10.1109/ICICV50876.2021.9388487.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. URL https://bitcoin.org/bitcoin.pdf.

O'Dwyer, K. J., & Malone, D. (2014). Bitcoin mining and its energy footprint. In *25th iet irish signals & systems conference 2014 and 2014 china-ireland international conference on information and communications technologies (issc 2014/ciict 2014)* (pp. 280–285). http://dx.doi.org/10.1049/cp.2014.0699.

Oham, C., Michelin, R. A., Jurdak, R., Kanhere, S. S., & Jha, S. (2021). B-FERL: Blockchain based framework for securing smart vehicles. *Information Processing & Management*, *58*(1), Article 102426. http://dx.doi.org/10.1016/j.ipm.2020.102426.

Perboli, G., Musso, S., & Rosano, M. (2018). Blockchain in logistics and supply chain: A lean approach for designing real-world use cases. *IEEE Access*, *6*, 62018–62028. http://dx.doi.org/10.1109/ACCESS.2018.2875782.

Putz, B., Dietz, M., Empl, P., & Pernul, G. (2021). EtherTwin: Blockchain-based secure digital twin information management. *Information Processing & Management*, *58*(1), Article 102425. http://dx.doi.org/10.1016/j.ipm.2020.102425.

Roşu, I., & Saleh, F. (2021). Evolution of shares in a proof-of-stake cryptocurrency. *Management Science*, *67*(2), 661–672. http://dx.doi.org/10.2139/ssrn.3377136.

Saad, M., Qin, Z., Ren, K., Nyang, D., & Mohaisen, D. (2021). E-PoS: Making proof-of-stake decentralized and fair. *IEEE Transactions on Parallel and Distributed Systems*, *32*(8), 1961–1973. http://dx.doi.org/10.1109/TPDS.2020.3048853.

Salha, R. A., El-Hallaq, M. A., & Alastal, A. I. (2019). Blockchain in smart cities: exploring possibilities in terms of opportunities and challenges. *Journal of Data Analysis and Information Processing*, *7*(3), 118–139. http://dx.doi.org/10.4236/jdaip.2019.73008.

Singhal, B., Dhameja, G., & Panda, P. S. (2018). *Beginning blockchain : A beginner's guide to building blockchain solutions*. Berkeley, CA: Apress.

Vincent, J. (2019). Bitcoin consumes more energy than Switzerland, according to new estimate. URL https://www.theverge.com/2019/7/4/20682109/bitcoin-energy-consumption-annual-calculation-cambridge-index-cbeci-country-comparison.

Wan, S., Li, M., Liu, G., & Wang, C. (2020). Recent advances in consensus protocols for blockchain: a survey. In *Wireless networks, Vol. 26* (pp. 5579–5593). Springer, http://dx.doi.org/10.1007/s11276-019-02195-0.

Xu, X., Sun, G., Luo, L., Cao, H., Yu, H., & Vasilakos, A. V. (2021). Latency performance modeling and analysis for hyperledger fabric blockchain network. *Information Processing & Management*, *58*(1), Article 102436. http://dx.doi.org/10.1016/j.ipm.2020.102436.

Yu, Z., Liu, X., & Wang, G. (2018). A survey of consensus and incentive mechanism in blockchain derived from P2P. In *2018 ieee 24th international conference on parallel and distributed systems (icpads)* (pp. 1010–1015). http://dx.doi.org/10.1109/PADSW.2018.8645047.

Yu, G., Zhang, L., Wang, X., Yu, K., Ni, W., Zhang, J. A., et al. (2021). A novel dual-blockchained structure for contract-theoretic LoRa-based information systems. *Information Processing & Management*, *58*(3), Article 102492. http://dx.doi.org/10.1016/j.ipm.2021.102492.

Zhang, J. (2019). Deploying blockchain technology in the supply chain. In C. Thomas, P. Fraga-Lamas, & T. M. Fernández-Caramés (Eds.), *Computer security threats* (pp. 1–16). IntechOpen, http://dx.doi.org/10.5772/intechopen.86530, URL https://www.intechopen.com/books/computer-security-threats/deploying-blockchain-technology-in-the-supply-chain.

Zhang, R., & Chan, V. W. K. (2020). Evaluation of energy consumption in block-chains with proof of work and proof of stake. *Journal of Physics: Conference Series*, *1584*, http://dx.doi.org/10.1088/1742-6596/1584/1/012023.

Zhao, Q., Chen, S., Liu, Z., Baker, T., & Zhang, Y. (2020). Blockchain-based privacy-preserving remote data integrity checking scheme for IoT information systems. *Information Processing & Management*, *57*(6), Article 102355. http://dx.doi.org/10.1016/j.ipm.2020.102355.