Optimizing the energy consumption of blockchain-based systems using evolutionary algorithms

Alofi, Akram; Bokhari, Mahmoud A. ; Bahsoon, Rami; Hendley, Robert

DOI:
10.1109/TSUSC.2022.3160491

Optimizing the Energy Consumption of Blockchain-based Systems Using Evolutionary Algorithms: A New Problem Formulation

Akram Alofi, Mahmoud A. Bokhari, Rami Bahsoon and Robert Hendley

Abstract—Blockchain technology has gained recognition in industrial, financial, and various technological domains for its potential in decentralizing trust in peer-to-peer systems. A core component of blockchain technology is a consensus algorithm, most commonly Proof of Work (PoW). PoW is used in blockchain-based systems to establish trust among peers; however, it does require the expenditure of an enormous amount of energy that affects the environmental sustainability of blockchain-based systems. Energy minimization, whilst ensuring trust within blockchain-based systems that use PoW, is a challenging problem. The solution has to consider how energy consumption can be minimized without compromising trust, whilst still ensuring, for instance, scalability, security, and decentralization. In this paper, we represent the problem as a subset selection problem of miners in a blockchain-based system. We formulate the problem of blockchain energy consumption as a Search-Based Software Engineering problem with four objectives: energy consumption, carbon emission, decentralization, and trust. We propose a model composed of multiple fitness functions. The model can be used to explore the complex search space by selecting a subset of miners that minimizes the energy consumption without drastically impacting the primary goals of the blockchain technology (i.e., security/trustworthiness and decentralization). We integrate our proposed fitness functions into five evolutionary algorithms to solve the problem of blockchain miners selection. Our results show that the environmental sustainability of blockchain-based systems (e.g. reduced energy use) can be enhanced with little degradation in other competing objectives. We also report on the performance of the algorithms used.

Index Terms—Search-Based Software Engineering, Blockchain, Mining, Optimization, Evolutionary Algorithms, Sustainability.

1 Introduction

In 2008, a new decentralized cryptocurrency was proposed, Bitcoin [1] that relies on blockchain technology. Since then, blockchain has attracted considerable interest from the research and industrial communities, and it has been applied in a variety of fields, including finance, manufacturing, and academia [2]. Despite the great potential of blockchain technology, there is an important issue regarding its energy consumption. In the context of the existing debate concerning sustainability and global warming, such a perspective could result in constraining or postponing the global adoption of this technology [3]. Therefore, the question of optimizing and finding solutions for this issue is currently receiving much attention.

Optimization is a fundamental technique in all branches of applied mathematics, engineering, medical science, economics, and other sciences. The most recent advancements in the previous few decades have largely relied on meta-heuristic algorithms. Indeed, meta-heuristic algorithms are used in the great majority of modern optimization techniques in all important disciplines of engineering and science, as well as industrial applications. Also, meta-heuristic algorithms, such as Particle Swarm Optimization, Genetic Algorithms, etc. have become increasingly powerful in tackling hard optimization problems [4].

The amount of energy consumed by blockchain-based systems is critical. According to [5], one Bitcoin transaction uses approximately the same amount of energy as that consumed by the average British household in eight weeks. Moreover, as of 31 March 2021, Cambridge Bitcoin Electricity Consumption Index (CBECI) estimated that the Bitcoin network consumes 137.20 terawatt-hours of electricity per year [6].

Due to the huge amount of energy that can be consumed by blockchain-based systems, researchers have proposed more sustainable and energy-efficient mechanisms for trustworthy verification. Solutions include new consensus algorithms, regulatory mechanisms, and fiscal policies, as well as limiting the use of these systems. However, research has not utilized Search-Based Software Engineering (SBSE) techniques to solve the problem of blockchain energy consumption. In SBSE, software engineering problems are converted into a search problem, and then search-based optimization algorithms are used to find optimal and near-optimal solutions.

Optimizing the energy consumption by selecting miners in a blockchain-based system is an SBSE problem that can be considered as an instance of a subset selection problem. This

- *A. Alofi is with the School of Computer Science, The University of Birmingham, Birmingham, United Kingdom and also with the Computer Science Department, Jamoum University College, Umm Al-Qura University, Jamoum, Kingdom of Saudi Arabia.*
  E-mail: ama848@cs.bham.ac.uk
- *M. Bokhari is with the Computer Science Department, Taibah University, Medina, Kingdom of Saudi Arabia and also was with Optimization and Logistics, School of Computer Science, The University of Adelaide, Adelaide, Australia.*
  E-mail: mabokhari@taibahu.edu.sa
- *R. Bahsoon and R. Hendley are with the School of Computer Science, The University of Birmingham, Birmingham, United Kingdom.*
  E-mail: {r.bahsoon, r.j.hendley}@cs.bham.ac.uk

Manuscript received Jun xx, 2021; revised March xx, 2022.

has a time complexity of $O(2^n)$ and is an NP-hard problem. As the number n increases, the number of possible solutions increases exponentially. Approximation algorithms can be used to generate good solutions to such problems. One category of approximate algorithms is evolutionary algorithms (EAs), which evolve a set of optimal solutions. Although several optimization solutions use heuristic or meta-heuristic techniques, the problem proposed in this study has not been previously reformulated using meta-heuristics through evolutionary computing. Therefore, this study intends to show the possibility of reformulating the miners' subset selection problem in blockchain-based systems as an SBSE problem. Also, it can help researchers and developers to reformulate the problem into a search problem, as needed.

Similarly to the majority of real-world applications, blockchain-based systems require trade-offs. In such systems, many objectives can be balanced which can be viewed as a distinct optimization challenge. In blockchain-based systems, there are conflicting objectives that include security, scalability, energy consumption, performance, decentralization, and trust. Although different well-developed optimization techniques are available and show promise to address optimization problems in many fields, minimizing the energy consumption of blockchain-based systems has not been formulated as an optimization problem and solved using EAs.

In this paper, we reformulate the problem of blockchain-based energy consumption as an SBSE problem: miners subset selection problem. We represent the problem as a set of participating miners within the system. Each miner consumes energy and emits carbon to add blocks and has a reputation score. The features of miners in a set, constitute a decentralization score, and as the score decreases, the system becomes a centralized system. With such trade-offs, we propose the use of EAs to optimize the energy consumption of blockchain-based systems by selecting miners that minimize the energy use and carbon emissions while maximize other conflicting objectives. In this work, we use four different fitness functions: energy versus reputation, energy versus carbon versus reputation, energy versus decentralization versus reputation, and energy versus carbon versus decentralization versus reputation.

The main contributions of this paper are summarized as follows:

- We formulate the problem of selecting miners within blockchain-based systems as a SBSE problem.
- A novel optimization model for the problem of selecting miners within blockchain-based systems is proposed using four different fitness functions for optimizing the energy consumption of blockchain-based systems.
- We conducted an experimental evaluation to show the efficacy of the proposed model in saving energy.
- A comparison among EAs is presented to analyze their performance in solving the problem of selecting miners within blockchain-based systems.

The rest of the paper is organized as follows. We briefly give the background of blockchain technology in Section 2. Section 3 introduces the related work. The details of our optimization problem are presented in Section 4. The experiment design is explained in Section 5. We present our results and discussion in Section 6. Finally, Section 7 presents our conclusions and potential future work leading from this paper.

2 Blockchain Background

The appearance of Bitcoin has resulted in the widespread adoption of blockchain technology. In blockchain-based systems, encrypted ledgers are maintained within a database that is publicly distributed. Blockchain-based systems consistently accumulate data over a wide network among nodes in a trusted environment that does not require third-party contributions. This innovative technology has drawn a great deal of interest for both business and academic purposes, thanks to offering a range of qualities that include privacy, reliability, anonymity, and decentralization [7]. Currently, blockchain has been incorporated into a wide range of applications, such as Internet of Things (IoT) services, Finance initiatives, and Supply Chain Economics [2].

Blockchain-based systems can best be understood as a sequence of blocks similar to public ledgers that accept and maintain transaction data, which is interconnected by a reference hash belonging to the previous block (hash block). The process begins with a genesis block that is also called a 'starting' block. Each block is made up of a block body that consists of both transactions and a transaction counter, together with a header. The header incorporates a range of metadata including timestamp, previous block hash, Merkle tree root hash, nonce, nBits, and the block version [7].

Three forms of blockchain-based systems can be considered to exist: private, public, and consortium [7]. As the name suggests, private blockchains are managed by a designated user or organization and a list of predefined validators. However, the content of blocks can be visible to any node without permission or a list of predefined nodes with permission. A public blockchain not only publicly displays the content but allows anyone to participate in verifying transactions and adding a new block to the blockchain-based system. In contrast, consortium blockchains are structured to allow a set of determined participants, such as universities, to confirm transactions and add blocks to the system. Nevertheless, blocks can be accessed by restricted groups or can be open to everyone. In regards to the energy consumed by miners in each type of blockchain, public blockchain consumes a huge amount of energy because of the large number of miners participating in mining blocks. In contrast, private and consortium blockchains could be more efficient due to the limited number of miners. Table 1 shows comparison of public, private, and consortium blockchain.

One of the building blocks of blockchain-based systems is a consensus algorithm that makes decisions on the way agreements are established among all nodes on the verifying network and for appending a new block. The most primitive, commonly used consensus algorithm in blockchain-based systems is Proof of Work (PoW) [8].

In PoW, every node in a network calculates a hash value for the block header, which constantly changes. To achieve a consensus, the calculated value must be equal to or smaller

TABLE 1
A Comparison of Public, Private, and Consortium Blockchain

	Public	Private	Consortium
Consensus Determination	All miners	One organization	A selected number of nodes
Consensus Process	Permissionless	Permissioned	Permissioned
Centralization	No	Yes	Partially
Transactions Reading	Public	Can be restricted or public	Can be restricted or public
Transaction Processing Speed	Slow	Faster than Public	Faster than Public
Identity	Anonymous	Predefined participants	Predefined participants
Immutability	Almost impossible to tamper	Could be tampered	Could be tampered
Propagating Transactions	Low efficient	High efficient	High efficient
Energy Efficiency	High	Low	Moderate

than a particular predefined value. In a decentralized network, every participant must continuously undertake the calculation of the hash value by employing a number of nodes until the target is achieved. The nodes employed for calculating the hash are referred to as miners, with the PoW procedures being referred to as mining [7]. To solve a PoW puzzle and add suitable blocks, each miner needs enough computing power to find the hash quickly. This can be achieved by adding a random value called a nonce to a set of transaction that will be in a new block of the blockchain [7]. Therefore, considerable computational resources are needed for the PoW mechanism of those applications using this algorithm.

3 Related Work

Due to the excessive energy consumption of PoW, several consensus algorithms have been proposed to reduce energy consumption and improve environmental sustainability by reducing the number of miners in the competition to find the nonce. These algorithms track the miners' behaviors over a period of time and then calculate their reputation or trust values. These values are used to allow miners to add blocks. In [9], the authors propose a new consensus algorithm, called Proof of Trust (PoT), which selects miners randomly based on a trust graph built from the miners' network. A miner's trust value is considered a waiver for mining difficulty in PoW to add blocks. Proof of Reputation (PoR) is a consensus algorithm suggested in [10] in which the reputations of nodes are evaluated based on transaction activities, assets, and consensus participation for a node. Unlike these studies, our work does not change the architecture of the blockchain-based systems that rely on PoW, it changes the mechanism of choosing the appropriate miners that improve the environmental sustainability of mining blocks.

Since the consensus algorithm is a primitive component of blockchain-based systems, several studies propose alternative consensus algorithms to reduce the energy consumption of such systems. For example, Proof of Stake (PoS) [11] which uses miners' stake for deciding which miners should add the next block. Other techniques require miners to invest in specific hardware, such as high storage devices (e.g. Proof of Space [12]), or special Intel-based hardware (e.g. Proof of Luck [13]). Our approach does not propose a new consensus algorithm, rather it is used within the most common consensus algorithm (i.e., PoW). In addition, it does not require investing in specific hardware.

Several optimization models are proposed in the literature to reduce the environmental impact related to energy consumption. The problem of scheduling is one of the problems that has been formulated as a multi-objective optimization problem for optimizing energy consumption in many areas, such as heterogeneous computing systems [14], cloud computing [15], wireless sensor networks [16], and multicore processors systems [17]. Also, another energy-efficient optimization problem is formulated in various works regarding the offloading process. These studies aim to reduce the energy consumption of offloading processes in diverse domains including cloud computing [18] and mobile edge computing for the IoT [19]. Clustering techniques are employed to optimize energy consumption as well in different fields, such as cloud computing [20] and wireless sensor networks [21].

There is other work that solves blockchain-based system problems using EAs. For example, [22] proposes a Pareto-based technique to detect major influencers in a blockchain-based system. In addition, [23] proposes a transaction selection process using a combination of large deviation theory and Lyapunov optimization.

SBSE techniques have been utilized successfully to improve the non-functional properties of software. The study by [24] applied Genetic Improvement (GI) of software for trading-off energy consumption with the functional properties of software running on a Raspberry PI; the study presented in [25] utilizes *in-vivo* optimization using GI to achieve a trade-off between the energy consumption of Rebound (an animation library for Android, written in Java) and its output accuracy; the study [26] implements a multi-objective approach to optimize the energy use of Android applications by changing 'GUIs' color palettes; in [27] they fix object-oriented and energy anti-patterns by finding an optimal set of refactoring sequences to maximize the number of fixed anti-patterns.

Additionally, software runtime and memory consumption have been optimized using SBSE. For instance, the study [28] has applied GI on the Viola-Jones algorithm (a face detection algorithm in the openCV library) to trade-off its functionality with its runtime; the study [29] has utilized a multi-objective approach to speed-up the runtime of shader software by degrading its output graphics.

Our work is different from the above studies as none of these studies solves or discusses the most serious problem of blockchain based-systems, energy consumption, using SBSE techniques. In addition, we trade-off two of the main objectives related to the environmental sustainability dimension: energy consumption and carbon emission. We compromise these objectives with other non-functional properties

of blockchain-based systems, namely decentralization and trustworthiness. Table 4 [1] shows a summary of each study in this section representing the problem solved by the study, the optimization technique used, and its area.

4 Optimization Problem Formulation

Similarly to many real-world systems, blockchain-based systems have trade-offs among different objectives that are considered as one kind of optimization problem. According to [30], there are many conflicting blockchain objectives, such as trust and energy consumption that can be used for optimizing blockchain-based systems.

We posit that trust provisioning within blockchain-based systems is expensive, both computationally and in terms of energy. We consider energy consumed for managing trust within these systems to be an optimization problem. The problem is represented as a subset selection problem of miners participating in a blockchain-based system. To solve the above problem, in which there is a trade-off between energy consumption and other conflicting objectives, we present a novel optimization model that can enhance the environmental sustainability of blockchain-based systems. Our model is generally applicable to scenarios where miners are predefined and controlled. Consortium and private Blockchain-based systems can benefit from our model as the participating miners are often managed and predefined. Public blockchain-based systems can still benefit from our model if a global standard or policy for selecting miners to mine blocks is specified.

Blockchain-based systems, especially those that use a Proof of Work consensus algorithm, have a scalability issue [31] that limits the capability of these systems to handle a large amount of transaction data in a short time. Since a dynamic optimization could increase the overall mining process overhead that may affect the scalability of blockchain-based systems, our model employs a static optimization style that performs optimization first then applies. While researchers attempt to shorten the time needed to create a new block [32], we believe that a dynamic optimization model can increase the time for adding blocks to the chain. Applying a dynamic optimization may cause a delay in the processing of mining blocks. In other words, a large number of transactions will have to wait for a long time because of the time spent for the optimization model to select optimal miners and for adding the transactions to the chain, especially for blockchain-based systems with a large number of users. Although we employ a static optimization model, our model can be employed to perform a dynamic optimization that can select optimal miners during run-time (i.e., select optimal miners after each mined block).

In this paper, we reformulate the problem of minimizing the energy consumption and carbon emissions of blockchain-based systems by reducing the number of miners. Moreover, this formulation includes maximizing the trust level of these systems not only by selecting miners with high reputation values but also by the degree of decentralization in the blockchain network where decentralized trust is fundamental to the operations of these systems. We follow the definition of trust in [33] that describes trust as "the firm belief in the competence of an entity to act dependably, securely, and reliably within the specified context". Also, we use the reputation definition that is stated in [34]. The authors describe reputation as "an expectation about an agent's behavior based on information about its past behaviour". We consider a blockchain-based system as an entity and a miner as an agent. There is an inherent trade-off between the number of miners, energy consumption, decentralization, and miners' reputations within blockchain-based systems [35]. The more miners a blockchain network has, the more energy is consumed, the greater the levels of carbon emissions, and the more decentralized and trusted it becomes. Furthermore, better decentralization of miners leads to greater resistance against censorship of individual transactions and, consequently, greater trust in the system.

4.1 Solution Representation

Solution representation determines how the problem is structured in the EAs, as well as the genetic operators that can be used. In the proposed model, the chromosome representation is an array of nodes which represents a set of miners in a blockchain network. The length of chromosomes (number of genes) is exactly equal to the number of miners that participate in the mining process. Each gene X_i holds a *Boolean* value which determines whether a miner is included.

4.2 Optimization Model

In this model, we devise four objective functions that are mathematically formulated to minimize the total energy consumption and the total carbon emissions produced by blockchain-based systems. These also maximize trust levels based on maximizing the degree of decentralization and the reputation values for miners within blockchain-based systems. A list of notations used and their description are presented in Table 5 [2].

4.2.1 Energy Consumption Objective

This paper focuses on enhancing the sustainability of blockchain-based systems through the saving of energy used in computing procedures by miners, which accounts for the bulk of blockchain-based systems' energy consumption. Power is a measurement of the rate at which energy is used or work is performed by a system over a period of time [36]. From this definition and due to the relationship between power, energy, and time, the energy consumption for each miner EM ($kilowatt - hour$) can be calculated by:

$$EM = \frac{\sum_{i=1}^{mD}(P_i \times T_i)}{1000} \quad (1)$$

where P_i is the amount of power used by a mining device i that is needed for all mining device components including processor and memory ($watt$), T_i is the hours of participating in the blockchain network per day ($hours$), and mD is the number of mining devices since one miner could have more than one device.

1. Available in the online supplemental material (see Appendix A)

2. Available in the online supplemental material (see Appendix B)

As the optimization objective is to minimize the total energy consumption ET ($kilowatt - hour$) of all participating miners in a Pareto front's solution, the smaller the energy value is, the fitter the solution is. We optimize the energy consumption as follows:

$$\text{Minimize:} \qquad ET = \sum_{i=1}^{m} X_i \times EM_i \tag{2}$$

where m is the total number of miners that compose the blockchain network, and X_i is the value of each gene in a solution representation. It can be either '1', which denotes the miner is selected for participation in the mining process for the next block, or '0', which denotes a non-selected miner.

4.2.2 Carbon Emission Objective

The carbon emission of electricity can be defined as the greenhouse gas emitted for producing or using a certain amount of electricity, which indicates that lowering the energy use by blockchain-based systems will actually reduce greenhouse gas emissions. Thus, the carbon emissions caused by the electricity used by a mining device can be defined as:

$$CM = EF \times EM \tag{3}$$

where *CM* is greenhouse gas emissions produced by a miner in grams (g), *EF* is the emission factor of electricity in the miner's location (gCO_2eq/kWh), and EM is the energy consumption for each miner ($kilowatt - hour$) calculated using Equation 2.

We optimize the total carbon emission CT generated by all participating miners in a Pareto front's solution as follows:

$$\text{Minimize:} \qquad CT = \sum_{i=1}^{m} X_i \times CM_i \tag{4}$$

4.2.3 Decentralization Objective

In distributed systems, decentralization means that systems do not rely on a central party among connected and distributed nodes or peers [35]. This guarantees that a single authority or a group of authorities cannot control the assets in the system or impose any change without consent from other users. In blockchain-based systems, one way of quantifying decentralization is based on the number of miners that participate in the mining process. Specifically, it is useful to look at the number of miners, or how many organizations control the nodes, and their power expressed in hashrate. A network's destiny is controlled by the hashrate power held by miners. Thus, there is no benefit of having 1000 miners competing if one miner has a 51% hashrate in the network. This is because this miner would then have the chance to control the whole network. The key point is to look at which individual has the highest hashrate or creates the most blocks. Decentralization is important for how the system is controlled. When a system has a high degree of decentralization, it means the system has greater strength against attacks and tampering, which leads to a high level of trust in the system [35].

It is critical to have scientific measurements of decentralization in order to assess the level of decentralization for blockchain-based systems. Several fields have used entropy for the quantification of the randomness or uncertainty of a specific mechanism or event [35]. Taking a blockchain-based system as a source of information, modeling can be used with the system serving as a random variable. In this case, the quantity of information a source puts out represents the quantity of uncertainty that exists before the release of information. In blockchain-based systems, estimations can be made of how probable it is that a miner can mine the next block, on the basis of its hashrate. Following the models proposed in [35] and [37], we can calculate the self-information of the event mining blocks for a miner to use with Shannon's entropy [38].

Since decentralization is one of the core features in blockchain, we want to use this valuable feature as one objective of our model. We use Shannon's entropy to quantify decentrality D based on the distribution of miners' hashrate to prevent one miner from mining all blocks and taking control of the blockchain-based system (i.e., 51% attack). The optimization of this objective is defined as:

$$\text{Maximize:} \qquad D = -\sum_{i=1}^{m} X_i \times (FH_i \times \log_2 FH_i) \tag{5}$$

where m is the number of miners in a blockchain-based system, and FH_i is the fraction of the hashrate of a miner in a Pareto front's solution. The FH_i is calculated using the following:

$$FH_i = \frac{h_i}{h_t} \tag{6}$$

where each miner's hashrate is represented by h_i, and the total hashrate of participating miners in the solution is h_t.

4.2.4 Reputation Objective

In our model, the number of miners will be reduced, so we need to support the PoW consensus algorithm by increasing the trust level for blockchain-based systems. The trust level can be raised by calculating the reputation value for each miner. We can use certain trustworthiness evaluation models to compress a miners historical activities into a reputation value for each miner. Since building a trust or reputation model is not an essential contribution of this paper, we adopt a simple model inspired by the ideas of PoW and PoS.

In our optimization model, we use a sigmoid function to evaluate the trustworthiness of miners within a blockchain network after each published block. We collect two features about each miner and use them to calculate their reputation values. The first feature is the number of blocks a miner has added to the blockchain while the second is the stake the miner has. Similarly to PoW, we assume that the miner will not assault the network after doing a lot of work with significant requirements. Furthermore, the miner's ownership of the amount of currency should be a protection against attacks on the network because miners do not want to lose their coins, as with PoS. In this model, the reputation value for the miner is the sum of the sigmoid function for

each feature. Thus, the reputation value for each miner RM within a blockchain network can be calculated as:

$$RM = \sum_{i=1}^{B} \left(\frac{1}{1+e^{-b}} + \frac{1}{1+e^{-s}} \right) \tag{7}$$

where B is the total number of mined blocks in the blockchain, b is the number of blocks mined by a miner, and s is the total of fees and rewards the miner has.

The last objective to maximize is the total reputation RT of participating miners in a Pareto front's solution, which in turns maximizes the trustworthiness of the blockchain network. It is worth mentioning that the level of trust of a Pareto front's solution does not follow the number of miners. However, the highest trust of a blockchain-based system can achieve when all miners are participating in mining processes. There are some Pareto front's solutions that show a lower level of trust with a high number of miners compared with other solutions that have a low number of miners.

$$\text{Maximize:} \qquad RT = \sum_{i=1}^{m} X_i \times RM_i \tag{8}$$

4.2.5 Fitness Functions Constraints

Equations (2), (4), (5), and (8) share same constraints, as follows:

$$H_c < TL\% \sum_{i=1}^{m} X_i \times h_i - H_c$$
$$\sum_{i=1}^{m} X_i > 1$$
$$X_i \in \{1, 0\}$$

where h_i is the hashrate for a miner i in the blockchain network, H_c is the hashrate for a current miner that will be compared to other miners' hashrate, and TL is the percentage tolerance level that must be identified by a decision-maker for the system. The decision-maker can determine the TL to be 50% or less.

These constraints ensure that a miner's hashrate should be less than the TL, such as 50% or 30%, of the total hashrate for all other miners in the individual solution. When a malicious miner has a total hashing power above 50% or 30% of the whole network's hashing powers, a double-spending attack or a selfish mining attack can be launched [39], [40]. Therefore, this constraint ensures avoiding such a vulnerability. Also, they ensure that more than one miner should participate in the mining process to prevent centralization.

We use the above objectives to form four fitness functions: energy versus reputation, energy versus carbon versus reputation, energy versus decentralization versus reputation, and energy versus carbon versus decentralization versus reputation. In each fitness function, we have at least one pair of a conflicting objective.

5 Experiment Design

5.1 Research Questions

In this paper, our proposed model aims to improve the environmental sustainability of blockchain-based systems by using evolutionary algorithms for finding a set of miners. The study endeavors to answer four research questions:
RQ1: To what extent can our optimization model reduce the energy consumption of a blockchain-based system?
RQ2: To what extent can our optimization model reduce the carbon emission of a blockchain-based system?
RQ3: Are the selected evolutionary algorithms effective to solve our blockchain miner selection problem compared with Random Search (RS)?
RQ4: Among the used algorithms, which algorithm can achieve the best performance?

5.2 Evaluation Procedure

Now, we present the evaluation procedure used to answer our research questions.

5.2.1 Energy Consumption and Carbon Emission

To answer **RQ1 and RQ2**, we compare each algorithm's best solution in terms of energy use and carbon emission, and the degradation in the conflicting objective(s) compared to the original solution. The original solution is the complete set of miners within a blockchain-based system.

5.2.2 Performance Metrics

To compare algorithms' performance, we use the hypervolume metric, which computes the d-dimensional volume of the dominated portion of the objective space by the non-dominated solutions from a reference point [41]. In other words, the performance metric in this paper means the algorithm's ability to evolve non-dominated solutions that cover as much as possible of the solution space. The reference point that we use is the worst possible value for each objective (i.e., zero for maximization and max double for minimization problems). We use this metric since it compares algorithms in terms of diversity and convergence. This metric is widely used in the literature for the evaluation of algorithms' performance and for the solution selection procedures [42]. The higher the hypervolume value of an algorithm, the better the performance.

To answer **RQ3**, we compare the selected algorithms' hypervolume with the hypervolume of RS's non-dominated set. The comparison includes showing statistical differences between RS and other algorithms using the right-tailed Wilcoxon rank-sum test [43]. This conservative non-parametric test makes no assumptions about the datasets' distribution. The null hypothesis states that the hypervolume values of algorithm X are greater than RS's hypervolume values. We use the Wilcoxon test because most of the resulting datasets have non-normal distributions. The statistical technique used to determine whether an algorithm's hypervolume values come from a normal distribution is the Shapiro-Wilk test [44].

We then use the Vargha and Delaney $\hat{A}_{12}$ effect size to measure the approximate differences between the RS

performance and the selected algorithms. $\hat{A}_{12}$ is a non-parametric measure and calculates the proportional difference between two datasets [45]. For interpreting the effect size, this approach measures the quantity of the difference in four ranges: no difference (0.5), a negligible difference (up to 0.56), a small effect (up to 0.64), a medium difference (up to 0.71), a large difference (larger than 0.71). This approach calculates the expected probability that algorithm 1 performs better than algorithm 2. For instance, if $\hat{A}_{12} = 0.8$, then algorithm 1 is expected to outperform algorithm 2, 80% of the time.

To answer **RQ4**, we conduct a pairwise comparison between every pair of the selected algorithms using Wilcoxon rank-sum test and $\hat{A}_{12}$ effect size.

5.3 Selected Evolutionary Algorithms

As we introduce a new optimization problem (blockchain miner selection problem), we integrate our proposed fitness functions into five EAs that each have different mechanisms to preserve solution diversity. For example, Non-dominated Sorting Genetic Algorithm II (NSGA-II) creates niches by computing a crowding distance for each solution and uses the crowding distance in its selection operator to promote diversity [46]. To diversify solutions, Strength Pareto Evolutionary Algorithm 2 (SPEA2) [47] uses an external archive to store the non-dominated solutions. For each solution, it calculates how many solutions dominate it and the number of solutions it dominates. SPEA2 also uses a nearest neighbor density estimation technique to guide the search efficiently. Similarly to SPEA2, Pareto Archived Evolution Strategy (PAES) [48], which is a mutation-only algorithm, uses a d-dimensional archive (d = the number of objectives) as a reference set when it creates new solutions. To promote diversity, PAES divides the objective space into grids and places each solution in a certain grid according to the solution's objective values. A crowding measure is computed using the density of solutions in each grid. The crowding measure is used in ranking the non-dominated solutions in a way that prefers non-dominated solutions belonging to the least crowded regions.

One of the main performance indicators is the hypervolume, which computes the dominated proportion of the search space by the found solutions [41]. The greater the hypervolume is, the better performing the algorithm is. Indicator-based Evolutionary Algorithm (IBEA) [49] uses the hypervolume indicator to rank solutions. It calculates how much volume each solution contributes to the overall Pareto front's hypervolume. Solutions with higher hypervolume values are preferred. As a result, this process maximizes the final Pareto front domination of the search space.

We also use Non-dominated Sorting Genetic Algorithm III (NSGA-III) which is an improved version of NSGA-II for many-objective problems [50]. To preserve diversity, the NSGA-III algorithm uses a set of well-spread reference points that represents interesting directions in the fitness landscape and virtually represents the Pareto front. As the search process progresses, the algorithm updates the set, as well as niches created around these reference points. The use of predefined points divides the search space into multiple targeted searches for the algorithm instead of one massive search space. This alleviates the problem of computing a diversity score for every solution by selecting solutions from different niches instead of computing the crowding distance. In addition, it reduces the massive number of non-dominated solutions in many-objective problems, as each optimal solution corresponds to a targeted search segment.

We use the algorithms discussed above as they have different mechanisms for preserving solution diversity, which helps to navigate the search space efficiently [51]. We focus on the native algorithms that are implemented within the MOEA Framework and that support all functionality provided by the MOEA Framework. The selected algorithms are well-suited for solving similar problems to ours [52], [53], [54], [55], [56]. These algorithms can be used with binary-coded solutions. In this paper, the solution encoding is Boolean (Binary-coded).

TABLE 2
Implementation Details

Variable	Value
Number of Miners	160
Network Total Block Number	4073
Bitcoin Network Hashrate	107,611,000.0 TH/s
Bitcoin Reward	6.25 BTC
Bitcoin Fee	0.00028188 BTC
Bitcoin Average Transaction Size	250 Byte
Mining Device Hashrate	110 TH/s
Mining Device Power	3,250 Watt
Miner Running Time	24 Hours
Percentage of Tolerance Level	50%

5.4 Implementation Details

The details of the implementation used to run our experiments are presented in the sections below. A summary of the implementation details is presented in Table 2.

5.4.1 Bitcoin Simulator Settings

Simulation methods are used in multiple fields of science. The simulations enable us to obtain insight into a system's behavior and simplify the deployment and implementation of protocols. Simulations allow the investigation of large-scale systems with a large number of nodes using one machine and also to get findings in a reasonable time. Within large-scale blockchain networks, there are difficulties in procuring information related to the entire network, except where nodes offer information regarding themselves. Also, it is not usually possible for the actual behavior within a large-scale network to be observed. For this reason, it is neither feasible and practical to undertake experimentation within large-scale blockchain networks. Although the case study and evaluation were conducted in a controlled and simulated environment, the evaluation was careful to emulate those dynamics that can stress systems at scale.

Here, we use a blockchain simulation framework called Bitcoin-Simulator [8] that uses real and artificial data, such as the distribution of miners' hashrates and locations. It is a widely used simulator for the blockchain environment. Bitcoin-Simulator simulates the working of blockchain-based systems that use the PoW consensus algorithm and their network layers. Thousands of nodes and events can be tracked by the simulator. It replicates the PoW process

TABLE 3
The Distribution of Miners' Locations and their Hashrates Percentages.

Country	Hashrate Percentage	Country	Hashrate Percentage
Canada	0.8%	Kazakhstan	6.2%
China	65.1%	Malaysia	4.3%
France	0.2%	Norway	0.5%
Germany	0.6%	Paraguay	0.3%
Iceland	0.4%	Russia	6.9%
India	0.1%	Thailand	0.3%
Iran	3.8%	United States	7.4%
Italy	0.3%	Venezuela	0.4%

for miners within a blockchain network by assigning each miner a particular mining hashrate and location. We utilize the simulator to collect data that are used to implement our optimization model. The data collection involves running the simulator to mine 4073 blocks which is equivalent to one month of mining. There are 160 miners in the simulation. We have set the percentage tolerance level in the blockchain-based system to be 50% to prevent miners from performing a double-spending attack.

To replicate a real-world scenario of a blockchain-based system, we use the basic properties of Bitcoin's network, such as the hashrate, rewards, and fees as shown in Table 2 [3]. We use Bitcoin as it is the most widely known blockchain-based system [57]. We determine the distribution of miners' locations and their hashrate percentages based on information retrieved from CBECI published in [6]. Table 3 shows the distribution of miners' locations and their hashrates percentages of Bitcoin's network hashrate that are divided into 16 countries where each country has ten miners.

5.4.2 MOO Model Assumptions

In blockchain networks, we cannot estimate accurately how much electricity is used for mining operations, as it's impossible to determine how many mining machines are in a network or which machines are active at any given time [58], [59]. In order to determine the number of mining devices in a blockchain network, we first assume that all miners use the most efficient mining device and that as a miner's hashrate increases, their number of devices increases. We base our assumption on the fact that using inefficient devices leads to leaving the network as a consequence of not receiving profits from successful mining [58], [59]. In addition, we do not assume that a miner would have a high number of traditional devices that use CPUs and GPUs due to their inefficiencies compared to the current state-of-art Application-Specific Integrated Circuit (ASIC). Consequently, the number of devices for each miner is found by dividing the hashrate for each miner by the hashrate for the selected mining device type. The power of this device is also used to calculate the energy consumption of each miner. We use the hashrate of Antminer S19 produced by Bitmain Technology Holding Company (Bitmain) [4]. Its hashrate can reach 110 TH/s, and its mining power is 3,250 $watt$.

Moreover, we assume that miners try to mine blocks for 24 hours because they want to gain profit following the same assumption published in [59], [60]. For calculating the carbon emission, we first use the distribution of miners' locations retrieved from CBECI and set into the simulator. Then, we use miners' locations to calculate the carbon emission produced by each miner using emissions factors published for miners' countries in [61].

5.4.3 Experiment Settings

We integrated our proposed fitness models discussed in Section 4.2 with five evolutionary algorithms. We use the Random Search (RS) as a baseline for our comparison. For the algorithm implementations, we used a Java-based Multi-Objective Evolution Algorithm framework (MOEA Framework) [5]. For each algorithm, we leave all variation operators and variation probabilities at their default values (see Table 6 and 7 for these values [6]. Each algorithm is run with 40,000 fitness evaluations. To account for the stochastic nature in the algorithms used, we run each algorithm 100 times. All experiments were performed on a Windows 10 machine with 24GB memory and Intel i7-6700 CPU clocked at 3.4GHz.

6 Results and Discussion

In this section, we present the results of our experiments. First, we group our experiments based on the proposed fitness functions presented in Section 4.2. Then, we present our results and a discussion of **RQs1-4**.

To investigate the performance of the algorithms on the real-world blockchain miners selection problem, we need to compute the true Pareto front. Similarly to [62], [63], we compute the Pareto front by combining the outcomes of 500 independent runs of the algorithms for each proposed fitness function.

6.1 Objectives Space Results

6.1.1 Energy and Reputation Objective Space

Figure 1 shows the approximated Pareto front found by each algorithm during the 100 runs in blue and the Pareto front in black. The x-axis shows the energy consumption, and the y-axis presents the reputation score calculated by Equation 7. As can be seen, NSGA-II, SPEA2, IBEA, and NSGA-III consistently find better non-dominated solutions from the Pareto fronts. Clearly, PAES's non-dominated solutions are distant from the Pareto front, which shows that having only a mutation operator promotes exploitation over exploration. This is because mutation operators exploit the neighboring areas of the current solution whereas crossover operators create jumps in the search space to better explore it [64]. RS is the worst performing algorithm when minimizing the energy use and maximizing the reputation. NSGA-II, SPEA2, and NSGA-III obtain a better spread than IBEA. This is because the IBEA algorithm uses the hypervolume indicator in its selection operator. The majority of its non-dominated solutions (85+%) occupy the regions of the search space where the hypervolume is maximized. This behavior of IBEA has also been observed in the rest of the experiments.

3. Information was retrieved from https://blockchair.com/bitcoin on November 30, 2020.

4. https://www.bitmain.com

5. MOEA Framework version 2.13 available at http://moeaframework.org, accessed on December 10, 2020.

6. Available in the online supplemental material (see Appendix C)

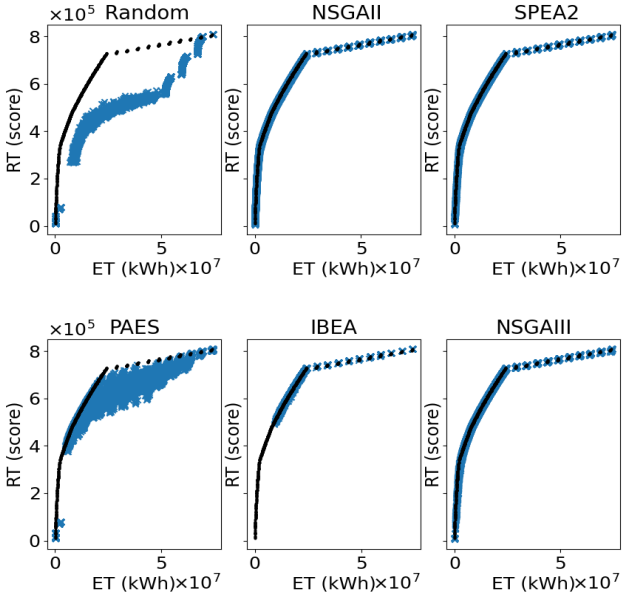

Fig. 1. The results of trading-off energy with reputation using Five algorithms. The Pareto front is shown in black.

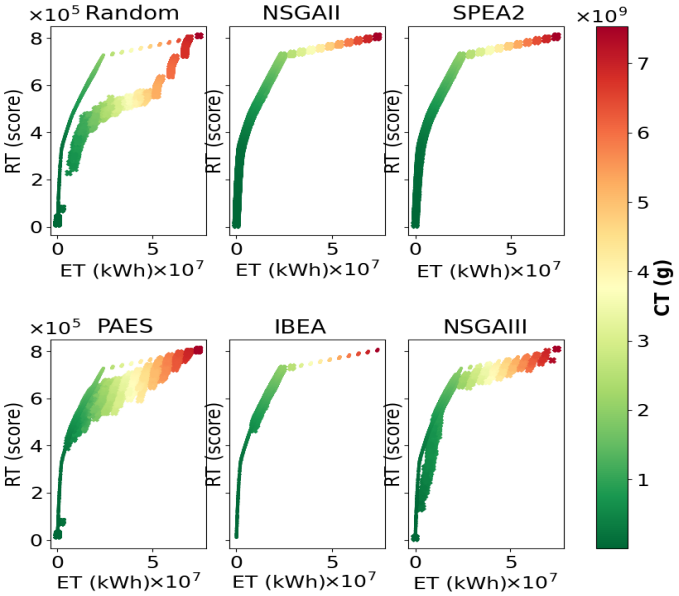

Fig. 2. The results of trading-off energy with carbon and reputation using five algorithms. The dot markers show Pareto front.

6.1.2 Energy, Carbon, and Reputation Objective Space

Figure 2 shows the approximated Pareto front found by each algorithm in the 100 runs and the computed Pareto front. The results are color-coded by the carbon objective values calculated by Equation 4, The energy use and the reputation score are represented by the x and y-axes, respectively. As can be seen, the non-dominated solutions created by NSGA-II and SPEA2 cover larger portions of the computed Pareto front compared to other algorithms. Interestingly, NSGA-III's non-dominated solutions are slightly distant from the Pareto front, and they are more scattered than those of NSGA-II, SPEA2, and IBEA on the energy and reputation dimensions. In addition, since the IBEA algorithm uses the hypervolume indicator in its selection operator, the majority of its non-dominated solutions (85+%) occupy the regions of the search space where the hypervolume is maximized. Similarly to the results of energy versus reputation experiments, PAES and RS performed poorly compared to other algorithms in terms of covering the Pareto front.

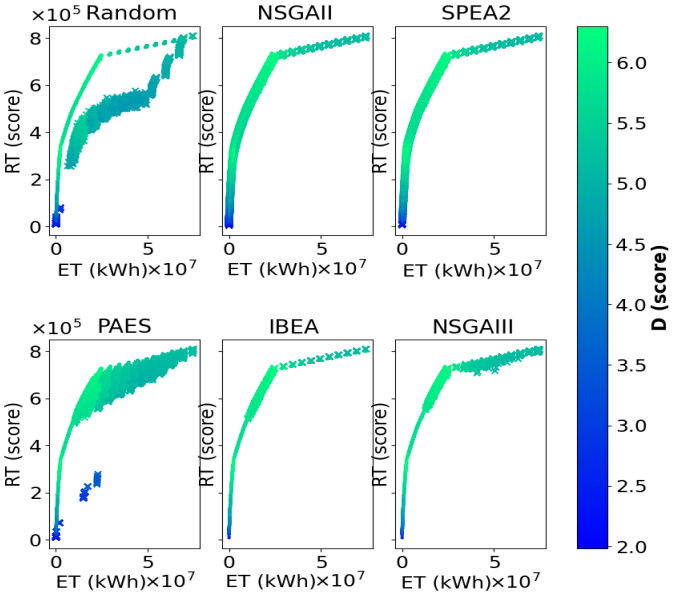

Fig. 3. The results of trading-off energy with decentralization and reputation using five algorithms. The Pareto front is shown using the dot marker.

6.1.3 Energy, Decentralization, and Reputation Objective Space

Figure 3 presents the results of minimizing the energy use while maximizing the decentralization and the reputation of the set of miners. The x-axis and y-axis represent the energy use and reputation score while the color scale of each point represents the decentralization score calculated by Equation 5. The overall results of the algorithms are similar to those in Figure 2, except that the NSGA-III algorithm covers fewer regions of the Pareto front. In addition, it can be observed that its solutions at the end of the spectrum, where the reputation score is maximized, are slightly distant from the Pareto front compared to NSGA-II, SPEA2, and IBEA.

6.1.4 Energy, Carbon, Decentralization, and Reputation Objective Space

Figure 4 shows the many-objective optimization experiment results. The x-axis, y-axis, and z-axis represent the energy use, reputation, and decentralization scores, respectively. The results are color-coded by carbon values. As can be seen, the NSGA-II and SPEA2 non-dominated sets include more solutions of the Pareto front. However, as the problem dimensionality increases, their ability to consistently find Pareto front's solutions degrades (i.e., they have more distant solutions from the Pareto front than IBEA and NSGA-III). On the other hand, IBEA non-dominated solutions are

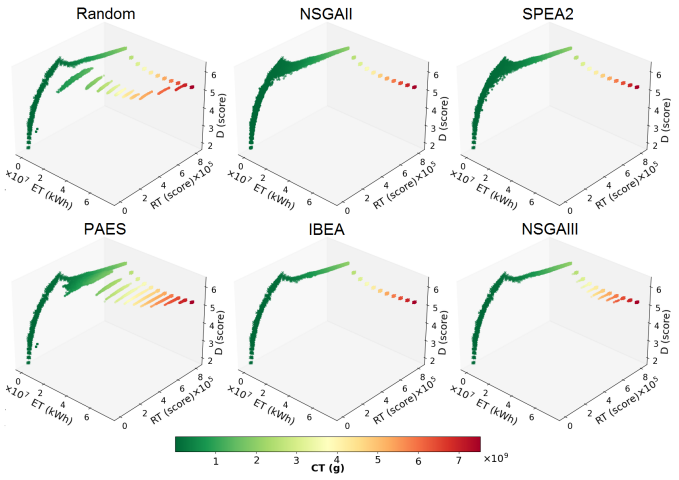

Fig. 4. The results of trading-off energy with carbon, decentralization, and reputation using five algorithms. The Pareto front is shown using dot marker.

less diverse (in terms of objective values), but they reside on the Pareto front. The NSGA-III non-dominated set covers slightly more regions of the Pareto front than IBEA's non-dominated solutions. The PAES and RS algorithms found the lowest number of Pareto front's solutions. The former is a mutation-based algorithm which effectively explores the neighbors of the promising solutions. However, as the problem dimensionality increases, the algorithm's effectiveness decreases.

6.2 Research Questions Answers

6.2.1 *Energy Consumption and Carbon Emission*

To answer **RQ1 and RQ2**, we compare the original solution's objective value (i.e., all miners included) to the objective values of the solutions found using EAs. The reported results are the ratio of the objective score of a solution to the objective score of the original solution. Overall, using the proposed fitness functions helps the optimizers to explore the search space and find optimal solutions concerning energy consumption and carbon emissions (i.e., efficient solutions for energy consumption and carbon emissions). Indeed, energy savings and low carbon emissions are achieved with some degradation in the conflicting objective(s). It is worth mentioning that Pareto-based algorithms are used to produce non-dominated solutions to the decision-makers.

In regards to energy savings, using the first fitness function (i.e., energy vs. reputation) to explore the search space, IBEA's best solution in terms of energy consumption improves the energy efficiency by 88% at the cost of only 40% of overall reputation. The other algorithms substantially reduce energy use; however, the reputation decrease by more than 95%. It is worth noting that other algorithms (except RS) managed to find solutions with similar energy efficiency to IBEA's best solution. We notice that IBEA's non-dominated set is very limited compared to those of other MOEAs. Although the degradation in the conflicting objectives (i.e., reputation and decentralization) with energy use is considerable, solutions with such objective values can be used in private or consortium blockchain-based systems. This is because miners are already known to organizations employing such kinds of blockchains.

In our model, we intend to reduce carbon emission by balancing with other conflicting objectives. Although the carbon emission rate produced by miners seems to strictly follow the increase of energy consumption at the same rate in figures 2 and 4, this does not mean that miners that consume high energy will produce high carbon emissions compared with other miners that consume low energy. In some cases, we can have a miner that consumes a low amount of energy but is located in a country with a high carbon intensity that leads the miner to produce high carbon emissions and vice versa. As a result, energy consumption and carbon emission can be considered as conflicting objectives. Having the carbon emission results in figures 2 and 4 that follow the energy increase shows that the optimizer has favorably selected miners from the same regions with the same carbon intensity.

Using our optimization model can reduce the amount of carbon emission in some solutions by more than 90% compared to the traditional blockchain-based systems that use PoW. For example, using the second fitness function (i.e., energy vs. carbon vs. reputation) to explore the search space, IBEA's best solution in terms of carbon emission reduces the amount of carbon emission by 92% at the cost of only 40% of overall reputation with energy consumption equal to 12% compared to the original solution. Although the other algorithms substantially reduce carbon emission, the reputation decreases by more than 95% except for PAES that decreases the reputation by 52%. Similarly to energy savings, the solutions that substantially reduce the overall reputation can be used in private or consortium blockchain-based systems.

6.2.2 *Performance Analysis*

For answering **RQ3**, we compare the hypervolume of the non-dominated set of each algorithm with the RS's non-dominated set's hypervolume. We use the two-tailed Wilcoxon rank-sum test with a threshold of $p \leq 0.05$ to conduct the comparisons. The null hypothesis is algorithm X's hypervolume is significantly greater than the hypervolume produced by the RS algorithm. To compute the difference, we use Vargha and Delaney effect size. In addition, we conducted a comparison between every pair of algorithms (pairwise comparison) to answer **RQ4**.

Figure 5 presents the results of the statistical test and the effect size for each algorithm's performance using the four proposed fitness functions. In each cell, the label S denotes a significant difference, whereas label I indicates a non-significant difference. The cell color shows the effect size.

As can be seen in Figure 5, across all fitness functions, each algorithm's hypervolume is significantly greater than that of RS (see the first column of every heatmap). In addition, the difference between RS performance and that of other algorithms, is large. This is consistent with the visual representation of the algorithms' non-dominated set in Figures 1, 2 and 3.

Now, we answer **RQ4**. In fitness function 1 (i.e., energy vs. reputation) and fitness function 3 (i.e., energy vs. decentralization vs. reputation), NSGA-II's performance is the

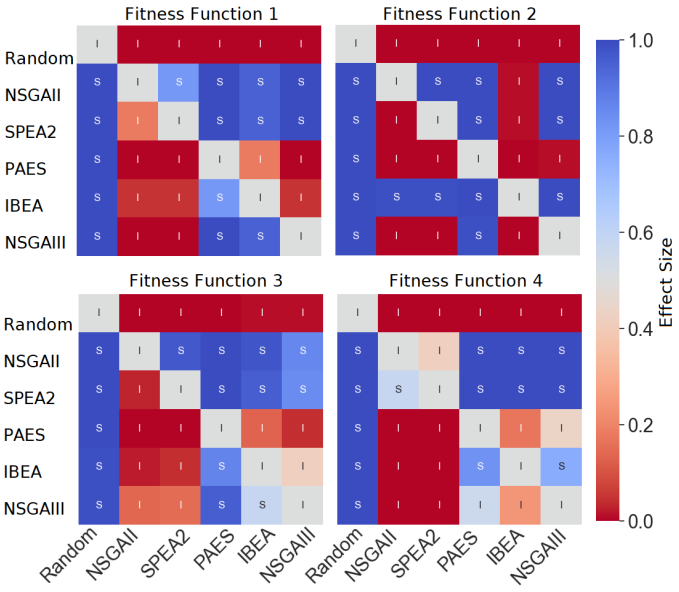

Fig. 5. Algorithms effect sizes and P-values. The letter S indicates significant differences, and the letter I denotes insignificant differences.

best among all algorithms which indicates that it produced the most diverse, non-dominated set that covers the largest portion of the search space among other non-dominated sets. PEAS has the second-worst performance, this due to its mechanism in exploring the search space. Among all algorithms used, IBEA's performance is the best in exploring the search space using the fitness function 2 (i.e., energy vs. carbon vs. reputation). We have investigated its non-dominated solutions, and we have found that they are centered in the region where the hypervolume is maximized. This is due to the algorithm selection preference among solutions which prefers a larger hypervolume. However, as can be seen in Figure 3, such a mechanism restricted the solution diversity in the fitness landscape.

Interestingly, SPEA2 has the best performance among other algorithms in exploring the solution space using fitness function 4 (energy vs. carbon vs. decentralization vs. reputation). However, it is worth noting that in terms of run-time, SPEA2 has the second-worst run-time after the PAES algorithms. Our results indicate that the expensive mechanism in determining the strength of solutions enables SPEA2 to outperform other algorithms in our proposed many-objective problem.

Although NSGA-III is designed for many-objective problems, it does not perform well in our many-objective problem. We conjecture that the NSGA-III light-weight nicheing mechanism, which is based on reference points, is not as effective as other algorithms' (i.e., SPEA2, NSGA-II, and IBEA) expensive diversity-preservation mechanisms. It mainly depends on reference points being created to virtually represent how the Pareto front would look in the objective space. The quality of the created niches dramatically influences the performance of the algorithm. It is worth mentioning that the performance of NSGA-III has been shown to be worse than NSGA-II and SPEA2 in [65].

7 Conclusion

Blockchain technology is widely considered as one of the most important of recent developments. However, it has a critical weakness: the mining network's use of resources and excessive energy consumption. This will have a significant environmental effect and could prevent the widespread adoption of this technology. Building an optimal balance between the multiple criteria involved is an important problem.

In this paper, we have reformulated this problem as a multi-objective optimization problem. We attempt to minimize energy consumption while considering the trade-off between decentralization and reputation of miners within blockchain-based systems. To solve the problem, we have proposed four different fitness functions. Our results show that energy usage was reduced by up to 88% with a 40% reduction in reputation (a non-functional property). Moreover, using private blockchain-based systems, where miners are known, can save energy by more than 90%. For evaluating our proposed model, we have compared five different evolutionary algorithms with different diversity-preservation mechanisms. The comparisons revealed that there is no one algorithm that is consistently superior using its default settings.

Current research work seeks to reduce the energy consumption of blockchain-based systems using different methods. However, most of these methods focus on proposing alternative consensus algorithms, such as PoS and PoT. These methods are limited because they do not assist decision-makers in choosing the best solution for their preferred criteria. They instead guess what potential solutions might be interesting.

As with every model, our proposed model has some limitations. For instance, the current model is static, and we used it in an off-line optimization scenario (i.e., optimize first, then deploy). However, our model can be used in dynamic optimization scenarios, where the environment changes overtime. In addition, for measuring energy consumption, we have used an estimation model which abstracts away systems' interactions that can affect energy estimations. Besides, we have assumed that all miners have the same device properties when competing to add blocks. It has been shown in SBSE that different hardware platforms and software systems' interactions can affect energy measurements [66]. However, such models can still produce accurate results [25].

In future work, we plan to combine evolutionary algorithms with learning algorithms to create self-adaptive approaches that deal with scenarios where miners or mining policies would change over the lifetime of the blockchain-based system operation. Also, we plan to investigate how a dynamic optimization model could provide better efficiency than static optimization taking into consideration scalability and mining process overhead. Besides, We will explore Nakamoto's coefficient and other metrics, such as the Gini coefficient, for measuring the degree of decentralization as an objective of our dynamic optimization model. In addition, we intend to conduct a sensitivity analysis on evolutionary algorithms' parameters that can impact the quality of the produced results.

REFERENCES

[1] S. Nakamoto. (2008) Bitcoin: A peer-to-peer electronic cash system. Accessed March 2, 2019. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics and Informatics*, vol. 36, pp. 55 – 81, 2019.

[3] J. Truby, "Decarbonizing bitcoin: Law and policy choices for reducing the energy consumption of blockchain technologies and digital currencies," *Energy Research & Social Science*, vol. 44, pp. 399 – 410, 2018.

[4] P. Singh and S. K. Choudhary, "Introduction: Optimization and metaheuristics algorithms," in *Metaheuristic and Evolutionary Computation: Algorithms and Applications, Studies in Computational Intelligence*. Springer, 2021, vol. 916, pp. 3–33.

[5] A. De Vries, "Bitcoin's energy consumption is underestimated: A market dynamics approach," *"Energy Research & Social Science"*, vol. 70, p. 101721, 2020.

[6] Cambridge Centre for Alternative Finance. (2020) Cambridge Bitcoin Electricity Consumption Index. Accessed December 31, 2020. [Online]. Available: https://www.cbeci.org

[7] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[8] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, 2016, p. 3–16.

[9] L. Bahri and S. Girdzijauskas, "When trust saves energy: A reference framework for proof of trust (pot) blockchains," in *Companion Proceedings of the The Web Conference (WWW '18)*, 2018, p. 1165–1169.

[10] Q. Zhuang, Y. Liu, L. Chen, and Z. Ai, "Proof of reputation: A reputation-based consensus protocol for blockchain based systems," in *Proceedings of the 2019 International Electronics Communication Conference (IECC '19)*, 2019, p. 131–138.

[11] S. King and S. Nadal. (2012) Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. Accessed October 14, 2019. [Online]. Available: https://www.peercoin.net/whitepapers/peercoin-paper.pdf

[12] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, "Proofs of space," in *Proceedings of the Conference on Advances in Cryptology (CRYPTO 2015)*, 2015, pp. 585–605.

[13] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: An efficient blockchain consensus protocol," in *Proceedings of the 1st Workshop on System Software for Trusted Execution (SysTEX '16)*, 2016, pp. 1–6.

[14] M. Mezmaz, N. Melab, Y. Kessaci, Y. Lee, E.-G. Talbi, A. Zomaya, and D. Tuyttens, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 11, pp. 1497–1508, 2011.

[15] X.-F. Liu, Z.-H. Zhan, K.-J. Du, and W.-N. Chen, "Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO '14)*. ACM, 2014, p. 41–48.

[16] J.-W. Lee and J.-J. Lee, "Ant-colony-based scheduling algorithm for energy-efficient coverage of wsn," *IEEE Sensors Journal*, vol. 12, no. 10, pp. 3036–3046, 2012.

[17] I. Ahmad, S. Ranka, and S. U. Khan, "Using game theory for scheduling tasks on multi-core processors for simultaneous optimization of performance and energy," in *2008 IEEE International Symposium on Parallel and Distributed Processing*, 2008, pp. 1–6.

[18] E. V. Dinesh Subramaniam and V. Krishnasamy, "Energy aware smartphone tasks offloading to the cloud using gray wolf optimization," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 3979–3987, 2021.

[19] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. Shen, "Energy efficient dynamic offloading in mobile edge computing for internet of things," *IEEE Transactions on Cloud Computing*, vol. 9, no. 3, pp. 1050–1060, 2021.

[20] M. Askarizade Haghighi, M. Maeen, and M. Haghparast, "An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing iaas platforms," *Wireless Personal Communications*, vol. 104, no. 4, pp. 1367–1391, 2019.

[21] S. A. Sert, H. Bagci, and A. Yazici, "Mofca: Multi-objective fuzzy clustering algorithm for wireless sensor networks," *Applied Soft Computing*, vol. 30, pp. 151–165, 2015.

[22] J. Gillett, S. Rahnamayan, M. Makrehchi, and A. A. Bidgoli, "A pareto front-based metric to identify major bitcoin networks influencers," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion (GECCO '20)*, 2020, p. 1395–1401.

[23] H. Shi, S. Wang, and Y. Xiao, "Queuing without patience: A novel transaction selection mechanism in blockchain for iot enhancement," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7941–7948, 2020.

[24] B. R. Bruce, J. Petke, M. Harman, and E. T. Barr, "Approximate oracles and synergy in software energy search spaces," *IEEE Transactions on Software Engineering*, vol. 45, no. 11, pp. 1150–1169, 2019.

[25] M. A. Bokhari, B. Alexander, and M. Wagner, "In-vivo and offline optimisation of energy use in the presence of small energy signals: A case study on a popular android library," in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18)*, 2018, p. 207–215.

[26] M. Linares-Vásquez, G. Bavota, C. E. B. Cárdenas, R. Oliveto, M. Di Penta, and D. Poshyvanyk, "Optimizing energy consumption of guis in android apps: A multi-objective approach," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015)*, 2015, p. 143–154.

[27] R. Morales, R. Saborido, F. Khomh, F. Chicano, and G. Antoniol, "Earmo: An energy-aware refactoring approach for mobile apps," *IEEE Transactions on Software Engineering*, vol. 44, no. 12, pp. 1176–1206, 2018.

[28] B. R. Bruce, J. M. Aitken, and J. Petke, "Deep parameter optimisation for face detection using the viola-jones algorithm in OpenCV," in *Search Based Software Engineering*, F. Sarro and K. Deb, Eds., 2016, pp. 238–243.

[29] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. Rinard, "Managing performance vs. accuracy trade-offs with loop perforation," in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering (ESEC/FSE '11)*, 2011, p. 124–134.

[30] A. Alofi, M. A. Bokhari, R. Hendley, and R. Bahsoon, "Selecting miners within blockchain-based systems using evolutionary algorithms for energy optimisation," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion)*, 2021, p. 291–292.

[31] A. Chauhan, O. P. Malviya, M. Verma, and T. S. Mor, "Blockchain and scalability," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2018, pp. 122–128.

[32] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16 440–16 455, 2020.

[33] T. Grandison and M. Sloman, "A survey of trust in internet applications," *IEEE Communications Surveys Tutorials*, vol. 3, no. 4, pp. 2–16, 2000.

[34] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities," in *Proceedings of the 33rd annual Hawaii international conference on system sciences*, 2000, pp. 9–pp.

[35] S. P. Gochhayat, S. Shetty, R. Mukkamala, P. Foytik, G. A. Kamhoua, and L. Njilla, "Measuring decentrality in blockchain based systems," *IEEE Access*, vol. 8, pp. 178 372–178 390, 2020.

[36] P. Hewitt, *Conceptual Physics*, 13th ed. Boston, MA, USA: Pearson, 2021.

[37] K. Wu, B. Peng, H. Xie, and Z. Huang, "An information entropy method to quantify the degrees of decentralization for blockchain systems," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2019, pp. 1–6.

[38] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[39] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 2567–2572.

[40] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security*, 2014, pp. 436–454.

[41] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
[42] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon, "Performance indicators in multiobjective optimization," *European Journal of Operational Research*, 2020.
[43] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull*, vol. 1, no. 6, pp. 80–83, 1945.
[44] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.
[45] A. Vargha and H. D. Delaney, "A critique and improvement of the cl common language effect size statistics of mcgraw and wong," *Journal of Educational and Behavioral Statistics*, vol. 25, no. 2, pp. 101–132, 2000.
[46] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
[47] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.
[48] J. Knowles and D. Corne, "The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, vol. 1, 1999, pp. 98–105.
[49] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature - PPSN VIII*, 2004, pp. 832–842.
[50] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
[51] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen *et al.*, *Evolutionary algorithms for solving multi-objective problems*, 2nd ed. New York, NY, USA: Springer, 2007.
[52] M. Hort and F. Sarro, "The effect of offspring population size on nsga-ii: A preliminary study," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '21)*. ACM, 2021, p. 179–180.
[53] Z. Li and X. Li, "A multi-objective binary-encoding differential evolution algorithm for proactive scheduling of agile earth observation satellites," *Advances in Space Research*, vol. 63, no. 10, pp. 3258–3269, 2019.
[54] P. Back, A. Suominen, P. Malo, O. Tahvonen, J. Blank, and K. Deb, "Towards sustainable forest management strategies with moeas," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference (GECCO '20)*. ACM, 2020, p. 1046–1054.
[55] H. Noguchi, T. Harada, and R. Thawonmas, "Parallel differential evolution applied to interleaving generation with precedence evaluation of tentative solutions," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '21)*. ACM, 2021, p. 706–713.
[56] J. E. Fieldsend and K. Alyahya, "Visualising the landscape of multi-objective problems using local optima networks," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '19)*. ACM, 2019, p. 1421–1429.
[57] G. Fournier and F. Petrillo, "Architecting blockchain systems: A systematic literature review," in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW '20)*, 2020, p. 664–670.
[58] A. De Vries, "Bitcoin's growing energy problem," *Joule*, vol. 2, no. 5, pp. 801 – 805, 2018.
[59] ——, "Renewable energy will not solve bitcoin's sustainability problem," *Joule*, vol. 3, no. 4, pp. 893 – 898, 2019.
[60] C. Stoll, L. Klaaßen, and U. Gallersdörfer, "The carbon footprint of bitcoin," *Joule*, vol. 3, no. 7, pp. 1647 – 1661, 2019.
[61] Carbon Footprint Ltd. (2020) Country specific electricity grid greenhouse gas emissions factors. Accessed November 02, 2020. [Online]. Available: https://www.carbonfootprint.com/international_electricity_factors.html
[62] M. S. Mahbub, M. Wagner, and L. Crema, "Multi-objective optimisation with multiple preferred regions," in *Australasian Conference on Artificial Life and Computational Intelligence*, 2017, pp. 241–253.
[63] U. Bhowan, M. Zhang, and M. Johnston, "Multi-objective genetic programming for classification with unbalanced data," in *Australasian Joint Conference on Artificial Intelligence*, 2009, pp. 370–380.
[64] E. Osaba, R. Carballedo, F. Diaz, E. Onieva, I. De La Iglesia, and A. Perallos, "Crossover versus mutation: A comparative analysis of the evolutionary strategy of genetic algorithms applied to combinatorial optimization problems," *The Scientific World Journal*, vol. 2014, 2014.
[65] B. Changaival, G. Danoy, D. Kliazovich, F. Guinand, M. R. Brust, J. Musial, K. Lavangnananda, and P. Bouvry, "Toward real-world vehicle placement optimization in round-trip carsharing," in *Proceedings of the 2019 Genetic and Evolutionary Computation Conference (GECCO '19)*, 2019, p. 1138–1146.
[66] M. A. Bokhari, B. Alexander, and M. Wagner, "Towards rigorous validation of energy optimisation experiments," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference (GECCO '20)*, 2020, p. 1232–1240.
[67] D. Hadka. (2015) MOEA framework user guide (Version 2.6). Accessed October 14, 2019. [Online]. Available: http://www.moeaframework.org

PLACE PHOTO HERE

Akram Alofi received the Bachelor's degree in computer science (First Class Honours) from Umm Al-Qura University, Makkah, Saudi Arabia and received the Master's degree in information technology (Golden Key Award) from Flinders University, Adelaide, Australia. He is currently working toward the PhD degree in the School of Computer, University of Birmingham, United Kingdom. His research interests include software engineering, optimization, trust management, and blockchain.

PLACE PHOTO HERE

Mahmoud A. Bokhari is an Assistant Professor at the School of Computer Science, Taibah University, Saudi Arabia. He has done his PhD studies in Computer Science at the University of Adelaide, Adelaide, Australia. His research interests include green software engineering, search-based software engineering, ubiquitous computing, and software testing.

PLACE PHOTO HERE

Rami Bahsoon is a Senior Lecturer (Associate Professor) at the School of Computer Science, University of Birmingham, United Kingdom. Bahsoon conducts research in the fundamentals of self-adaptive and managed software architectures and its application to emerging paradigms, such as cloud, microservices, IoT, CPS, etc. His investigations have also looked at self-aware software architectures, economics-driven software architectures, and technical debt management in software. He co-edited four books on Software Architecture, including Economics-Driven Software Architecture; Software Architecture for Big Data and the Cloud; Aligning Enterprise System, and Software Architecture. He holds a PhD in Software Engineering from University College London (2006) on software architectures and was a MBA Fellow in Technology at London Business School (2003-2005). He is a fellow of the Royal Society of Arts, Associate Editor of IEEE Software, and Editor of Wiley Software Practice and Experience.

PLACE PHOTO HERE

Robert Hendley is a lecturer at the School of Computer Science, University of Birmingham, United Kingdom. His research interests include information visualization, information networks, user interface design, plan recognition, adaptive hypertext, and intelligent tutoring systems.

Supplemental Materials: Optimizing the Energy Consumption of Blockchain-based Systems Using Evolutionary Algorithms: A New Problem Formulation

Akram Alofi, Mahmoud A. Bokhari, Rami Bahsoon and Robert Hendley

APPENDIX A

Appendix A provides a table that summarizes each study discussed in the Related Work Section. Table 4, shows the reference number of the study, the problem solved, the optimization technique used, and its area.

TABLE 4
A Summary of Related Work

Reference	Problem Considered	Optimization Technique	Area
[9]	Energy consumption	New consensus algorithm	Blockchain technology
[10]	Energy consumption	New consensus algorithm	Blockchain technology
[11]	Energy consumption	New consensus algorithm	Blockchain technology
[12]	Energy consumption	New consensus algorithm	Blockchain technology
[13]	Energy consumption	New consensus algorithm	Blockchain technology
[14]	Energy consumption	Scheduling with multi-objective optimization	Heterogeneous computing
[15]	Energy consumption	Scheduling with multi-objective optimization	Cloud computing
[16]	Energy consumption	Scheduling with multi-objective optimization	Wireless sensor network
[17]	Energy consumption	Scheduling with multi-objective optimization	Multi-core processors system
[18]	Energy consumption	Offloading with a meta-heuristic algorithm	Cloud computing
[19]	Energy consumption	Offloading with stochastic optimization	Mobile edge computing
[20]	Energy consumption	Clustering with Meta-heuristic algorithms	Cloud computing
[21]	Energy consumption	Clustering with multi-objective optimization	Wireless sensor networks
[22]	Bitcoin network's influencers	Multi-objective optimization	Blockchain technology
[23]	Bitcoin utility	Multi-objective optimization	Blockchain technology
[24]	Energy consumption	Genetic Improvement	Applications
[25]	Energy consumption	Genetic Improvement	Mobile application
[26]	Energy consumption	Multi-objective optimization	Mobile application
[27]	Energy consumption	Multi-objective optimization	Mobile application
[28]	Execution time	Deep parameter optimization	Face detection
[29]	Performance	Loop perforation	Applications

APPENDIX B

Appendix B provides a table that lists all notations used in this paper. A list of notations used and their description are presented in Table 5.

TABLE 5
A List of Notations

Notation	Description
EM	The energy consumption for each miner ($kilowatt - hour$)
P	The amount of power used by a mining device ($watt$)
T	The hours of participating in the blockchain network per day ($hours$)
mD	The number of mining devices
ET	The total energy consumption of all participating miners in a Pareto front's solution ($kilowatt - hour$)
m	The number of miners that compose the network of a blockchain-based system
X	The value of each gene in a Pareto front's solution representation ($X \in \{1, 0\}$)
CM	The greenhouse gas emissions produced by a miner ($gram$)
EF	the emission factor of electricity in the miner's location (gCO_2eq/kWh)
CT	The total carbon emission generated by all participating miners in a Pareto front's solution (gCO_2eq/kWh)
D	The degree of decentralization
FH	The fraction of the hashrate of a miner in a solution
h	The hashrate of a miner
h_t	The total hashrate of all participating miners in a Pareto front's solution.
RM	The reputation value for a miner
B	The total number of mined blocks in the blockchain
b	The number of blocks mined by a miner
s	The total of fees and rewards a miner has
RT	The total reputation of all participating miners in a Pareto front's solution
H_c	The hashrate for a miner that will be compared to other miners' hashrate in a solution
TL	The percentage of tolerance level in a blockchain-based system

APPENDIX C

Appendix C provides two tables related to the parameters for the chosen algorithms in this paper. Table 6 and Table 7 present the parameters notation related to each of the used algorithms and the value of these parameters for each algorithm, respectively. The notations and the values are retrieved from the MOEA framework manual [67]

TABLE 6
A List of Notations for Evolutionary Algorithms Parameters

Parameter	Description
populationSize	The size of the population
sbx.rate	The crossover rate for simulated binary crossover
sbx.distributionIndex	The distribution index for simulated binary crossover
pm.rate	The mutation rate for polynomial mutation
pm.distributionIndex	The distribution index for polynomial mutation
offspringSize	The number of offspring generated every iteration
k	Crowding is based on the distance to the k-th nearest neighbor
archiveSize	The size of the archive
bisections	The number of bisections in the adaptive grid archive
indicator	The indicator function (e.g., hypervolume, epsilon, crowding)
divisions	The number of divisions
epsilon	The ϵ values used by the ϵ-dominance archive, which can either be a single value or a comma-separated array (this parameter is optional)

TABLE 7
The values of the Parameters for the Used Algorithms

Parameter	Random	NSGA-II	SPEA2	PAES	IBEA	NSGA-III
populationSize	160	160	160	-	160	160
sbx.rate	-	1.0	1.0	-	1.0	1.0
sbx.distributionIndex	-	15.0	15.0	-	15.0	15.0
pm.rate	-	$1/N$	$1/N$	$1/N$	$1/N$	$1/N$
pm.distributionIndex	-	20.0	20.0	20.0	20.0	20.0
offspringSize	-	-	100	-	-	-
k	-	-	1	-	-	-
archiveSize	-	-	-	100	-	-
bisections	-	-	-	8	-	-
indicator	-	-	-	-	hypervolume	-
divisions	-	-	-	-	-	4
epsilon	Problem dependent	-	-	-	-	-