Chen *et al. J Wireless Com Network* (2021) 2021:146
https://doi.org/10.1186/s13638-021-02018-x

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Miner revenue optimization algorithm based on Pareto artificial bee colony in blockchain network

Yourong Chen[1,2], Hao Chen[2], Meng Han[3*], Banteng Liu[1,2], Qiuxia Chen[1], Zhenghua Ma[2] and Zhangquan Wang[1]

*Correspondence: mhanresearch@outlook.com
[3] Binjiang Institute, Zhejiang University, Hangzhou 310052, China
Full list of author information is available at the end of the article

Abstract

In order to improve the revenue of attacking mining pools and miners under block withholding attack, we propose the miner revenue optimization algorithm (MROA) based on Pareto artificial bee colony in blockchain network. MROA establishes the revenue optimization model of each attacking mining pool and revenue optimization model of entire attacking mining pools under block withholding attack with the mathematical formulas such as attacking mining pool selection, effective computing power, mining cost and revenue. Then, MROA solves the model by using the modified artificial bee colony algorithm based on the Pareto method. Namely, the employed bee operations include evaluation value calculation, selection probability calculation, crossover operation, mutation operation and Pareto dominance method, and can update each food source. The onlooker bee operations include confirmation probability calculation, crowding degree calculation, neighborhood crossover operation, neighborhood mutation operation and Pareto dominance method, and can find the optimal food source in multidimensional space with smaller distribution density. The scout bee operations delete the local optimal food source that cannot produce new food sources to ensure the diversity of solutions. The simulation results show that no matter how the number of attacking mining pools and the number of miners change, MROA can find a reasonable miner work plan for each attacking mining pool, which increases minimum revenue, average revenue and the evaluation value of optimal solution, and reduces the spacing value and variance of revenue solution set. MROA outperforms the state of the arts such as ABC, NSGA2 and MOPSO.

Keywords: Block withholding attack, Blockchain, Pow, Mining cost

1 Introduction

With the continuous development of information technology, information security issues in our daily lives are becoming more and more important [1]. Blockchain technology relies on maintaining a reliable distributed database through distributed ledgers, consensus mechanisms, smart contracts, cryptography and other means, and can solve the security issues such as information tampering [2, 3]. Therefore, blockchain technology is taken high attention from industry and academia.

At present, blockchain is widely used in education, auditing, human resource internet of Things (IoT), electronic voting, medical care, intelligent transportation and many other fields [4, 5]. For example, in the medical field, attackers can obtain the personal health information (PHI) through hospitals, schools, laboratories and other ways. And there are also some problems such as inconsistent information format and difficult reliability proof. Therefore, Bentov Iddo et al. [6] propose a typical blockchain case in the medical field, that is, a personal health information system based on blockchain. The system combines a proof-of-work (POW) algorithm to manage personal health and other information. It effectively not only solves the problems such as information format and reliability proof, but also facilitates the search of patients or medical institutions.

In the POW algorithm [7, 8], the miner first calculates the Merkle root of the block transaction set and fills the block header with the previous block hash value, block version number and other information. Then, it sets the random number Nonce to zero. The miner adds 1 to the random number Nonce and calculates the hash value of the current block based on the information in the block header. If the leading zero of the block hash value meets the difficulty requirement, that is, the miner completes the SHA256 mathematical puzzle, the miner will send the searched random number and block information to other nodes and obtain miner revenue after the blockchain network performs verification. The revenue consists of fixed revenue and variable transactions related to the number of transactions. If the miner fails to find that its block hash value meets the difficulty requirement within a certain period of time, the miner needs to update the transaction set of the timestamp and the block body, and perform the search of the number Nonce again. At the same time, considering the influence of the entire network's computing power on the block generation time, the blockchain network can flexibly adjust the difficulty value of block mining according to the entire network's computing power information. In the actual process, the POW algorithm takes about 10 minutes to generate a block. If each miner competes for mining through its own computing power, most miners will not obtain stable revenue. In order to increase the possibility of obtaining stable revenue, miners join the mining pool to ensure their own revenue through cooperative mining. The mining pool is composed of miners and a mining pool manager [9, 10]. The miner carries out mining work according to the workload certification requirements issued by the mining pool manager and sends the mining results to the mining pool manager in the form of proof of part work or proof of full work. The mining pool manager estimates the mining capacity of each miner by counting the proof of work reported by the miners and publishes the proof of full work to the blockchain network to compete with the current blockchain network. If the mining pool manager successfully obtains the rewards, it will allocate the revenue according to the mining ability of each miner, so that each miner in the mining pool can obtain a certain revenue.

But the mining pools are vulnerable to block withholding attack. Block withholding attack means that malicious miners always choose to send proof of part work to the mining pool manager and discard the proof of full work [11, 12]. The attack causes a waste of computing power in the attacked mining pool, resulting in a decrease in its total revenue. At the same time, it helps malicious miners get revenue from attacked mining pools by relying on the proof of part work. Since the malicious miner who carries out the block withholding attack always submits proof of part work, the mining pool manager only

detects that the total revenue of the mining pool decreases and finds that its mining pool is suffering from block withholding attack. But it cannot judge the malicious miners in the mining pool.

We believe that attacking and defensive algorithms promote the development of each other. Studying the new block withholding attack algorithm that maximizes the revenue of attackers can help us to understand the nature of block withholding attack and promote the continuous development of defense algorithms against block withholding attack [13]. The study of block withholding attack has a certain practical significance. Therefore, we propose the mining revenue optimization algorithm (MROA) of miners in PoW-based blockchain networks. The main contributions are as follows:

1. MROA establishes revenue optimization model of each attacking mining pool and revenue optimization model of entire attacking mining pools under block withholding attack with the mathematical formulas such as attacking mining pool selection, effective computing power, mining cost and revenue.
2. MROA solves the model by using the modified artificial bee colony algorithm based on the Pareto method. That is, MROA initializes the population. The employed bee operations include evaluation value calculation, selection probability calculation, crossover operation, mutation operation and Pareto dominance method, and can update each food source. The onlooker bee operations include confirmation probability calculation, crowding degree calculation, neighborhood crossover operation, neighborhood mutation operation and Pareto dominance method, and can find the optimal food source in multidimensional space with smaller distribution density. The scout bee operations delete the local optimal food source that cannot produce new food sources to ensure the diversity of solutions.
3. MROA can obtain the composition plan of each attacking mining pool. The plan increases the entire revenue of attacking mining pool while ensuring the revenue of each attacking mining pool and its miners as much as possible.

The rest of the paper is organized as follows: We introduce the related work in Sect. 2. We introduce the principles of our algorithm in Sect. 3. We explain how the algorithm is implemented in Sect. 4. We analyze the simulation results of our algorithm in Sect. 5. Finally, we summarize the paper and illustrate the future work in Sect. 6.

2 Related work

Considering the harmfulness of block withholding attack to the actual mining process, many scholars study the block withholding attack. Some scholars focus on considering the characteristics of the block withholding attack between two mining pools, and studding on the attack strategy of the attacking mining pool, so as to maximize the revenues of the attacking mining pool. Wenbai Li et al. [14] establish a mining pool game model from the perspective of system rewards and punishments, and analyze the attack penetration rate and betrayal rate of the mining pool under the Nash equilibrium. Yang Tian et al. [15] establish an iterative prisoner's dilemma model, which is solved by an indefinite value strategy. Considering that there are only two mining pools and only one mining pool allowed to initiate an attack, Rui Qin et al. [16] propose the optimal strategy and

attack conditions of block withholding attack. Nisarg Shah et al. [17] prove that the game of two mining pools attacking each other is a Nash equilibrium and calculate the wasted computing power when it reaches the equilibrium state. Considering the two mining pools being able to freely choose to cooperate or block withholding attack, Wu Di et al. [18] establish the revenue matrix of each mining pool and find the Nash equilibrium from the perspective of pure strategy and hybrid strategy to maximize the revenue of the two mining pools. Hu Qin et al. [19, 20] use ZD (zero determinant) strategy to optimize the strategy of the mining pool to find the Nash equilibrium in the scenario where two mining pools carry out block withholding attack against each other. However, references [14–20] do not consider the problem of block withholding attack among mining pools with uneven distribution of computing powers in the actual process.

Therefore, some scholars focus on the problem of block withholding attack among multiple mining pools and obtain the computing power allocation scheme of the mining pools. Rajani Singh et al. [21] establish a dynamic game model among mining pools and propose two algorithms of cooperative mining strategy and noncooperative mining strategy to maximize the revenue of the mining pool. Considering the dynamic influence of malicious miners on the mining pool, Kim Seonggeun et al. [22] propose an evolutionary game theory based on blockchain. When the miner attacks mining pool, Luu Loi et al. [23] propose an algorithm for calculating revenue of miners and finding the Nash equilibrium. Considering block withholding attack among multiple mining pools, Shajari Mehdi et al. [24] use the tile coding algorithm of reinforcement learning to analyze the influence of miners' migration on the computing power of mining pools. WANG Tiantian et al. [25] consider the behavior of block withholding attack among multiple mining pools as an iterative prisoner's dilemma model, and use the gradient algorithm to adjust the mining strategy of the mining pool. Considering the situation where the attacking mining pool and other mining pools collude to attack the attacked mining pool in the environment of multiple mining pools, Bag Samiran et al. [26] propose a sponsored block withholding attack strategy of attacking mining pool to maximize its own revenue. But the above references do not consider factors such as mining cost in the model establishment.

Some scholars focus on the combination of block withholding attack, selfish mining attack, 51% attack and other attack algorithms to increase the revenue of attacking mining pool. Based on the traditional block withholding attack, Jaewoo So [27] propose a block withholding attack combined with selfish mining, that is, the miner who initiates the block withholding attack carry out selfish mining on the withholding block. Dong Xuewen et al. [28] propose a self-sustaining block withholding attack, that is, the attacking mining pool not only carries out a selfish mining attack on the attacked mining pool, but also assigns some computing power to carry out block withholding attack. Ke Junming et al. [29] propose an intermittent block withholding attack based on the dynamic adjustment of the block mining difficulty value. That is, when the difficulty value of the whole blockchain network is high, the attacking mining pool sends computing power to carry out block withholding attack, and when the difficulty value drops, the attacking mining pool turns the block withholding attack into honest mining. Chang Sangyoon et al. [30] propose an uncle block attack strategy for the block reward of Ethereum, that is, the uncle-block attack strategy requires the attacking mining pool to submit all

reserved blocks when other miners submit blocks to help the attacking mining pool obtain revenue. Considering the blockchain network deployment, Wang Yilei [31] propose a hybrid block withholding attack to determine the optimal attack strategy based on the blockchain network. But the above references mainly optimize the revenue of a single attacking mining pool, and do not consider the revenue of the entire attacking mining pools.

In conclusion, many scholars use a variety of algorithms to optimize the computing power scheme of mining pool. But they do not consider the cost of honest mining, the cost of block withholding attack and the dynamic block withholding attack among multiple mining pools. Therefore, we propose the MROA algorithm. In the preliminary work, we have achieved certain study results on the block withholding attack. To overcome the fast-changing block withholding attacks among multiple mining pools composed of miners in the blockchain system, we propose a mining pool computing power allocation algorithm, which significantly improves the revenues of mining pools with block withholding attacks [32], but the algorithm does not consider the selection of the miners during the model establishment. Then, considering the selection factors of the miners in the attacking mining pool, we propose a novel anti-attack mining revenue optimization algorithm to improve the revenues of both the attacking mining pools and miners under block withholding attack [33], but in conference paper, the principle description and the experimental simulation work are simple.

3 Algorithm

The assumptions are as follows:

1. Attacking mining pool can freely choose between block withholding attack and honest mining. Honest mining pools cannot carry out block withholding attack, but they can carry out honest mining.
2. The miners in the blockchain network can freely choose among attacking mining pools.
3. In order to help attacking mining pools obtain revenue, miners carry out honest mining or block withholding attack according to the requirement of their own attacking mining pools.
4. Miner shares the total revenue of its mining pool according to the computing power provided by himself.

As shown in Fig. 1, miners carry out honest mining or block withholding attack on another mining pool according to the instructions of their own attacking mining pool. The mining pool manager sends an attacking message to all its miners against an honest mining pool. If a miner accepts the block withholding attack task, it sends a receiving message to the mining pool manager to carry out block withholding attack. Then, it regularly reports attack status messages and accepts the revenue allocated by the mining pool manager. Other message passing and signaling are consistent with the message communication process of the POW algorithm. Considering that both honest miners and attacking miners are trying to maximize the revenue of their mining pools, there are two issues that need to be resolved. The first is how to establish a

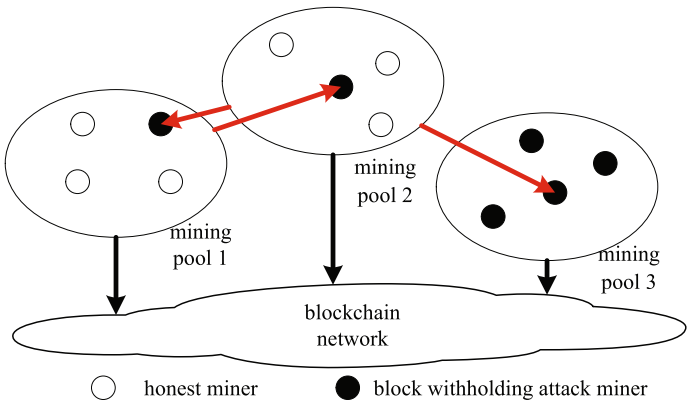

Fig. 1 Schematic diagram of block withholding attack

revenue optimization model of each attacking mining pool and revenue optimization model of entire attacking mining pool under block withholding attack with mathematical formulas such as miner selection formula and revenue formula. The second is how to solve the model by using the modified artificial bee colony algorithm based on the Pareto method and find a reasonable miner work plan for each attacking mining pool.

3.1 Model establishment

Let x_{ijk} represent the correlation among miner i, the attacking mining pool j which miner i belongs to and other mining pool k. If $x_{ijk} = 1$ and $j = k$, miner i in the attacking mining pool j and carries out honest mining. If $x_{ijk} = 1$ and $j \neq k$, the attacking mining pool j let miner i join other mining pool k and carries out block withholding attack. At the same time, the correlation meets the following conditions:

$$\sum_{j}\sum_{k} x_{ijk} = 1, \forall i \tag{1}$$

Let w_{ij} represent the indicator whether miner i carries out honest mining in attacking mining pool j which miner i belongs to. It is

$$w_{ij} = \begin{cases} 1 & j = k \text{ and } x_{ijk} = 1 \\ 0 & \text{others} \end{cases} \tag{2}$$

Let s_{ijk} represent the indicator whether miner i carries out block withholding attack on other mining pool k according to the requirement of attacking mining pool j which miner i belongs to. It is

$$s_{ijk} = \begin{cases} 1 & j \neq k \text{ and } x_{ijk} = 1 \\ 0 & \text{others} \end{cases} \tag{3}$$

Let y_j represent the total computing power that attacking mining pool j can use for honest mining. It is

$$y_j = \sum_i c_i w_{ij} \tag{4}$$

where c_i represents effective computing power provided by miner i. Then, the total computing power of each attacking mining pool j subjected to block withholding attack a_j and the total computing power of the attacking mining pool j used to carry out the block withholding attack v_j are

$$a_j = \sum_k \sum_i c_i s_{ikj}, v_j = \sum_{\mathrm{k}} \sum_{\mathrm{i}} \mathrm{c_i s_{ijk}} \tag{5}$$

Since each attacking mining pool j allocates computing power to carry out block withholding attack, the real honest mining revenue R_j^H of each attacking mining pool j needs to be allocated according to the computing power y_j and v_j. So R_j^H is

$$\begin{aligned} R_j^H &= \frac{y_j}{\sum_j y_j + y_H} * \frac{y_j + v_j}{y_j + a_j + v_j} \\ &= \frac{\sum_i c_i w_{ij}}{\sum_j \sum_i c_i w_{ij} + y_H} * \frac{\sum_i c_i w_{ij} + \sum_k \sum_i c_i s_{ijk}}{\sum_i c_i w_{ij} + \sum_k \sum_i c_i s_{ikj} + \sum_k \sum_i c_i s_{ijk}} \end{aligned} \tag{6}$$

where y_H represents the computing power of honest mining pool. The real block withholding attack revenue R_j^W comes from the computing power of block withholding attack $v_{jk}, \forall k$ used to attack other mining pool k in attacking mining pool j. So R_j^W is

$$\begin{aligned} R_j^W &= \sum_k \left(\frac{y_k}{\sum_j y_j + y_H} * \frac{v_{jk}}{y_k + a_k + v_k} \right) \\ &= \sum_k \left(\frac{\sum_i c_i w_{ik}}{\sum_j \sum_i c_i w_{ij} + y_H} * \frac{\sum_i c_i s_{ijk}}{\sum_i c_i w_{ik} + \sum_j \sum_i c_i s_{ijk} + \sum_j \sum_i c_i s_{ikj}} \right) \end{aligned} \tag{7}$$

We consider the mining pool needs to consume costs such as electricity and water to carry out honest mining or block withholding attacks. Due to the certain differences of resource costs in various regions, we set the cost of honest mining C_H and the cost of block withholding attack C_p according to the region of the miner and calculate the total revenue of the mining pool. The total revenue R_j is the total computing power revenue of the mining pool j minus the cost consumed.

$$R_j = R_j^W + R_j^H - y_j C_H - v_j C_p \tag{8}$$

Since miner shares the total revenue of its mining pool according to the computing power provided by himself, we let R_i^c represent each miner's revenue. It is

$$R_i^c = R_j \frac{c_i}{y_j + v_j} \tag{9}$$

Then, we transform formula (9) into the revenue optimization model of each attacking mining pool j, which is based on the condition that miners have fixed computing power and the revenue of the mining pool is evenly allocated.

$$\begin{aligned} &\max(R_j) \\ \text{s.t. formulas}&(1)-(8), \forall i,k \\ &x_{ijk} \in \{0,1\}, \forall i,k \end{aligned} \tag{10}$$

At the same time, we consider that the maintainer of the attacking mining pool hopes that each miner in the attacking mining pool ensures that the difference in revenue is as small as possible while increasing the revenue, so as to achieve an overall increase in the revenue of the miners in the attacking mining pool. Therefore, we choose average miner revenue R_m^{av}, minimum miner revenue R_m^{min} and revenue miner variance var_m to establish revenue optimization model of entire attacking mining pools under block withholding attack.

$$\begin{aligned} &\max(R_m^{av} R_m^{\min}/\mathrm{var_m}) \\ \text{s.t. formulas}&(1)-(9), \forall i,j,k \\ &x_{ijk} \in \{0,1\}, \forall i,j,k \end{aligned} \tag{11}$$

3.2 Model solution

Formulas (10) and (11) are game problems of individual and entire attacking mining pools, and the direct solution is more complex. Currently, Newton algorithm, gradient descent algorithm and other optimization algorithms for solving the model are complicated in calculation and difficult to solve the nonlinear optimization problem. Genetic algorithm, ant colony algorithm, simulated annealing algorithm and other traditional artificial intelligence algorithms tend to fall into local optimal solutions within a limited number of iterations. Reinforcement learning and some other learning algorithms achieve goal optimization through continuous learning of optimal strategies, but they require a large amount of data sample in the optimization process. Artificial bee colony algorithm means that an intelligent optimization algorithm realizes model optimization by simulating the honey-collecting process of bees [34]. The algorithm expresses the solution in the process of model optimization in the form of a food source and finds the optimal solution through operations such as employed bee operations, onlooker bee operations and scout bee operations. Among them, employed bee operations select food sources to generate new food sources and perform evaluation value calculation to retain food sources with better evaluation values. Onlooker bee operations calculate the selection probability of each food source based on the evaluation value and select the food source to generate a new food source with the roulette method. Then, they perform the evaluation value calculation operation to retain the foods with good evaluation values. Scout bee operations eliminate the food sources that fall into the local optimal solution. Therefore, the artificial bee colony algorithm can find the global optimal solution through optimization operations in different stages and has a faster convergence speed. The artificial bee colony algorithm can solve the single-objective optimization problem. But according to formula (10), each attacking mining pool wants to maximize its own

revenue, which is a multi-objective optimization problem. Therefore, we propose the modified artificial bee colony algorithm based on Pareto. The specific solution process is as follows.

3.2.1 Population initialization

Let N_w, N_c and N_v , respectively, represent the number of miners, the number of attacking mining pools and the number of honest mining pools. Then, we use the array as food sources. The number of rows in the array represents miner's serial number. The first column in each food source represents the attacking mining pool which the miner belongs to, and the second column represents the attacked mining pool which the miner is currently in. The random initialization of each food source is that MROA generates an array of zero values and repeats the following operations for N_w times until the initialization of food source is completed: It randomly selects the natural number from 1 to N_c and replaces zero value in the first column of the array with the random value. Then, it randomly selects the natural number from 1 to $N_c + N_v$ and replaces the zero value in the second column of the array with the random value.

3.2.2 Employed bee operation

MROA calculates the total computing power that each attacking mining pool can use for honest mining and block withholding attack by formula (4) and formula (5). In order to select the food source with the largest evaluation value as the optimal food source x_{sta}, MROA combines the following formula to calculate the evaluation value of each food source f_m.

$$f_m = R_m^{av} R_m^{\min} / \mathrm{var_m} \tag{12}$$

Then, MROA uses the evaluation value of each food source to calculate selection probability P_m^{se} in crossover operation and mutation operation.

$$P_m^{se} = f_m / \sum_{m=1}^{SN} f_m \tag{13}$$

According to the optimal food source x_{sta} and selection probability of each food source, MROA crosses the food sources. That is, it takes the optimal food source x_{sta} as a fixed parent food source, and determines another paternal food source x_{alt} by the selection probability and roulette method. Letting that current row is the first row, MROA carries out the following operations for N_w times in turn until the crossover operation of the two food sources is completed: it randomly generates a crossover factor η_1; if the crossover factor η_1 is larger than the threshold $\eta_{1\mathrm{thr}}$, it will select the current row in food source x_{sta}; otherwise, it will select the current row in food source x_{alt}; the number of current row adds 1.

The crossover operation that produces a new food source with two original food sources can increase the diversity of food sources, but it is also necessary to change the food source in a favorable direction through the mutation operation of the selected food source. Therefore, mutation operation can help MROA to reduce that the revenue gap among mining pools, that is, letting that current row is the first row, MROA carries out the following

operations for N_w times until the mutation operation of the food source is completed: it randomly generates a mutation factor φ_1; if the mutation factor φ_1 is larger than the threshold $\varphi_{1\text{thr}}$, it uses the formula (8) to obtain the revenue of each attacking mining pool in the food source and replaces the current row according to formula (14), which makes the mutation result of food source beneficial to the current attacking mining pool of lowest revenue; Otherwise, it does not change; the number of current row adds 1.

$$\mathrm{x}_{\mathrm{new}}^{\mathrm{k}} = [\mathrm{R}_{\mathrm{Last}}, \mathrm{R}_{\mathrm{First}}] \tag{14}$$

where $\mathrm{x}_{\mathrm{new}}^{\mathrm{k}}$ represents kth row in new food source, R_{Last} represents the attacking mining pool of lowest revenue, R_{First} represents the mining pool of largest revenue.

In conclusion, MORA generates a large number of new food sources after evaluation value calculation, selection probability calculation, crossover operation and mutation operation of food sources. Therefore, MROA uses the method of elite retention to compare the quality of new and old food sources. If the evaluation value of new food source is greater than that of the old food source, the new food source will replace the old food source. Otherwise, the old food source will remain unchanged. At the same time, MROA realizes the Pareto dominance method through formula (15). If MROA finds that food source κ and food source λ have a dominant relationship in the revenue of each attacking mining pool, MROA will put the dominant food source into the non-dominated solution set QF_1.

$$R_i(\kappa) \geq R_i(\lambda) \text{ and } \exists R_\varepsilon(\kappa) > R_\varepsilon(\lambda), i = 1, \ldots \varepsilon \ldots, N_c \tag{15}$$

3.2.3 Onlooker bee operation

In the non-dominated solution set QF_1, some solutions are concentrated in a certain region. Some solutions are sparse. Therefore, it is necessary to calculate the crowding degree of food sources and obtain space metric of food sources with their neighbors. According to the metric, MROA achieves the non-dominated solution set QF_1 maintenance, that is, it calculates the revenue of each attacking mining pool in the non-dominated solution set QF_1 and sorts the food sources according to the calculation result. MROA finds the corresponding revenues of attacking mining pools $R_j(\mathrm{x}_{\mathrm{m}+1})$ and $R_j(\mathrm{x}_{\mathrm{m}-1})$ according to the sorting result and uses formula (16) to calculate the crowding degree d_i of the food sources.

$$d_i = \begin{cases} \infty, & i = 1 \text{ and } s_{\mathrm{QF}_1} \\ \sum\limits_{j=1}^{N_c} \frac{|R_j(\mathrm{x}_{\mathrm{m}+1}) - R_j(\mathrm{x}_{\mathrm{m}-1})|}{R_j^{\max} - R_j^{\min}}, & 1 < i < s_{\mathrm{QF}_1} \end{cases} \tag{16}$$

where $R_j^{\max}$ and $R_j^{\min}$, respectively, represent the maximum and minimum revenue of attacking mining pool j. According to the calculated crowding degree, MROA uses formula (17) to calculate the confirmation probability P_i^{con} of the ith food source in the non-dominated solution set QF_1.

$$P_i^{\mathrm{con}} = d_i / \sum_{m=1}^{s_{\mathrm{QF}_1}} d_i \tag{17}$$

Then, MROA determines the food source x_ψ based on the confirmation probability of each food source. MROA finds the new food source by carrying out the same

neighborhood crossover operation and neighborhood mutation operation for the food source x_{ψ}. The neighborhood crossover operation, neighborhood mutation operation and elite retention are basically the same as those of the employed bee operation. MROA can obtain the non-dominated set QF_2 through the Pareto dominance method of the new food source. If the number of food sources in the set QF_2 is below the threshold value, MROA supplements some food sources through the initialization of food sources. When the algorithm iteration is completed, it uses the non-dominated solution set QF_2 as the solution set for maximum revenue of multiple mining pools, and we take the food source with the largest evaluation value as the optimal solution.

3.2.4 Scout bee operation

MROA records the number of times ς_i that food source i cannot produce new food sources. If $\varsigma_i > \varsigma_{thr}$, then MROA obtains a new food source x_m^n by formula (18) to replace the corresponding food source in the set.

$$x_m^n = x_{max} + \kappa x_{max} - x_{\sigma} \tag{18}$$

where x_{max} represents the least crowding degree of food source in the non-dominated solution set QF_2, x_{σ} represents the food source with the largest similarity to food source x_{max}. κ represents an $N_w * 1$ array composed of 0 or 1 random numbers. If there is a negative number in food source x_m^n, the number will take absolute value and replace the negative number.

4 Algorithm implementation

As shown in Fig. 2, the specific implementation steps of MROA are as follows:

Step 1: MROA initializes the parameters such as number of miners N_w, number of attacking mining pools N_c, number of honest mining pools N_v, total number of food sources in the population SN, mutation factor threshold φ_{1thr}, and number of iterations item $= 0$.

Step 2: MROA initializes all food sources in the population. That is, it generates SN arrays of $N_w * 2$ dimensions whose values are all zero. It repeats the random selection of natural number from 1 to N_c and replaces the zero value with the random value in the first column of the array. Then, it repeats the random selection of natural number from 1 to $N_c + N_v$ and replaces the zero value with the random value in the second column of the array until the assignment of the arrays is completed.

Step 3: MROA uses formulas (12) and (13) to calculate the evaluation value and selection probability and selects the food source with largest evaluation value as a fixed parent food source x_{sta}. m=1.

Step 4: MROA selects a food source x_{alt} with the roulette method for crossover. The current row sets the first row. Then, it carries out the following operations for N_w times in turn until the crossover operation of food sources is completed to obtain a new food source: It randomly generates a crossover factor η_1; if the crossover factor η_1 is greater than the threshold value η_{1thr}, it will select the current row in x_{sta}; otherwise, it will select the current row in the food source x_{alt}; the number of current row adds 1.

Step 5: Current row sets the first row. Then, MROA carries out the following operations for N_w times in turn until the mutation operation of food source is completed:

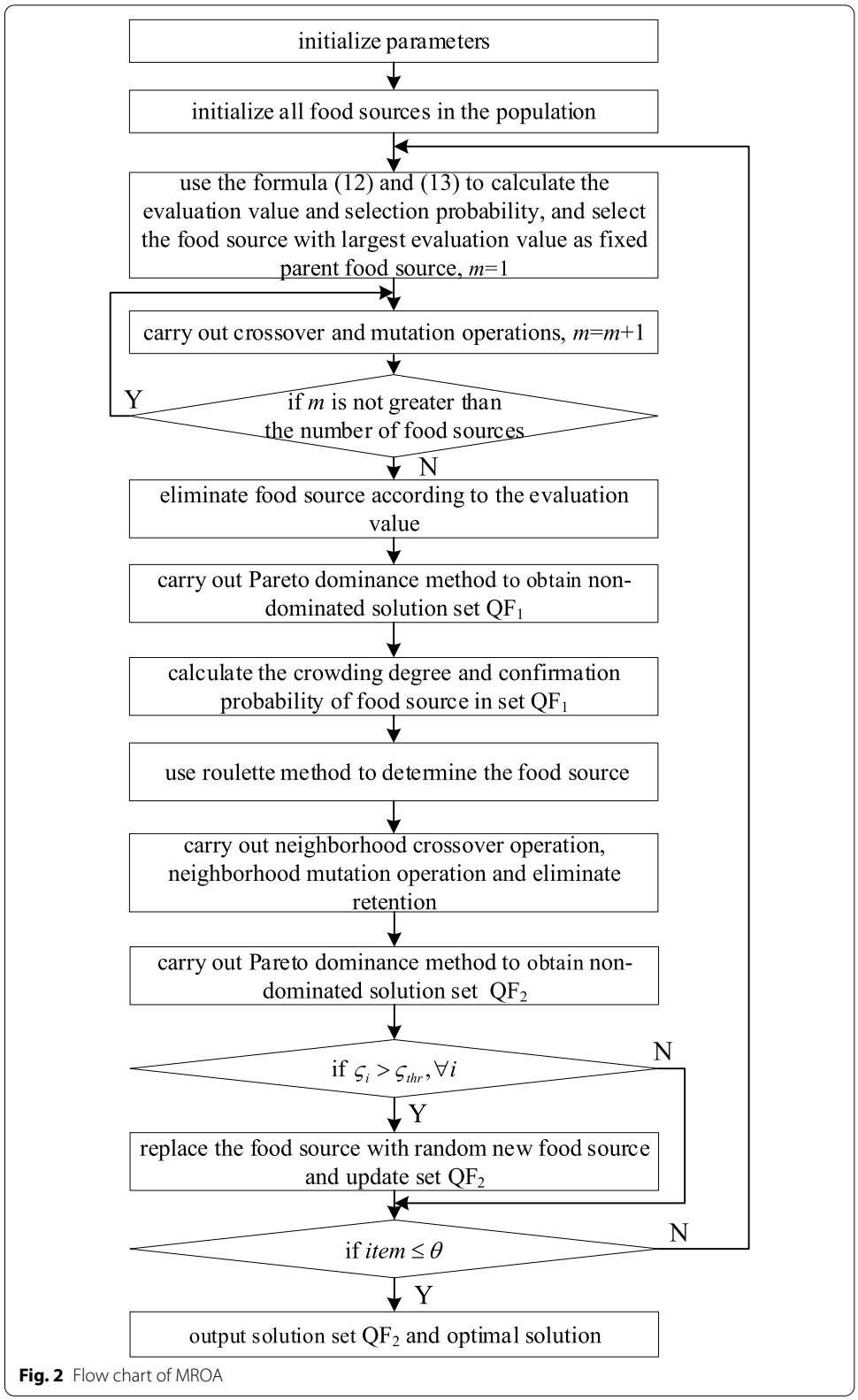

Fig. 2 Flow chart of MROA

It randomly generates a mutation factor φ_1; if the mutation factor φ_1 is greater than threshold value $\varphi_{1\text{thr}}$, it uses formula (8) to obtain the revenue of each attacking mining pool in the food source. Then, it replaces the current row with the row in x^k_{new} by formula (14); otherwise, it does not change; the serial number of current row adds 1. $m = m + 1$. If m is not greater than the number of food sources, skip to step 4. Otherwise, skip to step 6.

Step 6: If evaluation value of new food source is larger than original food source, MROA replaces the original food source with new food source. Otherwise, the original food source remains unchanged. And MROA carries out the Pareto domination method to get the non-dominated solution set QF_1.

Step 7: MROA uses formula (16) to calculate the crowding degree of the non-dominated solution set QF_1 and sorts in descending order. Then, it calculates confirmation probability by crowding degree and uses the roulette method to determine the food source for each onlooker bee operation.

Step 8: MROA carries out neighborhood crossover operation and neighborhood mutation operation and elite retention on set QF_1.

Step 9: MROA obtains the non-dominated solution set QF_2 by carrying out the Pareto dominance method on set QF_1. If the number of food sources in the set QF_2 is below the threshold value, MROA supplements some food sources through the initialization of food sources.

Step 10: If the number of times ς_i of food source i is larger than threshold ς_{thr}, it means that food source i in set QF_2 cannot generate a new food source. Then, MROA replaces the food source i with random new food source and updates set QF_2. The number of iterations item = item + 1. If item $\leq \theta$, skip to step 3. Otherwise, MROA obtains the solution set QF_2 to maximize the revenue of multiple attacking mining pools, and we take the food source with the largest evaluation value as the optimal solution.

According to the above flow chart, the pseudo-code of MROA is as follows, and the time complexity of MROA is analyzed on the basis of the pseudo-code. MROA mainly includes four parts, such as population initialization, employed bee operation, onlooker bee operation, and scout bee operation. The first part is to initialize the food source, that is, its time complexity is $\Theta(SN)$. The second part is that MROA performs crossover operation, mutation operation and Pareto dominance method of food sources, that is, its time complexity is $\Theta(SN^2N_c\theta)$. The third part is that MROA performs crowding calculation of food sources, neighborhood crossover operation, neighborhood mutation operation and Pareto dominance method on food sources, that is, its time complexity is $\Theta(SN^2N_c\theta)$. The fourth part is that MROA updates the food source in time, that is, its time complexity is $\Theta(\theta SN)$. In summary, the time complexity of MROA is $\Theta(SN^2N_c\theta)$. At present, the time complexity of classic multi-objective optimization algorithms such as NSGA2 (non-dominated sorting genetic algorithm II) [35] and MOPSO (multiple objective particle swarm optimization) [36] is $\Theta(SN^2N_c\theta)$, so the time complexity of the MROA is the same as that of other classic multi-objective optimization algorithms, without increasing the complexity of the algorithm.

Algorithm 1 Miner revenue optimization algorithm based on Pareto artificial bee colony in blockchain network (MROA).

Require: Information such as the number of mining pools and the number of miners in the network;
Ensure: Miner work plan for each attacking mining pool;
$N_c = 6$, $N_v = 1$, $SN = 20$, $\theta = 50$, $\eta_{1thr} = 0.05$, $\varphi_{1thr} = 0.02$, ...;
Generate SN arrays of $2N_w$ dimensions arrays as food sources;
for $l = 1$ to θ **do** //Employed bee operation
    Use the formula (12) and (13) to calculate the evaluation value and selection probability, and select the food source with the largest evaluation value as fixed parent food source x_{sta};
    for $m = 1$ to θ **do**
        Select the *m*th food source with the roulette method for crossover and mutation operation;
    end for
    Calculate the evaluation value of the food source and eliminate the food source;
    for $i = 1$ to SN **do**
        for $j = 1$ to SN **do**
            for $k = 1$ to N_c **do**
                Combine the *i*th food source and the *j*th food source to carry out the Pareto dominance method on the revenue of the *k*th attacking mining pool;
            end for
        end for
    end for//Onlooker bee operation;
    Use the formula (16) and (17) to calculate the crowding degree and confirmation probability;
    Select the food source with the roulette method for neighborhood crossover and neighborhood mutation operation;
    Calculate the evaluation value of the food source and update the food source;
    for $i = 1$ to SN **do**
        for $j = 1$ to SN **do**
            for $k = 1$ to N_c **do**
                Combine the *i*th food source and the *j*th food source to carry out the Pareto dominance method on the revenue of the *k*th attacking mining pool;
            end for
        end for
    end for
    for $i = 1$ to SN **do** //Scout bee operation;
        if $(\varsigma_i > \varsigma_{thr})$ **then**
            Combine the formula (18) to generate a new food source to update the food source;
        end if
    end for
end for
Select the food source with the largest evaluation value as the optimal solution;

5 Results and discussion

5.1 Simulation parameters and performance parameters

To analyze the performance of MROA, we refer to the distribution of mining pools in the real Bitcoin network in the btc126.com website. According to the statistics of btc126.com, the computing power distribution diagram of mining pools in the Bitcoin network is shown in Fig. 3. The attacking mining pools mainly refer to the AntPool, ViaBTC, BTC.com, and SlushPool mining pool in the real Bitcoin network. The miner in the attacking mining pool carries out honest mining or block withholding attack on other mining pools. The honest mining pool refers to the remaining mining pools in the real Bitcoin network. The miner in the honest mining pool carries out honest mining. According to the above algorithm simulation environment and references [18, 37, 38], we use Table 1 to analyze the influence of cost parameters of honest mining and block withholding attack on the average mining pool revenue and average miner revenue. Moreover, we select NSGA2, MOPSO and ABC (artificial bee colony) [34] as comparison algorithms and calculate the minimum mining pool revenue, mining pool revenue variance, average mining pool revenue, minimum miner revenue, miner revenue variance, average miner revenue, the evaluation value and spacing value of optimal solution when the number of attacking mining pools and miners change. Among them, NSGA2, MOPSO and ABC all choose formula (12) as weight value. The

Table 1 Simulation parameter

Parameter name	Value	Parameter name	Value
Number of honest mining pools N_v	1	Initial computing power of each miner c_i	0.1
Number of attacking mining pools N_c	6	Computing power of honest mining pool	0.5
Maximum number of iterations θ	50	Crossover factor threshold $\eta_{1\text{thr}}$	0.05
Block withholding attack cost C_p	0.02	Network reward value	1000
Mutation factor threshold $\varphi_{1\text{thr}}$	0.02	Number of onlooker bees	10
Number of food sources SN	20	Honest Mining cost C_H	0.05

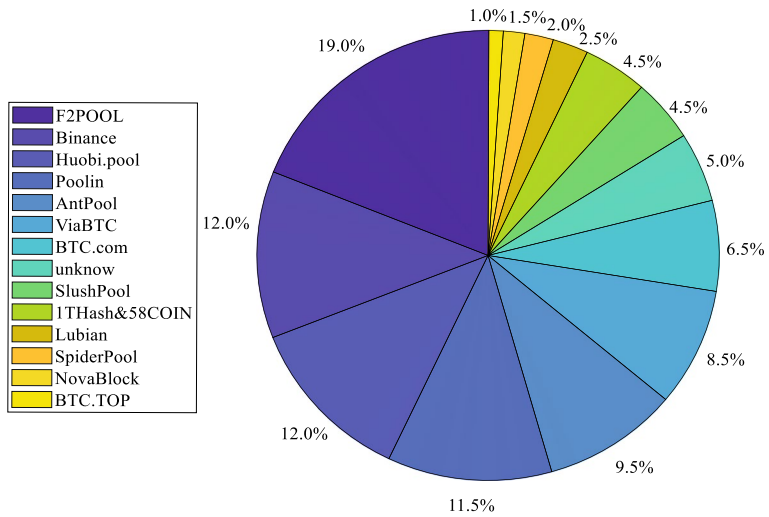

Fig. 3 The distribution diagram of computing power of mining pools in real Bitcoin network

NSGA2 optimizes the mining pool revenue through a fast non-dominated sorting strategy of elite reservation and selects the solution with the largest evaluation value as the optimal solution in the solution set. The MOPSO optimizes the mining pool revenue through double set and adaptive grid method and also selects the optimal solution according to the evaluation value. ABC is a classic bee colony algorithm. In order to find the optimal solution through iterations, ABC calculates the evaluation value of food source by formula (12). The average mining pool revenue or miner revenue means the attacking mining pools or miners obtain the average revenue in the optimal solution. The minimum mining pool revenue or miner revenue means attacking mining pools or miners obtain the minimum revenue in the optimal solution. The revenue variance of mining pools or miners means attacking mining pools or miners to obtain the revenue variance in the optimal solution. The spacing value S_P means the minimum standard deviation from each solution to other solutions in the output solution set.

5.2 Analysis of simulation results

5.2.1 Algorithm convergence and optimal scheme analysis

We select the maximum number of iterations 100, number of miners 120, and other parameters in Table 1 to obtain the non-dominated solution set in MROA. Then, we

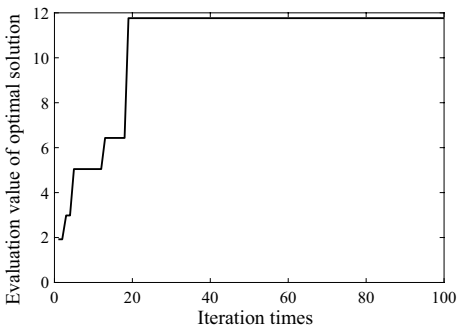

Fig. 4 Convergence chart of evaluation value of food source

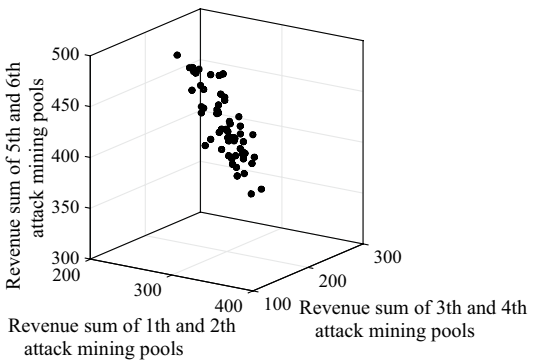

Fig. 5 Distribution map of non-dominated solution set in MROA

analyze the convergence of MROA. As shown in Fig. 4, MROA selects the food source with the largest evaluation value to carry out crossover operation with other food sources, which ensures that the offspring can inherit the current optimal food source as much as possible. And MROA carries out the mutation operation to change the food source in a favorable direction. In the onlooker bee operations, MROA updates the food source with low evaluation value through crowding calculation, Pareto domination and other operations. In the scout bee operations, MROA gives up the local optimal food sources, which cannot generate new food sources. These operations can quickly find the current optimal food source, so MROA can find the optimal evaluation value of food source about 23 iterations and the actual running time is 10.864s. As shown in Fig. 5, according to the revenue of the six attacking mining pools, we select the revenue sum of two attacking mining pools as a coordinate value to obtain the distribution of the solution set in the three-dimensional space. In the process of iteration, MROA considers the mutation operation to ensure the overall revenue of attacking mining pools while reducing the revenue gap among attacking mining pools. Then, MROA ensures the uniformity and convergence of non-dominated solution set by the congestion calculation, Pareto dominance method, and other operations. Therefore, the distribution of the solution set of MROA in the three-dimensional space is relatively uniform and can form the Pareto front. We calculate

the evaluation value of the food source in the non-dominated solution set according to formula (12) and select the food source with the largest evaluation value as the final output solution.

As shown in Fig. 6, it is the allocation of the computing power of each mining pool in the optimal solution of MROA, where x and y coordinate axes represent the serial number of each mining pool, z coordinate axis represents the allocation of computing power used by each mining pool for honest mining or block withholding attack. Attacking mining pool 1–6 can freely choose between block withholding attack and honest mining. Mining pool 7 is an honest mining pool, which cannot carry out block withholding attack, but they can carry out honest mining. The attacking mining pool reserves computing power 0.4 in its own mining pool for honest mining. In order to ensure the miner's revenue inside the mining pool, the attacking mining pool widely attacks other mining pools, so each of them uses around computing power 1.3 for block withholding attack. Since the honest mining pool cannot carry out block withholding attack, it is more likely to be attacked by attacking mining pools. So the total computing power of the honest mining pool subjected to block withholding attack is 1.6.

5.2.2 Influence of cost parameters on MROA

We select honest mining cost 0.01, 0.02, 0.03, 0.04 and 0.05, number of attacking mining pools 6, 7, 8, 9, 10, number of miners 60, 90, 120, 150, other parameters in Table 1 to analyze the influence of honest mining cost on the average mining pool revenue and the average miner revenue.

As shown in Figs. 7 and 8, when the number of attacking mining pools is 6 and the honest mining cost and block withholding attack cost are 0.01, the average mining pool revenue and average miner revenue reach the maximum revenue under the reasonable allocation in MROA (average mining pool revenue is 138.5 and average miner revenue is 14.48). With the increase in honest mining cost, attacking mining pools and miners need to consume more cost for honest mining, which leads to the decrease in average mining pool revenue and miner revenue, so the average mining pool revenue and miner revenue in MROA gradually decrease. When the number of attacking mining pools is 6, the average mining pool revenue is the sum of miner revenue in

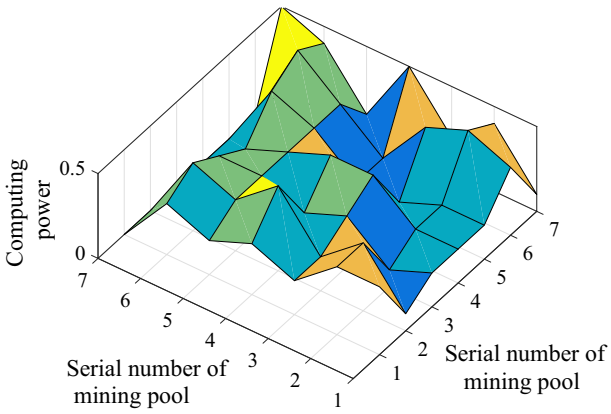

Fig. 6 The allocation of computing power of each mining pool in the optimal solution in MROA.

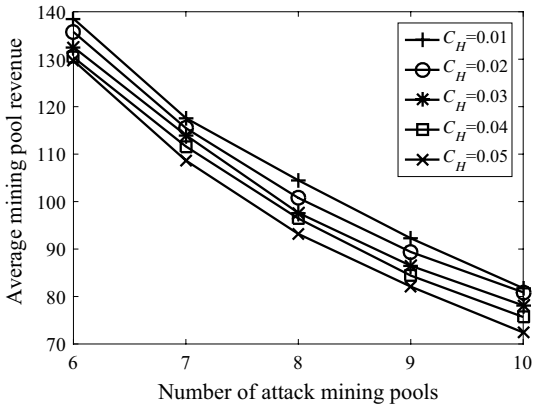

Fig. 7 Influence of honest mining cost on the average mining pool revenue

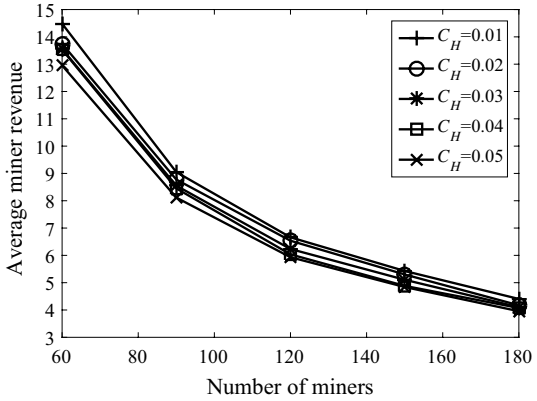

Fig. 8 Influence of honest mining cost on the average miner revenue

the attacking mining pools, and the number of attacking mining pools is more. So the average mining pool revenue decreases by 1.62% and the average miner revenue decreases by 2.7% when honest mining cost increases by 0.01. However, the increase in the number of attacking mining pools directly leads to more choices for miners to join the mining pool, and the increase in the number of miners directly leads to an increase in the mining pools that can be used for block withholding attack. Moreover, MROA can find the optimal plan according to current situation and reasonably allocate the computing power of miners, so as to deal with block withholding attack from other mining pools. Then, it results in a decline of average mining pool revenue by 1.6%, 1.9%, 2.8%, 2.8%, 2.9%, which gradually stabilizes the declining trend, and results in a decline of average miner revenue by 2.7%, 2.6%, 2.8%, 2.8%, 2.6%, which is no significant difference in the decline. Therefore, due to the continued increase in honest mining costs, the average mining pool revenue is significantly affected more than average miner revenue. As the number of attacked mining pools continues to increase, the declining trend of average mining pool revenue is gradually stable. But the increase in the number of miners has less influence on the average miner revenue. The decline range of the average miner revenue is basically unchanged.

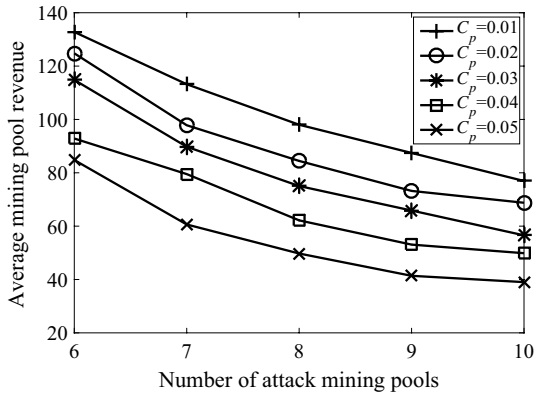

Fig. 9 Influence of block withholding attack cost on the average mining pool revenue

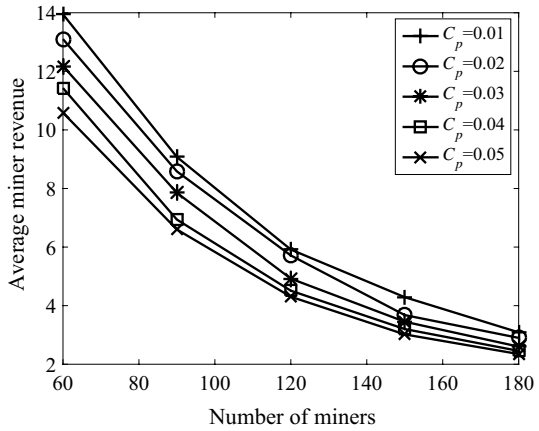

Fig. 10 Influence of block withholding attack cost on the average miner revenue

We select block withholding attack cost 0.01, 0.02, 0.03, 0.04 and 0.05, honest mining cost 0.03, number of attacking mining pools 6, 7, 8, 9, 10, number of miners 120 and other parameters in Table 1 to analyze the influence of block withholding attack cost on the average mining pool revenue and average miner revenue.

As shown in Figs. 9 and 10, when the number of attacking mining pools is 6, and the honest mining cost and block withholding attack cost are 0.01, average mining pool revenue and average miner revenue reach maximum revenue under the reasonable plan in MROA. (The average mining pool revenue is 132.79 and the average miner revenue is 13.94.) With the increase in block withholding attack cost, attacking mining pools and miners both need to spend more on block withholding attack, which leads to the decline of average mining pool revenue and miner revenue. When the number of attacking mining pools is 6, with each 0.01 increase of block withholding attack cost, the average mining pool revenue decreases by 10% on average, and the average miner revenue only decreases by 6.6%. With the increase in the number of attacking mining pools or the number of miners, MROA can find the optimal solution according to the attack situation among mining pools and reasonably allocate the computing power of miners. It

results in the average mining pool revenue decreased by 10%, 14%, 15%, 16% and 15%, and the declining trend is gradually stable. It also results in the average miner revenue decreased by 6.6%, 7.6%, 7.5%, 8.3%, 6.5%, which are with little difference. Therefore, with the increase in block withholding attack cost, average mining pool revenue is significantly affected more than average miner revenue. At the same time, with the increase in the number of mining pools, the declining trend of average mining pool revenue is stable. But the influence of the increasing number of miners is less. The decline range of the average miner revenue remains unchanged.

5.2.3 Algorithm performance comparison

Algorithm performance comparison under the change of number of attacking mining pools

We select the number of attacking mining pools 6, 7, 8, 9, 10, number of miners 120 and other parameters in Table 1 to analyze the minimum mining pool revenue. As shown in Fig. 11, when the network reward value is the same, as the number of attacking mining pools increases, the real mining revenue of the MROA, NSGA2 and MOPSO mining pools decreases, which directly reduces the minimum revenue of the algorithm. ABC chooses the direct optimization evaluation value to ensure the maximum social efficiency and does not consider the factor of mining pool revenue. ABC choses the direct optimization evaluation value to ensure maximum social efficiency and ignores factors such as mining pool revenue. Therefore, the minimum mining pool revenue in ABC is low, and there is a certain fluctuation. In order to ensure the quality of the non-dominated solution set in the process of optimizing the revenue of the dominated pool, NSGA2 uses the traditional crossover operation, mutation operation and Pareto dominance method. However, traditional optimization strategies can easily obtain the local optimal solution in complex situations. Although MOPSO selects particles in each iteration to ensure the quality of the non-dominated solution set, its selection process is uncertain and the crossover and mutation operations are simple. MROA not only adds the minimum mining pool revenue to the evaluation value and improves the common crossover and mutation operations to ensure the quality of the solution, but also adds the Pareto domination calculation, crowding calculation and other operations to avoid

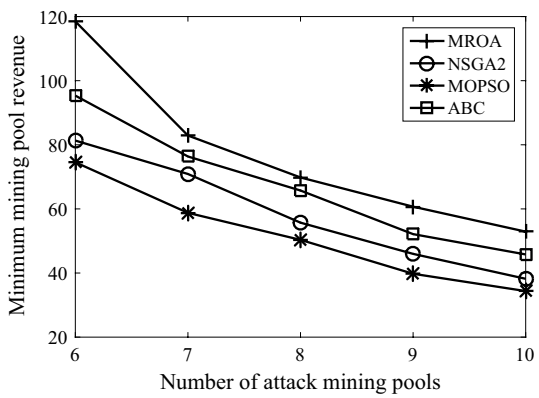

Fig. 11 Influence of the number of attacking mining pools on the minimum mining pool revenue

falling into a local optimal solution. MROA is significantly larger than NSGA2, MOPSO and ABC in terms of the minimum mining pool revenue.

We select the number of attacking mining pools 6, 7, 8, 9, 10, number of miners 120 and other parameters in Table 1 to analyze the average mining pool revenue. As shown in Table 2, with the increase in the number of attacking mining pools, the average mining pool revenue gradually decreases because the network rewards have the same value. MROA can be quickly realized by the modified artificial bee colony algorithm based on Pareto domination in finding the optimal solution. ABC, NSGA2 and MOPSO use the same evaluation function of MROA in the selection of a final solution and also allocate the current iterative network revenue reasonably. MOPSO has the worst effect because of the uncertainty in the process of particle selection. Mining pool average revenue of MROA is slightly larger than that of NSGA2, MOPSO and ABC. Since attacking mining pools freely choose block withholding attack and honest mining based on the current revenue situation, the average revenue of attacking mining pools is 32.5% larger than the average revenue of honest mining pools.

We select the number of attacking mining pools 6, 7, 8, 9, 10, number of miners 120 and other parameters in Table 1 to analyze the mining pool revenue variance. As shown in Fig. 12, MROA takes mining pool revenue variance into account. MROA uses mutation operation to the revenue gap among attacking mining pools and uses the Pareto dominance method to search for a multi-objective space solution set. It searches for the computing power allocation plan that can improve the revenue of all mining pools as much as possible. It also avoids falling into the local optimal solution and improves the convergence speed of the algorithm. ABC takes the evaluation value of algorithm as the only objective and mainly considers the maximization of group revenue. Its randomness in the solution process is large. The crossover operation and mutation operation in NSGA2 and MOPSO are simple, and their convergence speeds are slow. They still stay in the local optimal solution with low number of iterations. Therefore, with the limited number of iterations, MROA is lower than NSGA2, MOPSO and ABC in terms of the mining pool revenue variance. MROA decreases the revenue variance of attacking mining pools and has good fairness.

We select the number of attacking mining pools 6, 7, 8, 9, 10, number of miners 120 and other parameters in Table 1 to analyze the evaluation value of optimal solution. As shown in Fig. 13, as the number of aggressive mining pools increases, the decrease in mining pool revenue variance is smaller than the decrease in average revenue and minimum revenue, resulting in the decrease of the evaluation values of optimal solutions in

Table 2 Influence of number of mining pools on the average mining pool revenue

Number of attacking mining pools	Average mining pool revenue			
	MROA	NSGA2	MOPSO	ABC
6	132.7	131.5	104.7	132.4
7	112.5	111.0	91.2	111.7
8	98.5	95.5	78.0	97.8
9	85.9	84.4	64.4	84.7
10	77.4	75.2	59.9	76.9

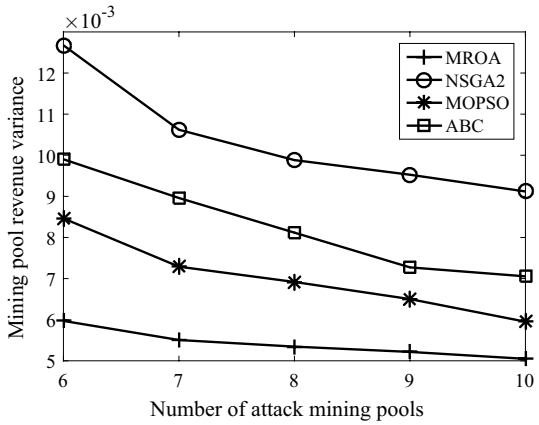

Fig. 12 Influence of number of attacking mining pools on the mining pool revenue variance

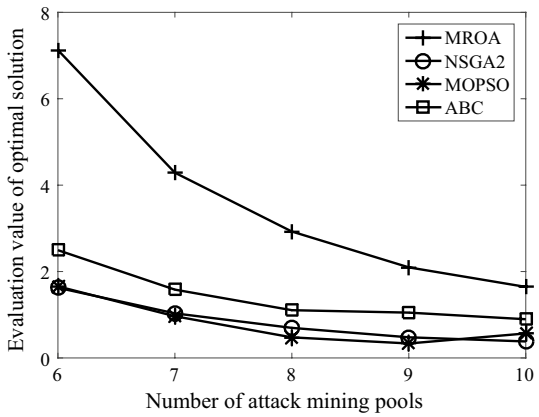

Fig. 13 Influence of number of attacking mining pools on the evaluation value of optimal solution

MROA, NSGA2, MOPSO and ABC. MROA improves the average and minimum mining pool revenues and reduces the mining pool revenue variance. According to the revenue average allocation of mining pools (formula (9)), MROA reduces the revenue gap of the miners as much as possible and improves the revenue of the average miner and the minimum miner. In terms of the evaluation value of the optimal solution, MROA is better than NSGA2, MOPSO and ABC.

We select the number of attacking mining pools 6, 7, 8, 9, 10, number of miners 120 and other parameters in Table 1 to analyze S_P of mining pools. As shown in Fig. 14, due to the increase in the number of attacking mining pools, the number of optimization targets and the difference in the non-dominated solution sets increases, which make the increasing of spacing value S_P in MROA, NSGA2, MOPSO and ABC. MROA not only uses the elite retention to retain the optimal solution during each iteration, but also determines the search direction for finding the optimal solution with mutation operation. In order to find a non-dominant solution set with a balanced distribution, it uses onlooker bee operations such as crowding calculation and Pareto dominance method. However, NSGA2 and MOPSO obtain the non-dominated solution set by optimizing

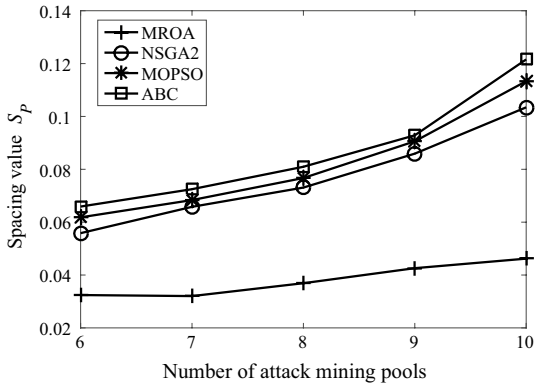

Fig. 14 Influence of number of attacking mining pools on the spacing value S_P

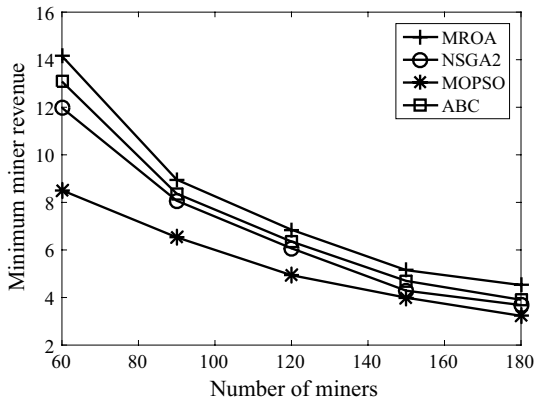

Fig. 15 Influence of number of miners on the minimum miner revenue

the mining pool revenue. ABC is a single-objective optimization algorithm and only obtains the latest generation of food source set. In terms of the spacing value, MROA is lower than NSGA2, MOPSO and ABC.

Algorithm Performance Comparison under the Change of Number of Miners

We select the number of miners 60, 90, 120, 150, 180 and other parameters in Table 1 to analyze the minimum miner revenue. As shown in Fig. 15, when the number of miners increases, in order to increase the revenue and reduce the influence of block withholding attack of other mining pools on its own mining pool, the attacking mining pools use part of their computing power for block withholding attack. It makes the percentage of honest mining computing power in the network's total computing power decrease and results in a gradual decrease in the total revenue that the entire network obtains. Therefore, the minimum miner revenue of MROA, NSGA2 and MOPSO gradually decreases. When the number of miners is 60, 80 and 120, the computing power in the network is relatively small. In the process of finding the optimal solution, MROA takes the minimum revenue of the miner as one of the evaluation parameters of the food source, so that the optimal solution in the iterative process can ensure the minimum revenue of miners. The convergence speeds of NSGA2 and MOPSO are relatively slow, and they only obtain the local optimal solution after 50 iterations. ABC improves the miner revenue by

directly optimizing the evaluation value, so the average miner revenue is better than that of NSGA2 and MOPSO. But ABC does not consider the revenue competition between mining pools, resulting in low revenue of some mining pools and miners. Therefore, the minimum miner revenue in MROA is larger than that in NSGA2, MOPSO and ABC. In terms of the minimum miner revenue, MROA is better than NSGA2, MOPSO and ABC. When the number of miners is 150 and 180, there is a large amount of computing power in the blockchain network, and each attacking mining pool has more computing power, which can be allocated to increase its own revenue. The allocation of computing power for each algorithm is roughly similar. Therefore, the minimum miner revenue of the MROA is slightly larger than that of the NSGA2, MOPSO and ABC algorithms, and the difference in the minimum miner revenue of each algorithm is gradually reduced.

We select the number of miners 60, 90, 120, 150, 180 and other parameters in Table 1 to analyze the average miner revenue. As shown in Table 3, as the number of miners increases, the average miner revenue of MROA, NSGA2, MOPSO and ABC has shown a gradual decline. In terms of the average miner revenue, MROA and ABC are slightly larger than NSGA2 and MOPSO. The ABC focuses on the average revenue of miners in the optimization process with formula (12), which is the same evaluation function of MROA; therefore, the average revenue of the ABC and the average revenue of MROA are not much different. The average miner revenue in MROA, NSGA2 and ABC is similar and obviously larger than that of MOPSO. Since attacking mining pools freely choose block withholding attack and honest mining, the average revenue of attacking mining pools is 31.2% larger than the average revenue of honest mining pools. The specific reason is the same as the influence of number of mining pools on the average mining pool revenue. Please refer to section 5.2.3.

We select the number of miners 60, 90, 120, 150, 180 and other parameters in Table 1 to analyze the miner revenue variance. As shown in Fig. 16, due to the increase in the number of miners, each attacking mining pool has enough computing power to allocate and increase its own revenue, so the miner revenue variance in each algorithm gradually decreases. When calculating the evaluation value of food sources, MROA not only takes the miners revenue variance as one of the evaluation parameters, but also improves the traditional crossover operation and elite retention. And MROA uses mutation manipulation and Pareto dominance method to ensure that the revenue gap of miners remains within a certain range. ABC considers the miner revenue, but its optimization algorithm is only the traditional single-objective bee colony algorithm, and its optimization effect is limited. In the iterative processes of NSGA2 and MOPSO, the main purpose is

Table 3 Influence of number of miners on average miner revenue

Number of miners	Average miner revenue			
	MROA	NSGA2	MOPSO	ABC
60	13.4	13.3	9.8	13.3
90	9.0	9.0	7.0	9.0
120	6.7	6.6	5.4	6.7
150	5.1	5.1	4.3	5.0
180	4.0	4.0	3.4	4.0

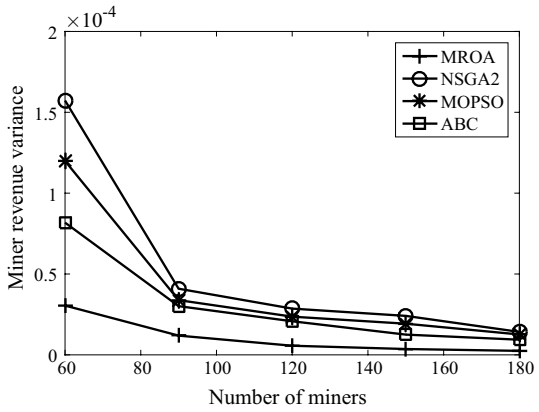

Fig. 16 Influence of number of miners on the miner revenue variance

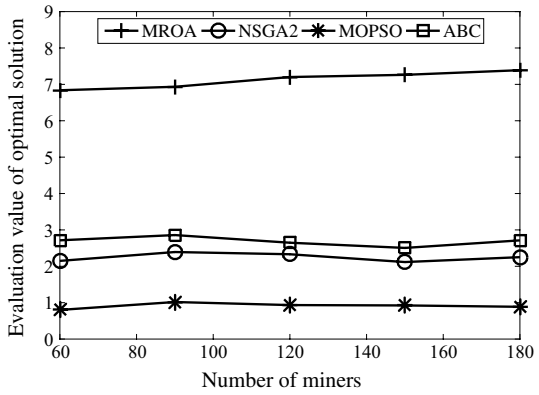

Fig. 17 Influence of number of miners on the evaluation value of optimal solution

to optimize the revenue among mining pools, and there is a local optimal solution in the optimization result. In terms of the miner revenue variance, MROA can balance the miner revenue and have better fairness.

We select the number of miners 60, 90, 120, 150, 180 and other parameters in Table 1 to analyze the evaluation value of optimal solution. As shown in Fig. 17, due to the increase in the number of miners, miner revenue variance, average miner revenue and minimum miner revenue have decreased accordingly. Each algorithm maintains a stable state in the evaluation value of the optimal solution. MROA can increase the minimum and average revenue of miners and reduce miner revenue variance with the same number of miners. Because ABC uses the evaluation value as the only objective to evaluate the food source, it can get a larger evaluation value in the iterative solution process. In terms of the evaluation value of optimal solution, MROA is better than NSGA2, MOPSO and ABC.

We select the number of miners 60, 90, 120, 150, 180 and other parameters in Table 1 to analyze the spacing value S_P of mining pools in MROA, NSGA2, MOPSO and ABC. As shown in Fig. 18, due to the increase in the number of miners, the disposable computing power in a single mining pool has increased significantly, and the difference among

the non-dominated solution sets of each algorithm decreases. The spacing value S_P of each algorithm gradually decrease. Moreover, MROA can better balance the distribution of non-dominated sets. In terms of spacing value S_P, MROA is lower than NSGA2, MOPSO. The specific reason is the same as the influence of number of attacking mining pools on the spacing value. Please refer to section 5.2.3.

6 Conclusion

This paper proposes the miner revenue optimization algorithm (MROA) based on Pareto artificial bee colony in blockchain network. According to block withholding attack, MROA establishes revenue optimization model of each attacking mining pool and revenue optimization model of entire attacking mining pools with the mathematical formulas such as attacking mining pool selection, effective computing power, mining cost and revenue. Secondly, we propose a modified artificial bee colony algorithm based on the Pareto method to maximize the model, which includes employed bee operations, onlooker bee operations and scout bee operations. That is, MROA initializes algorithm parameters and population. The employed bee operations include the evaluation value calculation, selection probability calculation, crossover operation, mutation operation and Pareto dominance method, and can update each food source. The onlooker bee operations include the confirmation probability calculation, crowding degree calculation, neighborhood crossover operation, neighborhood mutation operation and Pareto dominance method, and can find the food source in multidimensional space with smaller distribution density. The scout bee operations delete the local optimal food source that cannot produce new food sources to ensure the diversity of solutions. Finally, we analyze the influence of honest mining cost and block withholding attack cost on the algorithm revenue and compare the difference of minimum revenue, average revenue, revenue variance, spacing value and evaluation value of optimal solution in ABC, NSGA2, MOPSO and MROA.

The simulation results show that under the condition that the number of attacking mining pools and miners change, MROA increases the minimum revenue, average revenue and evaluation value of optimal solution, and reduces the revenue variance and spacing value of the solution set. The overall performance of MROA is better than NSGA2, MOPSO and ABC. But it does not consider the miner as an independent

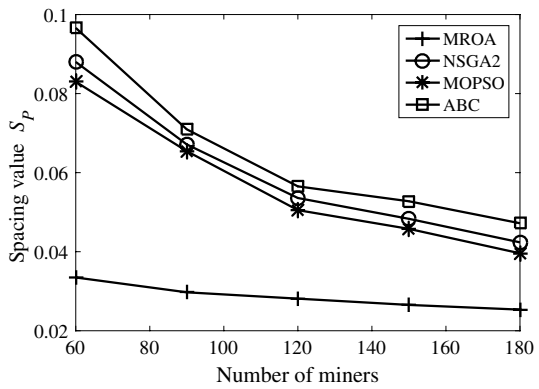

Fig. 18 Influence of number of miners on the spacing value S_P

individual against other miners. Therefore, the next stage is to study the game problem among miners under block withholding attack with game theory.

Abbreviations
IoT: Internet of Things; PHI: Personal health information; POW: Proof of work; MROA: Mining revenue optimization algorithm of miners in PoW-based blockchain networks; NSGA2: Non-dominated sorting genetic algorithm; MOPSO: Multiple objective particle swarm optimization; ABC: Artificial bee colony.

Acknowledgements
None.

Authors' contributions
YC wrote the entire article. YC, MH and QC are responsible for the algorithm principles. HC and BL are responsible for the algorithm simulation. ZW is responsible for the translation of the paper. ZM is responsible for the layout and guidance of the paper. All authors read and contributed in the writing and approved the final manuscript.

Funding
This work was supported by the Public Welfare Technology Application and Research Projects of Zhejiang Province of China under Grant No. LGG19F010011 and Grant No. LGF19F010005, the Natural Science Foundation of Zhejiang Province of China under Grant No. LQ18F030006, and the Special fund project for basic scientific research operation fees of provincial universities of Zhejiang Shuren University under Grant No. 2021X2018.

Availability of data and materials
The dataset used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for publication
All authors agree to submit this version and claim that no part of this manuscript has been published or submitted elsewhere.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]College of Information Science and Technology, Zhejiang Shuren University, Hangzhou 310015, China. [2]School of Computer Science and Artificial Intellgence, Changzhou University, Changzhou 213164, China. [3]Binjiang Institute, Zhejiang University, Hangzhou 310052, China.

Received: 7 November 2020 Accepted: 22 June 2021
Published online: 06 July 2021

References

1. Z. Cai, X. Zheng, J. Yu, A differential-private framework for urban traffic flows estimation via taxi companies. IEEE Trans. Ind. Inf. **15**(12), 6492–6499 (2019)
2. M. Swan, *Blockchain: Blue Print for a New Economy* (O'Reilly Media Inc, USA, 2015)
3. L. Ge, X. Ji, T. Jiang, Y. Jiang, Security mechanism for internet of things information sharing based on blockchain technology. J. Comput. Appl. **39**(2), 458–463 (2019)
4. A. P. Joshi, M. Han, Y. Wang, A survey on security and privacy issues of blockchain technology. Math. Found. Comput. **1**(2), 121–147 (2018)
5. D. Laufenberg, L. Li, H. Shahriar, M. Han, An architecture for blockchain-based collaborative signature-based intrusion detection system. In: Proceedings of the 20th Annual SIG Conference on Information Technology Education, pp. 169–169 (2019)
6. I. Bentov, C. Lee, A. Mizrahi, M. Rosenfeld, Proof of activity: extending bitcoin's proof of work via proof of stake. ACM SIGMETRICS Perform. Eval. Rev. **42**(3), 34–37 (2014)
7. Y. Yong, F. Wang, Blockchain: the state of the art and future trends. Acta Autom. Sin. **42**(4), 481–494 (2016)
8. X. Ling, C. Wu, S. Ji, M. Han, Http/2 dos: an application-layer dos attack towards http/2 protocol. In: International Conference on Security and Privacy in Communication Systems, pp. 550–570 (2017)
9. J. Han, J. Zou, H. Jiang, Q. Xu, Research on mining attacks in bitcoin. J. Cryptol. Res. **5**(5), 470–483 (2018)
10. X. Zheng, Z. Cai, Privacy-preserved data sharing towards multiple parties in industrial IOTs. IEEE J. Sel. Areas Commun. **38**(5), 968–979 (2020)
11. D. K. Tosh, S. Shetty, X. Liang, C. A. Kamhoua, K. A. Kwiat, L. Njilla, Security implications of blockchain cloud with analysis of block withholding attack. In: 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), pp. 458–467 (2017)

12. S. Zhu, W. Li, H. Li, L. Tian, G. Luo, Z. Cai, Coin hopping attack in blockchain-based IoT. IEEE Internet Things J. **6**(3), 4614–4626 (2018)
13. X. Xu, S. He, M. Han, R.M. Parizi, G. Srivastava, Budget feasible roadside unit allocation mechanism in vehicular ad-hoc networks. In: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), pp. 1–5 (2020)
14. W. Li, M. Cao, Y. Wang, C. Tang, F. Lin, Mining pool game model and nash equilibrium analysis for pow-based blockchain networks. IEEE Access **8**(8), 101049–101060 (2020)
15. T. Yang, Z. Xue, Game theory among mining pools in blockchain system. Commun. Technol. **52**(05), 1189–1195 (2019)
16. R. Qin, Y. Yuan, F. Wang, Optimal block withholding strategies for blockchain mining pools. IEEE Trans. Comput. Soc. Syst. **7**(3), 709–717 (2020)
17. C. Alkalay-Houlihan, N. Shah, The pure price of anarchy of pool block withholding attacks in bitcoin mining. Proc. AAAI Conf. Artif. Intell. **33**, 1724–1731 (2019)
18. D. Wu, X. Liu, X. Yan, R. Peng, G. Li, Equilibrium analysis of bitcoin block withholding attack: a generalized model. Reliab. Eng. Syst. Saf. **185**(5), 318–328 (2019)
19. C. Tang, Z. Yang, Z. Zheng, Z. Chen, L. Xiang, Game dilemma analysis and optimization of pow consensus algorithm. Acta Autom. Sin. **43**(9), 1520–1531 (2017)
20. Q. Hu, S. Wang, X. Cheng, A game theoretic analysis on block withholding attacks using the zero-determinant strategy. In: Proceedings of the International Symposium on Quality of Service., pp. 1–10 (2019)
21. R. Singh, A. D. Dwivedi, G. Srivastava, A. Wiszniewska-Matyszkiel, X. Cheng, A game theoretic analysis of resource mining in blockchain. Cluster Comput. **1–12** (2020)
22. S. Kim, S. Hahn, Mining pool manipulation in blockchain network over evolutionary block withholding attack. IEEE Access **7**(7), 144230–144244 (2019)
23. L. Luu, R. Saha, I. Parameshwaran, P. Saxena, A. Hobor, On power splitting games in distributed computation: the case of bitcoin pooled mining. In: 2015 IEEE 28th Computer Security Foundations Symposium, pp. 397–411 (2015)
24. A. T. Haghighat, M. Shajari, Block withholding game among bitcoin mining pools. Future Gen. Comput. Syst. **97**(8), 482–491 (2019)
25. T. Wang, S. Yu, B. Xu, Research on proof of work mining dilemma based on policy gradient algorithm. J. Comput. Appl. **39**(5), 1336–1342 (2019)
26. S. Bag, S. Ruj, K. Sakurai, Bitcoin block withholding attack: analysis and mitigation. IEEE Trans. Inf. Forensics Secur. **12**(8), 1967–1978 (2017)
27. J. SO, A new attack scheme on the bitcoin reward system. IEICE Trans. Fundam. Electron. **102**(1), 300–302 (2019)
28. X. Dong, F. Wu, A. Faree, D. Guo, Y. Shen, J. Ma, Selfholding: a combined attack model using selfish mining with block withholding attack. Comput. Secur. **87**(11), 1–11 (2019)
29. J. Ke, P. Szalachowski, J. Zhou, Q. Xu, Z. Yang, Ibwh: an intermittent block withholding attack with optimal mining reward rate. In: International Conference on Information Security, pp. 3–24 (2019)
30. S. Chang, Y. Park, S. Wuthier, C. Chen, Uncle-block attack: blockchain mining threat beyond block withholding for rational and uncooperative miners. In: International Conference on Applied Cryptography and Network Security, pp. 241–258 (2019)
31. Y. Wang, G. Yang, T. Li, L. Zhang, Y. Wang, L. Ke, Y. Dou, S. Li, X. Yu, Optimal mixed block withholding attacks based on reinforcement learning. Int. J. Intell. Syst. **9**(9), 1–17 (2020)
32. Y. Chen, H. Chen, M. Han, B. Liu, Q. Chen, T. Ren, A novel computing power allocation algorithm for blockchain system in multiple mining pools under withholding attack. IEEE Access **8**(8), 155630–155644 (2020)
33. H. Chen, Y. Chen, M. Han, B. Liu, Q. Chen, Z. Ma, A novel anti-attack revenue optimization algorithm in the proof-of-work based blockchain. In: International Conference on Wireless Algorithms, Systems, and Applications, pp. 40–50 (2020)
34. G. Li, L. Cui, X. Fu, Z. Wen, N. Lu, J. Lu, Artificial bee colony algorithm with gene recombination for numerical function optimization. Appl. Soft Comput. **52**(52), 146–159 (2017)
35. B. Zhao, Y. Xue, B. Xu, T. Ma, J. Liu, Multi-objective classification based on NSGA-II. Int. J. Comput. Sci. Math. **9**(6), 539–546 (2018)
36. R. Sivaranjani, S.M.M. Roomi, M. Senthilarasi, Speckle noise removal in SAR images using multi-objective PSO (MOPSO) algorithm. Appl. Soft Comput. **76**(76), 671–681 (2019)
37. X. Chen, B. Xu, C. Mei, Y. Ding, K. Li, Teaching-learning-based artificial bee colony for solar photovoltaic parameter estimation. Appl. Energy **212**(212), 1578–1588 (2018)
38. A. Singh, K. Deep, Artificial bee colony algorithm with improved search mechanism. Soft. Comput. **23**(23), 12437–12460 (2019)

Publisher's Note