Block Size Optimization for Block Chain Applications based on POW Algorithm using Differential Evolution Algorithm

Ali Izadi[1], Hamidreza Fardad[2*]
1- Department of Electrical and Computer Engineering, Majlesi Branch, Islamic Azad University, Majlesi, Iran.
Email: ali.ezadi.1991@gmail.com
2- Department of Electrical and Computer Engineering, Nikabad Branch, Islamic Azad University, Nikabad, Iran.
Email: hr.fardad@iau.ac.ir (Corresponding author)

Received: 2 September 2022 Revised: 13 October 2022 Accepted: 9 November 2022

ABSTRACT:
Block chain technology, known by popular crypto currencies mainly Bitcoin, is widely used in several fields, not only in the financial sector but also in the healthcare industry as well as supply chain management sectors. To produce an efficient block chain, consensus algorithms are an essential part of it. Proof of Work (POW) and Proof of Stake (POS) protocols are widely used in many block chain networks such as Bitcoin, Ethereum, and Litecoin. Block size optimization solves the trade-off between maximum revenue and becomes an alternative payment system. In this research, differential evolution algorithm was used to determine the size of the blocks. The simulation of this research has been done in MATLAB environment. In order to evaluate the proposed method, two scenarios of the number of miners and the type of request have been used. In this research, statistical criteria have been used to evaluate the proposed method. The criteria used in this research include maximum, minimum, average and standard deviation criteria. The results showed that in the scenario of the number of miners, the best answer was obtained in the case where the number of miners was equal to 3. In the request type scenario, the dependency of the requests has been checked. In this case, requests are considered dependent and non-dependent. The results showed that in the non-dependency mode, the request processing time has decreased because in this mode the requests can be executed simultaneously with each othe. in the dependency mode, the requests have priority over each other. In this case until the desired request is not executed, the next request cannot be executed.

KEYWORDS: Block Chain Technology, Bitcoin, Block Chain, Proof of Work, Differential Evolution Algorithm.

1. INTRODUCTION

Block chain technology, known by popular crypto currencies mainly Bitcoin, is widely used in several fields, not only in the financial sector but also in the healthcare industry as well as supply chain management sectors [1]. It is a type of distributed ledger technology that maintains an identical copy of the transactions of participants in a network [2]. In short, block chain is the merging of chains of each block, and the blocks are connected by keeping the hash value of the previous block [3]. Each block contains certain information: size, number of transactions, transactions themselves and block title [4].

To create new blocks in the chain, all parties in the chain network agree to a common agreement to write transactions in the ledger [5]. There are many protocols for the agreement of participants, which are called general consensus algorithms [6]. To produce an efficient block chain, consensus algorithms are the essential part [7]. Proof of Work (POW) [8] and Proof of Stake (POS) [9] protocols are widely used in many block chain networks such as Bitcoin, Ethereum, and Litecoin [10]. Block size as well as block creation time are focused on achieving Bitcoin scalability as key parameters [11]. Block size plays an important role in determining the overall performance of the proposed system [12].

Block size optimization [13] solves the trade between maximum revenue and becomes an alternative payment system [14]. In addition, in selecting the block size between security and speed in cryptographic creation, there is a cryptographic hashing function [15]. If the block size is smaller, then security becomes an important issue; If the block size is larger, the speed of its conversion and transfer becomes an issue [16].

Paper type: Research paper
DOI: 10.30486/mjtd.2023.1976630.1008
How to cite this paper: A. Izadi and H.R. Fardad, “**Block Size Optimization for Block Chain Applications based on POW Algorithm using Differential Evolution Algorithm**”, Majlesi Journal of Telecommunication Devices, Vol. 11, No. 4, pp. 205-213, 2022.

In this article, we have focused on determining the optimal block size and the goal of improving the block production time in POW systems. For this reason, the delay and block generation time are considered as an objective function. Make Note that transaction cost and security are outside the scope of this approach. Therefore, in this article, differential evolution algorithm is used to determine the size of the blocks.

Next, in the second part, the background of the research will be given, in the third part, the proposed method is stated. The results of the proposed method are stated in the fourth section. Finally, the conclusion is given in the fifth section

2. LITERATURE REVIEW

Cao and colleagues analyzed and discussed the consensus process in block chain based on POW, POS and DAG. They compared the main performance in terms of average time to generate a new block, confirmation delay, TPS, and probability of confirmation failure, and showed the impact of computing power on POW, possible balance, and new transactions entering the DAG [17]. Based on the existing consensus algorithm, Wu et al presented an improved hybrid consensus algorithm based on PBFT algorithm and POS algorithm. The proposed algorithm reduces the number of consensus nodes to a fixed value with verifiable pseudo-random sorting and witnesses the interaction between nodes [18]. In order to improve the efficiency and security of the block chain hashing algorithm, Fu and his colleagues presented an optimization plan for the block chain hashing algorithm based on PRCA. This scheme combines the block chain with an active configurable computer to improve the performance of the block chain hash function [19]. Khan and colleagues presented their efforts to address a detailed study of performance and scalability constraints for an E^2-voting system [20]. Singh and his colleagues proposed a method in which multi-objective optimization is performed and 40 different solutions are obtained based on transaction selection time and block construction time [21]. Aygun and his colleagues proposed a multi-objective optimization (OP) problem by minimizing block generation and transmission time [22]. In 2022, Francis and colleagues focused on the delay associated with proof-of-work (POW) based on block chain networks, whereby participants add new information to a distributed ledger through consensus to confirm transactions [23].

.

3. SUGGESTED METHOD

In this research, a hierarchical process has been used to allocate work to miners, and the allocation conditions have been stated in the first stage. In this section, the needs of the block chain network and the relationships related to these environments are given. In the second step, the allocation operation is done by differential evolution algorithm. In this section, operations such as the initial population, determination of the objective function, improvement of the initial population are performed. Population improvement will be done by two operations, mutation and combination. In the mutation operation, populations are changed by three other populations. In the combination operation, the mutated population is combined with the current population. In the following, each of the desired sections is described.

3.1. Allocation Of Blocks

In this section, the process of allocating blocks to miners is shown. This process is shown in Fig 1.

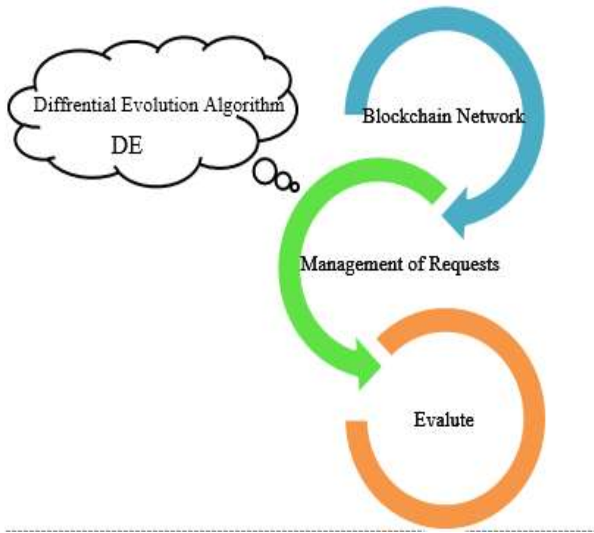


Fig. 1. Allocation of blocks.

As shown in Fig. 1, a three-step process is used to allocate blocks to miners. In the first stage, the explanations related to the block chain network are given and the method of receiving and processing requests is explained. In the second part, there is the process of handling requests. In other words, requests are managed in this section. In order to manage the requests, the has been used. In the final stage, the blocks are evaluated. In the following, each of the desired sections will be explained.

3.2. Blockchain Network

In every block chain program, miners connect with other miners through the network using network bandwidth. Each miner receives a number of transactions from a pool that must be verified before being stored in a block. The block chain network is Shown in Fig. 2, This network shows all the miners that are connected through a peer-to-peer network.

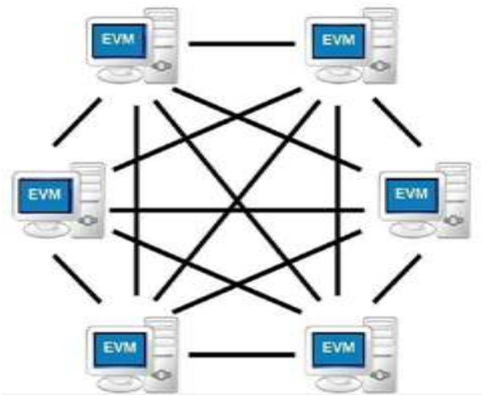


Fig. 2. Block chain network structure.

Miners send transactions in the form of blocks to other miners for the mining process. For this, miners receive transactions in moderation equal blocks from the pool. The miner tries to clear all transactions from the pool in order to achieve less transfer time and less block composition time. This is because depending on the type of application, millions of transactions can be generated that remain waiting to be mined. This problem for miner M1 is shown in Fig, 3.

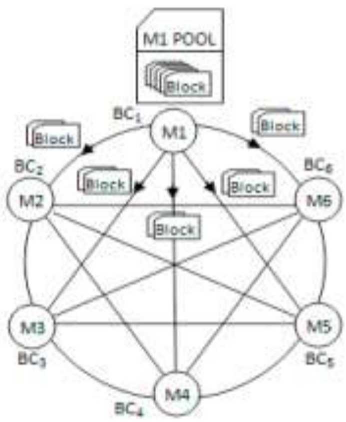


Fig. 3. Miner M1's problem in determining block size.

3.2.1. block combination time and transfer time

Block composition time and transfer time are two performance measurement factors in block chain networks. The time required by the miner to create a block is known as block composition time. Miner transfer time is the time required to transfer blocks. Block composition time includes the overhead of each block (block preprocessing time), which is determined based on the Merkel tree construction time. Merkel tree is an essential part of block chain technology that enables efficient and secure verification of transactions. Block composition time and transfer time depends on the number of transactions in each block (block size). that these two factors are contradictory in terms of performance. To solve the above problem, the size of the block should be determined in such a way that the transfer time and block combination time are minimized. The block construction time is calculated from equation (1):

$$T_b = \frac{S_{mem}}{S_b} * (T_{overhead} + (\frac{Tot_t}{MerkleTree_t} * T_{MerkleTree}) \tag{١}$$

In relation (1),S_{mem} indicates the size of the pool memory, $S_b$indicates the size of each block$S_b S_{mem}$indicates the total blocks produced. .$T_{overhead}$ overhead time, Merkle tree indicating the number of transactions in each block which is considered constant in this research. The value determined for it is equal to 10, Stindicates the size of each transaction, and Tot indicates the number of transactions, which is obtained from the relation$S_{b/}$ S_t. Tot / Merkle tree shows the number of Merkle trees that have been built. The time required to build a Merkle tree with 10 transactions is equal toT_{merkle}, which is considered 0.02 in this research. As mentioned above, block size is another important parameter for network propagation and transmission delay. The delay that depends on the block size is formulated in the following equation

$$T_d = h * (T_p + \frac{S_b}{R} * N_T) \tag{٢}$$

In relation (2), h is the depth of the block chain network tree and R is the bandwidth of the node. In this research, R is considered equal for all nodes (miners). NT is the number of miners connected to that node, which is calculated according to equation (3).

$$N_T = (n-1) * P_c \tag{٣}$$

Equation (4) will be used to calculate the depth of the tree.

$$\mathrm{h} = \log_P(n * (N_T - 1) + 1) \tag{٤}$$

As mentioned in (4), the target problem includes two objective functions that must be minimized. To convert two objective functions into one objective function, the weighting method is used, which is given in (5).

$$\mathrm{o} = \vec{w} * T_d + (1 - \vec{w}) * T_b \tag{٥}$$

In this research, the size of the blocks is to be adjusted by the differential evolution algorithm in such a way that the above fitness function is optimized.

3.3. Management of Requests

As mentioned in the previous section, differential evolution algorithm will be used to manage requests. The uses a multi-step process to adjust the size of the blocks, and these steps are shown in Fig. 4.

Fig. 4. Steps of differential evolution algorithm.

As shown in Figure (4), the differential evolution algorithm for adjusting the block size consists of three processes: initial population, fitness, and population improvement, which will be explained in the following sections.

3.3.1. Initial population

One of the important parts of the differential evolution algorithm is related to the primary population part. In this section, the coding of the problem is shown. The initial population can be used in two cases shown in Fig. 5.

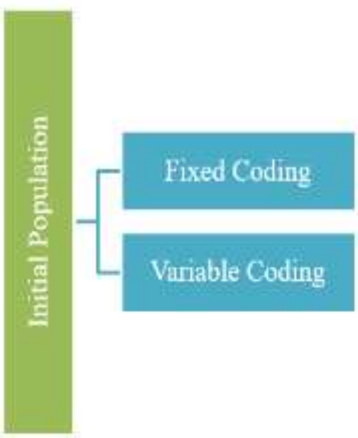


Fig. 5. Coding of the initial population.

As shown in Fig. 5, the initial population can be considered as fixed and variable. In the fixed mode, the format of all the populations is constant, but in the variable method, the format of the populations is variable. In this research, a fixed format is used to show the initial population. The format used in this research is as follows.

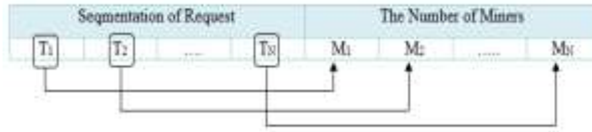


Fig. 6. Initial population format in fixed coding mode.

As shown in Fig. 6, each request is assigned to one of the available miners. The point used in this research is the way of coding the requests. For a better understanding of the coding problem, pay attention to the following example.

Example: Suppose there are three tasks in this research and each task is divided into five parts. Table (1) shows this process.

Table 1. Basic schematic of requests.

	Block 1	Block 2	Block 3	Block 4	Block 5
T1	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}
T2	T_{21}	T_{22}	T_{23}	T_{24}	T_{25}
T3	T_{31}	T_{32}	T_{33}	T_{34}	T_{35}

After determining the blocks related to each task, the following process is used to number the requests. A single number is assigned to each task. The numbering method is from top to bottom. The desired process is shown below.

Table 2. Allocation of code to requests.

	Block 1	Block 2	Block 3	Block 4	Block 5
T1	1	2	3	4	5
T2	6	7	8	9	10
T3	11	12	13	14	15

As shown in Table 2, each request is assigned a unique number. The important point regarding the allocation of requests to miners is the type of requests. Requests can be used in two ways shown in Fig. 7.

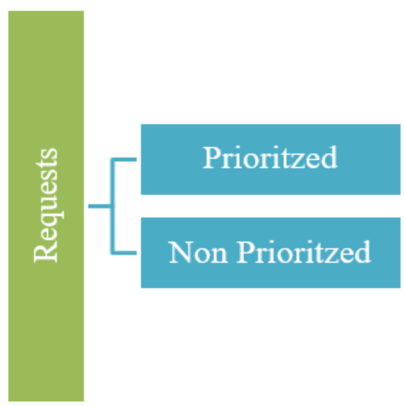


Fig. 7. Types of requests.

As shown in Fig. 7, requests can be considered as prioritized or non-prioritized. In the following, the difference between priority and non-priority requests is shown. for prioritized requests Fig. 8, is considered.

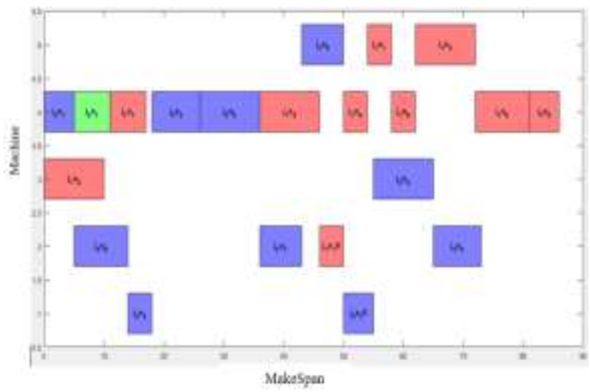


Fig. 8. Prioritized requests.

In Fig 9, it is considered fornon-prioritized requests.

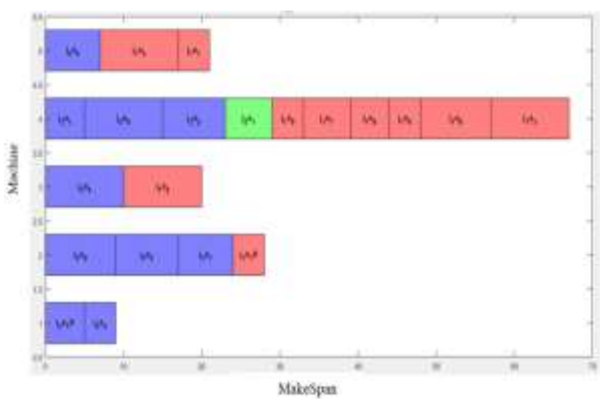


Fig. 9. Non-prioritized Requests.

As shown in Fig. 9, non-prioritized requests are less than prioritized requests in terms of processing time. Because in this case the requests can be executed at the same time.

3.3.2. Fitness of the Population

The fitness function is used to show the superiority of one population over another population. The fitness function is also known as the utility function. The fitness function can be used in single-objective or multi-objective modes. In single-objective mode, the algorithm seeks to optimize one objective, which can be minimum or maximum, but in multi-objective mode, the algorithm seeks to optimize several objectives. that these goals can be opposite to one another. In this research, the single target mode has been used. The fitness function used in this research is given below.

$$Fit = \alpha * T_P + \beta * T_M \qquad (۶)$$

In relation (6), Fit indicates the utility function, T_P indicates the time of processing requests by miners, T_M indicates the maximum time spent to complete requests by miners, and α β represents the weight coefficients of the objective function.

3.3.3. POPULATION IMPROVEMENT

In this research, the two operators shown in Fig. 10 have been used to improve the population.

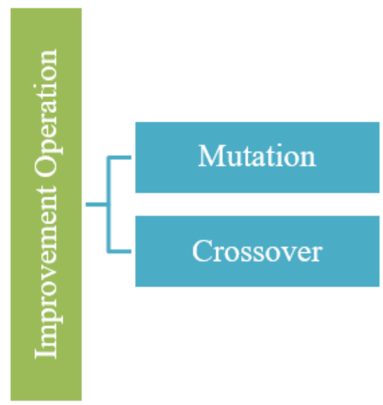


Fig., 10. Improvement operation.

In order to improve the population in the mutation algorithm, relation (7) is used.

$$\overrightarrow{U_i(t)} = x_{i_1}(t) + \beta\left(x_{i_2}(t) - x_{i_3}(t)\right) + \alpha(\text{fbest}(t) - x_i(t)) \qquad (۷)$$

In relation (7), $U_i(t)$represents the mutation vector, x_{i_1},x_{i_2}andx_{i_3} represent three selected populations that are randomly are selected The selected populations must be different from each other, in other words, the condition i≠i1≠i2≠i3 must be fulfilled. fbest represents the best population at the moment t andx_i represents the current population that is supposed to improve. After the mutation operation there is the combination operation. Current population and mutant population are used in combination operation. The relationship of the combination operation is as follows.

$$X_{ij}^{i} = \begin{cases} U_{ij}(t) & \text{if} \quad j \in J \\ X_{ij}(t) & \text{otherwise} \end{cases} \qquad (۸)$$

In relation (8), X_ij^i represents the improved population, U_ij represents the mutation vector and X_ij represents the current population. If the desired index is in the J vector, the desired value is selected from the mutation vector, otherwise from the current population. The indices in the vector J are chosen randomly.

3.3.4. Stop condition

In evolutionary algorithms, stopping conditions are used to complete the optimization steps. In this research, the repetition method is used to complete the optimization steps. In other words, in this case, the algorithm ends when it has been executed for a specified number of times.

4. RESULTS AND DISCUSSION

In this section, the specifications of the simulation environment are examined from the point of view of software and hardware. After examining the specifications of the simulation environment, explanations related to the problem scenarios are given. The problem scenarios are compared in the sections of block size, number of miners and number of requests. Requests are grouped into two prioritized or non-prioritized categories. Also, in this chapter, the statistical parameters for the evaluation of the proposed method are stated in order to measure the stability of the algorithm in the face of a large volume of requests. Finally, at the end of the chapter, a comparison between the proposed method and the methods carried out in this field is given.

4.1. SIMULATION ENVIRONMENT

In Fig. 11 the details of the simulation environment are examined from the hardware and software point of view.

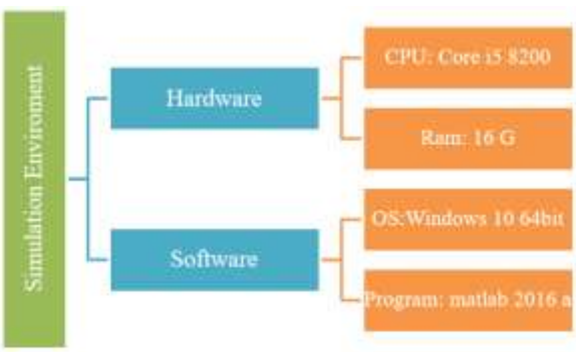


Fig. 11. Characteristics of the simulation environment.

4.2 Evaluation Scenarios

In this section, the evaluation scenarios of the proposed method are described. The scenarios used in this research are given in Fig. 12.

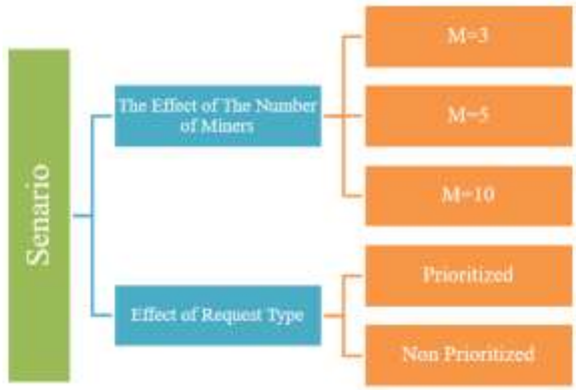


Fig. 12. Problem scenario.

As shown in Fig. 12, two scenarios of the number of miners and the type of request have been used to evaluate the proposed method.

4.3. Evaluation Criteria

In this research, statistical criteria are used to evaluate the proposed method. The criteria used in this research include maximum, minimum, average and standard deviation criteria. In order to calculate the statistical criteria, the proposed method has been repeated 30 times and based on the repetitions, the statistical criteria are calculated. Below are the explanations related to these criteria:

• **Maximum**: Maximum is the highest value obtained in 30 repetitions. This criterion is to show the difference of the obtained answers from the lowest value to the highest value.

• **Minimum:** This criterion indicates the lowest value obtained by the algorithm in 30 iterations. The lower this criterion is, it indicates the acceptable efficiency of the proposed method.

• **Average:** This criterion shows the average values obtained in 30 repetitions. The closer the value of this criterion is to the minimum, it indicates the stability of the algorithm in solving the optimization problem. Because in this case, the algorithm has been able to achieve the minimum solution in most cases.

• **Standard deviation:** This criterion is used to show the difference of the obtained values from the average. The closer this criterion is to zero, it indicates the acceptable efficiency of the proposed method.

4.4. Test Results

As mentioned in the previous section, the results of the proposed method are measured based on different scenarios. The method described in the article [22] was used to evaluate the proposed method. Below are the results of the proposed method.

4.4.1 The effect of the number of miners

In this section, the effect of miners on the processing time of requests in three modes of 3, 5 and 10 miners has been investigated. The number of requests in this experiment is equal to 3 requests, and each request can be divided into 10 separate blocks. Table 3 shows the results of this experiment.

Table 3. The results of the impact of the number of miners.

		Minimum	Maximum	Average	Standard Deviation
[22]	M=3	89	105	96.53	5.64
	M=5	71	82	76.16	3.49
	M=10	90	105	96.63	4.83
Our Method	M=3	87	100	92.46	4.04
	M=5	65	73	68.6	2.49
	M=10	88	93	90.43	1.77

As shown in Table 3, with the increase in the number of miners, the request processing time decreases (by increasing the number of miners from three to five miners), but in the case of increasing the number of miners from five miners to 10 miners, Transfer costs from miners to pools increase, therefore processing time has increased. Figure (13) shows the superiority of the proposed method over the reference method [22].

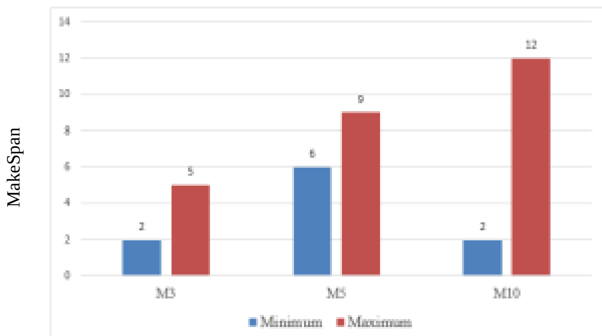
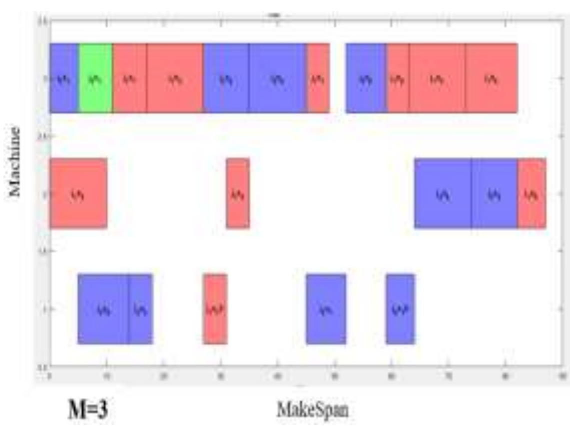


Fig. 13. The degree of superiority of the proposed method compared to the reference method [22].

The Gantt chart of the proposed method for different states of miners is shown in Fig. 14.



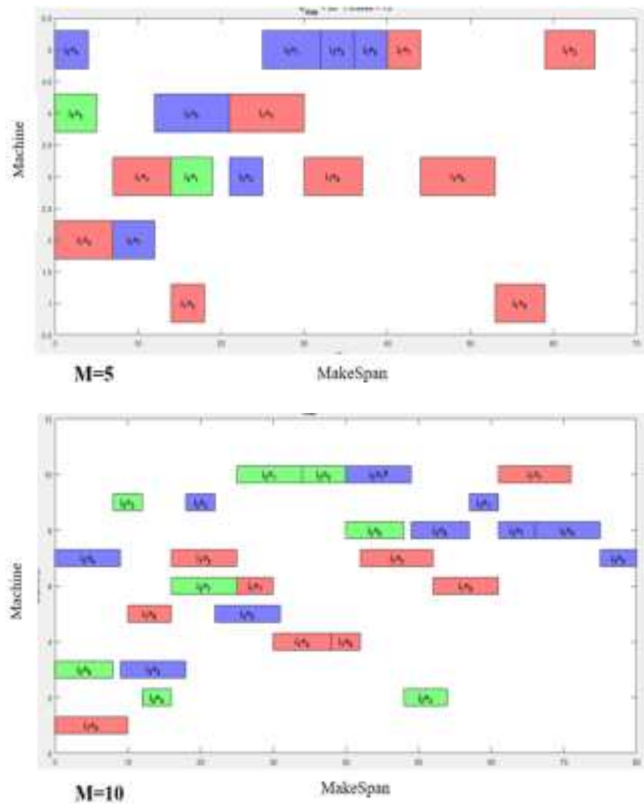


Fig 14. Gantt chart of the proposed method for different states of miners.

The convergence diagram of the algorithm in different states of miners is shown in Fig. 15.

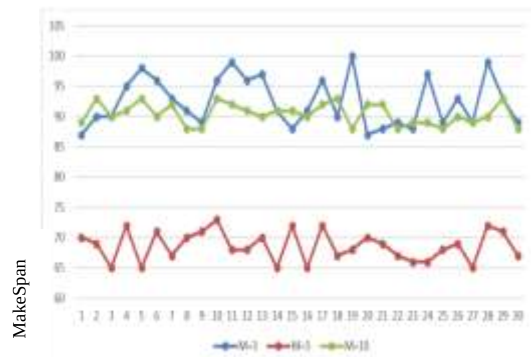


Fig. 15. Convergence diagram of the proposed method for different states of miners.

As shown in **Fig 15**, the best solution was obtained in the case where the number of miners was equal to 3.

4.4.2 The effect of the type of requests

In this test, the dependency of requests is checked. In this case, requests are considered dependent and non-dependent. The number of requests in this experiment is equal to 5 and the number of blocks of each request is equal to 7 at most. Also, the number of miners in this test is 3.

Table 4. The results of the impact of the type of requests

		Minimum	Maximum	Average	Standard Deviation
[22]	Non Dependent	75	85	80.63	3.32
	Dependent	87	102	95.46	4.70
Our Method	Non Dependent	71	78	74.23	2.45
	Dependent	79	91	84.1	4.01

As shown in Table 4, in the non-dependency mode, the request processing time has been reduced because in this mode the requests can be executed simultaneously with each other, in case in the dependency mode Requests have priority over each other and the next request cannot be executed until the desired request is executed.The superiority of the proposed method compared to the reference method [22] is shown in Fig. 16.

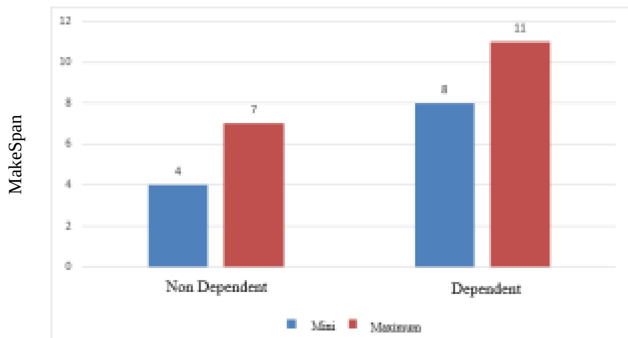
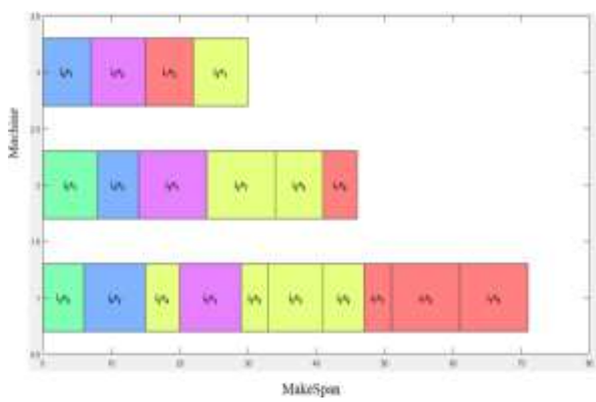
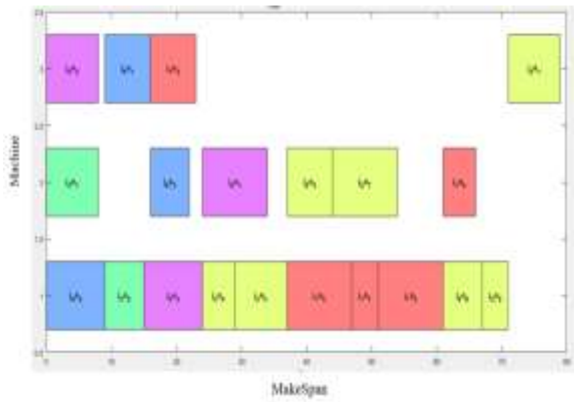


Fig. 16. The degree of superiority of the proposed method compared to the reference method [22].

The Gantt chart of the proposed method in non-dependency and with dependency mode is shown in Fig. 17.



Non-dependency



With dependence

Fig. 17. Gantt chart of the proposed method in the request type mode.

As shown in Fig. 17, in with dependency mode, requests are dependent on each other, but in non-dependency mode, requests can be executed simultaneously. The convergence diagram of the proposed method in two states of with dependency and non-dependency is shown in Fig. 18.

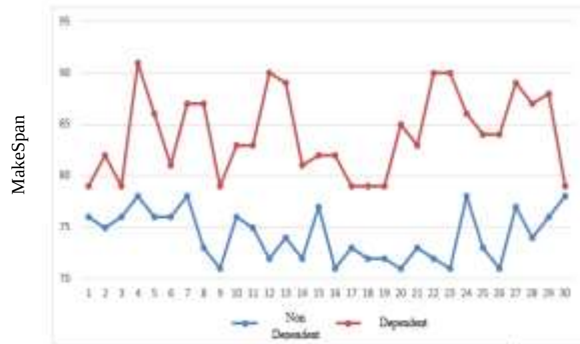


Fig. 18. Convergence diagram of the proposed method in the request type mode.

5. CONCLUSION

In this article, block size division method is used to assign requests to miners. To adjust the block size, a multi-stage process has been used, in the first stage, the requirements of the block chain network and in the second stage, the management of requests have been stated. In this article, differential evolution algorithm is used to allocate requests to miners. The scenarios used in this research include the number of miners and the type of requests. The criteria used in this research include minimum, maximum, average and standard deviation criteria. The implementation results have shown the superiority of the proposed method compared to the reference method [22].

REFERENCES

[1] Reddy BS, Sharma GV, **"Optimal transaction throughput in proof-of-work based blockchain**

networks". *In Multidisciplinary Digital Publishing Institute Proceedings*, 2019; 28 (1):6.

[2] Ghimire S. **"Analysis of bitcoin cryptocurrency and its mining techniques".** *PhD*, University of Nevada, Las Vegas,2019.

[3] Bach LM, Mihaljevic B, Zagar M. **"Comparative analysis of blockchain consensus algorithms"**. *In: 2018 41st International Convention on Information and Communication Technology*, Electronics and Microelectronics (MIPRO);2018. pp. 1545-1550.

[4] Yang D, Long C, Xu H, Peng S. **"A review on scalability of blockchain",** *In the 2nd International Conference on Blockchain Technology*; 2020. pp. 1-6.

[5] Khan KM, Arshad J, Khan MM**. "Investigating performance constraints for blockchain based secure e-voting system"**. *Future Generation Computer Systems* 2020; 105: 13-26.

[6] Jiang S, Wu J. **"Bitcoin mining with transaction fees: a game on the block size"**. *In: 2019 IEEE International Conference on Blockchain (Blockchain)*; 2020. pp. 107-115.

[7] Gemeliarana IG, Sari RF. **"Evaluation of proof of work (POW) blockchains security network on selfish mining"**. *In:2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*; 2019. pp.126-130.

[8] A. Litke, D. Anagnostopoulos, and T. Varvarigou, **''Blockchains for supply chain management: Architectural elements and challenges towards a global scale deployment'',** *Logistics*, vol. 3, no. 1, p. 5, Jan. 2020.

[9] J. Al-Jaroodi and N. Mohamed, **''Blockchain in industries: A survey'',** *IEEE Access*, vol. 7, pp. 36500–36515, 2020.

[10] F. Casino, T. K. Dasaklis, and C. Patsakis, **''A systematic literature review of blockchain-based applications: Current status, classification and open issues'',** *Telematics Inform.*, vol. 36, pp. 55–81, Mar. 2019.

[11] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, **''A survey on consensus mechanisms and mining strategy management in blockchain networks'',** *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[12] Monrat AA, Schelén O, Andersson K. **"A survey of blockchain from the perspectives of applications"**, *challenges, and opportunities*. Vol. 7, IEEE Access. 2021. p. 117134–51.

[13] A. Manimuthu, R. V. Sreedharan, R. G, and D. Marwaha, **''A literature review on bitcoin: Transformation of crypto currency into a global phenomenon'',** IEEE Eng. Manage. Rev., vol. 47, no. 1, pp. 28–35, 1st Quart. 2020.

[14] Y. Yuan and F.-Y. Wang, **''Blockchain and cryptocurrencies: Model, techniques, and applications'',** *IEEE Trans. Syst. Man, Cybern., Syst*., vol. 48, no. 9, pp. 1421–1428, Sep. 2019.

[15] H. Yu, Z. Yang, and R. O. Sinnott, **''Decentralized big data auditing for smart city environments leveraging blockchain technology'',** *IEEE Access*, vol. 7, pp. 6288–6296, 2020.

[16] R. A. Memon, J. P. Li, and J. Ahmed**, ''Simulation model for blockchain systems using queuing theory'',** *Electronics*, vol. 8, no. 2, p. 234, Feb. 2020.

[17] Cao B, Zhang Z, Feng D, Zhang S, Zhang L, Peng M, et al. **"Performance analysis and comparison of POW, PoS and DAG based blockchains".** *Digit Commun Networks [Internet]*. 2020 Nov; 6(4):480–5.

[18] Wu Y, Song P, Wang F. **"Hybrid Consensus Algorithm Optimization: A Mathematical Method Based on POS and PBFT and Its Application in Blockchain"**. Vol. 2020, *Mathematical Problems in Engineering*. 2020.

[19] Fu J, Qiao S, Huang Y, Si X, Li B, Yuan C. **"A Study on the Optimization of Blockchain Hashing Algorithm Based on PRCA"**. *Security Commun Networks [Internet]*. 2020 Sep 14; 2020:1–12.

[20] Khan KM, Arshad J, Khan MM. **"Investigating performance constraints for blockchain based secure e-voting system"**. *Futur Gener Comput Syst [Internet]*. 2020 Apr; 105:13–26.

[21] Singh, N., Vardhan, M,. **"Multi-objective Optimization of Block Size Based on CPU Power and Network Bandwidth for Blockchain Applications"**. In: Nath, V., Mandal, J.K. (eds) *Proceedings of the Fourth International Conference on Microelectronics, Computing and Communication Systems. Lecture Notes in Electrical Engineering*, vol 673. Springer, Singapore. (2021).

[22] Aygün B, Arslan H. **"Block size optimization for POW consensus algorithm based blockchain applications by using whale optimization algorithm"**. *Turkish J ElectrEngComputSci* [Internet]. 2022; 30:406–19.

[23] Francesc W, Sergio B-M and Paolo D**, "End-to-End Latency Analysis and Optimal Block Size of Proof-of-Work Blockchain Applications",** arXiv:2202.01497v1 [cs.NI] 3 Feb 2022.