Neural Computing and Applications
https://doi.org/10.1007/s00521-023-08695-7

ORIGINAL ARTICLE

An improved Henry gas optimization algorithm for joint mining decision and resource allocation in a MEC-enabled blockchain networks

Reda M. Hussien[1] · Amr A. Abohany[1] · Nour Moustafa[2] · Karam M. Sallam[3]

Received: 28 October 2022 / Accepted: 16 May 2023

Abstract
This paper investigates a wireless blockchain network with mobile edge computing in which Internet of Things (IoT) devices can behave as blockchain users (BUs). This blockchain network's ultimate goal is to increase the overall profits of all BUs. Because not all BUs join in the mining process, using traditional swarm and evolution algorithms to solve this problem results in a high level of redundancy in the search space. To solve this problem, a modified chaotic Henry single gas solubility optimization algorithm, called CHSGSO, has been proposed. In CHSGSO, the allocation of resources to BUs who decide to engage in mining as an individual is encoded. This results in a different size for each individual in the entire population, which leads to the elimination of unnecessary search space regions. Because the individual size equals the number of participating BUs, we devise an adaptive strategy to fine-tune each individual size. In addition, a chaotic map was incorporated into the original Henry gas solubility optimization to improve resource allocation and accelerate the convergence rate. Extensive experiments on a set of instances were carried out to validate the superiority of the proposed CHSGSO. Its efficiency is demonstrated by comparing it to four well-known meta-heuristic algorithms.

Keywords Blockchain · Classification · Mobile edge computing · Meta-heuristics · Henry gas algorithm · Resource allocation

1 Introduction

Blockchain networks (BNs) have been receiving a lot of attention lately due to the increasing popularity of cryptocurrencies like Bitcoin [11]. BNs are decentralized peer-to-peer networks that provide anonymity, auditability, and secure operations without requiring trust from a third party [77]. Because of these capabilities, BNs have been employed in various areas, such as smart manufacturing [39], smart grid [69], the Internet of Things (IoT) [79], and supply chain [8]. To implement BNs, three major technologies are used: (1) a peer-to-peer network; (2) public key cryptography with hash functions; and (3) a program (Blockchain protocol, such as consensus). To address the synchronization problem in traditional distributed databases, blockchain technologies use a distributed consensus algorithm and combine peer-to-peer networks with cryptography, mathematics, algorithms, and economic models to create an integrated infrastructure across multiple fields.

✉ Amr A. Abohany
amrabohany8@gmail.com

Reda M. Hussien
reda_mabrouk@fci.kfs.edu.eg

Nour Moustafa
nour.moustafa@unsw.edu.au

Karam M. Sallam
karam.sallam@canberra.edu.au

[1] Faculty of Computers and Information, Kafrelsheikh University, Kafrelsheikh 33511, Egypt

[2] University of New South Wales (UNSW Canberra) Canberra, Canberra, ACT 2612, Australia

[3] School of IT and Systems, Faculty of Science and Technology, University of Canberra, Bruce, ACT 2601, Australia

Published online: 07 June 2023

"In essence, a blockchain is a digital public ledger that records all digital transactions in a data structure known as 'Completed Transaction Blocks' or in chronological order, which is then stored across a distributed network. The block body consists of a transaction counter and transactions. The transaction counter records the number of transactions, while the list of transactions recorded by the block is simply referred to as the transactions. The maximum number of transactions that can be stored in a block is determined by both the block size and the size of each transaction. During the process of verifying and validating transactions, the system checks whether the initiator has sufficient balance to complete the transaction, or it may be fooled by a double-spending mechanism [50], where the same input amount is used in two or more different transactions [32]. Blockchain users, also known as 'BUs', are peers whose computational power is used to mine for blocks [37]. Once these BUs verify and validate a transaction, it is included in a block."

In order to publish a block, BUs must expend a significant amount of their computing resources to solve a computational puzzle, known as the mining process. The BUs who are able to solve the puzzle first become the winning BUs and are given a small incentive to create a new block. Once a block has been created, a consensus mechanism is used by all peers in the network to verify the new block [72]. The mining process involves a complex anti-collision hash function query, which typically requires significant resources and high computation capacities of BUs. However, since resources are often limited, this can lead to issues with resource allocation for BUs and create challenges in managing the growth of the network.

Mobile edge computing (MEC) is a promising field that aims to enhance the computing capabilities of IoT devices by offloading processes to a MEC server [55, 68]. Numerous studies have been conducted on MEC-operated wireless blockchain networks (WBNs). For instance, [44] established a MEC-operated WBN and introduced an alternative control methodology of multipliers for resource management. However, if the earnings of the MEC service provider (MSP) are insufficient, they may choose not to provide computing services to BUs, which could ultimately lead to the WBNs being unable to function.

In a related study, [47] proposed a framework for managing resources in MEC-operated WBNs using deep learning. Additionally, [31] introduced two bidding frameworks for resource management in MEC-operated WBNs: the multi-demand framework and the constant-demand framework. In the former, an auction mechanism is designed to achieve optimal social welfare, while in the latter, an estimation method is introduced to simultaneously consider computational efficiency, individual rationality, and truthfulness.

Furthermore, [73] explored the interaction between cloud/fog providers and BUs in a proof of work-based BN using a game-theoretic approach. However, it should be noted that these studies [31, 47, 73] do not take into account the transfer delay time between the participating BUs and MEC server. If there are many BUs offloading processes onto the MEC server simultaneously, they may experience significant overlap, resulting in high transportation delays [29]. Therefore, future research should consider the impact of transfer delay time on resource management in MEC-operated WBNs to further optimize the allocation of computing resources and improve network performance.

Recently, multiple studies of resource management in BNs have been proposed. Researchers mainly consider two basic problems: mining decisions and resource allocation. The first is to determine whether a BU contributes to the mining process or not, and the last is to decide the number of resources assigned to each subscribed BU.

Rizun [54] conducted a study on mining decisions, taking into account the block space supply curve and the mempool demand curve to demonstrate how a subscribing BU selects transactions to maximize their profit without a block size limit. By considering both the orphaning risk and revenue from transaction fees, the study found that the block size corresponding to the intersection point of these two curves is the optimal size for maximizing mining profit.

Kiayias et al. [34] introduced a mining decision stochastic game in the Bitcoin network to account for the randomness of the mining process, where multiple BUs participate. BUs typically choose to engage in mining to obtain stable profits.

Houy [28] presented and analyzed two Bitcoin mining games for BUs. When deciding how many transactions to include in the block they are mining, BUs need to consider the trade-off between including more transactions to earn higher transaction fees, and including fewer transactions to reduce the time needed to propagate their block solution and increase the likelihood of their block being included in the blockchain first.

Some BUs may engage in harmful actions within the mining pool, leading to a waste of distributed computing resources and posing a risk to the effectiveness of BNs. To address this issue, a game-theoretic approach has been proposed in [66] to incentivize BUs to mine honestly. While these methods have been shown to be effective, they have only been applied to wired BNs. With the emergence of IoT devices, there has been increased attention on Wireless BNs (WBNs) operating on these devices [7]. However, IoT devices are unable to sustain the mining process on local machines.

Liu et al. [45] conducted an analysis on the dynamic selection of mining pools in BNs. The selection of mining pools was represented as an evolutionary game, and evolutionary stability was produced through theoretical analysis. Furthermore, Shijing Yuan et al. [76] proposed an optimization method for a blockchain-supported edge video streaming system. The method aims to determine the offloading mode and resource allocation to achieve an optimal balance between accuracy and energy consumption.

In this context, many researchers have utilized optimization algorithms in their studies, these methods have numerous advantages [56], including (1) self-regulation; (2) flexibility to dynamic changes; (3) the ability to estimate multiple solutions simultaneously; and (4) not requiring bounded mathematical characteristics to be implemented. The family of optimization techniques includes, but is not limited to, differential evolution (DE) [6, 57–60], genetic methods [15, 42, 53], ant colony optimization algorithms [36, 52, 74], PSO [17, 26], gray wolf algorithm [46, 70], firefly optimizer [10, 81], flower pollination optimizer [1], whale optimization optimizer [2], artificial bee colony [78], binary slime mold optimizer [3], binary pigeon-inspired optimizer [12], cuckoo search optimizer [24], moth search optimizer [23], gain sharing knowledge-based optimization technique [5], diversified sine–cosine optimizer based on DE [25], light spectrum algorithm [4], binary light spectrum algorithm [4], and Aquila algorithm [9].

This paper differs from previous works in that it takes into account both MSP earnings and transportation postponement in a MEC-operated WBN. To increase the overall earnings of all BUs, the paper optimizes the BUs' mining decisions and estimates their resource allocation simultaneously using a modified version of the Henry gas solubility optimization (HGSO) called the Chaotic Henry single gas solubility optimization (CHSGSO) approach. When HGSO is utilized to address this issue, each individual refers to the resource allocations and mining decisions of the joining BUs. However, since not all BUs participate in mining, considering all BUs as individuals would lead to an overloaded search space and poor performance. To address this problem, the paper develops a new approach called CHSGSO, which generates a population with variable-length individuals that represent participating BUs.

The main contributions of this paper can be listed in the following points:

- We propose a modified HGSO, in which the resource allocation to only joining BUs is encoded as an individual. An adaptive strategy is designed to tune each individual size.
- A chaotic map has been integrated into the original HGSO to enhance the convergence rate.
- Comprehensive experiments are applied on a set of different instances to validate the superiority of CHSGSO.
- CHSGSO efficacy is then affirmed by doing a fair comparing with four well-known meta-heuristic methods.

The remainder of the paper is structured as follows. Section 2 introduces the problem formulation and the system model. Section 3 presents the proposed CHSGSO. The empirical results are investigated in Sect. 4. Finally, conclusions and some potential future works are presented in Sect. 5.

2 The problem formulation and system model

The MEC-enabled WBN is shown in Fig. 1, which has a group of n IoTDs, is considered as mining BUs that are involved in mining, where $N = \{1, 2, \ldots, n\}$. If a BU plans to join mining, it needs to pay for computing resources from the MEC Service Provider and then transfer its tasks to the MEC server for the mining process. For simplicity, the mining task is a 2-tuple: $\{B_i, C_i\}$, where B_i and C_i are the block size and the computation workload/intensity in CPU per bit, respectively.

In the considered BN, $d = \{d_1, \ldots, d_n\}$ represents the BUs mining decision, where $d_i = \{0, 1\}, (i \in N)$ means that the i^{th} BU chooses to join or not to join in mining. So,

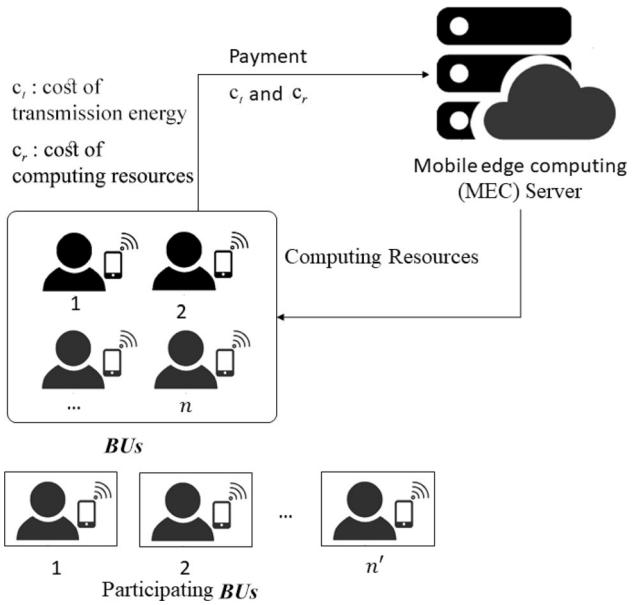

Fig. 1 A MEC-enabled WBN

the involved BUs number is $n' = \sum_{i \in n} d_i$. Additionally, the required resources to the joint BUs need to be allocated, i.e., computation resources $f = \{f_1, \ldots, f'_n\}$ and transmitting power (CPU Cycles/s) $p = \{p_1, \ldots, p'_n\}$ for all participating BUs.

For the considered BN, shown in Fig. 1, the mining task is successfully executed only after the completion of the following three phases.

Offloading phase In this phase, joint BUs concurrently transfer tasks to the MEC server with a transmission rate that is stated as:

$$R_i = b \log_2 \left(1 + \frac{p_i H_i}{\sigma^2 + \sum_{j \in n'/i} d_j p_j H_j} \right), \tag{1}$$

where H_i represents the ith participating BU's channel state information, and the term $\sum_{j \in n'/i} d_j p_j H_j$ represents the received interference from other BUs, b represents the channel bandwidth, and $sigma^2$ represents the power of background noise. For each BU, the task transmission time T_i^t and energy consumption E_i^t of ith BU can be formulated as:

$$T_i^t = \frac{B_i}{R_i}, \quad \forall i \in n', \tag{2}$$

and

$$E_i^t = p_i T_i^t, \quad \forall i \in n', \tag{3}$$

where R is the transmission rate, B is the size of the block, T is the task transmission and E is the transmitting energy consumption.

Mining phase

During this phase of the process, the MEC server is responsible for carrying out the mining tasks that have been transmitted by the participants. The energy consumption and time taken by the MEC server to execute the ith BU task are consequentially stated as:

$$T_i^m = \frac{B_i C_i}{f_i}, \quad \forall i \in n', \tag{4}$$

and

$$E_i^m = k_1 f_i^3 T_i^m, \quad \forall i \in n', \tag{5}$$

where k_1 is the efficient capacitance coefficient.

Propagation phase After completing the mining phase, if the BU executes its mining task faster than expected, the BU earns a reward. The probability of getting a reward, as a function of mining time, is given as:

$$P_i^m = \frac{k_2}{T_i^m}, \quad \forall i \in n', \tag{6}$$

where k_2 is scaling factor.

On the other hand, if the BU executes its mining task slowly, the BU will not get a reward. Because agreement may not be obtained, the block may be rejected. In blockchain networks, blocks are made using a Poisson process, with propagation time T_i^0 and with a constant mean rate (λ), is linearly proportional to their size B_i. The ith BU's orphaning probability is represented by:

$$P_i^0 = 1 - e^{-\lambda(\xi B_i + T_i^s)} = 1 - e^{-\lambda(T_i^0 + T_i^s)}, \quad \forall i \in n', \tag{7}$$

where ξ represents a delay factor and T_i^s represents starting time of mining. In this study, the mining task of the ith joining BU will be executed when it is received by the MEC server; hence $T_i^s = T_i^t$.

2.1 Profit model

As previously mentioned, if the mining task is executed successfully and fast enough, a reward can be obtained by BU. BUs are rewarded with a fixed reward of ω and a variable reward of ρB_i, where ρ is the variable reward factor. In addition, BUs consume definite computing and communication costs. Therefore, the ith BU profit is determined by:

$$F_i^{\mathrm{BU}} = (w + \rho B_i) P_i^r (1 - P_i^o) - \tau_1 E_i^t - \tau_2 f_i, \quad \forall i \in n', \tag{8}$$

where τ_1 and τ_2 represent the unit costs of the computation resources and transmission energy, respectively. The total profit of all BUs is determined by:

$$F^{\mathrm{BU}} = \sum_{i \in n'} F_i^{\mathrm{BU}}, \tag{9}$$

Additionally, while the MEC service provider earns a profit by selling computation resources to the BUs, it should pay for the costs of both the no-load E_o and mining energy consumption. So, the MSP profit is determined by:

$$F^{\mathrm{MSP}} = \sum_{i \in n'} (\tau_2 f_i - \tau_3 E_i^m) - \tau_3 E_0, \tag{10}$$

where τ_3 is the unit cost of consumed energy.

2.2 Problem formulation

For the investigated BN, the mining decision (d), transmission power (p), and computation resources (f) are all optimized at the same time to maximize the overall profit of all BUs. The profit model is expressed as a maximization problem as follows:

$$\begin{aligned} \max_{m,p,f} \quad & F^{\mathrm{BU}} = \textstyle\sum_{i\in n'} F_i^{\mathrm{BU}}, \\ \text{s.t.} \quad & C1: d_i \in \{0,1\}, \quad \forall i \in n', \\ & C2: f^{\min} \le f_i \le f^{\max}, \quad \forall i \in n', \\ & C3: p^{\min} \le p_i \le p^{\max}, \quad \forall i \in n', \\ & C4: \textstyle\sum_{i\in n'} f_i \le f^{\mathrm{total}}, \\ & C5: T_i^t + T_i^m + T_i^o \le T_i^{\max}, \quad \forall i \in n', \\ & C6: F^{\mathrm{MSP}} \ge 0. \end{aligned} \tag{11}$$

where $C1$ denotes that each BU can choose to join mining or not; $C2$ specifies the maximum and minimum computation resources assigned to each joint BU; $C3$ ensures that the transmission power assigned to each BU fall within the maximum and minimum allowable values; $C4$ represents the overall computation resources assigned for each BUs, involved in mining, cannot overtake the overall computation resources of the MEC server; $C5$ guarantees that the overall time of propagation, mining, and offloading can not overtake the limitation of the maximum time; and $C6$ guarantees that the MSP profit will be greater than zero.

In the studied BN, We are under the assumption that all BUs are identical and have the same transmission power and computation resource ranges.

3 Proposed improved Henry gas solubility optimization

The HGSO algorithm [27] is based on a physical property known as Henry's law [63], which governs the solubility of materials. Pressure and temperature are two factors that play a significant role in this rule. The solubility of gases reduces with increasing temperature. While the gas becomes more soluble at higher pressures.

3.1 Henry's law

Henry's law is a gas law that was formulated by William Henry in 1803 [40]. According to Henry's law, "When the temperature is constant, the amount of gas dissolved in a given type of liquid is directly proportional to its partial pressure above the liquid". Therefore, Henry's law is highly dependent on temperature. Staudinger and Roberts [63] proposed that the gas solubility (S_g) is directly proportional to the gas partial pressure (P_g), as described by the following equation:

$$S_g = H \times P_g, \tag{12}$$

where H represents Henry's constant, which is specific to a particular gaseous solvent composition at a given temperature, and the partial pressure of the gas is represented by P_g.

In addition, the effect of temperature dependence on the constants of Henry's law must be taken into account. The constants of Henry's law change with changes in the system temperature, which can be represented by the following equation of Van't Hoff as shown:

$$\frac{d\ln H}{d(1/T)} = \frac{-\nabla_{\mathrm{sol}} E}{R}, \tag{13}$$

where $\nabla_{\mathrm{sol}}E$ represents the dissolution enthalpy, R represents the constant of the gas and A and B are two factors for T dependence of H. So, Eq. 12 can be integrated as the following equation:

$$H(T) = \exp(B/T) \times A, \tag{14}$$

where H represents a function of parameters A and B. Alternatively, an expression can be created using H at the reference temperature $T = 298.15$ K.

$$H(T) = H^{\theta} \times \exp\left(\frac{-\nabla_{\mathrm{sol}} E}{R}\left(1/T - 1/T^{\theta}\right)\right), \tag{15}$$

The Van't Hoff equation is valid when $\nabla_{\mathrm{sol}}E$ is a constant, so, Eq. 15 can be reconstructed as the following equation:

$$H(T) = \exp\left(-C \times \left(1/T - 1/T^{\theta}\right)\right) \times H^{\theta}, \tag{16}$$

3.2 Inspiration source

Henry's law was first presented by J.W. Henry in 1800. Generally, the maximum amount of solute that can dissolve in a given amount of solvent at a given pressure or temperature is called solubility [48]. So, HGSO was inspired by Henry's law behavior. According to the above Eqs. 12 through 16, Henry's law can be utilized to estimate the solubility of low-solubility gases in liquids. In addition, pressure and temperature are the two parameters that affect solubility; at higher temperatures, gases are less soluble, but solids become more soluble. As for pressure, with increasing pressure, the solubility of gases increases [14].

3.3 Mathematical model of Henry gas solubility optimization

The mathematical procedures of the HGSO algorithm are described in this subsection as follows [27]:

Step 1: Equation 17 is used to create initial population of candidate solutions with N gases:

$$x_i^{(0)} = lb_i + r \times (ub_i - lb_i), \tag{17}$$

where $x_i^{(0)}$ is the initial position of the ith gas, and lb_i and ub_i are the position's lower and upper limits, respectively, for the ith candidate solution. r is a randomly generated real value in the range [0, 1].

Step 2: Candidates from the population are organized into groups that are referred to as clusters. Each cluster has

an equal number of candidates that have the same attributes. The equation referenced in Eq. 18 is used to initialize these properties:

$$H_j^{(0)} = l_1 \times rand_1,\ P_{i,j}^0 = l_2 \times rand_2,\ C_j^0 = l_3 \times rand_3, \tag{18}$$

where H_j^0 denotes the initial value of Henry's coefficient for jth cluster, $P_{i,j}^{(0)}$ represents the ith gas initial partial pressure in jth cluster, and $C_j^{(0)}$ represents the initial constant value of cluster j. $l1$, $l2$, and $l3$ are fixed values of 5×10^{-02}, 100, and 10^{-02}, respectively.

Step 3: The fitness value of each cluster's gas particles is computed, and the best $x_{j,best}$ cluster is assigned. In this stage, all candidate solutions are sorted according to fitness to get the global best solution x_{best}.

Step 4: As the applied partial pressure on gas particles changes during each iteration, Henry's coefficient $H_j^{(t+1)}$ is updated according to Eq. 19:

$$\begin{aligned} H_j^{(t+1)} &= H_j^{(t)} \times e^{-C_j \times (1/T^{(t)} - 1/T^{\theta})}, \\ T^{(t)} &= e^{(-t/t_{\max})}, \end{aligned} \tag{19}$$

where $H_j^{(t)}$ is Henry's constant for cluster j in iteration t, T^{θ} is a fixed parameter with value 298.15, $T^{(t)}$ represents the temperature at iteration t, and $t_{\max}$ is maximum iterations.

Step 5: During the tth iteration, Eq. 20 is used to change the solubility $S_{i,j}^{(t)}$ of the ith gas particle in the jth cluster:

$$S_{i,j}^{(t)} = K \times H_j^{(t+1)} \times P_{i,j}^{(t)}, \tag{20}$$

where $P_{i,j}^{(t)}$ represents the applied pressure on ith gas particle in jth cluster, and K is a fixed value.

Step 6: in this step, the ith gas particle position of the jth cluster is updated using Eq. 21 for iteration $t = t + 1$.

$$\begin{aligned} x_{i,j}^{(t+1)} &= x_{i,j}^{(t)} + F \times r_1 \times \gamma \times (x_{i,best} - x_{i,j}^{(t)}) \\ &\quad + F \times r_2 \times \alpha \times (S_{i,j}^{(t)} \times x_best - x_{i,j}^{(t)}), \\ \gamma &= \beta \times exp(\frac{-F_{best}^t + \epsilon}{F_{i,j}^t + \epsilon}), \epsilon = 0.05, \end{aligned} \tag{21}$$

where F is used for controlling search direction, γ is the interaction ability of gas within its cluster and α is effect of other gas on ith particle. r_1 and r_2 are randomly generated values between [0,1], and $\epsilon = 0.05$.

Step 7: since HGSO is heuristic algorithms, it may be optimized locally. Therefore, Eq. 22 is used to rank and number of worst solutions N_w for re-initialization:

$$N_w = N \times rand \times (c_2 - c_1) + c_1, \tag{22}$$

where N is the total number of individuals in the population and $rand$ is a random number between 0 and 1. c_1 and c_2 are constants that specify the percentage of worst solutions. Equation 17 is used to reinitialize the positions of the worst solutions selected in this process.

Algorithm 1 outlines the pseudo-code of the HGSO algorithm's step-by-step structure.

Algorithm 1 HGSO algorithm

Initialization: Number of gas particles N and type i, $H_j^{(0)}$, $P_{i,j}^{(0)}$, $C_j^{(0)}$, l_1, l_2 and l_3, using Eq. 18.
Initialize gas particles with random positions, $x_i^{(0)}, i \in [1 = 1, 2, ...N]$, using Eq. 17.
Divide gas particles into a number of gas types (cluster) with the Henry's constant value H_j.
Evaluate each cluster j.
Find the best gas particle $x_{i,best}^0$ in each cluster, and the best x_{best} in whole swarm.
while $t \leq t_{max}$ **do**
    for each gas particle **do**
        Update the positions of all gases Eq.21.
    end for
    Update Henry's coefficient of each gas type Eq. 19.
    Update solubility of each gas particle Eq. 20.
    Find the number for worst gas particles Eq. 22.
    update the best gas $x_{j,best}^{(t)}$, and best gas $x_{best}^{(t)}$.
    Update iteration counter $t = t + 1$.
end while
return x_{best}

3.4 Chaotic improved Henry gas solubility optimization

3.4.1 Chaotic systems

Dynamic systems are mathematical functions that describe the movement of a point in geometrical space over time. A dynamic system has a state for a given time and can be represented using a vector mathematical function with an appropriate state-space model. The evolution rule allows us to determine the next state of a dynamical system using the current state and its behavior. Most dynamic systems are deterministic, but some systems generate stochastic random events or have an incomplete description. A dynamic system can be completely modeled for predicting its future behavior by an analytical solution that is time-dependent. Dynamic systems can be further classified into two types: linear dynamic systems and nonlinear dynamic systems. A nonlinear dynamic system is a system whose output is not proportional to the changes made in the input. Linear dynamic systems are dynamic systems whose evaluation is a linear function, i.e., changes in the output are linearly proportional to the changes in the input. Chaotic systems are a type of nonlinear dynamic system. Chaotic maps are a field of study in mathematics where dynamic systems produce a random state that appears irregular but is governed by the initial seed conditions.

To analyze the chaotic behavior in dynamic systems, range bifurcation diagrams are often plotted. These diagrams illustrate the relationship between chaotic states and their corresponding control parameters. The Lyapunov exponent is a parameter that plays a crucial role in determining whether a chaotic map is useful in pseudo-random generators. This exponent is highly sensitive to slight changes in the seed parameters, such as initial conditions and control parameters. The idea of using chaotic systems instead of random processes has been noticed in several areas including computer science, economics, engineering, etc. [19, 33, 38, 51, 62, 64] One of these areas is the optimization theory. In random-based optimization algorithms, the role of randomness can be initialized using chaotic dynamics instead of random processes. Chaotic maps can be classified into two categories: 1D [30, 75, 80] and multi-dimension [13, 16, 21, 22, 67].

1D chaotic schemes have modest structure, and simple dynamic characteristics, and are easy to implement. To generate the Pseudo-random Sequence (PRS), only one variable and a few parameters are used. On the other hand, the 2D chaotic maps possess two variables and a greater number of control parameters. In the present study, the authors have used a 1D Sin map. The structure of the Sin map is defined as [23]:

$$x_{i+1} = \frac{\mu}{4}\sin(\pi x_i), \tag{23}$$

where μ is the control parameter with a range of $u \in [0, 4]$, As shown Fig. 3, one can see that only when the parameter $\mu \geq 3.57$ can chaotic behavior occur in the Sine map. The bifurcation diagram Fig. 2 depicts the possible state values of the system under each parameter. Corresponding to a value of the system parameter, if there are infinite state values, the system with the parameter has chaotic behavior.

As remarked in [35, 49, 61, 65], replacing a random variable with a chaotic sequence enhances the optimization algorithm's global convergence speed and exploration/exploitation.

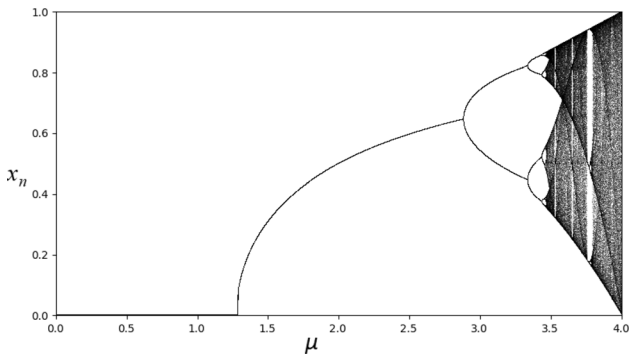

Fig. 2 The bifurcation diagram for the sin map

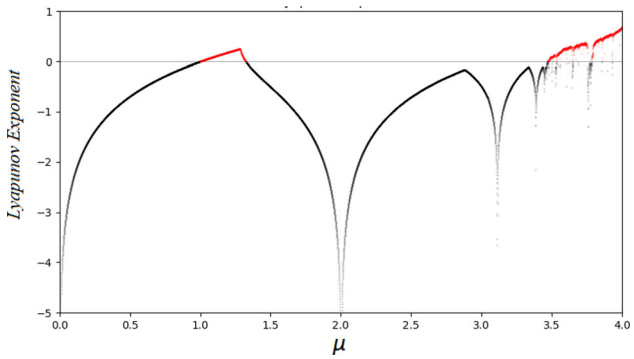

Fig. 3 The Lyapunov exponent for the sin map

In the HSGO, the standard procedure Algorithm 1, used for updating solutions, has at least two clusters and two gases in each cluster, and all gases have the same search space dimensions. As shown in Eq. 21, the balance between the exploration and exploitation phases is controlled by fine-tuning control parameters [27]:

1. the solubility of gas j in cluster i $S_{i,j}$, which is based on the time of iteration;
2. the ability γ of gas j in cluster i to interact with the gases in its cluster, which aims to transfers the search individuals from global to local phase and vice versa; and
3. the flag F that changes the direction of the search agent and provides $\pm ve$ diversity.

It can be observed that the system described by (11) is considered as a nonlinear optimization problem with mixed variables since p and f are continuous variables while d is binary. Consequently, this problem is hard to be solved by original HGSO algorithm (11). In this paper, the authors proposed a modified HGSO algorithm to handle the system (11). The steps of the proposed algorithm are described in the following subsections.

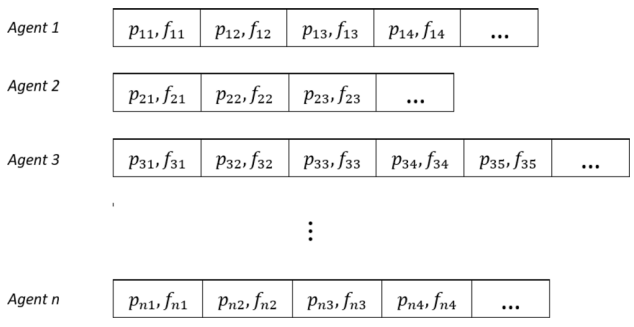

Fig. 4 Representation of proposed encoding schema and population of agents

3.4.2 Individual encoding

It should be noted that if a BU does not choose to join the mining process (i.e., $d_i = 0$), this BU does not require sending its task to the MEC server or acquiring computing resources for mining. In this situation, allocating resources for these BUs is not needed. Thus, the generally utilized encoding scheme in the original HGSO creates redundancy in the search space, which can degrade the performance. A new encoding schema is proposed to solve this problem.

In the proposed individual encoding 4, each individual contains continuous variables representing the resource allocations for each BU that joins mining. The length of each individual, which represents participating BUs only, is variable in size. Thus, the optimal solution is found in the 2D search space.

3.4.3 Agent structure

The proposed HGSO algorithms consist of different solutions called Agents. Each agent contains a single individual (i.e., single gas). Agents have different lengths depending on how many BUs are chosen at random to take part. The structure of the population of agents is presented in Fig. 4.

3.4.4 Initialization

In the initialization, the participating BUs are randomly chosen for each agent with a different number of BUs. Then, the resource allocation initial values of all BUs are randomly generated. The Fitness function value for each BU, total fitness, and degree of constraint violation are evaluated as presented in Algorithm 2. Then, the best agent is returned for further comparison. The process of the initialization phase is described in Algorithm 2.

Algorithm 2 Agents Initialization

```
for each gas agent do
    Randomly select participating BUs.
    Initialize agent with random positions (p,f)
    for each agent BU do
        Calculate Fitness using [8]
    end for
    Calculate Total Fitness using [9]
    Calculate degree of constraint violation based on 11
end for
Return the best agent
```

3.4.5 Update *p* and *f*

While searching for the best agent, the position (i.e., f_i and p_i) in iteration $t + 1$ of the ith agent is updated using Eq. 24 as follows:

$$x_{i,j}^{(t+1)} = F \times x_{i,j}^{(t)} \times (b1 + b2 \times \sin(\pi \times x_{i,j}^{(t)})) + F \times r_2 \times \alpha \times (S_{i,j}^{(t)} \times x_{i,j} - 1), \tag{24}$$

where the term $(b1 + b2 \times \sin(\pi \times x_{i,j}^{(t)}))$ provides the balance between exploration and exploitation by generating new positions due to its chaotic nature. $b1$ and $b2$ control the randomly generated position range.

3.4.6 Update mining decision *d*

For the mining decision optimization, we need to select and update BUs to be involved in the mining process. The following two steps are performed to optimize the mining decision:

1-Generate new gas agents In this step, all agents are sorted according to fitness, and the best agent is selected. Then, using the predetermined selection probability Ps_r, a number of the worst agents are chosen to be replaced by new agents with the same number of BUs as the best agent. The details of this process are described in Algorithm 3.

Algorithm 3 Agents Replacement

```
Sort agents according to evaluation metric.
Retrieve the number of participating BU of the best agent. n_best
Randomly select the number of worst agents with selection probability P_r^s
Replace selected worst agents with newly generated agents with n_best participating BU
```

2-Apply mutation To make the agents more diverse and give them a way to get out of a local optimum, a small number of BUs need to be replaced using a mutation operation with a probability of Pm_r that has already been set. The steps for applying mutation are listed in Algorithm 4.

Algorithm 4 Agent Mutation

```
for each agent do
    Sort participating BUs according to Fitness evaluation.
    Replace the worst BU with non-participating BU using mutation probability P_r^m
end for
```

4 Experiments and discussion

In this section, the results from different computational experiments with the proposed CHSGSO algorithm are compared with those obtained from other competing meta-heuristic algorithms in the literature. This section describes the parameter settings and performance measures adopted to validate the superiority of the proposed algorithm.

4.1 Algorithms for comparison

The performance of the proposed GHSGSO is compared with the following algorithms.

- ACO_{MV} [41]: ACO_{MV} is a continuous optimization algorithm that has a continuous relaxation and a categorical optimization approach. Together, these approaches enable ACO_{MV} to address resource allocation and mining decisions.
- DE [18]: The DE algorithm is one of the most robust evolutionary algorithm versions because of its fast convergence, simplicity, ease of use, and the same values of its parameters (population size, crossover rate, and scaling factor) can be used to tackle different optimization problems. DE was originally introduced to address continuous optimization problems. Modifications are needed to address the problems of both resource allocation and mining decisions.

  Integer constraints in mining decisions are handled by changing a continuous value to its nearest integer.
- DEMiDRA [71]: Every member shows the resource allocation of a joint BU and the resource allocation of all joint BUs generates the entire population. Then, the DE algorithm is used for resource allocation optimization. To optimize the mining decision, they have to choose BUs to join in mining and update the joining BU's number. Since the joining BU's number equals the population size, they modified the update of the joint BU's number into the organization of the population size and generated an adaptive approach. Additionally, a tabu approach is presented to avoid unfavorable BUs joining mining.
- BOToP [43]: Firstly, BOToP obtains the optimum solution of the mixed-variable optimization problem by solving a constrained modified bi-objective optimization problem. Second, DE is utilized to solve the original optimization problem with mixed variables to find the best solution.

4.2 Environment and parameter settings

In this paper, the proposed CHSGSO method was compared to four promising meta-heuristic methods. These methods are ACO_{MV}, DE, DEMiDRA, and BOToP. For every method, 30 independent runs were executed. Then, the mean values of profit were recorded over the 30 runs.

Table 1 Parameters setup for all methods

Algorithm	Parameter
All methods	Number of runs for each method = 30
	Maximum number of fitness evaluations FEs = 10000
ACO_{MV}	Population size = 60
	The influence of the best quality solution = 0.05099
	The search width = 0.6795
DE	Population size = 100
	Scaling factor = 0.9
	Crossover control parameter= 0.5
DEMiDRA	Population size =the number of joint BUs
	Scaling factor = 0.9
	Crossover control parameter = 0.5
BOToP	Population size = 30
	Scaling factor is selected from parameter pool: 0.6, 0.8, 1.0
	Crossover control parameter is selected from parameter pool: 0.1, 0.2, 1.0
CHSGSO	Population size =the number of joint BUs
	Agent selection probability = 0.015 to 0.23
	Mutation probability = 0.018 to 0.033
	$\alpha = \beta = 1.0$ and $K = 1.0$
	l_1 = 5E−02, l_2 = 1E+02, and l_3 = 1E−02

For a fair comparison, the maximum number of fitness evaluations was set at 10,000 for all methods. The common settings of all methods, along with the parameter settings for each method, are explained in Table 1.

The common setups of the considered BN are presented as follows. All BUs are allocated randomly in a square space of 1000 m × 1000 m and the MEC server is placed in the middle of this space. Ten examples with multiple numbers of BUs $(i.e., n = 50, 100, \ldots, 500)$ are studied to examine the proposed CHSGSO performance. Other setups are shown in Table 2.

To execute all the experiments in this paper, MATLAB was utilized on a computing environment with a Dual Intel® Xeon® Gold 5115 2.4 GHz CPU and 128 GB of RAM on the operating system Microsoft Windows Server 2019.

4.3 The effect of group size (GS)

By implementing different experiments, the effect of the group size *GS* on the proposed CHSGSO algorithm performance is verified. This is achieved by establishing the value of $GS = \{5, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 30\}$. Box plots of average accumulative profits for CHSGSO with different instances are presented in Fig. 5. The Friedman test was carried out to rank all the variants, with the results reported in Table 3 for the mean results. It can be seen from Table 3 that the variant with GS = 14 is the best-obtained result. Based on the Friedman test, the variant with GS = 14 is preferred and it will be used throughout the paper.

Table 2 The considered BN common settings

Parameter	Value
σ^2	174 dBm/Hz
X	1.8e+5 Cycles/bit
c_2	10 Token/G Cycles
α	0.005 Token/bit
f_i	[0.1, 2] G Cycles/s
λ	1/600
z	1.00e−4
B	10 MHz
$T_{i,\max}$	4 s
c_1	20 Token/J
E_0	70 J
$k1$	1e−27
c_3	3 Token/J
D_i	[1, 2] K bit
f^{total}	800 G Cycles/s
w	2
p_i	[0.01, 1] W
k_2	1

4.4 Comparison with four counterparts

The results obtained from CHSGSO and the competing algorithms over 30 independent runs are presented in Table 4, where "AVG" and "STD" denote the mean and standard deviation of the total earnings of all BUs. Sequentially, the ratios within the square brackets denote the improved rate of CHSGSO versus the rival algorithms. Wilcoxon's rank-sum test [20] is used for assessing the significance of the proposed CHSGSO against counterparts. Wilcoxon's rank-sum statistical analysis is conducted at a 0.05 significance level. Table 4, " $\approx''$, " $\downarrow''$, and " $\uparrow''$ show that CHSGSO performs similarly, worse than, and better than every counterpart, sequentially.

From Table 4, it is observed that the proposed CHSGSO achieves the best mean cumulative profits out of its four counterparts in each case. It should be noted that at $n \geq 200$, the mean cumulative gain achieved by CHANGES is much higher than that derived from three of its counterparts (ACOMV, DE, and BOToP) and slightly higher than that of the DEMiDRA algorithm. In terms of improving performance, CHICAGO outperforms its four counterparts in each case. Specifically, against DE, CHICAGO achieves greater than 100% performance enhancement in every case unless at $n = 50$. At $n \geq 200$, the enhancement rate of CHSGSO exceeds 200%. Against BOToP, *CHICAGO* achieves greater than 100% performance enhancement in every case unless at $n = 50$. At $n \geq 250$, the enhancement rate of CHSGSO exceeds 200%. Against ACOMV, when $n \geq 200$, CHSGSO obtains an improvement rate greater than 115%. In the end, CHSGSO always gets a slightly better rate than the DEMiDRA algorithm. As stated in Wilcoxon's rank-sum statistical analysis, CHSGSO is statistically more reliable than its four counterparts in each case.

Figure 6 presents the growth of the average accumulative gains derived from ACOMV, DE, DEMiDRA, BOToP and the proposed CHSGSO when $n = 50$, 100, 150, 200, 250, 300, 350, 400, 450, and 500. As shown in Fig. 6, CHSGSO achieves better average accumulative gains than all of its counterparts. Specifically, against ACOMV, DE, and BOToP, CHICAGO achieves higher average accumulative gains in all instances. Against DEMiDRA, CHICAGO achieves slightly higher average accumulative gains in all cases unless at $n = 150$ where the average cumulative gains of the algorithms are equal.

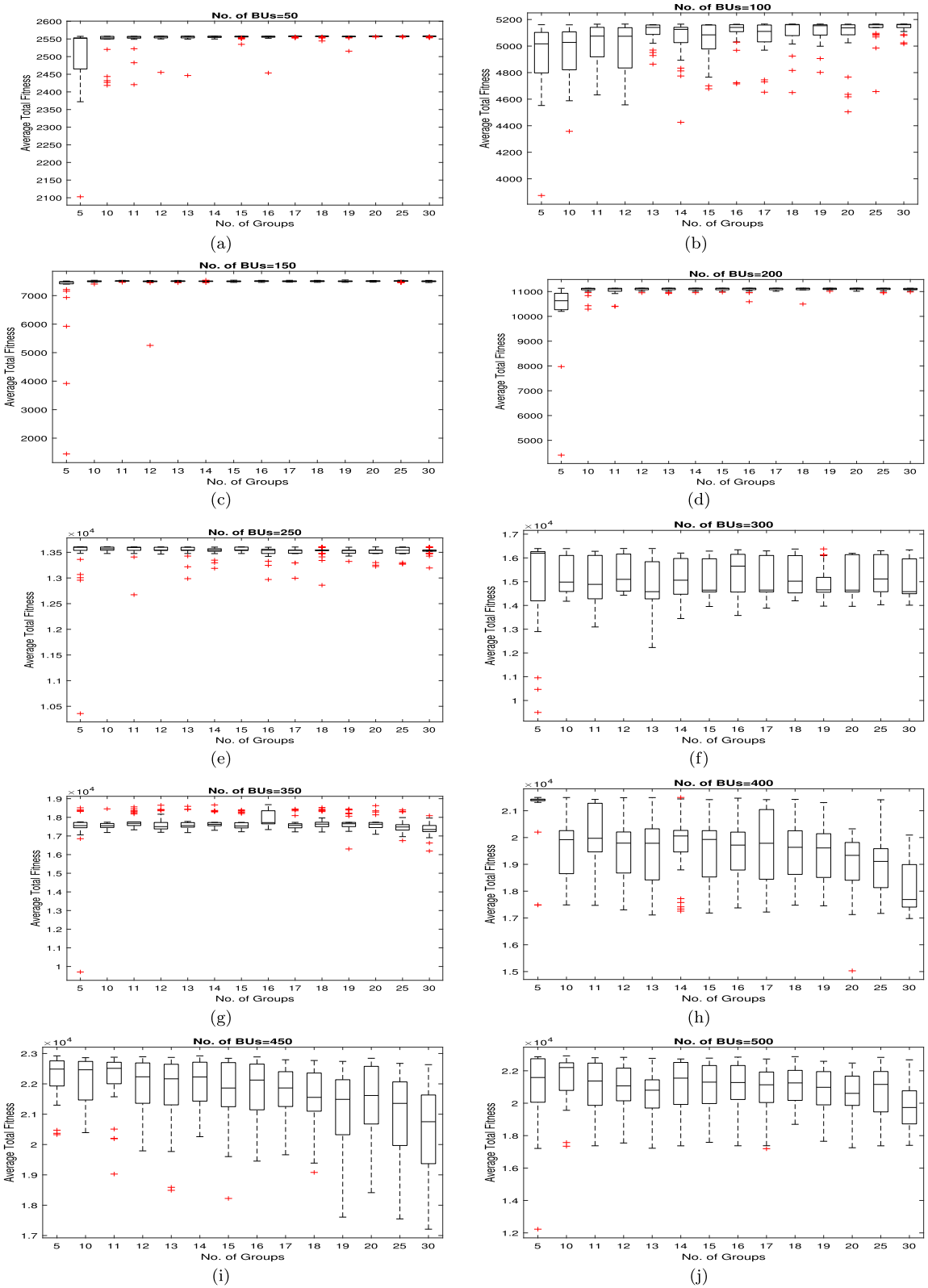

Fig. 5 Group size effect for **a** 50; **b** 100; **c** 150; **d** 200; **e** 250; **f** 300; **g** 350; **h** 400; **i** 450; and **j** 500 BUs

Table 3 Average rankings of the variants according to Friedman test

Variant	Ranking
GS = 05	10.2
GS = 10	7.3
GS = 11	6.7
GS = 12	6.8
GS = 13	8.7
GS = 14	5.7
GS = 15	7.2
GS = 16	5.9
GS = 17	7.8
GS = 18	6.9
GS = 19	8
GS = 20	7.3
GS = 25	7
GS = 30	9.5

4.5 Insights and real-world applications for proposed approach

One of the critical components of blockchain technology is the verification of transactions or blocks in and IoT and blockchain networks. This process demands an important amount of energy, making it a resource-intensive operation. Consequently, there is a continuous requirement for the optimization of gas usage in those networks. The gas optimization algorithm that has proven to be effective is the Improved Henry Gas Optimization method, which depends on the Henry Gas Law that uses the solubility of a gas in a liquid as directly proportional to the partial pressure of the gas, given a constant temperature. The proposed approach by predicting the gas consumption of the network

Table 4 Comparisons of CHSGSO against a few promising algorithms

n	ACO_{MV} AVG ± STD	DE AVG ± STD	DEMiDRA AVG ± STD	BOToP AVG ± STD	CHSGSO AVG ± STD
50	1.87e+3 ± 1.33e+1 [36.90%] ↑	1.42e+3 ± 3.41e+1 [80.28%] ↑	2.27e+3 ± 2.27e+2 [12.78%] ↑	1.61e+3 ± 7.18e+1 [59.00%] ↑	2.56e+03 ± 1.45e+00
100	2.86e+3 ± 1.97e+1 [79.37%] ↑	2.27e+3 ± 4.78e+1 [125.99%] ↑	4.74e+3 ± 3.60e+2 [8.22%] ↑	2.50e+3 ± 8.14e+1 [100.05%] ↑	5.13e+03 ± 1.36e+02
150	3.81e+3 ± 1.27e+2 [96.85%] ↑	2.91e+3 ± 5.69e+1 [157.73%] ↑	7.50e+3 ± 1.25e+1 [0.00%] ≈	3.12e+3 ± 1.14e+2 [140.38%] ↑	7.50e+03 ± 1.60e+01
200	5.11e+3 ± 2.57e+2 [117.22%] ↑	3.59e+3 ± 8.47e+1 [209.19%] ↑	1.00e+4 ± 1.86e+1 [11.00%] ↑	3.78e+3 ± 1.70e+2 [193.65%] ↑	1.11e+04 ± 4.12e+01
250	6.23e+3 ± 2.67e+2 [118.30%] ↑	4.24e+3 ± 8.73e+1 [220.08%] ↑	1.25e+4 ± 2.56e+2 [8.80%] ↑	4.35e+3 ± 1.27e+2 [212.64%] ↑	1.36e+04 ± 4.18e+01
300	6.81e+3 ± 2.55e+2 [121.73%] ↑	4.54e+3 ± 7.93e+2 [232.60%] ↑	1.48e+4 ± 4.18e+1 [2.03%] ↑	4.60e+3 ± 1.57e+2 [228.26%] ↑	1.51e+04 ± 1.01e+03
350	7.68e+3 ± 3.38e+2 [129.17%] ↑	5.04e+3 ± 1.28e+2 [249.21%] ↑	1.72e+4 ± 5.82e+1 [2.33%] ↑	5.23e+3 ± 1.44e+2 [236.52%] ↑	1.76e+04 ± 3.02e+02
400	8.12e+3 ± 3.09e+2 [145.07%] ↑	5.30e+3 ± 1.19e+2 [275.47%] ↑	1.95e+4 ± 1.03e+2 [2.05%] ↑	5.67e+3 ± 2.50e+2 [250.97%] ↑	1.99e+04 ± 1.34e+03
450	9.08e+3 ± 4.64e+2 [140.09%] ↑	5.82e+3 ± 1.35e+2 [274.57%] ↑	1.99e+4 ± 2.02e+2 [9.55%] ↑	6.09e+3 ± 2.37e+2 [257.96%] ↑	2.18e+04 ± 9.07e+02
500	9.38e+3 ± 4.80e+2 [123.88%] ↑	5.92e+3 ± 1.98e+2 [254.73%] ↑	1.99e+4 ± 1.69e+2 [5.53%] ↑	6.60e+3 ± 2.96e+2 [218.18%] ↑	2.10e+04 ± 1.46e+03
↑ / ↓ / ≈	10/0/0	10/0/0	10/0/0	9/0/1	

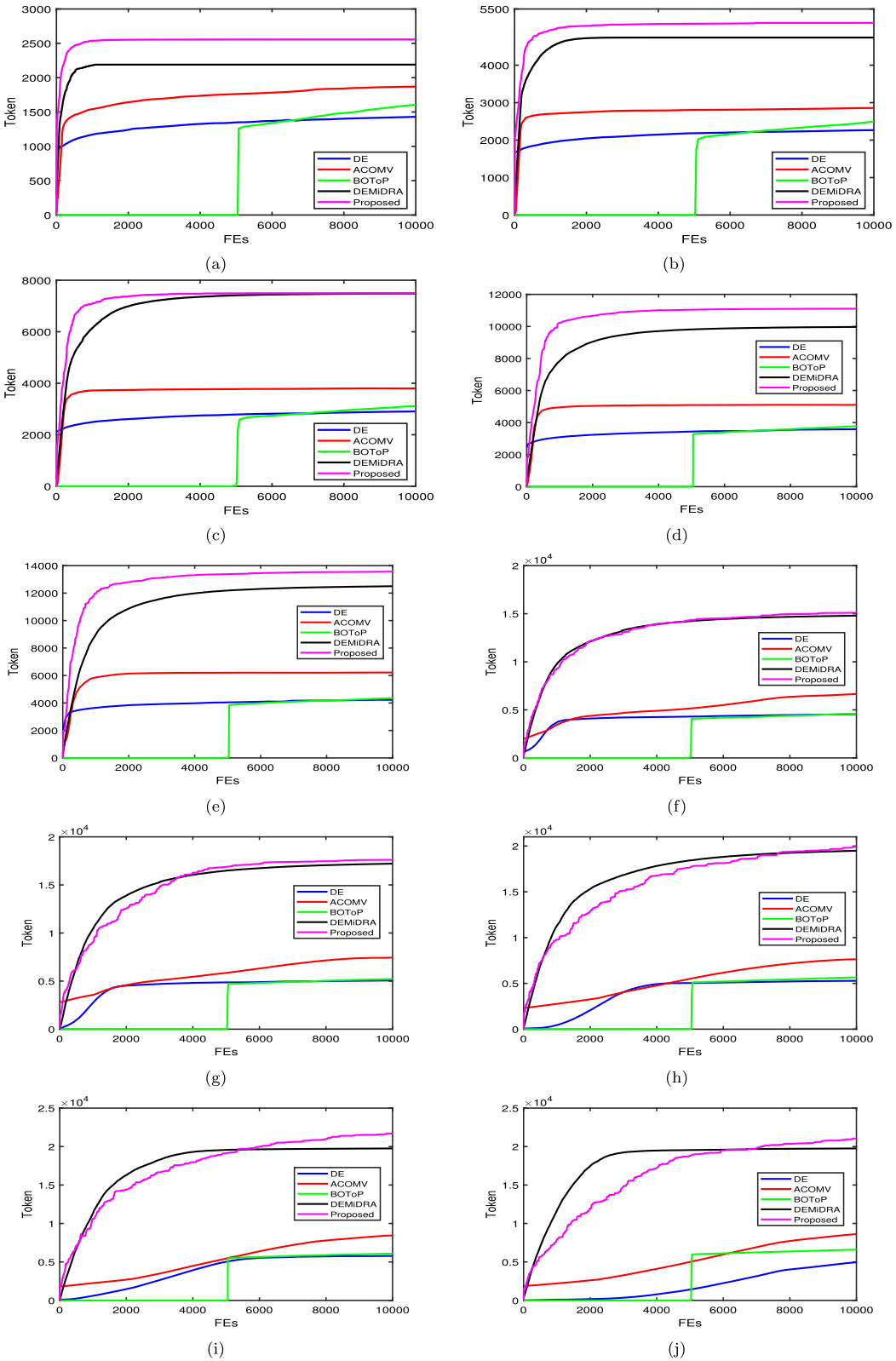

Fig. 6 Evolution of the mean total gain obtained by the proposed CHSGSO and other algorithms for **a** 50; **b** 100; **c** 150; **d** 200; **e** 250; **f** 300; **g** 350; **h** 400; **i** 450; and **j** 500 BUs

nodes, allowing more accurate estimations of gas fees required for transactions in the blockchain network.

The approach has various real-world applications, particularly in the financial sector, where blockchain networks are widely utilized. For instance, in a peer-to-peer (P2P) lending platform, users lend and borrow money from each other. The blockchain network that supports the platform should operate efficiently with minimal gas usage to assert that the platform's users benefit from the P2P lending platform. Moreover, the proposed approach in smart contract deployment allows developers to develop and deploy more efficient smart contracts that consume fewer resources while delivering optimal performance. For example, smart contracts deployed in the healthcare sector can record patient information and generate automated reminders, among other things, without consuming more gas than necessary.

Furthermore, the proposed algorithm is crucial for Proof-of-Stake (PoS) blockchain networks, where stakeholders who hold a significant number of tokens are responsible for verifying transactions instead of miners. Since PoS models do not require high computational power, the Improved Henry Gas Optimization method can significantly reduce gas usage for PoS-based blockchain networks. In conclusion, the proposed approach has proven to be an effective technique in reducing the gas fees required for transactions in blockchain networks. The use of this technique could significantly reduce the energy consumption of blockchain transactions, making blockchain networks more sustainable and environmentally friendly. The implementation of this method also has numerous real-world applications, particularly in finance, healthcare, energy, cybersecurity and IoT systems, where blockchain networks are widely employed.

5 Conclusions and future directions

In this study, an improved HGSO approach (termed CHSGSO) is presented to jointly improve the resource allocation and mining decisions for MEC-enabled wireless BNs. First, the resource allocation to only participating BUs is encoded, as the BU who decides to participate is encoded as an individual. An adaptive strategy is designed to tune each individual size. Following that a chaotic map was integrated into the original HGSO to improve the convergence rate. Finally, CHSGSO was implemented on a group of instances with various scales and compared to ACO, DE, DEMiDRA, and BOToP. The empirical outcomes verified the efficiency and significance of the proposed CHSGSO. It is noteworthy that this study considers that IoT devices are homogeneous in the utilized BNs.

In future studies, the heterogeneous BNs will be examined. Also, we will investigate the proposed algorithm's performance in solving other real-optimization problems and optimization problems with more than objective functions.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Data availability Data are available on request from the authors.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abdel-Basset M, El-Shahat D, El-Henawy I (2019) Solving 0–1 knapsack problem by binary flower pollination algorithm. Neural Comput Appl 31(9):5477–5495
2. Abdel-Basset M, El-Shahat D, Sangaiah AK (2019) A modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem. Int J Mach Learn Cybern 10(3):495–514
3. Abdel-Basset M, Mohamed R, Sallam KM, Chakrabortty RK, Ryan MJ (2021) BSMA: a novel metaheuristic algorithm for multi-dimensional knapsack problems: method and comprehensive analysis. Comput Ind Eng 159:107469
4. Abdel-Basset M, Mohamed R, Abouhawwash M, Alshamrani AM, Mohamed AW, Sallam K (2023) Binary light spectrum optimizer for knapsack problems: an improved model. Alex Eng J 67:609–632
5. Agrawal P, Ganesh T, Mohamed AW (2022) Solving knapsack problems using a binary gaining sharing knowledge-based optimization algorithm. Complex Intell Syst 8(1):43–63
6. Ali IM, Essam D, Kasmarik K (2018) An efficient differential evolution algorithm for solving 0–1 knapsack problems. In: 2018 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8
7. Ali MS, Vecchio M, Pincheira M, Dolui K, Antonelli F, Rehmani MH (2019) Applications of blockchains in the internet of things: a comprehensive survey. IEEE Commun Surv Tutor 21(2):1676–1717. https://doi.org/10.1109/COMST.2018.2886932

8. Azzi R, Chamoun RK, Sokhn M (2019) The power of a blockchain-based supply chain. Comput Ind Eng 135:582–592
9. Baş E (2023) Binary aquila optimizer for 0–1 knapsack problems. Eng Appl Artif Intell 118:105592
10. Baykasoğlu A, Ozsoydan FB (2014) An improved firefly algorithm for solving dynamic multidimensional knapsack problems. Expert Syst Appl 41(8):3712–3725
11. Böhme R, Christin N, Edelman B, Moore T (2015) Bitcoin: economics, technology, and governance. J Econ Perspect 29(2):213–38
12. Bolaji AL, Okwonu FZ, Shola PB, Balogun BS, Adubisi OD (2021) A modified binary pigeon-inspired algorithm for solving the multi-dimensional knapsack problem. J Intell Syst 30(1):90–103
13. Brindha M, Ammasai Gounden N (2016) A chaos based image encryption and lossless compression algorithm using hash table and Chinese remainder theorem. Appl Soft Comput 40:379–390. https://doi.org/10.1016/j.asoc.2015.09.055
14. Brown TL, LeMay HE, Bursten BE, Burdge JR (2002) Chemistry: the central science. Pearson Education, London
15. Changdar C, Mahapatra G, Pal RK (2015) An improved genetic algorithm based approach to solve constrained knapsack problem in fuzzy environment. Expert Syst Appl 42(4):2276–2286
16. Chen G, Mao Y, Chui CK (2004) A symmetric image encryption scheme based on 3d chaotic cat maps. Chaos Solitons Fractals 21(3):749–761. https://doi.org/10.1016/j.chaos.2003.12.022
17. Chih M (2015) Self-adaptive check and repair operator-based particle swarm optimization for the multidimensional knapsack problem. Appl Soft Comput 26:378–389
18. Das S, Suganthan PN (2010) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31
19. Dehkordi AA, Sadiq AS, Mirjalili S, Ghafoor KZ (2021) Nonlinear-based chaotic Harris hawks optimizer: algorithm and internet of vehicles application. Appl Soft Comput 109:107574. https://doi.org/10.1016/j.asoc.2021.107574
20. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18
21. Diab H (2018) An efficient chaotic image cryptosystem based on simultaneous permutation and diffusion operations. IEEE Access 6:42227–42244. https://doi.org/10.1109/ACCESS.2018.2858839
22. Diab H, El-semary AM (2018) Secure image cryptosystem with unique key streams via hyper-chaotic system. Signal Process 142:53–68. https://doi.org/10.1016/j.sigpro.2017.06.028
23. Feng Y, Wang GG (2022) A binary moth search algorithm based on self-learning for multidimensional knapsack problems. Futur Gener Comput Syst 126:48–64
24. García J, Maureira C (2021) A KNN quantum cuckoo search algorithm applied to the multidimensional knapsack problem. Appl Soft Comput 102:107077
25. Gupta S, Su R, Singh S (2022) Diversified sine-cosine algorithm based on differential evolution for multidimensional knapsack problem. Appl Soft Comput 130:109682
26. Haddar B, Khemakhem M, Hanafi S, Wilbaut C (2016) A hybrid quantum particle swarm optimization for the multidimensional knapsack problem. Eng Appl Artif Intell 55:1–13
27. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. Futur Gener Comput Syst 101:646–667
28. Houy N (2016) The bitcoin mining game. Ledger 1:53–68. https://doi.org/10.5195/ledger.2016.13
29. Huang PQ, Wang Y, Wang K, Liu ZZ (2020) A bilevel optimization approach for joint offloading decision and resource allocation in cooperative mobile edge computing. IEEE Trans Cybern 50(10):4228–4241. https://doi.org/10.1109/TCYB.2019.2916728
30. Hussain I, Shah T, Gondal MA (2013) Application of s-box and chaotic map for image encryption. Math Comput Model 57(9):2576–2579. https://doi.org/10.1016/j.mcm.2013.01.009
31. Jiao Y, Wang P, Niyato D, Suankaewmanee K (2019) Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks. IEEE Trans Parallel Distrib Syst 30(9):1975–1989. https://doi.org/10.1109/TPDS.2019.2900238
32. Karame G, Androulaki E, Capkun S (2012) Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin. IACR Cryptol ePrint Arch 2012(248)
33. Khokhar B, Dahiya S, Parmar KS (2021) Load frequency control of a microgrid employing a 2d sine logistic map based chaotic sine cosine algorithm. Appl Soft Comput 109:107564. https://doi.org/10.1016/j.asoc.2021.107564
34. Kiayias A, Koutsoupias E, Kyropoulou M, Tselekounis Y (2016) Blockchain mining games. In: Proceedings of the 2016 ACM conference on economics and computation, pp 365–382
35. Kohli M, Arora S (2018) Chaotic grey wolf optimization algorithm for constrained optimization problems. J Comput Des Eng 5(4):458–472. https://doi.org/10.1016/j.jcde.2017.02.005
36. Kong M, Tian P, Kao Y (2008) A new ant colony optimization algorithm for the multidimensional knapsack problem. Comput Oper Res 35(8):2672–2683
37. Kroll JA, Davey IC, Felten EW (2013) The economics of bitcoin mining, or bitcoin in the presence of adversaries. In: Proceedings of WEIS, vol 2013, p 11
38. Kumar S, Yildiz BS, Mehta P, Panagant N, Sait SM, Mirjalili S, Yildiz AR (2023) Chaotic marine predators algorithm for global optimization of real-world engineering problems. Knowl Based Syst 261:110192. https://doi.org/10.1016/j.knosys.2022.110192
39. Leng J, Yan D, Liu Q, Xu K, Zhao JL, Shi R, Wei L, Zhang D, Chen X (2019) Manuchain: combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing. IEEE Trans Syst Man Cybern Syst 50(1):182–192
40. Li Y, Zhang W, Wang L, Zhao F, Han W, Chen G (2017) Henry's law and accumulation of weak source for crust-derived helium: a case study of Weihe basin, china. J Nat Gas Geosci 2(5–6):333–339
41. Liao T, Socha K, de Oca MAM, Stützle T, Dorigo M (2013) Ant colony optimization for mixed-variable optimization problems. IEEE Trans Evol Comput 18(4):503–518
42. Lin FT (2008) Solving the knapsack problem with imprecise weight coefficients using genetic algorithms. Eur J Oper Res 185(1):133–145
43. Liu J, Wang Y, Xin B, Wang L (2021) A biobjective perspective for mixed-integer programming. IEEE Trans Syst Man Cybern Syst 52(4):2374–2385
44. Liu M, Yu FR, Teng Y, Leung VCM, Song M (2018) Computation offloading and content caching in wireless blockchain networks with mobile edge computing. IEEE Trans Veh Technol 67(11):11008–11021. https://doi.org/10.1109/TVT.2018.2866365
45. Liu X, Wang W, Niyato D, Zhao N, Wang P (2018) Evolutionary game for mining pool selection in blockchain networks. IEEE Wirel Commun Lett 7(5):760–763
46. Luo K, Zhao Q (2019) A binary grey wolf optimizer for the multidimensional knapsack problem. Appl Soft Comput 83:105645
47. Luong NC, Xiong Z, Wang P, Niyato D (2018) Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach. In: 2018 IEEE International Conference on Communications (ICC), pp 1–6, https://doi.org/10.1109/ICC.2018.8422743

48. Mohebbi V, Naderifar A, Behbahani R, Moshfeghian M (2012) Determination of Henry's law constant of light hydrocarbon gases at low temperatures. J Chem Thermodyn 51:8–11
49. Mukherjee A, Mukherjee V (2015) Solution of optimal power flow using chaotic krill herd algorithm. Chaos Solitons Fractals 78:10–21. https://doi.org/10.1016/j.chaos.2015.06.020
50. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. In: Decentralized business review, p 21260
51. Pietrych L, Sandubete JE, Escot L (2021) Solving the chaos model-data paradox in the cryptocurrency market. Commun Nonlinear Sci Numer Simul 102:105901. https://doi.org/10.1016/j.cnsns.2021.105901
52. Zg Ren, Zr Feng, Am Zhang (2012) Fusing ant colony optimization with Lagrangian relaxation for the multiple-choice multidimensional knapsack problem. Inf Sci 182(1):15–29
53. Rezoug A, Bader-El-Den M, Boughaci D (2018) Guided genetic algorithm for the multidimensional knapsack problem. Memet Comput 10(1):29–42
54. Rizun PR (2016) A transaction fee market exists without a block size limit
55. Saleh H, Saber W, Rizk R (2022) Mobile computation offloading in mobile edge computing based on artificial intelligence approach: a review and future directions. In: International conference on advanced machine learning technologies and applications. Springer, pp 593–603
56. Sallam KM, Elsayed SM, Sarker RA, Essam DL (2017) Landscape-based adaptive operator selection mechanism for differential evolution. Inf Sci 418:383–404
57. Sallam KM, Elsayed SM, Sarker RA, Essam DL (2017) Multimethod based orthogonal experimental design algorithm for solving CEC2017 competition problems. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE, pp 1350–1357
58. Sallam KM, Elsayed SM, Sarker RA, Essam DL (2018) Improved united multi-operator algorithm for solving optimization problems. In: 2018 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8
59. Sallam KM, Chakrabortty RK, Ryan MJ (2020) A two-stage multi-operator differential evolution algorithm for solving resource constrained project scheduling problems. Future Gener Comput Syst 108:432–444
60. Sallam KM, Elsayed SM, Chakrabortty RK, Ryan MJ (2020) Improved multi-operator differential evolution algorithm for solving unconstrained problems. In: 2020 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8
61. Sayed GI, Khoriba G, Haggag MH (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. Appl Intell 48(10):3462–3481. https://doi.org/10.1007/s10489-018-1158-6
62. Sharma S, Singh G (2023) Design and analysis of novel chaotic pelican-optimization algorithm for feature-selection of occupational stress. Procedia Comput Sci 218:1497–1505. https://doi.org/10.1016/j.procs.2023.01.128
63. Staudinger J, Roberts PV (1996) A critical review of Henry's law constants for environmental applications. Crit Rev Environ Sci Technol 26(3):205–297
64. Talatahari S, Azizi M (2020) Optimization of constrained mathematical and engineering design problems using chaos game optimization. Comput Ind Eng 145:106560. https://doi.org/10.1016/j.cie.2020.106560
65. Talatahari S, Farahmand Azar B, Sheikholeslami R, Gandomi A (2012) Imperialist competitive algorithm combined with chaos for global optimization. Commun Nonlinear Sci Numer Simul 17(3):1312–1319. https://doi.org/10.1016/j.cnsns.2011.08.021
66. Tang C, Li C, Yu X, Zheng Z, Chen Z (2020) Cooperative mining in blockchain networks with zero-determinant strategies. IEEE Trans Cybern 50(10):4544–4549. https://doi.org/10.1109/TCYB.2019.2915253
67. Tong X, Cui M (2009) Image encryption scheme based on 3d baker with dynamical compound chaotic sequence cipher generator. Signal Process 89(4):480–491. https://doi.org/10.1016/j.sigpro.2008.09.011
68. Wang K, Huang PQ, Yang K, Pan C, Wang J (2019) Unified offloading decision making and resource allocation in ME-RAN. IEEE Trans Veh Technol 68(8):8159–8172. https://doi.org/10.1109/TVT.2019.2926513
69. Wang S, Taha AF, Wang J, Kvaternik K, Hahn A (2019) Energy crowdsourcing and peer-to-peer energy trading in blockchain-enabled smart grids. IEEE Trans Syst Man Cybern Syst 49(8):1612–1623
70. Wang Y, Wang W (2021) Quantum-inspired differential evolution with grey wolf optimizer for 0–1 knapsack problem. Mathematics 9(11):1233
71. Wang Y, Chen CR, Huang PQ, Wang K (2021) A new differential evolution algorithm for joint mining decision and resource allocation in a MEC-enabled wireless blockchain network. Comput Ind Eng 155:107186
72. Xiong Z, Zhang Y, Niyato D, Wang P, Han Z (2018) When mobile blockchain meets edge computing. IEEE Commun Mag 56(8):33–39
73. Xiong Z, Feng S, Wang W, Niyato D, Wang P, Han Z (2019) Cloud/fog computing resource management and pricing for blockchain networks. IEEE Internet Things J 6(3):4585–4600. https://doi.org/10.1109/JIOT.2018.2871706
74. Yang X, Zhou Y, Shen A, Lin J, Zhong Y (2021) A hybrid ant colony optimization algorithm for the knapsack problem with a single continuous variable. In: Proceedings of the genetic and evolutionary computation conference, pp 57–65
75. Ye G (2010) Image scrambling encryption algorithm of pixel bit based on chaos map. Pattern Recogn Lett 31(5):347–354. https://doi.org/10.1016/j.patrec.2009.11.008
76. Yuan S, Li J, Wu C (2022) JORA: blockchain-based efficient joint computing offloading and resource allocation for edge video streaming systems. J Syst Arch 133:102740. https://doi.org/10.1016/j.sysarc.2022.102740
77. Yuan Y, Wang FY (2018) Blockchain and cryptocurrencies: model, techniques, and applications. IEEE Trans Syst Man Cybern Syst 48(9):1421–1428
78. Zhang S, Liu S (2019) A discrete improved artificial bee colony algorithm for 0–1 knapsack problem. IEEE Access 7:104982–104991
79. Zhang Y, Zhang P, Tao F, Liu Y, Zuo Y (2019) Consensus aware manufacturing service collaboration optimization under blockchain based industrial internet platform. Comput Ind Eng 135:1025–1035
80. Zhou Y, Bao L, Chen CP (2014) A new 1d chaotic system for image encryption. Signal Process 97:172–182. https://doi.org/10.1016/j.sigpro.2013.10.034
81. Zouache D, Nouioua F, Moussaoui A (2016) Quantum-inspired firefly algorithm with particle swarm optimization for discrete optimization problems. Soft Comput 20(7):2781–2799