Computer & Robotics

# Black Widow Optimization (BWO) Algorithm in Cloud Brokering Systems for Connected Internet of Things

Nasim Jelodari [a], Ali Asghar Pourhaji Kazem [a,*]

[a] *Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran*

Received 25 January 2022, Accepted 30 June 2022

## Abstract

The Internet of Things (IoT) now connects over nine billion devices. This number is predicted to approach 20 billion in the near future, and the number of things is rapidly expanding, implying that a large amount of data will be created. To handle the connected things, an infrastructure must be built. Cloud computing (CC) has become necessary in the analysis and data storage for IoT. A cloud broker, which is an intermediate in the infrastructure that controls connected things in cloud computing, is discussed in this study. An optimization problem is examined for maximizing the broker's profit and system availability while minimizing request response time and energy consumption. For this purpose, an objective function is proposed and solved using the Black Widow Optimization (BWO) algorithm. Subsequently, the obtained results are compared with the particle swarm optimization (PSO) algorithms. The results indicate that the BWO algorithm could outperform the PSO algorithm, and it can provide much better results considering different scenarios.

***Keywords:*** Internet of Things, Cloud Broker, Black Widow Optimization Algorithm, Optimization, Simulation

## 1.Introduction

In 1999, Kevin Ashton proposed the word "Internet of Things (IoT)," referring to it as "uniquely identified interoperable connected things" using radio-frequency identification (RFID) technology. In other words, IoT is a network of things that are wirelessly linked through smart sensors and can interact without human involvement. Several experimental IoT applications have already been created in healthcare, transportation, and automotive [1]. Furthermore, IoT may be thought of as a superset of connected objects that can be uniquely identified using current near-field communication (NFC) approaches [2]. IoT is proposed by the IoT European Research Cluster (IERC) as a self-configuring global network infrastructure based on open and interoperable communication protocols [3]. Compared to traditional information and communication technologies, the IoT is an intelligent and autonomous solution that allows enterprises to more simply and effectively react to customer needs [4]. However, the projections for the Internet of Things' influence on the internet and the economy are significant. Some estimate that up to 100 billion linked IoT devices and a worldwide economic impact of more than $11 trillion by 2025 [5]. Although the number of layers that make up an IoT ecosystem might grow over time, it operates on three core levels: the IoT device layer, the IoT gateway layer, and the IoT platform layer [6]. IoT is commonly regarded as the next generation of technology, having broad applications across almost every segment of the industry and the possibility to expand the degree of integration of end products, systems, and services. Fig.1 shows the development of IoT in different phases.

* Corresponding Author Email: a.pourhajikazem@iaut.ac.ir

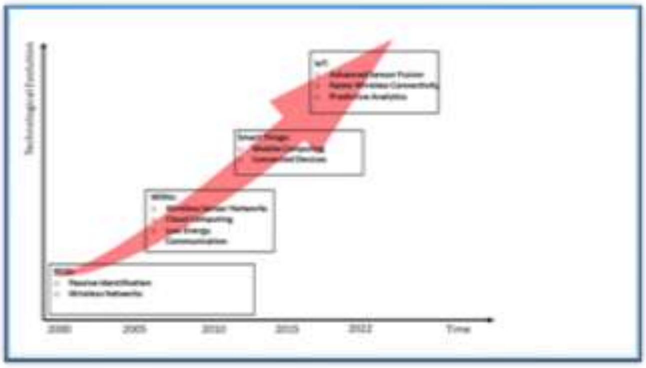

Fig.1. Evolution of the IoT

Cloud computing (CC) means providing processing services that include servers, data storage, databases, networks, software, analysis and artificial intelligence on the internet (cloud) platform. In a broad term, CC allows client devices to connect to remote physical servers, databases, and computers over the internet to access data and cloud applications. Clouds generally fall into one of two main categories of infrastructure or service. In terms of infrastructure, there are four different categories as follows:

- Private cloud
- Public cloud
- Hybrid Cloud
- Social cloud

However, the CC's services are categorized into three types: (i) Infrastructure as a Service (IaaS), when a cloud-based service includes processing resources such as server hardware, network bandwidth, or load balancing systems, the cloud is said to provide infrastructure. A well-known example of this is Amazon Web Services; (ii) Platform as a Service (PaaS), when a cloud provides an environment that users can use to develop software, what is provided is called a platform. Such a service is very convenient for users who want to focus only on the actual development of the application and do not have to bear the heavy burden of managing and configuring the hardware and software issues of the host system that cause cloud activity. Force.com can be called an example of this category; (iii) Software as a Service (SaaS) enables customers to perform software via the internet without installing or maintaining the software. One of the most critical problems the IoT will face in the future is data management and support, which will significantly impose pressure on the Internet infrastructure. A cloud computing infrastructure interface is utilized to make sense of the data provided by the Internet of Things to tackle this issue. Cloud Services Broker (CSB) acts as an intermediary between Cloud Service Providers (CSPs) and end-users of cloud services. A CSB's role is to connect users' requirements to the best available service. CSB considers qualitative criteria such as availability, reliability, throughput, response time, and energy consumption to select the best service between users and cloud service providers. The CSB is also expected to consider its maximum profit when choosing the best service. On the other hand, the user wishes that the service or system in question, in case of anticipated or unforeseen problems or having a single point of failure (SPOF) of the service or in the system in question, maximum availability. Furthermore, minimizing energy consumption is also an important issue due to the development of infrastructure and cloud computing hardware, and finally, the user expects response time to be minimized when submitting their request to service providers.

Given that the problem of this research is in the category of Non-deterministic Polynomial (NP)-Hard problems, it can be solved using heuristic and metaheuristic algorithms. In reality, the Greek word "meta," given inside the title, is employed to illustrate that these algorithms are "higher-level" heuristic algorithms distinguishing from problem-specific heuristics [7]. Irrespective of providing good results, metaheuristic algorithms do not offer optimized solutions. Generally speaking, metaheuristic algorithms can be classified into four main groups:

- **Evolutionary Algorithms (EAs):** They are very efficient heuristic search techniques based on Darwinian evolution. They combine the advantages of resilience and flexibility with the ability to capture global solutions to complicated optimization problems. Using EAs early in the optimization process makes the possibility of discovering a near-optimum relatively high [8]. The well-known EAs algorithms are genetic algorithm (GA) [9] and differential evolution (DE) [10].
- **Swarm Intelligence Algorithm (SI):** Swarm intelligence algorithms are, in general, nature-inspired algorithms inspired by the interactions of biological species such

as flocks of birds, ants, and fish. By identifying new combinations of values, these methods aid in the improvement of fitness functions in combinatorial and numerical optimization problems [11]. Particle swarm optimization (PSO) [12], Ant colony optimization (ACO) [13, 14], Whale Optimization Algorithm (WOA) [15], and also Grey Wolf Optimizer (GWO) [16, 17] are renowned swarm-based algorithms.

- **Physics-based Algorithms:** They are inspired by some famous physics laws, including (i) Newton's gravitational law such as artificial physics optimization (APO) [18]; (ii) Quantum mechanics like atomic orbital search (AOS) [19, 20] and Material Generation Algorithm (MGA) [21, 22]; (iii) Universe theory such as big bang-big crunch (BBBC) [23]; (iv) Electromagnetism such as electromagnetism-like heuristic (EM) [24] and crystal structure algorithm (CryStAl) [25-28]; and chaos game optimization (CGO) [29, 30].
- **Human and Animal Behaviour-based Algorithms:** They are inspired by some specific behaviours of humans in society and animals in nature. Some of the prominent algorithms in this category are Teaching–learning-based Optimization (TLBO) [31], Harmony Search (HS) [32, 33], and Human Behavior-Based Optimization (HBBO) [34].

Nonetheless, the current study employed the Black Widow Optimization (BWO) algorithm proposed by Hayyolalam and Pourhaji Kazem [35] to minimize the request-response time and energy consumption in CC while maximizing availability and broker profits for connected IoT servers. The current research work considers four different objectives simultaneously for the first time, which could be the novelty of this. Another novelty of this paper is employing the BWO algorithm in CC problems. The main contributions of this paper can be summarized as follows:

- Investigate a cloud brokering problem in a cloud IoT system considering the response time, broker profit, energy consumption, and system availability.
- A novel optimization problem is formulated to maximize the system availability and broker profit and minimize the system's energy consumption and response time. For the mentioned purpose, a novel recently proposed metaheuristic algorithm is employed.
- The obtained results by the BWO algorithm are compared with the PSO algorithm from the literature review.

The rest of the paper is divided into the following sections. In Section 2, the literature review is presented; and the inspiration and mathematical model of the BWO algorithm are presented in Section 3. Furthermore, Section 4 describes the optimization problem statement, including objective functions. Results and discussion are illustrated in Section 5. Finally, in section 6, the core findings of this study are presented as concluding remarks.

## 2. Literature Review

In recent years, many research works focused on cloud brokering in CC. Kessaci, et al. [36] proposed a multi-objective genetic algorithm (MO-GA) that optimizes a geographically dispersed cloud computing infrastructure's energy consumption, CO2 emissions, and profit earned. Additionally, the authors suggested a greedy heuristic to increase the number of scheduled applications to compare it to the MO-GA. Mehta, et al. [37] presented a multi-purpose cloud broker system to find cloud resources using standard and custom attributes and control these resources, in which the cloud broker provides an optimized list of cloud resources based on selected attributes to help the consumer select the most appropriate resources. Yildirim, et al. [38] proposed a genetic algorithm-based solution to find an optimal sensor node distribution. Elhoseny, et al. [39] presented a novel approach for optimizing the selection of virtual machines (VMs) in cloud-IoT health services applications to effectively handle a large quantity of data in an integrated industry. 4.0. Industry 4.0 applications demand the processing and analysis of large amounts of data that originate from various sources, such as sensor data, without the participation of humans. Mei, et al. [40] proposed a novel methodology to reduce the cost of cloud users.

Asghari, et al. [41] proposed a privacy-aware cloud service composition approach for QoS optimization in the IoT environment by presenting an IoT-based cloud service composition model based on the privacy level computing model and a novel hybrid evolutionary algorithm called SFLA-GA that combines shuffled frog leaping and genetic algorithms. Ye, et al. [42] presented an energy-efficient KnEA (EEKnEA) algorithm to minimize energy consumption and maximize load balance, resource utilization, and robustness in Cloud data centres. Le Berre, et al. [43] maximize the coverage of the area under time, the network's lifetime depending on coverage while minimizing the financial cost of the wireless sensor network (WSN). Sun, et al. [44] presented a new algorithm called complex alliance strategy with multi-objective optimization of coverage (CASMOC), which could effectively improve node coverage. Furthermore, they also presented the proportional connection of the energy conversion function between the working node and its neighbours, which he uses to schedule low-energy mobility nodes, resulting in network energy balance and resource optimization. However, regarding the competing characteristics of multi-tenant environments in cloud computing, Wei, et al. [45] suggested a cloud resource allocation model based on an imperfect information Stackelberg game (CSAM-IISG) in a cloud computing context utilizing a hidden Markov model (HMM). Dörterler, et al. [46] solved the virtual machine placement (VMP) problems by optimizing CPU utilization while minimizing energy consumption; the authors employed and compared four prominent multi-objective algorithms' performance on CloudSim software. Li, et al. [47] proposed an intermediary framework with multiple cloud environments to provide low-cost streaming big data computing services. A cloud service intermediary rents cloud resources from multiple providers and provide streaming processing services to users via multiple service interfaces. Butun, et al. [48] suggested and demonstrated the success of a cloud-centric, multi-level authentication as a service strategy that addresses scalability and time constraints. Rana, et al. [49] proposed a vital agreement scheme for fog supported IoT environment to ensure accountability and privacy. Pandey, et al. [50] demonstrated a particle swarm optimization (PSO) algorithm for scheduling cloud applications that accounts for computing and data transmission costs. Yadav, et al. [51] presented a modified fireworks algorithm combining opposition-based learning and differential evolution techniques to minimize makespan and cost and improve resource utilization. Rakrouki and Alharbe [52] introduced a technique for analyzing the current status of running tasks based on quality of service (QoS) predictions provided by an ARIMA model with a Kalman filter. Because workflow scheduling becomes increasingly complicated as the number of activities and virtual machines grows, it is considered an NP-hard problem; Arora and Banyal [53] proposed a new hybrid approach PSO–GWO algorithm to minimize the average total execution cost and average total execution time. Furthermore, Singh, et al. [54] developed a model and system approach to increase cloud resource and service availability for cloud users. The proposed model is then compared to the current model using Weibull and exponential distributions and quantitative and graphical analyses on test cycles. They demonstrated that the suggested approach provides high availability of cloud resources, ensuring the highest levels of reliability and availability for cloud users. Prakash and Budihal [55] provided high availability at instance and storage level in the cloud environment. Sun, et al. [56] proposed a dynamic data replication strategy with a brief survey of replication strategies suitable for distributed computing environments, including analyzing and modelling the relationship between system availability and the number of replicas. Apduhan, et al. [57] investigated IaaS service provisioning in hybrid cloud, which comprises private and public clouds, proposing a hybrid cloud framework to ameliorate the reliability and availability of IaaS services considering alternative services available through public clouds. S, et al. [58] proposed a novel algorithm for three tier cloud architecture to reduce the average execution time of the user tasks thereby increasing the machine availability time which finally leads to the uniform distribution of the workload in the cloud infrastructure. Mesbahi, et al. [59] proposed a ‘Reference Roadmap’ of reliability and high availability in cloud computing environments. Ma, et al. [60] proposed an efficient and effective approach to the problem of application deployment in multi-cloud to minimize overall deployment costs and response time by employing a hybrid NSGA-II approach with a local search method. The authors declared that experimental evaluations with benchmark datasets demonstrate that our proposed hybrid approach outperforms NSGA-II and SPEA2. Guerrero, et al. [61] asserted that most papers implemented a classic single-objective GA or NSGA-II in designing genetic-based optimizations that consider the emerging evolutions of fog computing, such us osmotic computing and service adaptation. Finally, Jafari and Rezvani [62] formulated the problem of joint optimization of energy consumption and latency in the form of a multi-objective problem and solve it using the non-dominant sorting genetic algorithm (NSGA-II) and Bees algorithm (BA). The simulation results show that NSGA-based methods have remarkable robustness compared to BA-based methods in terms of significant criteria such as energy consumption, time delay.

## 3. Black Widow Optimization (BWO) Algorithm

### *3.1. Inspiration*

The Black widow is nocturnal, with the female remaining inconspicuous during the day and spinning her web at night. In most cases, a female widow stays in the same region for most of her adult life. When a black female widow wants to mate, she sprays pheromone on selected net parts to attract the male. The first male to join the web reduces the attractiveness of females to competitors by reducing their web size. The female eats the male during or after mating and then transports the eggs to her egg sock. The offspring participates in sibling cannibalism after hatching the egg. The fit and powerful individuals survive as a result of this cycle that best one is the global optimum of the objective function. However, the male of the black widow spider is one of just two known creatures in which the male actively assists the female in sexual cannibalism. The female consumes the male entirely in around two out of every three cases during mating. Furthermore, for many days to a week, black widow spiderlings reside together on the maternal web, during which time sibling cannibalism is most prevalent. As a result, cannibalism is probably linked to demography and could have enormous consequences on the population. Density-dependent cannibalism may control population size, which may be essential in black widow spider populations. Cannibalism lowers the number of surviving spiderlings, although it may improve parental fitness if survivors are in better physical shape.

### *3.2. Mathematical Model*

The BWO algorithm commences with an initial population of spiders, demonstrating the potential solution candidates, considered a Black widow spider in the search space. A candidate widow matrix of size $N_{pop} \times N_{var}$ is generated with an initial population of spiders. These first spiders strive to reproduce the next generation in pairs. During or after mating, the female black widow consumes the male. She then transports the sperm she has stored in her sperm thecae to the egg sacs and releases them. Spiderlings emerge from egg sacs as early as 11 days after being laid. For many days to a week, they cohabitate on the maternal web, during which sibling cannibalism is seen. They are then swept away by the wind. Subsequently, the offspring are produced and repeated for $N_{var}/2$ times, which are mathematically presented as follows:

$$\begin{cases} y_1 = \alpha \times x_1 + (1-\alpha) \times x_2 \\ y_2 = \alpha \times x_2 + (1-\alpha) \times x_1 \end{cases} \quad (1)$$

Where $x_1$ and $x_2$ show parents; $y_1$ and $y_2$ elucidate offspring [35].

In the following step of the algorithm, cannibalism occurs in two miscellaneous stages: sexual cannibalism and sibling cannibalism. In the former, the female black widow consumes her spouse during or after mating, and in the latter, weak spiderlings consume weaker spiderlings.
Finally, the mutation stage occurred in the BWO algorithm, in which the Mutepop number of individuals from the population was chosen. Each of the selected solutions exchanges two members in the array at random. The mutation rate is used to compute Mutepop. However, in Fig. 2, the pseudo-code of the BWO algorithm is provided, and Fig. 3 presents the flowchart of this algorithm.

The "Big O notation," a prominent mathematical notation in science and mathematics, could be used to do a computational complexity study for a metaheuristic algorithm. For comparison, the algorithms' needed execution time and memory use were assessed. It should be mentioned that setting numerical values for an algorithm's complexity for testing is commonplace; nevertheless, finding a solution for analyzing run time concerns in such algorithms is another issue that should be considered. Other complexity procedures should be used in this case, as an algorithm's complexity can be described irrespective of computer or hardware constraints. "Big O notation" is a phrase used in computer science to explain the needed run time and memory use of algorithms, which are calculated for comparison reasons. Initially, the overall number of initial solution candidates is calculated (NP), and D is the dimension of the problem under consideration. The computational complexity of BWO's initialization phase is defined as O(NP×D), and the computational complexity of the objective function evaluation phase is computed as O(NP)×O(F(x)), where F(x) is the problem's objective function. The computational complexity of each line in BWO's primary search loop is equivalent to the number of iterations (MxIter).

Regarding female black widow eats the male during or after matin in the web, the position updating procedure for each solution candidate in the search space has a computational cost of O(MxIter×NP×D×2). Meanwhile, the mutation operator requires O(NP×D) time, and the crossover operator requires O(NP×D) time. Ultimately, the evaluation of the objective function in the BWO's primary search loop has a computational cost of O(MxIter×NP×D×2)×O(F(x)).

***Input:*** *Maximum number of iterations, rate of procreating, rate of cannibalism, rate of mutation*

***Output:*** *near-optimal solution for the objective function*

*// initilazation*

*1. The initial population of black widow spiders*

*Each pop is a D-dimensional array of chromosomes for a D-dimensional problem*

*// Loop until the terminal condition*

*1. Based on procreating rate, calculate the number of reproduction "nr";*
*2. Select the best nr solutions in pop and save them in pop1 ;*
*// Procreating and cannibalism*
*3. For i=1 to nr, do*
*4. Randomly select two solutions as parents from pop1 ;*
*5. Generate D children using equation1 ;*
*6. Destroy father ;*
*7. Based on the cannibalism rate, destroy some of the children (newly achieved solutions) ;*
*8. Save the remaining solutions into pop2 ;*
*9. End for*
*// Mutation*
*10. Based on the mutation rate, calculate the number of mutation children "nm";*
*11. For i=1 to nm, do*
*12. Select a solution from pop1 ;*
*13. Mutate randomly one chromosome of the solution and generate a new solution ;*
*14. Save the new one into pop3 ;*
*15. End for*
*// Updating*
*16. Update pop = pop2+pop3 ;*
*17. Returning the best solution ;*
*18. Return the best solution from pop ;*

Fig. 2. Pseudo-code of BWO [35].

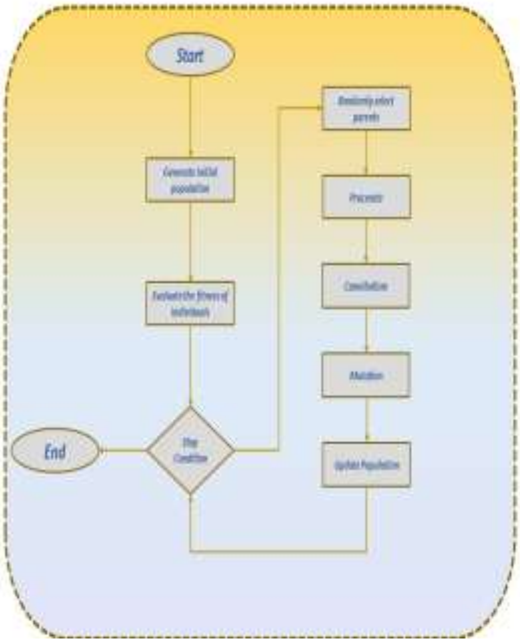


Fig.3. Flowchart of BWO [35].

## 3.Statement of the Optimization Problem

An optimization problem is one in which the objective is to discover the optimal solution among all possible options. A typical optimization problem is as follows:

- A function $f : B \to \mathbb{R}$ from some set B to the real numbers.
- An element $x_0 \in B$ such that $f(x_0) \leq f(x)$ for all $x \in B$ (minimization problem) or $f(x_0) \geq f(x)$ for all $x \in B$ (maximization problem).

Typically, *B* is a subset of Euclidean space, often characterized by a set of constraints, equalities, or inequalities that *B* members must meet. The search space or choice set of *f* is denoted by the domain *B*, and candidate solutions or viable solutions indicate the components of *B*. The term "objective function" refers to function *f*. An optimum solution is a possible solution that minimizes (or maximizes, if that is the aim) the objective function [63]. Table 1 shows the parameter settings of the BWO algorithm. Furthermore, all tests to evaluate the BWO's performance were conducted with 50 populations using a PC with the detailed parameters shown in Table 1, and 20 independent optimization runs were performed. In the current work, the model consists of N customers, a cloud broker, and M cloud service provider, as shown in Fig. 4. However, the cloud server must find the best configuration between customers and cloud service providers. Set $U = u_1, u_2, ..., u_N$ was used to identify the N customer, and the set $S = s_1, s_2, ..., s_M$ was used to identify the M cloud service provider in the proposed model. Each cloud service provider has a limited capacity to process customer requests, and the total number of requests made to a cloud service provider must exceed the number of requests from one customer. a binary variable called $b_{ij}$ to describe the process of a cloud server is used that is formulated in Eq. 2 as follows [64]:

$$b_{ij} = \begin{cases} 1, & if\ s_j\ handle\ the\ request\ from\ u_i \\ 0, & otherwise, \end{cases} \qquad (2)$$
$$i = 1, 2, ..., N \qquad j = 1, 2, ..., M$$

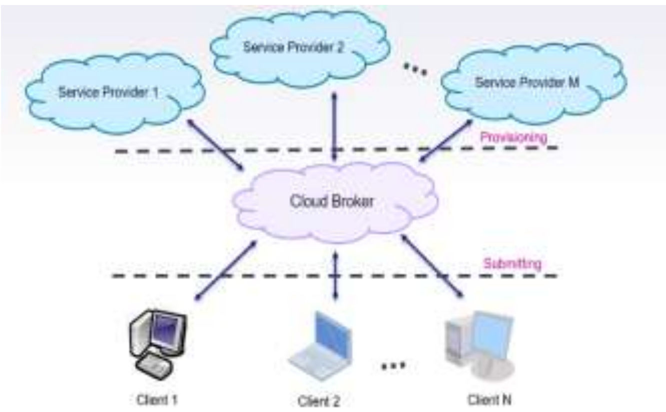


Fig. 4. Cloud model in the current study

Table 1

Details of the utilized system in the optimization process and the BWO parameters in the current study

| ***Name*** | ***Feature*** | ***Specification*** |
|---|---|---|
| Hardware | CPU<br>Frequency<br>RAM<br>Hard drive | CORE i7<br>2.8 GHz<br>8 GB<br>2 TB |
| Software | Operating system<br>Language | Windows 10<br>MATLAB R2020 |
| BWO Algorithm | pp<br>CR<br>Pm | 0.6<br>0.44<br>0.4 |
| *Pp: procreate rate; CR: cannibalism rate; Pm: mutation rate* | | |

Customers would also expect their requests to be processed in the shortest possible time when they send them to the cloud brokers and servers. Hence, *RT* is considered as the response time of requests and $L_{ij}$ is also considered as the delay between *ith* client and *jth* cloud server. This delay can be achieved using $L_{ij} = CT - AT$ , where *CT* is the current time and *AT* is the time that *ith* client's request reaches the *jth* cloud server. When the cloud server receives a request from a customer, it has to spend $T_j$ to execute the request. Consequently, the following objective function is to minimize the response time of requests [64]:

$$RT = \sum_{i=1}^{N} \sum_{j=1}^{M} b_{ij} (L_{ij} + T_j) \qquad (3)$$

When a customer submits their request to the cloud server through a cloud broker, the cloud broker manages the requests to find the best solution for customer satisfaction, and the cloud broker is also expected to make a profit. Therefore, the cloud broker's profit is also the second goal. However, $P_i$ is the price received from the ith customer, and Cj is the *jth* cloud server cost. Mathematically speaking, the following objective function is to maximize the cloud broker's profit [64]:

$$P = \sum_{i=1}^{N} \sum_{j=1}^{M} b_{ij} (P_i - C_j) \qquad (4)$$

To execute the request received from the customer, the cloud surveyor has to do the relevant work with the least energy consumption. Therefore, energy consumption is considered as another goal due to its importance in cloud computing. $E_j$ is assumed to be the amount of energy the *jth* cloud server uses to perform the task. Consequently, the third objective function for energy minimization can be formulated as Eq.5 [64]:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{M} b_{ij}.E_j \tag{5}$$

On the other hand, cloud services should always be available to users 24 hours a day. In a cloud environment, the constant availability of cloud services is a paramount concern for users and cloud servers. Therefore, maximizing the availability of cloud services provided by cloud servers is this research's next and fundamental goal. In a cloud computing environment, availability is the percentage of time cloud servers are available to their customers or users (to perform their tasks). Hence, system availability is obtained based on cloud servers using Eq.6 as follows [54]:

$$A_{sys} = \prod_{i=1}^{M} A_i \tag{6}$$

Where $A_i$ is the availability of the *ith* server obtained by Eq.7 [54]:

$$A = \frac{MTBF}{MTBF + MTTR} \tag{7}$$

Where *MTBF* shows the mean time between failure; and *MTTR* demonstrates the mean time to repair, which are calculated by Eqs.8 and 9, respectively [54]:

$$MTBF = \frac{Total\ Uptime}{number\ of\ \ failure} \tag{8}$$

$$MTTR = \frac{Total\ Downtime}{number\ of\ \ failure} \tag{9}$$

Nonetheless, considering the mentioned statements and a single-objective optimization problem, four objective functions are used to determine the utility function of the cloud broker as follows:

$$U = \omega_1.RT + \omega_2.E + \omega_3.P^{-1} + \omega_4.A^{-1} \tag{10}$$

Where $\omega_1$, $\omega_2$, $\omega_3$, and $\omega_4$ are the weighting factors of the response time of requests, the profit of the cloud broker, the system's total energy consumption, and the system's availability, respectively that their summations equal one ($\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$). *RT* shows the response time of requests of the system calculated by Eq. 3. *P* is the profit of the cloud broker that is calculated utilizing Eq. 4. *E* demonstrates the system's total energy consumption obtained by Eq. 5. *A* elucidates of the system's availability were identified by Eq. 6. Consequently, the optimization problem of the cloud broker is to minimize the utility function as follows [64]:

$$Min\ U = \omega_1 RT + \omega_2 E + \omega_3 P^{-1} + \omega_4 A^{-1} \tag{11}$$

$$Subject\ to \sum_{i=1}^{N} x_{ij} \le A_j \tag{12}$$

$$\sum_{j=1}^{M} b_{ij} \ge R_i \tag{13}$$

$$A_j, R_i \ge 0; \qquad i = 1, \dots, N; j = 1, \dots, M \tag{14}$$

Where $R_i$ is the total number of requests received from customer *i*; $A_j$ denotes the service provider *j's* capacity. If the total number of customer requests to service provider *j* is fewer than or equal to the capacity of service provider *j*, constraint (12) is fulfilled. Constraint (13) then determines if the total number of requests from client *i* to all service providers is higher than or equal to the total number of requests from client *i*. Finally, constraint (14) determines if the number of requests and capacity are positive. Meanwhile, a normalization procedure is conducted because the units of variables are not the same. Normalization means the scaling down of the data set such that the normalized data falls in the range between 0 and 1. Mathematically, the Normalization equation is represented in Eq. 15:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{15}$$

However, The simulated cloud brokering is expected to identify the best deals between five customers and seven service providers with the most profit for the cloud broker and highest availability during the shortest response time and the lowest cloud energy usage. Small, large, and extra-large are the three sorts of instances. The kind of Instance determines an example's average execution time and price. The parameters utilized in the cloud brokering system are listed in Table 2.

Table 2
Cloud Brokering Parameters

| ***Parameters*** | ***Value*** |
|---|---|
| Instance Type | small, large, extra-large |
| Average execution time(s) | 980 ± 71, 616 ± 61, 697 ± 13 |
| Hourly price | 0.1, 0.125, 0.143 |
| Instance prices | Amazon EC2 pricing history |

## 4.Result and Discussion

To evaluate the capability of the proposed methodology in the current paper, all outcomes are compared to that obtained by Kumrai, et al. [64] using the PSO algorithm. It is noteworthy that seeing that the PSO algorithm could outperform the nondominated sorting GA-II (NSGA-II) algorithm in the mentioned study, the current study considers the results of the PSO algorithm for comparison purposes. However, four different scenarios have been considered to evaluate the proposed approach. In the first scenario, the proposed approach was evaluated regarding the response time of requests. For this purpose, the weight coefficient of the response time of requests ($\omega_1$) is equal to one, and the other weight coefficients are equal to zero. In other words, only the response time of requests' parameters will be used in the objective function. Fig. 5 shows the result of the response time of requests. As can be inferred from Fig. 5, it is clear that the BWO algorithm can outperform the PSO algorithm in dealing with reducing the response time of requests from customers. Furthermore, after 200 iterations during the optimization process, the BWO algorithm could calculate the lower value of the response time of requests rather than that of the PSO algorithm, accounting for approximately 0.1 (s); consequently, the BWO algorithm could be deemed as an appropriate algorithm to minimize the response time of requests. Furthermore, the BWO algorithm could converge to the optimum result as quickly as the PSO algorithm in the first scenario.
Furthermore, the proposed approach concerned energy consumption in the second scenario. For this purpose, the weight coefficient of energy consumption is equal to one, and the other weight coefficients are equal to zero. Only the energy consumption parameter has been deemed in the objective function. Fig. 6 elucidates the energy consumption considering the BWO and PSO algorithms; the BWO algorithm shows more rapid convergence behaviour than the PSO algorithm in the second scenario, indicating its proper performance. Moreover, the BWO algorithm has the lowest energy consumption, registered at slightly less than 0.1. In stark contrast, at the end of the 200th iteration, the PSO algorithm found 0.3 for energy consumption more than BWO in the second scenario. The profit of the cloud broker is shown in Fig. 7. It is understood that the BWO algorithm could outperform the PSO algorithm in the third scenario, reaching roughly 0.98, while the PSO algorithm reaches precisely 0.8 at the end of the $200^{th}$ iteration. In other words, the BWO algorithm could be deemed superior optimization to provide better profit for cloud brokers. Finally, Fig. 8 shows the result of system availability; preferable system availability is provided by the BWO algorithm than the PSO algorithm, indicating its superior performance in the fourth scenario. After 200 iterations, the BWO algorithm could reach approximately 0.98 for the fourth scenario, while the PSO could calculate much less value for the system availability. Overall, the proposed objective function and the BWO algorithm provide a much better and more reasonable solution than previous methods and algorithms.

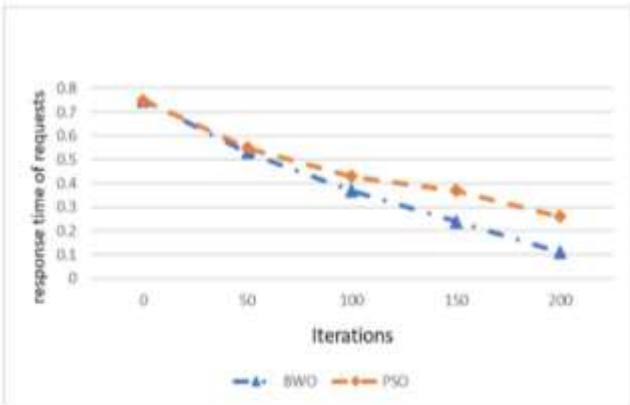


Fig. 5. The results of the response time of requests

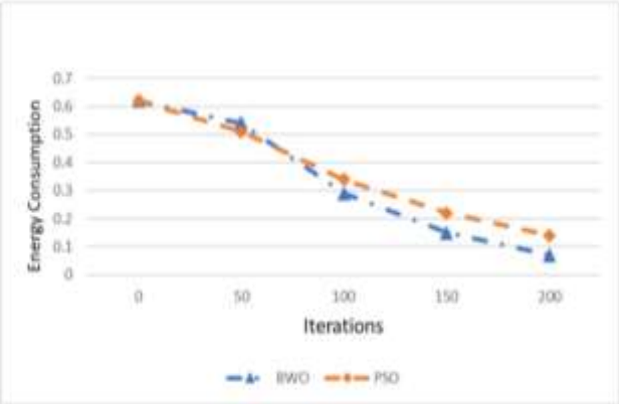


Fig. 6. The results of energy consumption

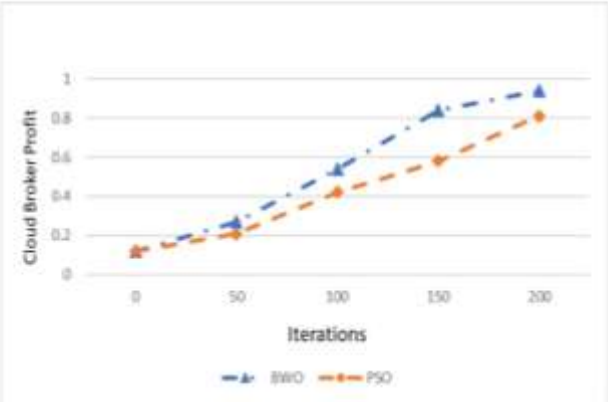


Fig. 7. The results of cloud broker profit

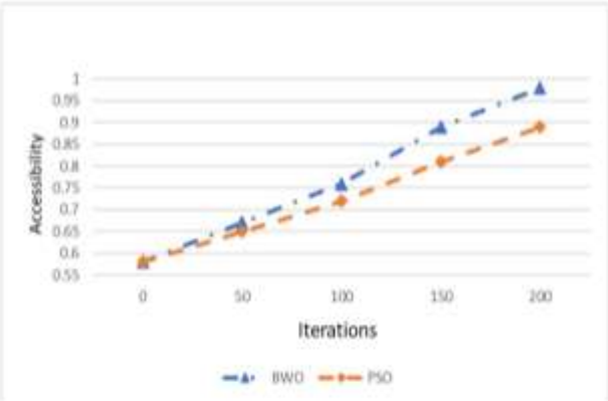


Fig. 8. The results of system availability

Fig. 9 shows the convergence history of the BWO algorithm in dealing with the mentioned problems. As can be seen, the BWO algorithm could converge to the optimum solution, which in the 129$^{th}$ iteration, the BWO algorithm could reach the optimum result. In other words, the proposed approach for the cloud brokerage system has a good convergence speed and has been able to converge to the optimal solution. However, since, like other metaheuristic algorithms, the BWO is an approximate algorithm, the stability of this algorithm should be analyzed in 20 independent optimization runs. Fig. 10 shows the convergence history of 20 independent optimization runs for the BWO. Examination of the stability test results shows the excellent stability of the proposed approach, and the algorithm is converged to the optimal solution. To better evaluate the stability test result, the standard deviation of the objective functions obtained from 20 times the algorithm has been calculated, equal to 0.0029. Given that this value is quite close to zero, it is concluded that the 20 values obtained from the 20 times the algorithm is executed are pretty close to each other, which means a high level of stability for the proposed approach.

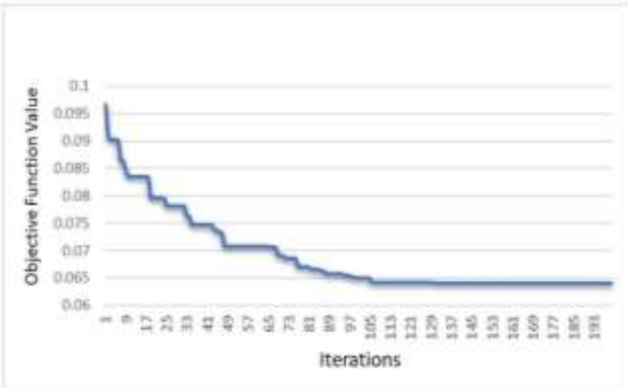


Fig. 9. convergence history of the BWO algorithm

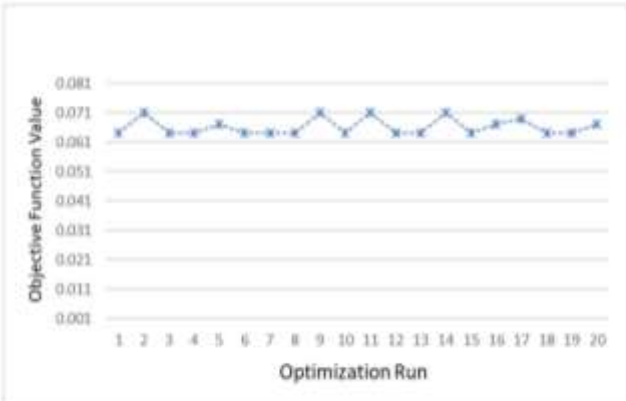


Fig. 10. convergence history of 20 independent optimization runs for the BWO

## 5.Conclusion

The proposed algorithm tries to provide the most appropriate proposal for users' requests from cloud servers to maximize the profit of the cloud server and system availability and minimize response time and energy consumption. The simulation was implemented in MATLAB software to evaluate the proposed approach, and the obtained results were compared to that of the PSO algorithm. Assess the results obtained from the simulation of the proposed approach indicates that it performs better than the particle swarm optimization algorithm in terms of all four parameters of cloud server profit, system availability, response time and energy consumption. The experimental results also showed that the proposed approach is at a perfect level of convergence and stability. Future research should evaluate the mentioned problems using different metaheuristic algorithms such as the Grey wolf optimization algorithm (GWO) and Chaos Game Optimization (CGO).

## References


[1] [1] S. Li, L.D. Xu, S. Zhao, The internet of things: a survey, Information systems frontiers, 17(2) (2015) 243-259.

[2] [2] S.M. Besen, The European telecommunications standards institute: A preliminary analysis, Telecommunications policy, 14(6) (1990) 521-530.

[3] [3] S. Samal, B. Acharya, P.K. Barik, Internet of Things (IoT) in agriculture toward urban greening, in: AI, Edge and IoT-based Smart Agriculture, Elsevier, 2022, pp. 171-182.

[4] [4] A. Rejeb, Z. Suhaiza, K. Rejeb, S. Seuring, H. Treiblmaier, The Internet of Things and the circular economy: A systematic literature review and research agenda, Journal of Cleaner Production, (2022) 131439.

[5] [5] K. Rose, S. Eldridge, L. Chapin, The internet of things: An overview, The internet society (ISOC), 80 (2015) 1-50.

[6] [6] D.K. Sharma, S. Bhargava, K. Singhal, Internet of Things applications in the pharmaceutical industry, in: An Industrial IoT Approach for Pharmaceutical Industry Growth, Elsevier, 2020, pp. 153-190.

[7] [7] I. Boussaïd, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Information Sciences, 237 (2013) 82-117. https://doi.org/https://doi.org/10.1016/j.ins.2013.02.041

[8] [8] B. Galván, D. Greiner, J. Periaux, M. Sefrioui, G. Winter, Parallel Evolutionary Computation for solving complex CFD Optimization problems: a review and some nozzle applications, Parallel Computational Fluid Dynamics 2002, (2003) 573-604.

[9] [9] J.H. Holland, Genetic Algorithms and Adaptation, in: O.G. Selfridge, E.L. Rissland, M.A. Arbib (Eds.) Adaptive Control of Ill-Defined Systems, Springer US, Boston, MA, 1984, pp. 317-333.

[10] [10] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, Journal of Global Optimization, 11(4) (1997) 341-359. https://doi.org/10.1023/A:1008202821328

[11] [11] B.M. Angadi, M.S. Kakkasageri, S.S. Manvi, Computational intelligence techniques for localization and clustering in wireless sensor networks, in: Recent Trends in Computational Intelligence Enabled Research, Elsevier, 2021, pp. 23-40.

[12] [12] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39-43.

[13] [13] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1) (1996) 29-41. https://doi.org/10.1109/3477.484436

[14] [14] Z.E. Ahmed, R.A. Saeed, A. Mukherjee, S.N. Ghorpade, 10 - Energy optimization in low-power wide area networks by using heuristic techniques, in: B.S. Chaudhari, M. Zennaro (Eds.) LPWAN Technologies for IoT and M2M Applications, Academic Press, 2020, pp. 199-223.

[15] [15] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Advances in Engineering Software, 95 (2016) 51-67. https://doi.org/https://doi.org/10.1016/j.advengsoft.2016.01.008

[16] [16] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Advances in Engineering Software, 69 (2014) 46-61. https://doi.org/https://doi.org/10.1016/j.advengsoft.2013.12.007

[17] [17] M. Azizi, S. Talatahari, A. Giaralis, Active Vibration Control of Seismically Excited Building Structures by Upgraded Grey Wolf Optimizer, IEEE Access, 9 (2021) 166658-166673.

[18] [18] L. Xie, J. Zeng, Z. Cui, General framework of artificial physics optimization algorithm, in: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE, 2009, pp. 1321-1326.

[19] [19] M. Azizi, Atomic orbital search: A novel metaheuristic algorithm, Applied Mathematical Modelling, 93 (2021) 657-683. https://doi.org/https://doi.org/10.1016/j.apm.2020.12.021

[20] [20] M. Azizi, S. Talatahari, A. Giaralis, Optimization of Engineering Design Problems Using Atomic Orbital Search Algorithm, IEEE Access, 9 (2021) 102497-102519. https://doi.org/10.1109/ACCESS.2021.3096726

[21] [21] S. Talatahari, M. Azizi, A.H. Gandomi, Material generation algorithm: a novel metaheuristic algorithm for optimization of engineering problems, Processes, 9(5) (2021) 859.

[22] [22] M. Azizi, M.B. Shishehgarkhaneh, M. Basiri, Optimum design of truss structures by Material Generation Algorithm with discrete variables, Decision Analytics Journal, (2022) 100043.

https://doi.org/https://doi.org/10.1016/j.dajour.2022.100043
[23] [23] O.K. Erol, I. Eksin, A new optimization method: big bang–big crunch, Advances in Engineering Software, 37(2) (2006) 106-111.
[24] [24] Ş.İ. Birbil, S.-C. Fang, An electromagnetism-like mechanism for global optimization, Journal of global optimization, 25(3) (2003) 263-282.
[25] [25] N. Khodadadi, M. Azizi, S. Talatahari, P. Sareh, Multi-Objective Crystal Structure Algorithm (MOCryStAl): Introduction and Performance Evaluation, IEEE Access, 9 (2021) 117795-117812. https://doi.org/10.1109/ACCESS.2021.3106487
[26] [26] M. Azizi, S. Talatahari, P. Sareh, Design optimization of fuzzy controllers in building structures using the crystal structure algorithm (CryStAl), Advanced Engineering Informatics, 52 (2022) 101616. https://doi.org/https://doi.org/10.1016/j.aei.2022.101616
[27] [27] B. Talatahari, M. Azizi, S. Talatahari, M. Tolouei, P. Sareh, Crystal structure optimization approach to problem solving in mechanical engineering design, Multidiscipline Modeling in Materials and Structures, (2022).
[28] [28] S. Talatahari, M. Azizi, M. Tolouei, B. Talatahari, P. Sareh, Crystal Structure Algorithm (CryStAl): A Metaheuristic Optimization Method, IEEE Access, 9 (2021) 71244-71261. https://doi.org/10.1109/ACCESS.2021.3079161
[29] [29] S. Talatahari, M. Azizi, Chaos Game Optimization: a novel metaheuristic algorithm, Artificial Intelligence Review, 54(2) (2021) 917-1004. https://doi.org/10.1007/s10462-020-09867-w
[30] [30] M. Azizi, U. Aickelin, H.A. Khorshidi, M.B. Shishehgarkhaneh, Shape and size optimization of truss structures by Chaos game optimization considering frequency constraints, Journal of Advanced Research, (2022). https://doi.org/https://doi.org/10.1016/j.jare.2022.01.002
[31] [31] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Computer-aided design, 43(3) (2011) 303-315.
[32] [32] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, simulation, 76(2) (2001) 60-68.
[33] [33] M. Azizi, S. Talatahari, M. Basiri, M.B. Shishehgarkhaneh, Optimal design of low-and high-rise building structures by Tribe-Harmony Search algorithm, Decision Analytics Journal, (2022) 100067.
[34] [34] S.-A. Ahmadi, Human behavior-based optimization: a novel metaheuristic approach to solve complex optimization problems, Neural Computing and Applications, 28(1) (2017) 233-244. https://doi.org/10.1007/s00521-016-2334-4
[35] [35] V. Hayyolalam, A.A. Pourhaji Kazem, Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems, Engineering Applications of Artificial Intelligence, 87 (2020) 103249. https://doi.org/https://doi.org/10.1016/j.engappai.2019.103249
[36] [36] Y. Kessaci, N. Melab, E.-G. Talbi, A Pareto-based metaheuristic for scheduling HPC applications on a geographically distributed cloud federation, Cluster Computing, 16(3) (2013) 451-468.
[37] [37] H.K. Mehta, P. Pawar, P. Kanungo, A two level broker system for infrastructure as a service cloud, Wireless Personal Communications, 90(3) (2016) 1135-1147.
[38] [38] K.S. Yildirim, T.E. Kalayci, A. Ugur, Optimizing coverage in a k-covered and connected sensor network using genetic algorithms, in: Proceedings of the 9th WSEAS international conference on evolutionary computing, Citeseer, 2008, pp. 21-26.
[39] [39] M. Elhoseny, A. Abdelaziz, A. Salama, A.e.-d. Riad, K. Muhammad, A. Kumar, A hybrid model of Internet of Things and cloud computing to manage big data in health services applications, Future Generation Computer Systems, 86 (2018). https://doi.org/10.1016/j.future.2018.03.005
[40] [40] J. Mei, K. Li, Z. Tong, Q. Li, K. Li, Profit maximization for cloud brokers in cloud computing, IEEE Transactions on Parallel and Distributed Systems, 30(1) (2018) 190-203.
[41] [41] P. Asghari, A. Rahmani, H. Haj Seyyed Javadi, Privacy-aware cloud service composition based on QoS optimization in Internet of Things, Journal of Ambient Intelligence and Humanized Computing, (2020). https://doi.org/10.1007/s12652-020-01723-7
[42] [42] X. Ye, Y. Yin, L. Lan, Energy-efficient many-objective virtual machine placement optimization in a cloud computing environment, IEEE Access, 5 (2017) 16006-16020.
[43] [43] M. Le Berre, F. Hnaien, H. Snoussi, Multi-objective optimization in wireless sensors networks, in: ICM 2011 Proceeding, IEEE, 2011, pp. 1-4.
[44] [44] Z. Sun, Y. Zhang, Y. Nie, W. Wei, J. Lloret, H. Song, CASMOC: a novel complex alliance strategy with multi-objective optimization of coverage in wireless sensor networks, wireless Networks, 23(4) (2017) 1201-1222.
[45] [45] W. Wei, X. Fan, H. Song, X. Fan, J. Yang, Imperfect information dynamic stackelberg game based resource allocation using hidden Markov for cloud computing, IEEE transactions on services computing, 11(1) (2016) 78-89.
[46] [46] S. Dörterler, M. Dörterler, S. Ozdemir, Multi-objective virtual machine placement optimization for cloud computing, in: 2017 International Symposium on Networks, Computers and Communications (ISNCC), IEEE, 2017, pp. 1-6.
[47] [47] H. Li, M. Dong, K. Ota, M. Guo, Pricing and repurchasing for big data processing in multi-clouds, IEEE Transactions on Emerging Topics in Computing, 4(2) (2016) 266-277.
[48] [48] I. Butun, M. Erol-Kantarci, B. Kantarci, H. Song, Cloud-centric multi-level authentication as a service for secure public safety device networks, IEEE Communications Magazine, 54(4) (2016) 47-53.
[49] [49] S. Rana, D. Mishra, R. Arora, Privacy-Preserving Key Agreement Protocol for Fog Computing Supported Internet of Things Environment, Wireless Personal Communications, 119(1) (2021) 727-747. https://doi.org/10.1007/s11277-021-08234-4

[50] [50] S. Pandey, L. Wu, S.M. Guru, R. Buyya, A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments, in: 2010 24th IEEE international conference on advanced information networking and applications, IEEE, 2010, pp. 400-407.
[51] [51] A.M. Yadav, K.N. Tripathi, S.C. Sharma, An enhanced multi-objective fireworks algorithm for task scheduling in fog computing environment, Cluster Computing, 25(2) (2022) 983-998. https://doi.org/10.1007/s10586-021-03481-3
[52] [52] M.A. Rakrouki, N. Alharbe, QoS-Aware Algorithm Based on Task Flow Scheduling in Cloud Computing Environment, Sensors, 22(7) (2022) 2632.
[53] [53] N. Arora, R.K. Banyal, A Particle Grey Wolf Hybrid Algorithm for Workflow Scheduling in Cloud Computing, Wireless Personal Communications, 122(4) (2022) 3313-3345. https://doi.org/10.1007/s11277-021-09065-z
[54] [54] H. Singh, S. Tyagi, P. Kumar, High availability and accessibility of services in cloud environment, in: 2018 4th International Conference on Computing Sciences (ICCS), IEEE, 2018, pp. 67-71.
[55] [55] M. Prakash, R. Budihal, IMPROVING HIGH AVAILABILITY IN CLOUD INFRASTRUCTURE AT INSTANCE AND STORAGE LEVEL, Australian Journal of Wireless Technologies, Mobility and Security, 1(1) (2017) 1-7.
[56] [56] D.-W. Sun, G.-R. Chang, S. Gao, L.-Z. Jin, X.-W. Wang, Modeling a dynamic data replication strategy to increase system availability in cloud computing environments, Journal of computer science and technology, 27(2) (2012) 256-272.
[57] [57] B. Apduhan, M. Younas, T. Uchibayashi, Improving reliability and availability of Iaas services in hybrid clouds, in: International Conference on Computational Science and Its Applications, Springer, 2015, pp. 557-568.
[58] [58] S. S, R. J, H.S. Guruprasad, Enhanced Load Balancing Algorithm in Three-Tier Cloud Computing, International Journal of Engineering Sciences & Emerging Technologies, 2 (2014) 296-301.
[59] [59] M.R. Mesbahi, A.M. Rahmani, M. Hosseinzadeh, Reliability and high availability in cloud computing environments: a reference roadmap, Human-centric Computing and Information Sciences, 8(1) (2018) 1-31.
[60] [60] H. Ma, A.S.d. Silva, W. Kuang, NSGA-II with Local Search for Multi-objective Application Deployment in Multi-Cloud, in: 2019 IEEE Congress on Evolutionary Computation (CEC), 2019, pp. 2800-2807.
[61] [61] C. Guerrero, I. Lera, C. Juiz, Genetic-based optimization in fog computing: Current trends and research opportunities, Swarm and Evolutionary Computation, 72 (2022) 101094. https://doi.org/https://doi.org/10.1016/j.swevo.2022.101094
[62] [62] V. Jafari, M.H. Rezvani, Joint optimization of energy consumption and time delay in IoT-fog-cloud computing environments using NSGA-II metaheuristic algorithm, Journal of Ambient Intelligence and Humanized Computing, (2021). https://doi.org/10.1007/s12652-021-03388-2
[63] [63] M. Azizi, A. Mousavi, R. Ejlali, S. Talatahari, Optimum design of fuzzy controller using hybrid ant lion optimizer and Jaya algorithm, Artificial Intelligence Review, 53 (2020) 1-32. https://doi.org/10.1007/s10462-019-09713-8
[64] [64] T. Kumrai, K. Ota, M. Dong, J. Kishigami, D.K. Sung, Multiobjective optimization in cloud brokering systems for connected Internet of Things, IEEE Internet of Things Journal, 4(2) (2016) 404-413.