future internet

MDPI

Article

Improving Quality Indicators of the Cloud-Based IoT Networks Using an Improved Form of Seagull Optimization Algorithm

Hamza Mohammed Ridha Al-Khafaji

Biomedical Engineering Department, Al-Mustaqbal University College, Hillah 51001, Babil, Iraq; hamza.alkhafaji@uomus.edu.iq

Abstract: The Internet of things (IoT) points to billions of devices located worldwide which are connected and share their data based on the Internet. Due to the new technologies that provide cheap computer chips and universal wireless networks, it is feasible that everything from a small tablet to a very large airplane will be connected to the Internet and will be a part of the IoT. In most applications, IoT network nodes face limitations in terms of energy source and cost. Therefore, the need for innovative methods to improve quality indicators that increase the lifespan of networks is evident. Here, a novel technique is presented to increase the quality of service (QoS) in IoT using an improved meta-heuristic algorithm, called the improved seagull optimization algorithm (ISOA), along with traffic management in these networks. Based on this subject, the traffic-aware algorithm can manage the sending of packets and increase the QoS provision in terms of time to a great extent. The performance evaluation of the proposed method and comparison with the previous methods demonstrated the accuracy and efficiency of this method and its superiority over the previous works.

Keywords: IoT; ISOA; QoS; quality indicators

Citation: Al-Khafaji, H.M.R. Improving Quality Indicators of the Cloud-Based IoT Networks Using an Improved Form of Seagull Optimization Algorithm. *Future Internet* **2022**, *14*, 281. https://doi.org/10.3390/fi14100281

Academic Editor: Sachin Sharma

Received: 20 August 2022
Accepted: 26 September 2022
Published: 29 September 2022

1. Introduction

The Internet of things (IoT) connects processing devices, mechanical devices, objects, digital machines, humans, or animals with unique identifiers (UID). This system can move information without human-to-computer or human-to-human interaction [1]. The thing concept in the IoT can be a person with an implanted heart rate monitoring device, an animal with a transponder biochip, a car with internal sensors to warn about low tire pressure, or an object with the ability to assign an IP address and refers to the movement of information in a network.

Today, organizations are one of the most common customers using IoT systems. This new technology helps organizations better understand customers, optimize operations and processes, improve customer service and decision-making, and increase business value.

An IoT environment contains several smart devices equipped with the Internet. These devices use embedded systems such as sensors, communication hardware, and processors to gather and send data and interact with the data they receive from the surroundings. IoT devices share incoming data through an IoT gateway or other edge device. These data are examined locally or in the cloud. Occasionally, these devices interconnect with other associated devices and perform certain actions based on the information they receive from each other [2]. These devices do most of the necessary processes independently, but people can also interact with these devices. For example, the user can set the device or give them the necessary commands and permission to access the data.

The networking, communication, and connectivity protocols utilized in Internet-enabled devices largely rely on the specified applications for the IoT device. The IoT may also employ soft computing to make data collection and processing easier and more dynamic [3]. In this research, most of the focus in the field of the IoT is directed toward

designing an energy-saving network by considering the average implementation time, makespan, network cost, and virtual machine computing cost.

Due to their large number, small size, and contingent placement, network nodes still rely on low-power batteries for energy [4]. In IoT networks, energy limitation is a significant issue [5]. Due to their use in harsh and inaccessible environments, network nodes cannot be recharged or replaced. Since IoT efficiency depends on network lifetime and coverage, energy-saving algorithms must be included in long-lived IoT networks [6]. Dynamic power management methods that reduce IoT networks' energy consumption after design and placement are crucial. The ideal IoT uses less energy through smart planning, receives data quickly and precisely over a long period, and requires no maintenance [7].

So far, many researchers have investigated the challenges of improving quality indicators of IoT networks, and in some of them, prioritization has been done between these challenges [8]. In this research, the improvement of qualitative indicators, including average implementation time, makespan, network cost, and virtual machine computing cost of the network will be investigated using an improved design of a seagull optimizer. The improved seagull optimization algorithm is used for minimizing the mentioned qualitative indicators under investigation to provide an energy-saving system with lower losses.

2. Literature Review

Recently, different works have been investigating IoT. To detect the virtual machines (VMs) experiencing overload and to maximize their use, Kaur et al. [9] introduced the bat algorithm and its hybridization with heuristic methodologies. In the suggested model, the heuristic techniques (HEFT and PEFT) seeding the bat algorithm parameters (frequency, velocity, loudness, and pulse rate) were used to optimize the time and cost of carrying out the processes. The model was put into practice in CloudSim, and the outcomes are examined in light of cutting-edge ACO and PSO metaheuristics.

Kavitha et al. implemented ant colony optimization (ACO) by effectively tweaking the parameters, replacing the first-come-first-serve (FCFS) virtual machine (VM) allocation strategy of Cloud computing in CloudSim [10]. In order to evaluate the effect of the Cloud's reaction times, an IoT-enabled healthcare setting was created that infrequently demands the capacity of the Cloud to process enormous volumes of data. By varying various parameters, including the number of ants, the potency of the pheromone, the number of VMs, the number of hosts, the strength of the computing power of processing elements, the number of users, and the size of the workloads, several experiments were conducted to determine the best VM allocation using ACO. According to experimental findings, the ACO leverages cloud resources more effectively than the default FCFS method.

For task scheduling in fog computing (TSFC), Abdel-Basset et al. suggested an energy-aware model based on the marine predator's algorithm (MPA) in order to better meet user-required QoS [11]. They put out the other two variations in addition to the conventional MPA. The first version, known as modified MPA (MMPA), changed the MPA so that it may be more effectively exploited by utilizing the most recent positions rather than the most recent best one. The second one had a better MMPA thanks to a ranking technique that included reinitialization, mutation toward the best, and reinitialization of the remaining half of the population after a certain number of iterations to get rid of local optima. Since the TSFC was regarded as a discrete problem while the MPA was developed to tackle continuous problems, the conventional MPA was transformed into a discrete one using the normalization and scaling phase. Based on several performance indicators, including energy consumption, makespan, flow duration, and carbon dioxide emission rate, the three variants were presented with additional metaheuristic and genetic algorithms.

To offer an energy-efficient task schedule within reasonable application completion durations, Ijaz et al. examined workflow scheduling in fog-cloud systems [12]. They proposed a two-phase scheduling approach called energy makespan multi-objective optimization. In order to schedule latency-sensitive tasks on fog resources and computationally complex tasks on cloud resources, the solution first models the problem as a multi-objective opti-

mization problem and calculates a trade-off between accuracy and conflicting objectives while earmarking fog and cloud resources. The deadline-aware stepwise frequency scaling strategy is modified to make use of the spare time between two already scheduled jobs on a single node in order to further cut down on energy consumption.

3. Methodology

Cloud computing is a model of users' access to services using their requirements regardless of the location and manner of service. Important firms such as Amazon, Microsoft, and Google financially support the cloud model, which entails the utilization of computer resources, including hardware and software, that are provided as a service through the network.

Present cloud computing schemes provide a wide range of on-demand virtual services. Cloud computing providers have categorized cloud services into three groups [13]: software as a service, platform as a service, and the Ministry of Construction as a service provide the necessary services [14].

Instead of buying hardware, customers buy their data center space and network equipment as a service. Platform as a service provides a computation stage that employs cloud structure. With this facility, the systems and environments needed may be employed by developers for the life cycle of the software [15]. Among them, web application development, testing, implementation, and hosting are mentioned.

Software as a service provides software over the Internet, thus eliminating the need to install software on customers' systems and making it easier to maintain and support them. Nowadays, the Internet has largely achieved its main goals. The main use of the Internet is to retrieve data and access services, while the host-to-host-based architecture is designed for applications such as long-distance networking. In recent years, the biggest challenge in the field of this data is the increase in service quality indicators in this field [16]. Various methods can be used to increase the quality of service (QoS) indicators.

In the present research, a novel developed design of the seagull optimization algorithm has been used so that the most optimal values can be obtained in the shortest processing time. This method consists of several parts, in the following order: problem modeling and expression of service quality parameters and calculation of these parameters, and use of the improved seagull optimization algorithm (ISOA) to improve these indicators; these parts are the constituent parts of the proposed method. Figure 1 presents the proposed model.

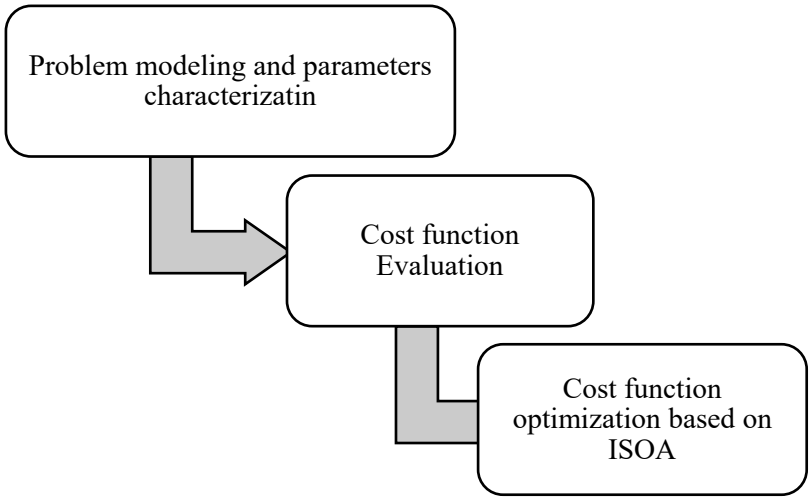

Figure 1. The introduced method.

In cloud environments, various processing resources are accessible to users, and they only pay for the resources they have used. Energy consumption and task scheduling are significant issues in cloud computing that many researchers have worked on. Task

scheduling is an NP-hard problem for which several bio-inspired techniques have been suggested. In these systems, the scheduling mechanism includes two steps: prioritizing tasks and choosing a virtual machine. Cloud computing is an architectural pattern regarding a network of clients that receive services from cloud devices, which have storage and computing capacity that allows them to share data and instances of cloud services to assign customers.

As a result, it is necessary to adopt the policies for managing the service and data in order to determine where and when to place data and services. The issue of fog service placement is a big problem in the field of fog services. In Figure 2, the fog calculation model is displayed graphically.

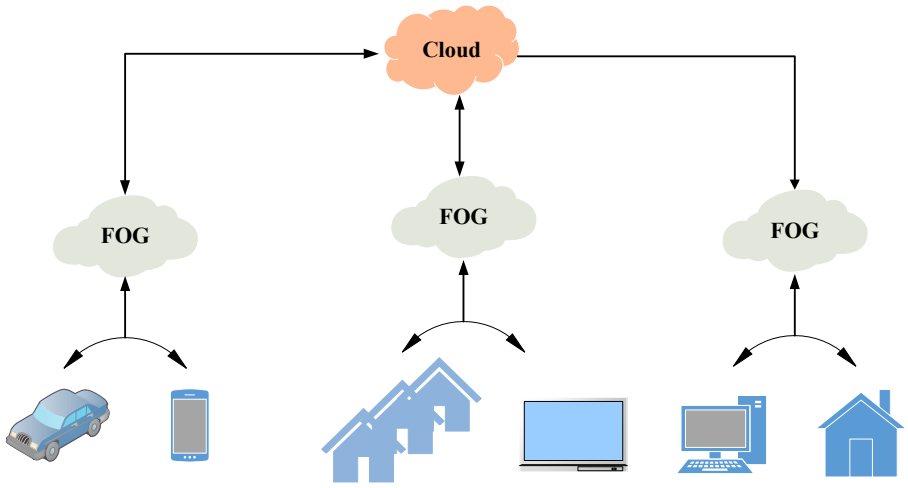

Figure 2. Fog calculation model.

3.1. Average Execution Time

The average execution time speaks about the typical Map/Reduce task execution time [17]. The parameters examined in this research include the average execution time (AET), which is given below:

$$AET = \frac{\sum_{j=1}^{N} P_{tN}(j)}{N} + \frac{\sum_{j=1}^{M} P_{tM}(j)}{M} \quad (1)$$

Based on the relationship above, P_{tN} and P_{tM} are the time spent for processing, and N and M define the map tasks quantity, and tasks reduction in this job, respectively, which, in this research, are the same as the time spent for processing.

3.2. Makespan

The critical time (makespan (M_{sp})) is the time needed to process all the tasks, which must be assigned to fog nodes in such a way that this amount is as low as possible to complete all the tasks. The makespan defines the time span of the Map/Reduce job from its beginning to its end. This term is mathematically formulated as follows:

$$M_{sp} = ft_r(M) \quad (2)$$

where $ft_r(M)$ defines the finished time for the reduced task, M.

3.3. Delay Time

The delay time term describes the disparity between the beginnings of the reduce and map tasks. This term can be mathematically formulated as follows:

$$T_D = St_N(N) + St_M(M) - ft_N(N) \quad (3)$$

where $ft_N(N)$ signifies the finished time of reduced task N, and $St_N(N)$ and $St_M(M)$ represent, in turn, the map task N beginning time of reduce task M.

3.4. Network Cost

The network cost (C_{net}) represents the network expenditure incurred when the reduced task acquires data from the intermediate output of the map task after the job task has already acquired it from storage. This term can be mathematically defined as follows:

$$C_{net} = T_D \times Network\ cost\ per\ unit\ time \tag{4}$$

where T_D describes the time delay.

3.5. Virtual Machine Computing Cost

This value shows the cost of using CPU, which may be evaluated in the form of the following relationship. Based on this relationship, V_m^{cost} indicates the cost of using CPU per second time unit. The P_{tN} has time taken for processing. The computation cost of the virtual machine can be formulated as follows:

$$V_m^{cost} = \left(\sum_{j=1}^{n_{vm}} P_{tN}(j) + \sum_{k=1}^{m_{vm}} P_{tM}(k) \right) \times VM\ cost\ per\ unit\ time \tag{5}$$

where $P_{tN}(j)$ and $P_{tM}(k)$ represent, in turn, the execution time of VM_j when running the map task, and the implementation time of VM_k when running the reduce task.

Based on these parameters, these four parameters should identify a minimum value. Based on these three parameters, the final fitness relationship will be as follows:

$$Fitness = w_1 \times AET + w_2 \times M_{sp} + w_3 \times C_{net} + w_4 \times V_m^{cost} \tag{6}$$

Based on this relationship, w_1, w_2, w_3, and w_4 represent proportionality coefficients that were employed for the summation of these 3 parameters. According to this relationship, we should have the following assumption:

$$w_1 + w_2 + w_3 + w_4 = 1 \tag{7}$$

When analyzing IoT-based applications, the defined variables are critical considerations. According to the equations, we can conclude that the mentioned variables are functions of the preceding independent variables, meaning that their values change when the independent variables change. For example, raising the quantity of the virtual machines for the same IoT application job while remaining inside the datacenter's capability may vary the makespan, average processing time, and other characteristics when more computational resources are needed by map or reduce processes.

4. Improved Version of Seagull Optimization Algorithm

Many issues around us need to be optimized in a better way for better design. Optimization problems refer to a group of problems that usually refer to a complex problem related to the industry [18].

When engineers can save development time and expense by utilizing optimization techniques, they can create better designs. Numerous engineering optimization issues are more complicated and challenging to address using traditional optimization techniques such as mathematical programming [19]. Nowadays, many optimization problems are considered to be non-polynomial degrees and NP-hard problems [20]. Optimization is widely used in applications, such as the industrial design of factory parts, structure design, task scheduling problems, and optimal clustering [21]. Among the available solutions for dealing with such problems, the world cup optimization (WCO) algorithm [22], the arithmetic optimization algorithm (AOA) [23], the Aquila optimizer (AO) [24], the cat

swarm optimization algorithm [25], and the seagull optimization algorithm (SOA) can be used [26].

4.1. Seagull Optimization Algorithm (SOA)

Seagulls are a particular species of bird that live in coastal habitats that belong to the family Laridae and suborder Lari. A seagull flying above the earth may be a breathtaking sight [27]. The tallest wing, similar to an airplane's wing, belongs to these enormous, winged birds, whose wingspan is 11 feet (3.4 m). Seagulls may occasionally stay in the air for continuous hours or even flip because of their large, powerful wings, which they employ to ride across ocean breezes. They sometimes float near the surface of the water; however, this condition leaves them open to aquatic predators.

They only congregate for mating and are seldom ever observed on land; during this period, they establish vast colonies on remote islands. They are also well-known to sailors since they occasionally pursue ships in search of food and trash. The primary food source for seagulls is fish, although they also consume insects, frogs, moles, and earthworms.

Seagulls cohabit in huge flocks. Every one of them includes a distinctive voice that they use to interact with others. They occasionally steal food from birds, animals, and even human hands.

Seagulls have also used other tactics to go after animals. They use their feet, for example, to make a rain shower noise to catch food, or they utilize breadcrumbs to lure fish. Another distinguishing element of seagulls is their migratory habits.

To avoid the cold and get to abundant food supplies, seagulls migrate north in the springtime and south in the fall. They may also migrate from the ground to higher levels or from one coast to another [28]. The following is an example of the migration algorithm.

Seagulls in a moving swarm catalyze the process. To prevent collisions, the starting points for each swarm are thought to be distinct.

To find the optimal answer, the seagulls strive to fly in the direction of their best chance of survival.

Following is a description of the SOA approach.

4.2. Migration

The migration is a technique to mimic the migration of a swarm of seagulls to another location. The algorithm's exploration component is comprised of this technique. Three requirements should be met for migration:

- The position of the swarms is adjusted depending on an extra parameter (A) to prevent collisions.

$$\begin{gathered} X = A \times \vec{X}_c(i), \\ i = 0,\ 1,\ 2,\ \ldots,\ Max(i) \end{gathered} \tag{8}$$

where $\vec{X}_N$ indicates the location to be prevented from the individuals' collusion, $X(i)$ identifies the agent's present position (i), and A shows the candidate's movement patterns that are described as follows:

$$A = S_c - \left(i \times \left(\frac{S_c}{Max\ (i)} \right) \right) \tag{9}$$

where S_c shows the frequency control of parameter A that it is in the range 0 and S_c.

(A) Knowledge of the surrounding neighbors: this section simulates the individual moving in the optimal direction based on the knowledge of the surrounding neighbors (good solution).

$$M_e = B \times (\vec{X}_b(i) - X(i)) \tag{10}$$

where $\vec{M}_e$ shows the location of the agents $\vec{X}_c(i)$, shown in comparison to the best-suited candidate $\vec{X}_b(i)$, and parameter B shows a stochastic factor. The algorithm balances exploitation and exploration in the manner described below:

$$B = 2 \times A^2 \times R \tag{11}$$

where R shows a stochastic variable between 0 and 1.

(B) Proceeding to the best answer side (search agent): this is an update stage for enhancing the best candidates, as shown in:

$$\vec{E}_e = \left| \vec{X}_N - \vec{M}_e \right| \tag{12}$$

where $\vec{E}_e$ shows the distinction between seagulls and the best answer.

4.3. Attacking

Throughout the migration, seagulls can modify the angle and speed of their attacks at any time. They can, however, keep their location in the air by using their wings and weight. This technique is the algorithm's exploitation element.

The seagulls attach from 3 different angles—*a*, *b*, and *c* planes—by the following equation:

$$\hat{a} = r \times cos(t) \tag{13}$$

$$\hat{b} = r \times sin(t) \tag{14}$$

$$\hat{c} = r \times t \tag{15}$$

where t shows a stochastic variable between 0 and 2π 0, and 2π and r illustrates the spiral turns' radius as shown in:

$$r = \alpha \times e^{\beta t} \tag{16}$$

where α and β define the spiral shape, and e indicates the natural logarithm base. Seagulls can update their location as follows:

$$\vec{X}_c(i) = (\vec{E}_e \times \hat{a} \times \hat{b} \times \hat{c}) + \vec{X}_b(i) \tag{17}$$

where $X_c(i)$ shows the best answer.

The SOA's early convergence and rapid convergence are two major downsides. The following develops an improvement to address these flaws (Figure 3).

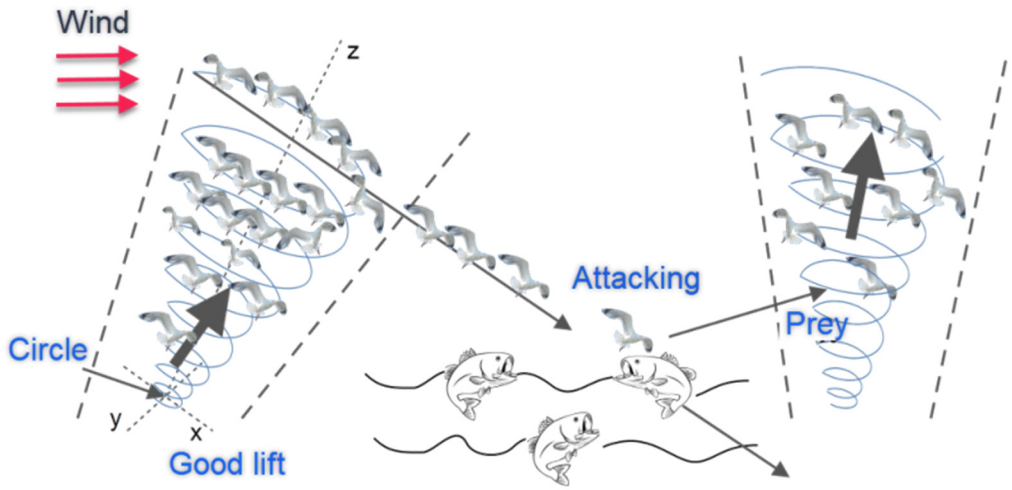

Figure 3. Method of migration and attacking of the seagulls.

4.4. The Improved SOA

The powerful technique known as Lévy flight (LF) is frequently utilized in meta-heuristics to address the issue of premature convergence [29]. In this technique, a stochastic methodology which is theoretically described as follows is used to effectively manage the local search:

$$Le(w) \approx w^{-1-\varrho} \tag{18}$$

$$w = \frac{A}{|B|^{1/\varrho}} \tag{19}$$

$$\sigma^2 = \left\{ \frac{\Gamma(1+\varrho)}{\varrho\Gamma((1+\varrho)/2)} \frac{\sin(\pi\varrho/2)}{2^{(1+\varrho)/2}} \right\}^{\frac{2}{\varrho}} \tag{20}$$

where ϱ shows the Lévy index that is between 0 and 2 (here, $\varrho = 3/2$ [29]), $A \sim N(0, \sigma^2)$ and $B \sim N(0, \sigma^2)$, $\Gamma(.)$ designates the Gamma distribution, w represents the step size, and $A/B \sim N(0, \sigma^2)$ indicates that the samples come from a Gaussian function with an average of 0 and a variance of σ^2, respectively.

By taking into account the aforementioned formulations, the SOA updating formula is as shown in:

$$\vec{E}_{el} = \vec{E}_e + \left| \vec{X}_N + \vec{M}_e \right| \times Le(\delta) \tag{21}$$

where $\vec{E}_{el}$ signifies the updated location of the search agent, $\vec{E}_e$.

The Singer technique is used as the following step to increase the pace of the state's convergence [30,31]. To use this method, the unidentified stochastic variables are transformed into the following regularly defined values:

$$\begin{aligned} r_{i+1} &= 1.07\left(7.9r_i - 23.3r_i^2 + 28.7r_i^3 - 13.3r_i^4\right) \\ A_{i+1} &= 1.07\left(7.9A_i - 23.3A_i^2 + 28.7A_i^3 - 13.3A_i^4\right) \\ t_{i+1} &= 1.07\left(7.9t_i - 23.3t_i^2 + 28.7t_i^3 - 13.3t_i^4\right) \end{aligned} \tag{22}$$

The fittest of them are retained as follows to create the finest agent-based best answer:

$$\vec{E}_{el} = \begin{cases} \vec{E}_{el} & F(\vec{E}_{el}) > F(\vec{E}_e) \\ \vec{E}_e & otherwise \end{cases} \tag{23}$$

Figure 4 presents the flowchart for the demonstrated ISOA.

The problem of increasing service quality indicators based on assigning tasks to fog processing nodes can be displayed in the majority of matrices where each matrix represents a solution candidate in ISOA, and the optimal solution in the latest iteration of the SOA represents the best solution matrix, which increases the quality of the indicators proposed based on the cost function, and each candidate creates a solution. In this way, the first matrix is randomly selected. However, this matrix is calculated during the algorithm process and based on the probability of choosing machines, and the machines that have a higher probability of being selected are chosen.

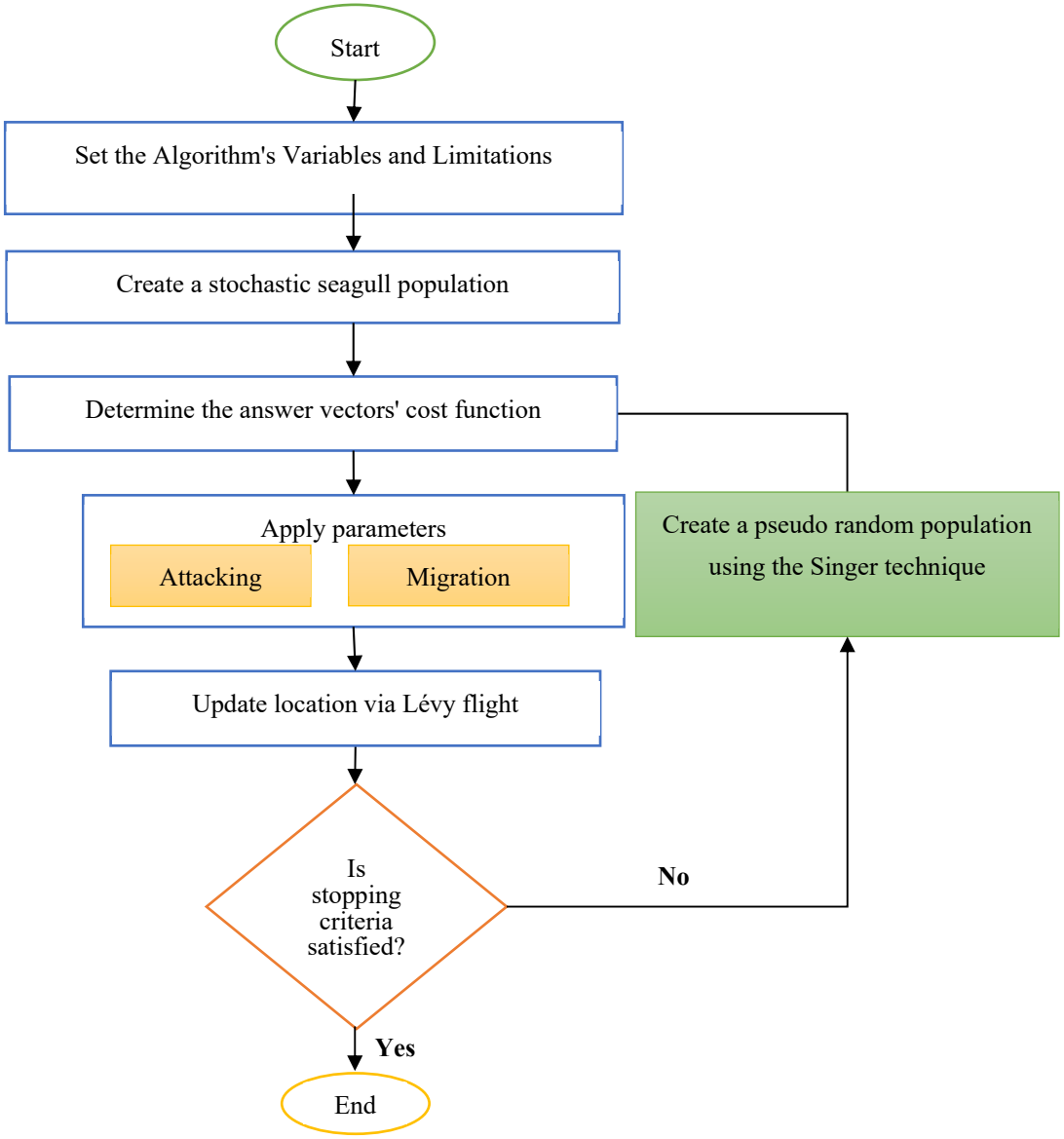

Figure 4. Block diagram of the ISOA.

5. Simulation Results

Cloud computing is a distributed infrastructure where data, computation, storage, and applications are dispersed locally between data-generating devices and the cloud. The concept of fog computing exists in both cloud systems and big data structures, where the problems of accessing information are increased. A distributed fog infrastructure is replacing a centralized cloud computing architecture. Both fog computing and cloud computing provide users with storage, application, and data. However, the most important and effective point in this field is the challenge of energy consumption in data centers and reducing the response time of machines in fog computing. Most of the workload in this form of network is the responsibility of data centers; due to the nature of the devices, which are not similar to the cloud computing environment, it has the same expectations as human users, so the control and management of the device in cloud computing is the responsibility of the data centers. Based on this, the results of the proposed algorithm will be discussed in this section.

The experimental environment is performed on Matlab R2017b on a 2 GHz processor Intel® Core™ i7 laptop, with 8 GB memory.

To assess the suggested technique, first, the results were tested using the suggested ISOA, and then it was validated through comparison with some published methods, including MRC [17], IoTSim-Osmosis (IOT/OS) [32], IoTSim-Stream (IOT/ST) [33], and the standard SOA-based technique. Here, the methods that have not used metaheuristics in their work have been extended by metaheuristics based on SOA to provide a fair comparison. To test the proposed solution, the ISOA parameters are defined in Table 1.

Table 1. Set parameter of the proposed ISOA in this study.

Parameter	Value
u	1
v	0.002
Iteration number	200 and 1000
population	50

The succeeding graphs related to the evaluation of the proposed method based on average execution time, makespan, and virtual machine computing cost.

Figure 5 displays the execution time for all comparative algorithms. It can be seen from Figure 5 that the execution time for the comparative methods is displayed in two hundred iterations, and although all algorithms are descending, the proposed ISOA was able to achieve the lowest possible value in the last iteration. Figure 6 shows the makespan of all the studied techniques.

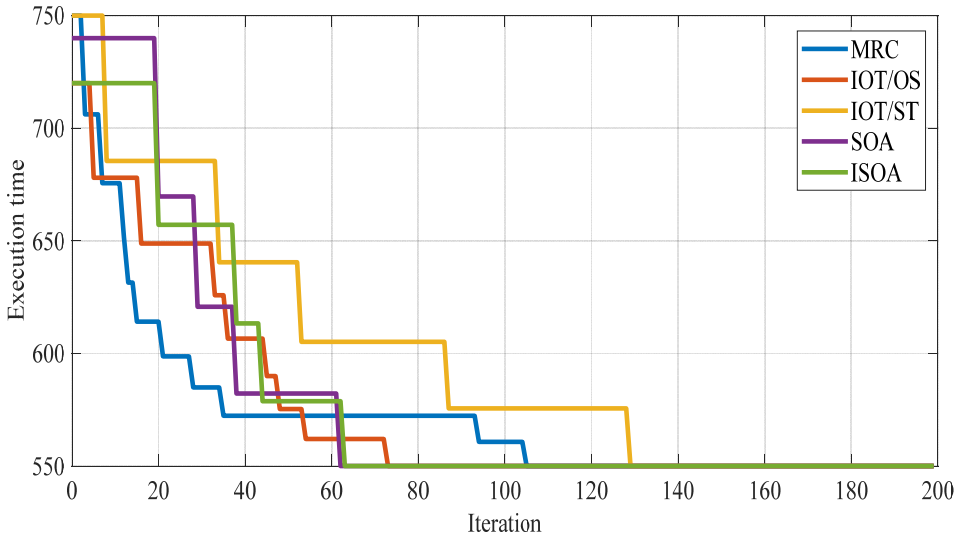

Figure 5. Execution time for the comparative algorithms.

Figure 6 shows the makespan of all studied techniques. It can be seen from Figure 6 that the amount of makespan for all algorithms is displayed in two hundred iterations, and the amount of makespan in the proposed algorithm is significantly decreased in the last iteration compared to the compared algorithm.

Figure 7 shows the virtual machine computing cost value for all comparative algorithms. As can be seen from Figure 7, in this graph, the virtual machine computing cost for all methods is shown in 200 iterations, and the proposed algorithm was able to achieve the lowest cost with a much lower amount in the last iteration.

Figure 8 displays the network cost for all comparative algorithms. It can be seen that the network cost is the same even if the VM number has changed. This is because, given the same workload, the data size produces the same network latency.

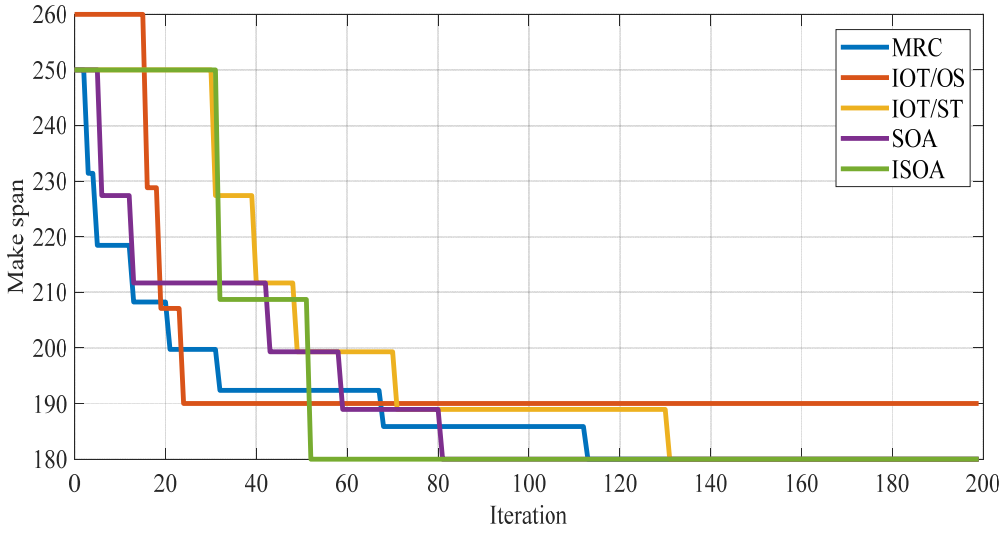

Figure 6. Makespan for the investigated methods.

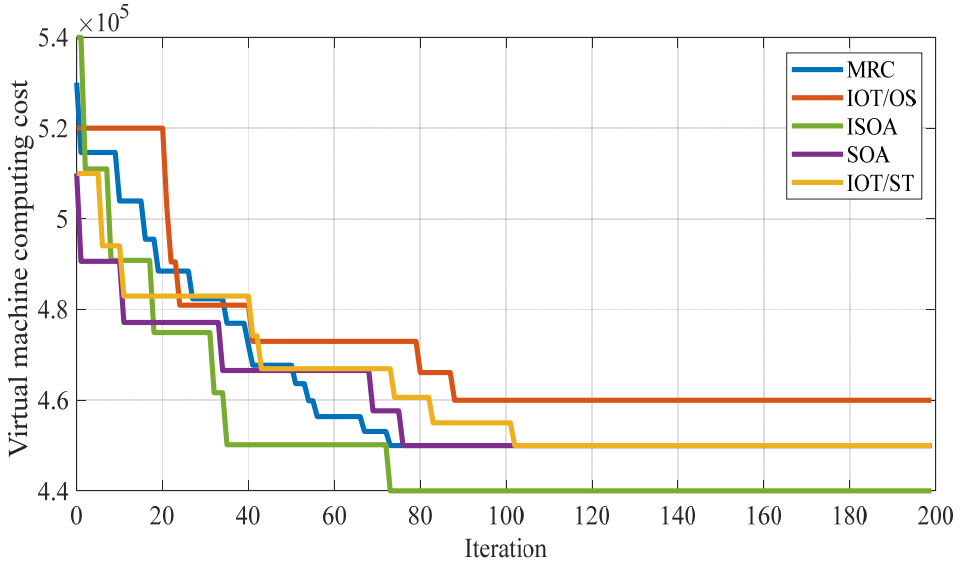

Figure 7. The amount of virtual machine computing cost for the comparative algorithms.

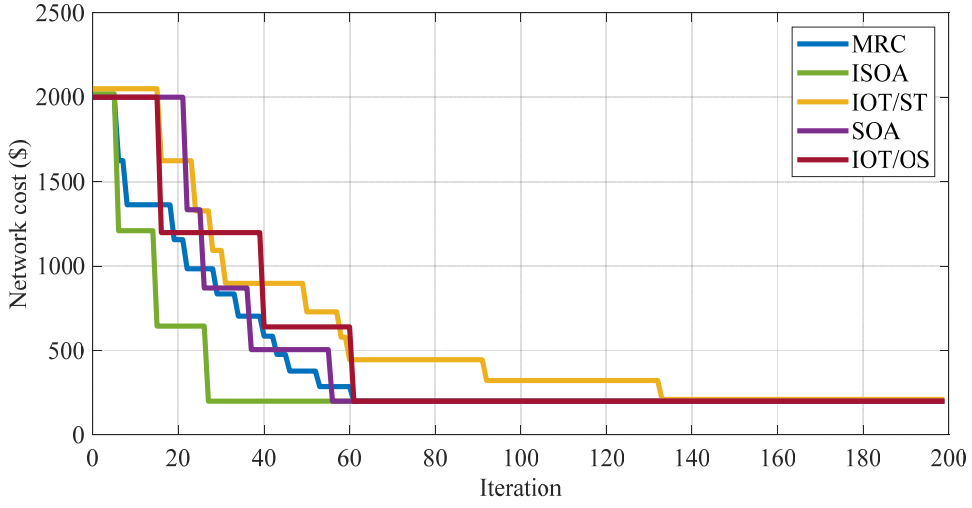

Figure 8. Network cost for all comparative algorithms.

Based on the results of all three-convergence analyses, several points can be pointed out. All methods have gone through an acceptable downward path to reach the solution. The proposed ISOA moves towards the optimum at a high speed and has a high convergence speed. The proposed algorithm can obtain the optimal solution with the least number of repetitions and calculations, and there is no need to repeat the algorithm many times. Although some of the methods, such as MRC and IOT/OS, have used parallel processing for this purpose, they still could not obtain a lower value than the proposed ISOA in the final result.

6. Conclusions

In this research, a new improved bio-inspired algorithm, called the improved seagull optimization algorithm (ISOA), was utilized for minimizing an objective function aimed at minimizing the average implementation time, makespan, network cost, and virtual machine computing cost. The results achieved by the suggested method were then compared with some other related state-of-the-art methods, including MRC, IoTSim-Osmosis (IOT/OS), IoTSim-Stream (IOT/ST), and the original seagull optimization algorithm-based technique (SOA). Simulation results showed that using the analysis study, the results of the suggested method indicated the superiority of the proposed ISOA-based method in this field. Although the proposed method indicated satisfying results for the research purpose, the following cases can be considered for future study: (1) using other modified metaheuristic algorithms to provide more optimal results; (2) using other parameters such as energy and response time; and (3) using combined methods with existing methods such as artificial neural networks and fuzzy logic.

Funding: This work was supported by Al-Mustaqbal University College (grant number: MUC-E-0122).

Data Availability Statement: The data shall be made available on request.

Conflicts of Interest: The author declares that he has no conflict of interest.

References

1. Alferaidi, A.; Yadav, K.; Alharbi, Y.; Razmjooy, N.; Viriyasitavat, W.; Gulati, K.; Kautish, S.; Dhiman, G. Distributed Deep CNN-LSTM Model for Intrusion Detection Method in IoT-Based Vehicles. *Math. Probl. Eng.* **2022**, *2022*, 3424819. [CrossRef]
2. Bahmanyar, D.; Razmjooy, N.; Mirjalili, S. Multi-objective scheduling of IoT-enabled smart homes for energy management based on Arithmetic Optimization Algorithm: A Node-RED and NodeMCU module-based technique. *Knowl.-Based Syst.* **2022**, *247*, 108762. [CrossRef]
3. Salih, K.O.M.; Rashid, T.A.; Radovanovic, D.; Bacanin, N. A comprehensive survey on the Internet of Things with the industrial marketplace. *Sensors* **2022**, *22*, 730. [CrossRef] [PubMed]
4. Koohang, A.; Sargent, C.S.; Nord, J.H.; Paliszkiewicz, J. Internet of Things (IoT): From awareness to continued use. *Int. J. Inf. Manag.* **2022**, *62*, 102442. [CrossRef]
5. Samie, F.; Bauer, L.; Henkel, J. From cloud down to things: An overview of machine learning in internet of things. *IEEE Internet Things J.* **2019**, *6*, 4921–4934. [CrossRef]
6. Seetharaman, A.; Patwa, N.; Saravanan, A.S.; Sharma, A. Customer expectation from industrial internet of things (IIOT). *J. Manuf. Technol. Manag.* **2019**, *30*, 1161–1178. [CrossRef]
7. Madhu, S.; Prasad, R.K.; Ramotra, P.; Edla, D.R.; Lipare, A. A Location-less Energy Efficient Algorithm for Load Balanced Clustering in Wireless Sensor Networks. *Wirel. Pers. Commun.* **2022**, *122*, 1967–1985. [CrossRef]
8. Ben-Daya, M.; Hassini, E.; Bahroun, Z.; Banimfreg, B.H. The role of internet of things in food supply chain quality management: A review. *Qual. Manag. J.* **2020**, *28*, 17–40. [CrossRef]
9. Kaur, A.; Kamboj, S.; Kaur, B.; Hrisheekesha, P.N. Hybrid Approach for Virtual Machine Optimization using BAT Algorithm in cloud. In Proceedings of the 2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST), Mohali, India, 17–18 December 2021; pp. 57–61.
10. Kavitha, K.; Sharma, S.C. Performance analysis of ACO-based improved virtual machine allocation in cloud for IoT-enabled healthcare. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5613. [CrossRef]
11. Abdel-Basset, M.; Mohamed, R.; Elhoseny, M.; Bashir, A.K.; Jolfaei, A.; Kumar, N. Energy-aware marine predators algorithm for task scheduling in IoT-based fog computing applications. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5068–5076. [CrossRef]
12. Ijaz, S.; Munir, E.U.; Ahmad, S.G.; Rafique, M.M.; Rana, O.F. Energy-makespan optimization of workflow scheduling in fog–cloud computing. *Computing* **2021**, *103*, 2033–2059. [CrossRef]
13. Haber, M.J.; Chappell, B.; Hills, C. *Cloud computing, in Cloud Attack Vectors*; Springer: Berlin, Germany, 2022; pp. 9–25.

14. Ke, M.; Gao, Z.; Wu, Y.; Gao, X.; Wong, K.K. Massive access in cell-free massive MIMO-based Internet of Things: Cloud computing and edge computing paradigms. *IEEE J. Sel. Areas Commun.* **2020**, *39*, 756–772. [CrossRef]
15. Marinescu, D.C. *Cloud Computing: Theory and Practice*; Morgan Kaufmann: Burlington, MA, USA, 2022.
16. Escamilla-Ambrosio, P.J.; Rodríguez-Mota, A.; Aguirre-Anaya, E.; Acosta-Bermejo, R.; Salinas-Rosales, M. Distributing computing in the internet of things: Cloud, fog and edge computing overview. In *NEO 2016*; Springer: Berlin, Germany, 2018; pp. 87–115.
17. Zeng, X.; Garg, S.K.; Strazdins, P.; Jayaraman, P.P.; Georgakopoulos, D.; Ranjan, R. IOTSim: A simulator for analysing IoT applications. *J. Syst. Archit.* **2017**, *72*, 93–107. [CrossRef]
18. Razmjooy, N.; Estrela, V.V.; Loschi, H.J.; Fanfan, W. *A Comprehensive Survey of New Meta-Heuristic Algorithms. Recent Advances in Hybrid Metaheuristics for Data Clustering*; Wiley Publishing: Hoboken, NJ, USA, 2019.
19. Xu, Y.; Wang, Y.; Razmjooy, N. Lung cancer diagnosis in CT images based on Alexnet optimized by modified Bowerbird optimization algorithm. *Biomed. Signal Process. Control* **2022**, *77*, 103791. [CrossRef]
20. Razmjooy, N.; Ashourian, M.; Foroozandeh, Z. *Metaheuristics and Optimization in Computer and Electrical Engineering*; Springer: Berlin, Germany, 2020.
21. Ramezani, M.; Bahmanyar, D.; Razmjooy, N. A new improved model of marine predator algorithm for optimization problems. *Arab. J. Sci. Eng.* **2021**, *46*, 8803–8826. [CrossRef]
22. Razmjooy, N.; Khalilpour, M.; Ramezani, M. A new meta-heuristic optimization algorithm inspired by FIFA world cup competitions: Theory and its application in PID designing for AVR system. *J. Control Autom. Electr. Syst.* **2016**, *27*, 419–440. [CrossRef]
23. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [CrossRef]
24. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
25. Ahmed, A.M.; Rashid, T.A.; Saeed, S.A.M. Cat swarm optimization algorithm: A survey and performance evaluation. *Comput. Intell. Neurosci.* **2020**, *2020*, 4854895. [CrossRef]
26. Dhiman, G.; Kumar, V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowl.-Based Syst.* **2019**, *165*, 169–196. [CrossRef]
27. Cao, Y.; Li, Y.; Zhang, G.; Jermsittiparsert, K.; Razmjooy, N. Experimental modeling of PEM fuel cells using a new improved seagull optimization algorithm. *Energy Rep.* **2019**, *5*, 1616–1625. [CrossRef]
28. Konzack, M.; Gijsbers, P.; Timmers, F.; van Loon, E.; Westenberg, M.A.; Buchin, K. Visual exploration of migration patterns in gull data. *Inf. Vis.* **2019**, *18*, 138–152. [CrossRef]
29. Li, X.; Niu, P.; Liu, J. Combustion optimization of a boiler based on the chaos and Levy flight vortex search algorithm. *Appl. Math. Model.* **2018**, *58*, 3–18. [CrossRef]
30. Yang, D.; Li, G.; Cheng, G. On the efficiency of chaos optimization algorithms for global optimization. *Chaos Solitons Fractals* **2007**, *34*, 1366–1375. [CrossRef]
31. Rim, C.; Piao, S.; Li, G.; Pak, U. A niching chaos optimization algorithm for multimodal optimization. *Soft Comput.* **2018**, *22*, 621–633. [CrossRef]
32. Alwasel, K.; Jha, D.N.; Habeeb, F.; Demirbaga, U.; Rana, O.; Baker, T.; Dustdar, S.; Villari, M.; James, P.; Solaiman, E.; et al. IoTSim-Osmosis: A framework for modeling and simulating IoT applications over an edge-cloud continuum. *J. Syst. Archit.* **2021**, *116*, 101956. [CrossRef]
33. Barika, M.; Garg, S.; Chan, A.; Calheiros, R.N.; Ranjan, R. IoTSim-Stream: Modelling stream graph application in cloud simulation. *Future Gener. Comput. Syst.* **2019**, *99*, 86–105. [CrossRef]