Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 09, No. 02, June 2023.

OPTIMAL ENERGY CONSUMPTION AND COST PERFORMANCE SOLUTION WITH DELAY CONSTRAINTS ON FOG COMPUTING

Zahra Mahmoudi, Elham Darbanian and Mohsen Nickray

(Received: 8-Dec.-2022, Revised: 9-Feb.-2023, Accepted: 24-Feb.-2023)

ABSTRACT

Cloud computing plays an essential role in the development of the Internet of Things, which provides-data processing and storage services. Fog computing, the evolution of cloud computing helps provide solutions to cloud-computing challenges, such as latency, location awareness and real-time mobility support. Fog computing fills the gap between the cloud and IoT devices within the close vicinity of IoT devices. So, computation, networking, storage, data management and decision-making occur along the path between the cloud and the IoT devices. The automatic and intelligent management of fog node resources and achieving an effective scheduling policy in the computing model are necessary requirements and will lead to the improvement of the overall performance of fog computing. Some optimization problems are modeled by mixed-integer nonlinear programming (MINLP). In this paper, a model; i.e., an MINLP optimization problem on fog computing, is designed. Our model has two goals: to increase cost performance as well as to reduce energy consumption. Cost performance is the price that users are charged as benefit/revenue. In other words, cost performance is defined as the ratio of the average data rate of each user to its cost. Then, the exact mathematical method with the GAMS program was used to prove its logical process. In the next step, we solved the model with genetic algorithm (GA), particle swarm optimization (PSO), simulated annealing+GA (SA+GA), teaching–learning-based optimization (TLBO), grey wolf optimizer (GWO), grasshopper optimization algorithm (GOA) and random method. According to the TOPSIS comparison, the SA+GA method with a value of 0.23 is the best one compared to other methods. Then come GWO, GA, TLBO, PSO and GOA methods, respectively.

KEYWORDS

Fog computing, Optimization, Mixed-integer nonlinear programming, MINLP, Energy consumption, Cost performance.

1. INTRODUCTION

According to the analysis carried out by Cisco by 2023, Internet of Things devices will make up about 50% of the devices in networks around the world. The Internet of Things (IoT) is one of the most influential technologies in the world. IoT deals with large amounts of data that are not easy to process and store. Cloud computing plays a significant role in the development of the Internet of Things, which provides data-processing and storage services. However, many of its applications encounter with cloud computing challenges, including latency, location awareness and real-time mobility support. Fog computing, that almost looks like the evolution of cloud computing, contributes to providing solutions to these challenges [1]. The growth of devices and subsequently the data of the Internet of Things is such that it is possible to exceed the capacity of information and this shows the necessity of using the models in the fog-computing technology infrastructure to process the data and as a complement to the cloud-computing model. On the other hand, fog nodes are the main entities of the fog-computing model and for this reason, the effective management of the resources of these nodes is of particular importance. Automation of operations in computer networks through the use of innovations can lead to increased productivity, reduced operational cost and better quality of service delivery. Therefore, the automatic and intelligent management of fog-node resources and achieving an effective scheduling policy in the computing model are necessary requirements and will lead to the improvement of the overall performance of fog computing. Fog computing fills the gap between the cloud and the IoT devices within the close vicinity of the IoT devices. So, computation, networking, storage, data management and decision-making occur along the path between the cloud and the IoT devices [2].

There is a topic called "optimization", which is related to maximization and minimization. The goal of

Z. Mahmoudi, E. Darbanian and M. Nickray are with Department of Computer Engineering and Information Technology, University of Qom, Qom, Iran. Emails: z.mahmoudi@stu.qom.ac.ir, e.darbanian@stu.qom.ac.ir and m.nickray@qom.ac.ir

optimization is finding the assignment of variables, maximizing or minimizing the value of a given function [3]. When the quantity to be optimized is expressed by just one objective function, it is deemed a uni-objective or single-objective problem. Otherwise, it is a multi-objective problem that must be optimized simultaneously [4].

It is a common misconception that most design or problem-solving activities should optimize a single goal; for example, maximizing profit or creating the lowest cost, even if there are several conflicting goals for optimization. However, the relationship between goals is usually complex and depends on available options. In addition, different goals are usually incomparable; therefore, combining them into one combined goal is challenging. Many decision-making and planning problems involve several conflicting goals that have to be studied simultaneously. Such problems are generally known as multi-criteria decision-making (MCDM) problems. Depending on the characteristics of the problem [5], MCDM can classify problems in many ways. Two main classes of MCDM are generally introduced: multiple-objective decision making (MODM) and multiple-attribute decision making (MADM) [6].

Many engineering and scientific optimization problems involve combinatorial and nonlinear relations. Some optimization problems are modeled by mixed-integer nonlinear programming (MINLP) that combines the capabilities of linear programming (LP) and nonlinear programming (NLP) [7].

Initially, the driving agent behind the development of the general algebraic modeling system (GAMS) was that mathematical-programming users are believing in optimization as a strong and subtle framework to solve real-life problems in science and engineering [8]. GAMS is a high-level modeling language for formulating models. Being composed of short algebraic statements, it is easily read by modelers and it formally looks like the generally-used programming languages [9].

Particle swarm optimization (PSO) is a population-based algorithm in which the movement of a flock of birds are simulated to find the optimum solution [10]. Genetic algorithm (GA) is one of the population-based optimization methods [11]. Simulated annealing (SA) is an algorithm to solve large combinatorial optimization problems [12]. Teaching–learning-based optimization (TLBO) is an algorithm for optimizing mechanical design problems. This method deals with the teacher's effect on learners. TLBO is a population-based method too [13]. A model, in order to perform optimization, is grey wolf optimizer (GWO) which models the hunting technique and the social hierarchy of grey wolves mathematically [14]. Another optimization algorithm is called the grasshopper optimization algorithm (GOA) [15].

In this paper, the single-objective model is proposed for minimizing the total energy of all mobile devices and maximizing the cost performance due to their service-latency limitations. Our problem is a mixed-integer nonlinear programming (MINLP) problem. Then, the proposed model with the exact mathematical method of GAMS is solved. Also, the problem with PSO, GA, SA+GA, TLBO, GWO, GOA and random, is solved in order to find the best solution. The research hypotheses in this paper include the solvability of the problem model using the exact mathematical method (GAMS) and meta-heuristic methods, examining our problem model that has an answer at an acceptable time with a high number of iterations, examining the existence of an optimal solution that optimizes the objective function (sum of energy-consumption values being minimum and cost performance values being maximum) and finally examining the environment in which resources should be allocated. In this paper, it is assumed that the number of fog nodes and users was limited. All users in the model, request for resource and resource allocation are applied as first-in, first-out (FIFO).

User satisfaction or user experience is one of the important criteria for SPs. Service latency is a measure of user satisfaction. Also, we must first ensure that the transmission quality between users and FNs is satisfied. To measure system performance, we consider mandatory revenue and price offers from users as the benefit/revenue of the SPs. The price offered by each user is related not only to the latency requirement, but also to data size. Each SP serves more than one user and therefore receives more than one offer. Two performance measures, user satisfaction and SP revenue, are essential for good resource allocation in fog computing. CP is defined as the ratio between each user's average data rate and its price cost, in unit of Mbps/sec/dollar. Because the actual amount of delay is strongly related to the amount of user data to transmit and process, the data rate is considered, rather than the pure delay. It also makes sense to use the users' monetary payment/offer for the respective fog computing service that they obtain for the cost factor. The cost-performance function for each user represents the quality of service for which the user pays [16]. Due to the increasing importance of energy consumption, fog-computing architecture is an effective solution to enable energy-efficient and low-latency mobile applications due to its low-latency and high-bandwidth connections with mobile devices as well as cloud servers, agile mobility and location-awareness support [17].

The contributions of our article are as follows:

- A computational framework (single-objective model) which considers both vertical and horizontal cooperation between fog nodes and mobile devices is proposed. Due to service-latency limitations, the total energy of all mobile devices is minimized and the cost performance is maximized at the same time. Then, the proposed model is solved with the exact mathematical method of GAMS and Baron Solver.
- A task can be offloaded by a mobile device to one of the fog nodes, the cloud, through a fog node or to the cloud server directly. The main problem among thirty problems (according to the central limit theorem (CLT)) is considered with three sizes (small, medium and large). By changing the number of mobile devices and fog nodes, problems are obtained with some different variables, which can be used to check the scalability of the problem. The purpose of this scenario is to measure the behavior of algorithms by considering the values of two objective functions and their execution times.
- Then, the problem is solved with eight methods (PSO, GA, SA+GA, TLBO, GWO, GOA, random and exact mathematical method) to find the best solution. Also, twelve other issues are considered with different numbers of mobile devices and fog nodes, in two main stages: the number of fixed fog nodes and the number of variable mobile devices and conversely (the number of fixed mobile devices and the number of variable fog nodes). Therefore, a more accurate analysis of the values of objective function is obtained with two indices of execution time and the value of the objective function.
- Extensive simulations are performed to evaluate cost performance and energy consumption and finally, the best algorithm is selected by using the technique for order of preference by similarity to ideal solution (TOPSIS) method.

The rest of this paper is arranged as follows. Related work is presented in Section 2. In Section 3, the network model is discussed. Problem solving is presented in Section 4. Evolution is presented in Section 5. Finally, in Section 6, the conclusion is presented.

2. RELATED WORKS

The related papers are presented as follows. Authors in [18] proposed a joint offloading decision and a framework for resource allocation optimization for MEC with algorithms of relaxing-optimization policy (ROP) and index branch-and-bound algorithm (IBBA). Their paper has disregarded the optimal transmission power assignment. The model only included communications. In [19], an IoT-based remote health-monitoring system implementation was presented, that included a demonstration of a smart e-health gateway called UT-GATE. Applications were user-centric, whereas services were developer-centric. Authors in [20] have studied an MCC system with multiple users, one CAP and one remote cloud server. The weighted total cost of energy, computation and maximum delay were minimized among all users. Also, a new approach to the joint task offloading and computation and communication resource allocation with share CAP, an efficient heuristic algorithm using semi-definite relaxation (SDR) and a new approach to randomization mapping were proposed by them. They assumed that there were several mobile-phone users, each with only one task. In [21], the authors have proposed min-max fairness in a mixed fog/cloud computing system by joint optimization of offloading decision-making and resource allocation using computation offloading and resource allocation (CORA) algorithms, as well as the bisection method for computation-resource allocation (BCRA) algorithm. They only considered orthogonal multiple access (OMA). Researchers in [22] have investigated a framework for optimization of computation offloading, computation resource allocation, resource block (RB) pattern assignment, transmit power allocation and a low-complexity general algorithm framework known as fireworks algorithm based on joint computation offloading and resource allocation algorithm (FAJORA) to decompose the problem into several sub-problems. If the original LTE standard were to be considered, support for these processes would be costly. The joint task assignment, communication rate and computation frequency allocation for a device-to-device (D2D)-enabled multi-helper MEC system was proposed by the authors in [23]. Also, to create a sub-optimal task assignment solution for the MINLP formulation and a benchmark scheme with fixed computation frequency and a greedy task assignment-based heuristic algorithm, they proposed a special convex-relaxation-based algorithm. This work only considered the users' cooperative computation under fixed energy supplies (e.g., batteries). In [24], a novel low-latency and trustworthy communication-computing system design was proposed to enable mission-critical applications by which the ultra-reliable low latency communications (URLLC) requirement has been formulated. In particular, relying solely on the average queue length did not meet the strict delay requirement for vehicle applications. In [25], a holistic strategy for a joint task offloading and resource allocation in a multi-cell MEC network was investigated by the researchers. To optimize the MINLP problem, the original problem was formulated into a resource-allocation problem with a fixed task offloading decision and a task offloading problem. This work did not consider the ultra-dense network and it was difficult to gain insight into the design of critical parameters. Authors in [26] decomposed the drone placement problem into two sub-problems and improved the latency ratio of the network. They placed drone base stations to the locations with higher user densities. In a dynamic network, each server had to process dynamically changing the amount of data load gathered in different clusters, which made the load on cluster servers unbalanced. In [27], by leveraging the vertical cooperation among devices, edge nodes and cloud servers with alternating-direction method of multipliers (ADMM) method and difference of convex functions (D.C.) programming, a three-tier cooperative computing network was inspected. The ADMM and D.C. programming-based methods were only sub-optimal. Authors in [28], using fog computing with low latency, performed the electric energy control in a microgrid. Their scheme proposed some services, including the proportional integral derivative (PID) controller and algorithm scheduling, to decrease consumers' bills and algorithms (FIFO and GA) using the PID calculations. In this work, just downlink was used. In [29], general joint computation offloading and resource allocation for the multiple-input multiple-output (MIMO)-based mobile cloud computing system considering perfect- channel state information (P-CSI) and imperfect-CSI (IP-CSI) were tackled by the researchers. In order to solve the underlying MINLP, the optimal and low-complexity algorithms were proposed. This work has discussed the network and communication resources individually, typically focusing independently on each of them. Authors in [30] inspected the optimization of offloading decision, local computation capability and computing resource allocation of fog node. The problem was decomposed into two independent sub-problems by them and an HGSA-based latency-minimum offloading decision algorithm was designed to tackle this MINLP problem with low complexity. Using traditional greedy search methods was challenging. In [31], the offline placement problem of IoT services supporting horizontal and vertical scaling in an edge computing environment was investigated. The authors formulated an MINLP problem and proposed linearization and genetic-based method to solve it. This article only minimized deadline violations due to limited resources at the edge. In [32], radio resource allocation between two network slices with heterogeneous performance metrics in fog radio access networks was investigated by the researchers and the problem as a Stackelberg game was modelled, where the global radio resource manager (GRRM) with a strong position acted as the leader and the local radio resource managers (LRRMs) of slices acted as followers. This work was inefficient in long-term resource allocation performance. Researchers in [33] proposed an advanced caching technique through which the energy efficiency and delays can be improved and an algorithm for load balancing in the advanced cached fog layer. It has become more challenging to connect and monitor many devices, the most critical feature of which is content security. In [34], the joint optimization of computation offloading decisions, service caching placement and system resource allocation was studied by the authors. The complicated MINLP problem was transformed to a pure 0-1 integer linear programming (ILP) problem and reduced-complexity algorithms were proposed. It may not work properly for multi-user systems. In [35], a declarative methodology, SecFog, was proposed, which may be used for quantitative assessment of the security level of multi-service application deployment to cloud-edge infrastructures. Also, an MINLP problem of placing application services for the purpose of assuring end-to-end delay constraints was formulated. The underlying model was limited to the security controls provided by the infrastructure. In [36], the authors studied the service-placement problem in fog computing using meta-heuristic approaches and proposed an improved parallel genetic algorithm. But, it seems to be costly to implement. In [37], researchers have proposed a multi-objective strategy including execution time, energy consumption and cost, based on a biogeography-based optimization algorithm, for MEC offloading to satisfied users' multiple requirements. Objective(s), network and environment of related works are shown in Table 1.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 09, No. 02, June 2023.

Table 1. Related works.

Ref.	**Objective(s)**	**Network**								**Environment**	
		MCC	**MFC**	**MEC**	**Fog/cloud system**	**Edge computing**	**Mobile edge-cloud computing**	**Fog computing**	**Cloud-edge infrastructures**	**Simulation**	**Real**
Vu et al., 2018 [18]	Energy and delay			*						*	
Rahmani et al., 2018 [19]	Energy efficiency, security, overall system intelligence, reliability performance, mobility and interoperability		*	*							*
Chen et al., 2018 [20]	Cost of energy, computation and delay	*								*	
Du et al., 2017 [21]	Cost, energy and delay				*					*	
Du et al., 2018 [22]	Computation-resource allocation, optimizing computation offloading, transmitting power allocation, resource block pattern assignment				*					*	
Xing et al., 2019 [23]	Energy and delay			*						*	
Liu et al., 2019 [24]	Power consumption and latency			*						*	
Tran and Pompili, 2018 [25]	Energy consumption			*						*	
Fan and Ansari, 2018 [26]	Latency				*					*	
Wang et al., 2019 [27]	Average task-duration subject			*						*	
Barros et al., 2019 [28]	Power demand and managing power production and response time				*					*	
Nguyen et al., 2019 [29]	Energy and delay						*			*	
Wang and Chen, 2020 [30]	Resource-allocation scheme, completion time and latency							*		*	
Maia et al., 2019 [31]	Potential violation of QoS requirements					*					*
Sun et al., 2019 [32]	Resource allocation							*		*	
Shahid et al., 2020 [33]	Energy and latency							*		*	
Bi et al., 2020 [34]	Energy and delay			*						*	
Forti et al., 2020 [35]	Security								*	*	
Wu et al., 2022 [36]	latency, cost and trust							*		*	
Li et al., 2022 [37]	time-energy consumption and cost			*						*	

3. Network Model

Figure 1 shows a three-layer fog computing system with N mobile devices and M cooperative fog nodes. A cloud server *V* accesses directly by mobile devices. A task can be offloaded by a mobile device to one of the fog nodes, the cloud, through a fog node or directly to the cloud server. The model used in our study is similar to that used in paper [17]. The explanation of the parameters is given in Table 2. At each time slot, mobile device i can request to offload a computing task. The mobile devices, fog. $I_i\left(D_i^i, D_i^0, C_i, t_i^r\right)$ nodes, as well as the cloud server meet the delay requirements and are eligible for work processing. So, there are three modes for processing tasks, including mobile devices or local mode, fog nodes mode and cloud server mode.

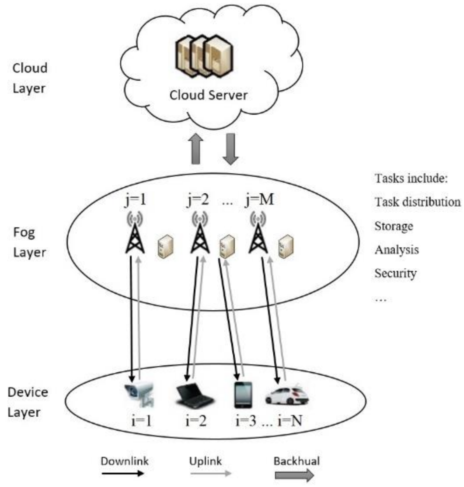


Figure 1. The architecture of fog-computing network.

Table 2. Explanation of parameters.

Parameters	Explanation
D_i^i	Input (including execution code and input data)
D_i^o	The lengths of output (result) data
C_i	The number of CPU cycles (required to execute the task)
t_i^r	The maximum delay requirement of the task
f_i^l	The processing rate of mobile devices i (cycles per second)
T_i^l	The time to perform the task
E_i^l	The energy consumed in the mobile devices (adequate to the CPU cycles required for task I_i) in local processing
v_i	The consumed energy per CPU cycle
R_j^u	The total uplink rate
R_j^d	The total downlink rate
R_j^f	CPU cycle rate
r_{ij}	Spectrum and computation resources for mobile device i
r_{ij}^f r_{ij}^d r_{ij}^u	The CPU cycle, downlink, uplink rates for task execution, input and output transmissions
e_{ij}^u	The energy consumption for transmitting a unit of data
e_{ij}^d	The energy consumption for receiving a unit of data
T_{ij}^f	The delay of mobile devices
E_{ij}^f	The energy consumed in the mobile devices in fog-node processing
$R_{(M+1)}^u$	The uplink rate by the cloud
$R_{(M+1)}^d$	The downlink rate by the cloud
$R_{(M+1)}^f$	The allocated CPU cycle rate by the cloud
w^c	The data rate between a fog node and the cloud server
f^c	The processing rate on the cloud server assigned to each task

3.1 Local Processing

The time to perform the task I_i when it is processed locally is:

$$T_i^l = C_i / f_i^l \tag{1}$$

where, C_i is the number of CPU cycles (required to execute the task) and f_i^l is the processing rate of mobile device i (cycles per second).

The energy consumed in the mobile device (E_i^l), adequate to the CPU cycles required for task I_i, is:

$$E_i^l = v_i C_i \tag{2}$$

and v_i denotes the consumed energy per CPU cycle.

3.2 Fog-node Processing

Fog node j has capabilities defined by a tuple (R_j^u, R_j^d, R_j^f). R_j^u, R_j^d, R_j^f are the total uplink rate, the total downlink rate and CPU cycle rate, respectively. If task I_i is processed at fog node j, this node will allocate spectrum and computation resources for mobile device i ($r_{ij} = (r_{ij}{}^u, r_{ij}{}^d, r_{ij}{}^f)$), where $r_{ij}{}^u$ is the uplink rate for output transmissions, $r_{ij}{}^d$ is the downlink rate for input transmissions for task execution and $r_{ij}{}^f$ is the CPU cycle. Hereon, the energy consumption of the mobile device is to transfer input to and receive output from the fog node j. The delay includes the time of task processing, transmitting input and receiving output at the fog node, so that they are $C_i / r_{ij}{}^f$, $D_i^i / r_{ij}{}^u$ and $D_i^o / r_{ij}{}^d$, respectively. D_i^i is input data and D_i^o is output data.

The delay $T_{ij}{}^f$ and the consumed energy E_{ij}^f of the mobile device are:

$$T_{ij}{}^f = D_i^i / r_{ij}{}^u + D_i^o / r_{ij}{}^d + C_i / r_{ij}{}^f \tag{3}$$

and

$$E_{ij}{}^f = E_{ij}{}^u + E_{ij}{}^d \tag{4}$$

where $E_{ij}{}^u = e_{ij}{}^u D_i^i$ and $E_{ij}{}^d = e_{ij}{}^d D_i^o$. The energy consumption for transmitting a unit of data and the energy consumption for receiving a unit of data are denoted by $e_{ij}{}^u$ and $e_{ij}{}^d$.

3.3 Cloud-server Processing

Assume that all fog nodes are connected to a public cloud server. If fog node j forwards task I_i to the cloud server, it will allocate resources for mobile device i, ($r_{ij} = (r_{ij}{}^u, r_{ij}{}^d, r_{ij}{}^f)$) and $r_{ij}^f = 0$. Fog node j sends the input data to the cloud server for processing after receiving the task. It then receives the result and sends it to the mobile device. Hereon, the consumed energy E_{ij}^c at the mobile device and the delay $T_{ij}{}^c$ are:

$$T_{ij}{}^c = D_i^i / r_{ij}{}^u + D_i^o / r_{ij}{}^d + (D_i^i + D_i^o) / w^c + C_i / f^c \tag{5}$$

and

$$E_{ij}{}^c = E_{ij}{}^f = E_{ij}{}^u + E_{ij}{}^d \tag{6}$$

where w^c is the data rate between a fog node and the cloud server and f^c is the processing rate on the cloud server assigned to each task.

Because the cloud is in the top tier, it cannot move its task to the top tier. This is accomplished by setting the corresponding latency to infinite: $T_{i(M+1)}^c = \infty$. The total energy consumption $E_{i(M+1)}^c$ is set as a constant. The binary offloading decision variable for task I_i is $x_i = (x_i^l, x_{i1}{}^f, \dots, x_{i(M+1)}{}^f, x_{i1}{}^c, \dots, x_{i(M+1)}{}^c)$, in which $x_i^l = 1$, $x_{ij}^f = 1$ and $x_{ij}^c = 1$ indicate that task I_i is processed at the mobile device, fog node j or the cloud server, respectively. Suppose that $h_i = (T_i^l, T_{i1}{}^f, \dots, T_{i(M+1)}{}^f, T_{i1}{}^c, \dots, T_{i(M+1)}{}^c)$. From (1), (3) and (5), the delay T_i when task I_i is processed is equal to:

$$T_i = h_i^T x_i \tag{7}$$

Suppose that $e_i = (E_i^l, E_{i1}^f, \dots, E_{i(M+1)}^f, E_{i1}^c, \dots, E_{i(M+1)}^c)$. From (2), (4) and (6), the consumed energy E_i of the mobile device when task I_i is processed is:

$$E_i = e_i^T x_i \tag{8}$$

Assuming that $e = (e_1, \dots, e_N)$ and $x = (x_1, \dots, x_N)$, the consumed energy E_i of the mobile device is as follows:

$$E = e^T x \tag{9}$$

The purpose of our paper is to minimize the total energy consumption of all mobile devices in the delay requirement, which is a joint offloading decision (x) and resource allocation ($r = \{r_{ij}\}$) problem:

$$\text{Min}_{x,r} e^T x \tag{10}$$

the restrictions of which are equal to:

$$\begin{cases} (\text{C1}) & T_i \le t_i^r, \forall i \in N \\ (\text{C2}) & \sum_{i=1}^{N} r_{ij}^f \le R_j^f, \forall j \in \text{M}^*, \\ (\text{C3}) & \sum_{i=1}^{N} r_{ij}^u \le R_j^u, \forall j \in \text{M}^*, \\ (\text{C4}) & \sum_{i=1}^{N} r_{ij}^d \le R_j^d, \forall j \in \text{M}^*, \\ r_{ij}^f, r_{ij}^u, r_{ij}^d \ge 0, \forall (i,j) \in N \times \text{M}^* \end{cases} \tag{11}$$

and

$$\begin{cases} (\text{C5}) \;\; x_i^l + \sum_{j=1}^{M+1} x_{ij}^f + x^c = 1, \forall i \in N, \\ x_i^l, x_{ij}^f, x_{ij}^c \in \{0,1\}, \forall (i,j) \in N \times \text{M}^* \\ (\text{C6}) \quad T_i^l < t_i^r \\ (\text{C7}) \quad T_{ij}^c < t_i^r \\ (\text{C8}) \quad T_{ij}^f < t_i^r \end{cases} \tag{12}$$

where (C1) is the delay requirement of tasks. Resource constraints at fog nodes are (C2), (C3) and (C4). Offloading decision constraints are (C5) and (C6) to (C8), indicating that the task delay for each mobile device should not exceed the maximum value (t_i^r).

According to the equations presented in [16], the revenue of the service provider (cloud) leads to better services for member users. Another factor in measuring cost performance is the price that users are charged as benefit/revenue. The price offered by each user depends on the delay requirement T_{ij} and data size D_i^i and D_i^o, where we assume a linear relationship between the price and the data size and an inversely linear relationship between the price and the delay requirement. Therefore, the offer from each user is equal to:

$$O_{ij} = f\left(D_i^i, D_i^o, T_{ij}\right) \tag{13}$$

where the function $f()$ must be a monotonic increasing function for D_i^i and D_i^o must be a monotonic increasing function for T_{ij}. For simplicity, the following function is used to define $f\left(D_i^i, D_i^o, T_{ij}\right)$:

$$O_{ij} = a \frac{D_i^i + D_i^o}{T_{ij}} \tag{14}$$

where a is a parameter with the unit of dollar/Mbps and O_{ij} is the price that the user of the mobile device i pays if it is compatible with V. Because V serves more than one user, it receives more than one offer. Revenue V is the sum of offers defined by all users. For simplicity, it is assumed that the cost of V is related to the power consumption of the transmission and its maintenance; in this work, it is fixed. In revenue V, the impact of fixed-service costs is ignored. The system objective in our article is named cost performance. Cost performance is defined as the ratio of the average data rate of each user to its cost in Mbps/sec/dollar.

The actual delay value is related to the size of the data D_i^i and D_i^o that must be transmitted and processed. Then, for the cost factor, use the user's payment/offer for the relevant fog calculation service that it acquires. As a result, the cost performance function is defined for combining two factors in one criterion for each user, which indicates the quality of services for which the user pays. The cost performance system (CP_{sys}) is equal to:

$$CP_{sys} = \frac{\sum_{u_i \in U} CP_{ij}}{N} \tag{15}$$

where CP_{ij} is the cost performance value for each mobile device i and is shown as follows:

$$CP_{ij} = x_{ij}^f \frac{\left(D_i^i + D_i^o\right) / T_{ij}^f}{O_{ij}} \tag{16}$$

The optimization problem of our article is shown below:

$$Max \quad CP = \sum_{ij} x_{ij}^f \frac{(D_i^i + D_i^o)/T_{ij}^f}{o_{ij}} \tag{17}$$

the restrictions of which are equal to:

$$\begin{cases} (C9) \ \sum_{ij} x_{ij}^f \times t_{ij}^f < t_i^r \\ (C10) \ \sum_{ij} x_{ij}^f \times r_{ij}^f \le R_j^f \\ (C11) \ \sum_{ij} x_{ij}^f \times r_{ij}^u \le R_j^u \\ (C12) \ \sum_{ij} x_{ij}^f \times r_{ij}^d \le R_j^d \end{cases} \tag{18}$$

where (17) is the system objective and shows the overall system cost performance for users. (C9) indicates that each user who has been served has a task delay less than the maximum delay requirement of the task (t_i^r). In (C10), (C11) and (C12), for each user who has been served, the CPU cycle, downlink and uplink rates for task execution and output and input transmissions of all users must be less than the CPU cycle rate (R_j^f), the total uplink rate and the total downlink rate.

4. PROBLEM SOLVING

The Experimental parameters are shown in Table 3 and are similar to the experimental parameters in [17]. GAMZ solves the problem as a single-objective. The weighted-sum method is used to turn our two-objective problem into a single-objective problem. Eq. (19) has two goals: to increase cost performance and reduce energy consumption. Since one of the goals of the problem is maximization, to achieve the ultimate goal of the problem, which is a minimization of the objective function, must be multiplied by -1. First, through the weighted sum method, two weights with a value of 0.5 are assigned to the objective function and are added up to become a minimized objective function. The choice of weight for the objective function depends on the priority and importance of the selected objective function. Because both cost performance and energy consumption are equally important in this paper, a factor of 0.5 is considered for both.

$$Min \quad (-0.5) \times \sum_{ij} x_{ij}^f \frac{\left(D_i^i + D_i^o\right)/T_{ij}^f}{o_{ij}} + 0.5 \times e^T x \tag{19}$$

In the next step, inside the GAMS, for a small model with five mobile devices, four fog nodes and a cloud, the model is entered into the program. Indices are displayed in GAMS with sets, which in this article include i for mobile devices at 5, j for fog nodes at 3, k for cloud server at 1 and m is an auxiliary index at 4. After entering the parameters, decision variables and sets, it is time to choose the appropriate solver. Both Eq. (10) and Eq. (17) are nonlinear and MINLP. Because the sum of Eq. (10) and (17) is nonlinear and MINLP, Eq. (19) is also nonlinear and MINLP. In this case, because our problem is MINLP, which is generally NP-hard to solve [38], the best solver is Baron, which solves these kinds of problems with a good track record. After execution, the results are as follows: the decision variables are quantified, the optimal answer is obtained and resources are allocated to the devices. The decision variable x^f determines which mobile is served by which device. The lower bound is set r_{ij}^u, r_{ij}^d and r_{ij}^f from 0.01, because in equations where r_{ij}^u, r_{ij}^d and r_{ij}^f are at the denominator of the fraction, they cannot have a value of zero. As described, the weighting method is used to balance the objects. Considering w1, w2 with values of 0.5 and 0.5, the lowest amount obtained for the objective function was -1624.439.

Table 3. Experimental parameters.

Parameters	Value
Number of mobile devices N	10
Number of fog nodes M	4
Number of cloud servers V	1
CPU rate (in mobile devices) f_i^l	0.5 Giga cycles/s
Processing energy consumption rate v_i	1000/730 J/Giga cycles
Input data size D_i^i	U(a, b) MB
Output data size D_i^o	U(c, d) MB
Required CPU cycles C_i	$\alpha_i \times D_i^i$
Unit transmission energy consumption to fog nodes e_{ij}^u ($\forall j \le M$)	0.142 J/Mb
Unit receiving energy consumption from fog nodes e_{ij}^d ($\forall j \le M$)	0.142 J/Mb
Unit transmission energy consumption to cloud server V ($e_{i(M+1)}^u$)	0.658 J/Mb
Unit receiving energy consumption from cloud server V ($e_{i(M+1)}^d$)	0.278 J/Mb
Delay requirement t_i^r	[1, 10]s
Processing rate (each fog node) R_j^f	10 Giga cycles/s
Uplink data rate (each fog node) R_j^u	72 Mbps
Downlink data rate (each fog node) R_j^d	72 Mbps
CPU rate (the cloud server) f^c	10 Giga cycles/s
Data rate between FNs and the cloud w^c	5 Mbps
A parameter with the unit dollar/Mbps a	1
Delay requirement T_{ij}	[6, 7]s

5. EVALUATION

In this section, the implementation of meta-heuristic algorithms including evaluation setup, experimental results and results' analysis, is presented.

5.1 Evaluation Setup

The proposed model was implemented by use of PSO [10], GA [11], SA+GA [12], TLBO [13], GWO [14], GOA [15] and random search method for thirty problems of different sizes to solve our single-objective problem (19). The proposed models are solved using meta-heuristic algorithms to solve problems in small, medium and large sizes. Because the problem is of the MINLP type, meta-heuristic algorithms have a good track record of solving this type of problem. Also, the model is based on population and the algorithms used in this paper are the same. Meta-heuristic algorithms have acceptable speed and accuracy in finding the optimal solution in resource-allocation models. These algorithms are scalable for resource-allocation optimization problems (for more iterations and larger populations) [39]-[40]. Programming and execution of algorithms are also carried out using MATLAB version (R2016a). The algorithms run on a 64-bit system with a 2.5 GHz processor and a 2 GB memory.

To investigate the performance of the proposed model, PSO [10], GA [11], SA+GA [12], TLBO [13], GWO [14], GOA [15] and random method have been used. Then, a comparison of the results obtained from each of these methods is performed. To do this, different instances of different sizes must first be designed. The parameters of each algorithm are given in Table 4.

Table 4. The parameters of the algorithms.

Algorithm	**Parameters**	**Values**
PSO	maximum number of iterations	100
	population size (swarm size)	50
	inertia weight	1
	inertia weight damping ratio	0.99
	personal rating coefficient	2
	global rating coefficient	2
GA	maximum number of iterations	100
	population size	50
	Cross-over percentage	0.8
	number of off-springs	2×round (cross-over percentage × population size/2)
	mutation percentage	0.3
	number of mutants	round (mutation percentage × population size)
	gamma	0.05
	mutation rate	0.02
	beta	0.5
SA+GA	maximum number of iterations	100
	maximum number of sub-iterations	10
	initial temp.	10
	temp reduction rate	0.99
	population size	50
	Cross-over percentage	2
	number of parents (off-springs)	2×round (cross-over percentage × population size/2)
	mutation percentage	0.3
	number of mutants	round (mutation percentage × population size)
	Cross-over inflation rate	0.05
	mutation rate	0.02
	mutation mode	"rand"
	eta	0.1
TLBO	maximum number of iterations	100
	population size	50
GWO	maximum number of iterations	100
	population size	50
GAO	maximum number of iterations	100
	population size	50

The computational complexities of PSO [10], GA [11], SA+GA [12], TLBO [13], GWO [14], GOA [15] are O(nlogn),O(nm),O(nm),O(ldg),O(nm) and O(m^3), respectively, where n is the number of population, m is the size of individuals, d is the number of subjects, g is the number of iterations and l is the number of learners.

In this research, instances in small, medium and large sizes are designed and the size of each parameter for each type of example is classified as follows (*i* for the number of mobile devices and *j* for the number of fog nodes):

- For small instances, the set $i = 5$ and $j = 3$ with 60 variables,
- For medium instances, the set $i = 15$ and $j = 5$ with 300 variables,
- For large instances, the set $i = 100$ and $j = 15$ with 6000 variables.

5.2 Experimental Results and Results' Analysis

The results related to the values of the objective function and the solution times for these algorithms are given in Table 5 and Table 6.

Table 5. The values of the objective function.

Problem type	No.	GA	PSO	SA+GA
Small problem	1	-3,624	-3,239	-7537250.18
	2	-5,137	-4,633	-26513.6
	3	-5,582	-4,840	-1128
	4	-3,045	-2,912	-15617.6
	5	-5,239	-4,782	-11922.9
	6	-3,488	-2,846	-608.9
	7	-4,823	-3,925	-579.9
	8	-3,256	-2,885	-69918.4
	9	-7,144	-6,547	-21484.3
	10	-6,739	-5,168	-11940.5
	11	-4,958	-4,151	-17363.5
	12	-5,786	-5,004	-5701.3
medium problem	13	-493,059	-435,452	-412371.2
	14	-626,800	-591,089	-446073.1
	15	-829,385	-781,866	-511573.6
	16	-743,971	-694,826	-648282.3
	17	-720,405	-685,978	-2437.3
	18	-534,126	-493,097	-397386.3
	19	-694,271	-562,966	-439611.8
	20	-495,543	-402,779	-211423.9
	21	-413,253	-354,656	-2813.9
	22	-748,903	-687,841	-315033.7
large problem	23	-3,433,470	-3,028,605	-49497370.4
	24	-3,737,970	-3,478,755	-19938346.9
	25	-3,891,427	-3,606,192	-31052647
	26	-3,759,422	-3,313,783	-14822865
	27	-3,027,824	-2,831,833	-32319789.1
	28	-3,487,582	-3,117,100	-29267467.9
	29	-2,517,827	-2,390,396	-27467320.4
	30	-4,728,847	-4,310,443	-32405902.4
Problem type	**TLBO**	**GOA**	**GOW**	**Random**
Small problem	-26848.6	-9506.1	-1436	-110.35
	-33697.5	-21977.9	-65731.6	-74.49
	-943.8	-1062.6	-1216.7	-221.76
	-2744.9	-5338.4	-13687.9	-11.25
	-4662.5	-8130.7	-984	-119.24
	-22366.2	-13209.1	-775.6	-67.47
	-1084.6	-7559.5	-3531.5	-32.69
	-78935	-4933.3	-33810.1	-84.93
	-1659.9	-17399.8	-12122.8	-210.67
	-7981.7	-14711.6	-24042.3	-203.19
	-8174.7	-759.4	-83975.9	-118.37
	-3080.8	-12755.8	-1591.3	-19.67
medium problem	-14450.6	-25405.9	-2501.9	309.11
	-501673.5	-35910	-71205.4	397.31
	-133464.3	-2098.5	-134118.1	100.33
	-136988.4	-165627.6	-93339.8	115.60
	-63350.8	-86799	-270145	220.36
	-199839.4	-82712.8	-75138.9	79.14

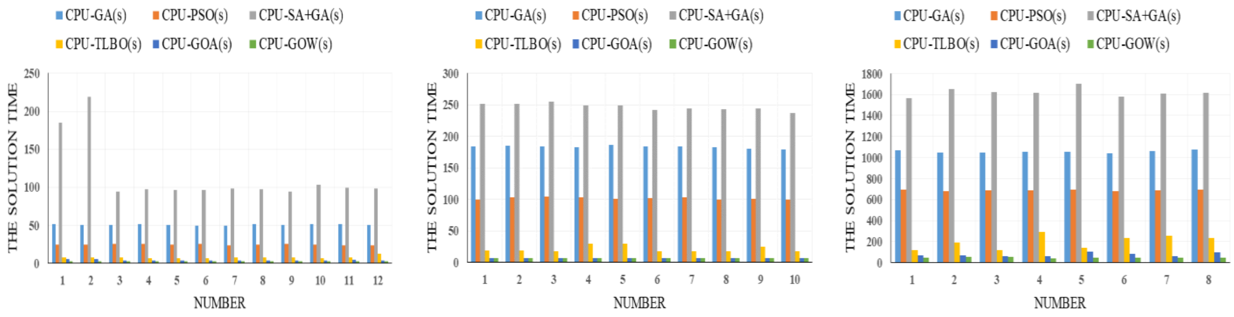
	-192184.3	-216008.7	-82738.5	19.35
	-148564.8	-206550.1	-138484.9	10.46
	-147154.4	-148449.6	-190762.6	24.89
	-166501.5	-70551.7	-65520.3	114.49
large problem	-3620749.4	-881734	-4128805	9.17
	-4048094.4	-858796.8	-504324.6	14.11
	-5092522.9	-830234.3	-776526	-25.21
	-4557760.8	-656639	-4638037	-7.82
	-2747554	-794339.2	-3261333.7	10.35
	-3628517.4	-361017.3	-3628227.1	-2.56
	-4323419	-946594.2	-4020436.5	-18.59
	-4323411	-1013361.6	-4952766.4	-4.90

In Table 5, the values of the objective function are shown. As can be seen in Table 5, the problem is investigated in small, medium and large dimensions. Cost is obtained at each stage and the larger the problem, the lower and more convergent the cost function becomes. Because the presence of all meta-heuristic algorithms at every stage, these algorithms try to get a more optimal answer. The solution times are indicated in Table 6 and Figure 2.

Table 6. The values of the solution time.

Problem type	No.	CPU-GA(s)	CPU-PSO(s)	CPU-SA+GA(s)	CPU-TLBO(s)	CPU-GOA(s)	CPU-GOW(s)
Small problem	1	51.11	24.77	185.5	7.3	5.3	2.9
	2	50.64	24.53	219.4	7.1	5.6	2.5
	3	50.37	25.25	94.2	7.4	3.5	2.6
	4	50.96	25.27	97.3	7	3.7	2.5
	5	50.44	24.48	96.3	7	3.9	2.5
	6	49.82	25.46	96.5	6.9	3.4	2.6
	7	49.88	23.69	98.1	7.5	3.6	2.5
	8	51.03	24.55	97.3	7.3	3.4	2.5
	9	50.81	25.29	93.9	7.4	3.4	2.8
	10	51.07	24.01	103.2	7	3.4	2.5
	11	51.24	23.34	99.8	7.1	4.8	2.5
	12	50.15	23.35	97.9	12.7	3.4	2.5
medium problem	13	184.12	99.56	251.5	18.7	7.1	6.4
	14	185.44	103.23	251.2	18.2	7.1	6.3
	15	183.34	103.83	255.2	17.6	6.9	7
	16	183.02	103.03	248.4	29.4	7.1	6.3
	17	186.32	100.16	248.6	29.8	7.1	6.4
	18	184.28	102.43	241.6	17.7	6.9	6.4
	19	184.06	102.99	244	17.8	7.1	6.3
	20	182.29	99.46	243	17.8	6.9	6.3
	21	180.19	100.85	244.7	25	6.9	6.3
	22	179.27	99.04	237.4	17.8	7	6.3
large problem	23	1068.48	693.68	1566.1	118.5	67.8	46.6
	24	1045.26	681.12	1651.1	193.1	67.5	58
	25	1049.38	687.39	1621.5	120.7	61.2	52.4
	26	1053.51	689.14	1617.7	294.2	60.1	44.2
	27	1054.37	694.76	1705.1	141.2	102.5	47.1
	28	1038.45	681.16	1577.4	237.2	83.7	50.3
	29	1062.94	686.94	1606.5	255.8	65.1	45.7
	30	1078.55	694.17	1616.1	238.2	99.6	46.4

In Table 6, due to the fact that small problems have fewer variables and parameters and the number of fogs and users is less, the execution time of the algorithm is shorter and they are executed faster than the next steps when the number of fogs and users increases. Finally, in larger problems, the number of variables reaches 6,000, which will certainly be solved in more time. It should be noted that to eliminate the uncertainty in the obtained outputs, each problem was performed three times and the average of these three problems was reported as the final answer variable.



a b c

Figure 2. The values of the solution time for a) small problems, b) medium problems and c) large problems.

The plot of a problem with fifty mobile users, ten fog nodes and a cloud sever is shown in Figure 3, while the plot of a problem with one hundred mobile users, ten fog nodes and a cloud sever is shown in Figure 4. As you can see in Figure 3, in a number of iterations below 10, the SA+GA [12] graph fluctuates upwards, which can be due to the lower stability of this algorithm in such number of iterations. In Figure 3 and Figure 4 in each step, the meta-heuristic algorithms perform the optimization process and try to reduce the total cost in each step and obtain a more optimal value. In fact, this leads to the most accurate answer, allocating resources of a certain size with the minimum delay to users through fog nodes or directly to the cloud server.

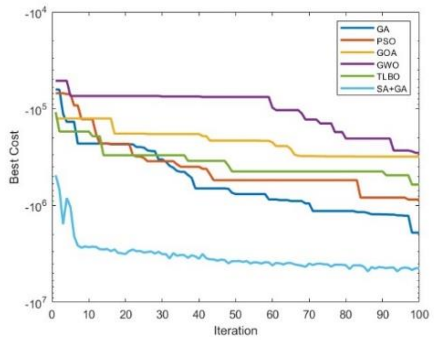


Figure 3. The value of the objective function in the number of iterations for fifty mobile users, ten fog nodes and a cloud sever.

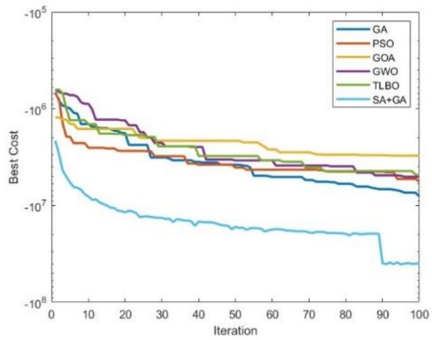


Figure 4. The value of the objective function in the number of iterations for one hundred mobile users, ten fog nodes and a cloud sever.

To study and compare the proposed algorithms accurately, decision-making with multiple criteria has been used. Multiple-attribute decision making (MADM) method is a multi-criteria decision-making method; by which TOPSIS method is used for comparing algorithms with various indicators. Since this method has been used in a variety of articles and ended in excellent and accurate results, it is also used here for comparison. In this model, two indicators of execution time (CPU time) and the value of the objective function are considered for comparison. Since not all the amounts can be taken into account due to their large number, the average of each column is obtained and then the values are entered into TOPSIS. On the other hand, because an algorithm is better in terms of each indicator, there is no such decision-making ability to choose a better algorithm. The result of TOPSIS method is shown in Table 7. According to Table 7 of the TOPSIS comparison, compared to other methods, the SA+GA method with a value of 0.23 is deemed the best. Then come GWO [14], GA [11], TLBO [13], PSO [10] and GOA [15] methods, respectively.

Table 7. The result of TOPSIS method.

Algorithms	Ranking
SA+GA	0.23
GWO	0.197
GA	0.167
TLBO	0.146
PSO	0.117
GOA	0.104

In Figure 5 (with ten fog nodes), as the number of mobile devices increases, the cost performance increases too. The highest cost performance is for the GWO [14], GOA [15], TLBO [13], SA+GA [12], PSO [10] and GA [11], respectively. GA has the lowest gradient, because it has a slower convergence. GWO has the highest cost performance increase because of its high convergence and iterability. With the increase in the number of users, the cost performance has increased as well, because more users try to get services and the size of input-output data (D_i^i and D_i^o) increases too and as the goal is to increase cost performance, the latency decreases at each stage.

In Figure 6 (with ten fog nodes), as the number of mobile devices increases, the energy consumption increases too. The highest energy consumption is for the GWO [14], GOA [15], TLBO [13], SA+GA [12], PSO [10] and GA [11], respectively. In Figure 6, because of the power of convergence and iterability, GWO has the highest and GA the lowest power consumption. Given the increase in the number of users, the increase in the need for services and the allocation of more resources, the number of users connecting to fog nodes has increased, while the delay may increase with regard to the constant number of fog nodes. So, energy consumption increases.

In Figure 7 (with forty mobile devices), as the number of fog nodes increases, the cost performance increases too. The highest cost performance is for the GWO [14], GOA [15], TLBO [13], SA+GA [12], PSO [10] and GA [11], respectively. With the number of fog 5, GOA has the highest cost performance, because the algorithm has lower convergence and TLBO performs better due to greater stability.

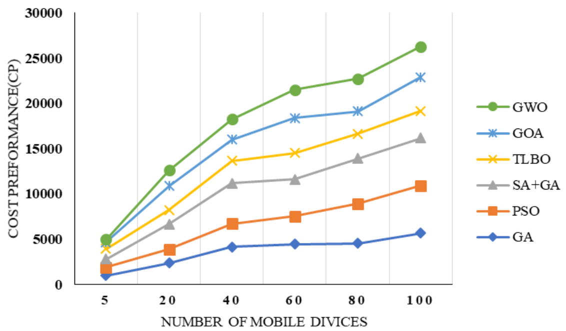


Figure 5. The cost performance with ten fog nodes.

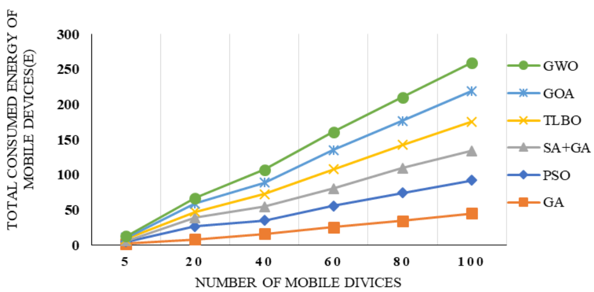


Figure 6. The energy consumption with ten fog nodes.

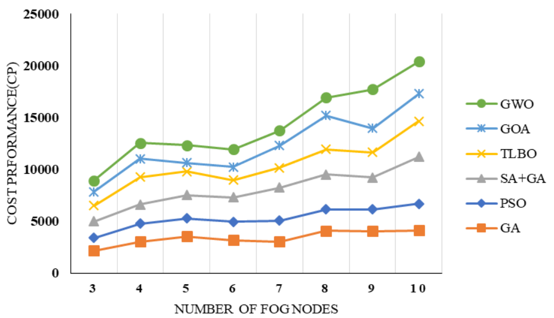


Figure 7. The cost performance with forty mobile devices.

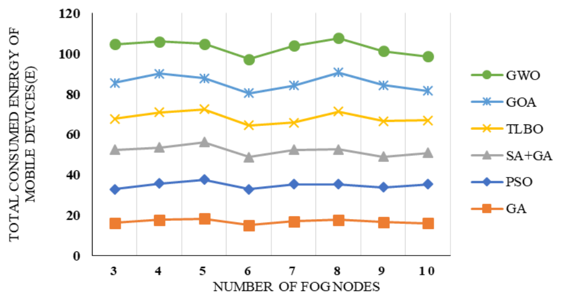


Figure 8. The energy consumption with forty mobile devices.

With the number of fog nodes of 9, the GOA again has the highest performance decline, which indicates the instability of this pattern compared to the rest. Figure 7 shows a slight upward slope of cost performance with increasing number of fog nodes. When the number of users is considered constant, with an increase of one fog node, the cost performance is slightly increased due to the fact that the size of input-output data (D_i^i and D_i^o) is slightly increased. As the number of available fog nodes increases, work is done faster and better load balancing occurs on the network, so better cost performance occurs at each step.

In Figure 8 (with forty mobile devices), as the number of fog nodes increases, the energy consumption increases too. The highest energy consumption is for the GWO [14], GOA [15], TLBO [13], SA+GA [12], PSO [10] and GA [11], respectively. All algorithms with the number of fog nodes of 6 have a reduction in consumed energy, which can be related to the number of users that have been set in the parameters and the initial amount of resources (R_j^u, R_j^d, R_j^f). With increasing the number of fog nodes, the changes in energy consumption remain constant, because the number of users has not changed, so the number of requests remains almost constant. When sending a request to fog nodes, the fog node that is closer to the user has a higher priority to communicate with the intended user, so changes in energy consumption remain almost constant.

6. CONCLUSIONS

Cloud computing plays an essential role in developing the Internet of Things, which provides data-processing and storage services. However, many of its applications suffer from cloud-computing challenges, such as latency, location awareness and real-time mobility support. Fog computing, which almost looks like the evolution of cloud computing, helps provide solutions to these challenges. In this paper, due to their service latency limitations, the total energy of all mobile devices is minimized and the cost performance is maximized at the same time. This generally leads to the increase in the quality of service (QoS) and in the quality of experience (QoE). A model is designed and the exact mathematical method with the GAMS program is used to prove its logical process. In the next step, the model is solved with GA, PSO, SA+GA, TLBO, GOA, GWO and random methods. To do this, sproblems are considered with three sizes: small, medium and large. The six main algorithms (GA, PSO, SA+GA, TLBO, GOA and GWO) are compared with two indicators; the value of the objective function and the execution time. According to the TOPSIS comparison, the SA+GA method with a value of 0.23 is the best one compared to other methods. Then come GWO, GA, TLBO, PSO and GOA methods, respectively. In this paper, the customer affairs is one of the practical applications of this modeling, in terms of meeting the needs of customers, improving the level of loyalty and establishing mutual, transparent, respectful relationships and satisfying them, which are among the priorities of the customer relationship center. Communication with customers and the method of responding to their needs and requests should be well managed. Also, the cost performance and energy consumption should be optimized. In project management, resource allocation means the distribution of available resources in the company (equipment, labor, budget, facilities, …etc.). In order to perform the tasks related to project management, two objectives of the problem and its solutions are implemented. Organizational resource management, in fact, seeks effective planning and control by modeling and providing solutions. All affairs related to receiving, producing and delivering to customers should be done for production, distribution and service companies with the highest cost performance and the lowest energy consumption. In fact, one of the concerns of this modeling is to provide the customer's desired service with the least delay and in a reasonable time. For future work, more goals can be added to the model to bring the system closer to reality. More attributes for comparison should be added. Also, the architecture of the fog computing network can be changed and newer technologies can be used in it. Customer service can be prioritized, too. New methods including machine learning and deep learning can be used to assign tasks intelligently and analyze data by fog computing. Also, neural networks can be used for the problems of load balancing and task scheduling.

REFERENCES

[1] G. Javadzadeh and A. M. Rahmani, "Fog Computing Applications in Smart Cities: A Systematic Survey," Wireless Networks, vol. 26, no. 2, pp. 1433–1457, 2020.

[2] A. Yousefpour et al., "All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete Survey," J. of Systems Architecture, vol. 98, pp. 289–330, 2019.
[3] R. W. Cottle and M. N. Thapa, Linear and Nonlinear Optimization, 1st Edn., New York, Springer, 2017.
[4] S.-A. N. Alexandropoulos, C. K. Aridas, S. B. Kotsiantis and M. N. Vrahatis, "Multi-objective Evolutionary Optimization Algorithms for Machine Learning: A Recent Survey," Chapter in Book: Approximation and Optimization, vol. 145, pp. 35–55, 2019.
[5] K. Deb and K. Deb, "Multi-objective Optimization," Chapter in Book: Search Methodologies, pp. 403–449, DOI: 10.1007/978-1-4614-6940-7_15, 2014.
[6] G. Colson and C. de Bruyn, "Models and Methods in Multiple Objectives' Decision Making," Math. Comput. Model., vol. 12, no. 10–11, pp. 1201–1211, 1989.
[7] P. Muts et al., "The Decomposition-based Outer Approximation Algorithm for Convex Mixed-integer Nonlinear Programming," J. Global Optimization, vol. 77, no. 1, pp. 75–96, 2020.
[8] M. R. Bussieck and A. Meeraus, "General Algebraic Modeling System (GAMS)," Proc. of Modeling Languages in Mathematical Optimization, vol. 88, pp. 137-157, 2004.
[9] A. Brooke, D. Kendrick, A. Meeraus, R. Raman and U. America, "The General Algebraic Modeling System," GAMS Development Corporation, 1050, 1998.
[10] Rajani, D. Kumar and V. Kumar, "Impact of Controlling Parameters on the Performance of MOPSO Algorithm," Procedia Computer Science, vol. 167, pp. 2132–2139, 2020.
[11] Li, Mincan et al., "Multi-task Allocation with an Optimized Quantum Particle Swarm Method," Applied Soft Computing, vol. 96, p. 106603, 2020.
[12] P. J. Van Laarhoven and E. H Arts, "Simulated Annealing: Theory and Applications," Springer, Part of the Book Series: Mathematics and Its Applications, vol. 37, pp. 7-15, 1987.
[13] R. V. Rao et al., "Teaching– Learning-based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems," Computer Aided Design, vol. 43, no. 3, pp. 303–315, 2011.
[14] S. Mirjalili et al., "Grey Wolf Optimizer," Advances in Eng. Software, vol. 69, pp. 46–61, 2014.
[15] S. Saremi, S. Mirjalili and A. Lewis, "Grasshopper Optimization Algorithm: Theory and Application," Advances in Eng. Software, vol. 105, pp. 30–47, 2017.
[16] Y. Gu, Z. Chang, M. Pan, L. Song and Z. Han, "Joint Radio and Computational Resource Allocation in IoT Fog Computing," IEEE Transactions on Vehicular Technol., vol. 67, no. 8, pp. 7475–7484, 2018.
[17] T. T. Vu et al., "Optimal Energy Efficiency with Delay Constraints for Multi-layer Cooperative Fog Computing Networks," IEEE Transactions on Communications, vol. 69, no. 6, pp. 3911–3929, 2021.
[18] T. T. Vu, N. Van Huynh, D. T. Hoang, D. N. Nguyen and E. Dutkiewicz, "Offloading Energy Efficiency with Delay Constraint for Cooperative Mobile Edge Computing Networks," Proc. of the 2018 IEEE Global Comm. Conf. (GLOBECOM), DOI: 10.1109/GLOCOM.2018.8647856, Abu Dhabi, UAE, 2018.
[19] A. M. Rahmani et al., "Exploiting Smart e-Health Gateways at the Edge of Healthcare Internet-of-Things: A Fog Computing Approach," Future Generation Computer Systems, vol. 78, pp. 641–658, 2018.
[20] M.-H. Chen, M. Dong and B. Liang, "Resource Sharing of a Computing Access Point for Multi-user Mobile Cloud Offloading with Delay Constraints," IEEE Trans. on Mobile Computing, vol. 17, no. 12, pp. 2868–2881, 2018.
[21] J. Du, L. Zhao, J. Feng and X. Chu, "Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems with Min-max Fairness Guarantee," IEEE Trans. on Communications, vol. 66, no. 4, pp. 1594–1608, 2018.
[22] J. Du, L. Zhao, X. Chu, F. R. Yu, J. Feng and Chih-Lin, "Enabling Low-latency Applications in LTE-A Based Mixed Fog/Cloud Computing Systems," IEEE Trans. on Vehicular Technol., vol. 68, no. 2, pp. 1757–1771, 2019.
[23] H. Xing, L. Liu, J. Xu and A. Nallanathan, "Joint Task Assignment and Resource Allocation for D2D-enabled Mobile-edge Computing," IEEE Trans. on Communications, vol. 67, no. 6, pp. 4193–4207, 2019.
[24] C.-F. Liu et al., "Dynamic Task Offloading and Resource Allocation for Ultra-reliable Low-latency Edge Computing," IEEE Trans. on Communications, vol. 67, no. 6, pp. 4132–4150, 2019.
[25] T. X. Tran and D. Pompili, "Joint Task Offloading and Resource Allocation for Multi-server Mobile-edge Computing Networks," IEEE Trans. Vehicular Technol., vol. 68, no. 1, pp. 856–868, 2019.
[26] Q. Fan and N. Ansari, "Towards Traffic Load Balancing in Drone-assisted Communications for IoT," IEEE Internet of Things J., vol. 6, no. 2, pp. 3633–3640, 2019.
[27] Y. Wang, X. Tao, X. Zhang, P. Zhang and Y. T. Hou, "Cooperative Task Offloading in Three-tier Mobile Computing Networks: An ADMM Framework," IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2763–2776, 2019.
[28] E. B. C. Barroset al., "Fog Computing Model to Orchestrate the Consumption and Production of Energy in Microgrids," Sensors (Basel), vol. 19, no. 11, p. 2642, 2019.
[29] N. T. Ti, L. B. Le and Q. Le-Trung, "Computation Offloading in MIMO Based Mobile Edge Computing Systems under Perfect and Imperfect CSI Estimation," IEEE Trans. on Services Computing, vol. 14, no. 6, pp. 2011–2025, 2021.

[30] Q. Wang and S. Chen, "Latency-minimum Offloading Decision and Resource Allocation for Fog-enabled Internet of Things Networks," Trans. on Emerging Telecomm. Techn., vol. 31, no. 12, p. e3880, 2020.

[31] A. M. Maia et al., "Optimized Placement of Scalable IoT Services in Edge Computing," Proc. of the 2019 IFIP/IEEE Sympo. on Integrated Network and Service Manag. (IM), pp. 189-197, Arlington, USA, 2019.

[32] Y. Sun, M. Peng, S. Mao and S. Yan, "Hierarchical Radio Resource Allocation for Network Slicing in Fog Radio Access Networks," IEEE Trans. on Vehicular Technol., vol. 68, no. 4, pp. 3866–3881, 2019.

[33] M. H. Shahid et al., "Energy and Delay Efficient Fog Computing Using Caching Mechanism," Computer Communications, vol. 154, pp. 534–541, 2020.

[34] S. Bi et al., "Joint Optimization of Service Caching Placement and Computation Offloading in Mobile Edge Computing Systems," IEEE Trans. on Wireless Comm., vol. 19, no. 7, pp. 4947–4963, 2020.

[35] S. Forti, G.-L. Ferrari and A. Brogi, "Secure Cloud-edge Deployments with Trust," Future Generation Computer Systems, vol. 102, pp. 775–788, 2020.

[36] B. Wu, X. Lvl, W. D. Shamsi and E. G. Dizicheh, "Optimal Deploying IoT Services on the Fog Computing: A Metaheuristic-based Multi-objective Approach," J. King Saud Univ.-Computer Inf. Sci., vol. 34, no. 10, pp. 10010-10027, 2022.

[37] H. Li, P. Zheng, T. Wang, J. Wang and T. Liu, "A Multi-objective Task Offloading Based on BBO Algorithm under Deadline Constrain in Mobile Edge Computing," Cluster Computing, DOI: 10.1007/s10586-022-03809-7, 2022.

[38] D. Bertsimas and J. N. Tsitsiklis, Introduction to Linear Optimization, ISBN: 1886529191, Athena Sci., 1997.

[39] M. Abdel-Basset et al., "Energy-aware Metaheuristic Algorithm for Industrial IoT Task Scheduling Problems in Fog Computing Applications," IEEE IoT J., vol. 8, no. 16, pp. 12638–12649, 2021.

[40] K. P. N. Jayasena and B. S. Thisarasinghe, "Optimized Task Scheduling on Fog Computing Environment Using Meta Heuristic Algorithms," Proc. of the 2019 IEEE Int. Conf. on Smart Cloud (SmartCloud), pp. 53-58, DOI: 10.1109/SmartCloud.2019.00019, 2019.

ملخص البحث:

تلعبُ الحوسبة السّحابية دوراً أساسياً في تطوُّر إنترنت الأشياء، التي توفّر خدمات معالجة البيانات وتخزينها. حيث تساعد الحوسبة الضّبابية المتطورة من الحوسبة السّحابية في إيجاد حُلولٍ للتّحدّيات التي تواجه الحوسبة السّحابية، مثل التّأخير والوعي بالموضِع ودعم حَرَكية الزّمن الحقيقي. وتعمل الحوسبة الضّبابية على ملء الفجوة بين السّحابة وأجهزة إنترنت الأشياء بالقرب من تلك الأجهزة. وهكذا فإنّ الحوسبة، والتّشبيك، وتخزين البيانات، وإدارتها، واتّخاذ القرار تتمّ جميعها على طول المسار بين السّحابة وأجهزة إنترنت الأشياء. وإنّ الإدارة الأوتوماتيكية والذّكية لمصادر العُقد الضّبابية وتحقيق سياسة جدْولةٍ فعّالة في نموذج الحوسبة هما متطلّبان ضروريان ويقودان الى تطوير الأداء الإجمالي للحوسبة الضّبابية.

في هذه الورقة، تمّ تصميم نموذجٍ له هدفان هما: تحسين الأداء المتعلّق بالتّكلفة من جهةٍ، وتوفير استهلاك الطّاقة من جهةٍ ثانية. ويُعرَّف الأداء الخاص بالتّكلفة بأنّه النّسبة بين معدَّل البيانات للمُسْتخدِم وبين التّكلفة المتعلّقة بتلك البيانات. وقد تمّ استخدام طريقةٍ رياضيةٍ دقيقةٍ مع برنامج (GAMS) لإثبات العملية المنطقية للنّموذج. وفي الخطوة التّالية، قُمنا بحلّ النّموذج باستخدام عدّة طُرق. وبينت نتائج المقارنة أنّ (SA+GA) كانت أفضل طُرق الحلّ بقيمةٍ بلغت (0.23) مقارنة بالطُّرق الأخرى، وجاءت بعدها الطُّرق (GWO؛ GA؛ TLBO؛ PSO؛ GOA)، على الترتيب. وقد أدّى استخدام النّموذج المقترح الى الحصول على أعلى أداءٍ للتّكلفة بأقلّ استهلاكٍ للطّاقة.