IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 December 2022, accepted 27 January 2023, date of publication 31 January 2023, date of current version 3 March 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3241240*

RESEARCH ARTICLE

# Multi-Objective Grey Wolf Optimizer Algorithm for Task Scheduling in Cloud-Fog Computing

**FATEN A. SAIF, ROHAYA LATIP, (Member, IEEE), ZURINA MOHD HANAPI, (Member, IEEE), AND KAMARUDIN SHAFINAH**

Department of Communication Technology and Networks, Faculty of Computer Science And Information Technology, Universiti Putra Malaysia (UPM), Serdang, Selangor 43400, Malaysia

Corresponding authors: Faten A. Saif (f-saif500@hotmail.com), Kamarudin Shafinah (shafinah@upm.edu.my), and Rohaya Latip (rohayalt@upm.edu.my)

This work was supported in part by the University Putra Malaysia, and in part by the Ministry of Education (MOE) Malaysia.

**ABSTRACT** The revolution of IoT and its capabilities to serve various fields led to generating a large amount of data for processing. Tasks that require an instant response, especially with sensitive delay tasks send to the fog node due to the close distance, and the complex tasks transfer to the cloud data center for its huge computation and storage. However, sending tasks to the fog decreases the transmission delay. Still, it increases the energy consumption of the end users, while transferring tasks to the cloud reduces users' energy consumption but increases the transmission delay due to the long distance; besides, assigning tasks to appropriate resources compatible with task requirements. These are the main challenges in cloud-fog computing that need to improve. Thus, this study proposed a Multi-Objectives Grey Wolf Optimizer (MGWO) algorithm to reduce the QoS objectives delay and energy consumption and held in the fog broker, which plays an essential role in distributing tasks. The simulation result verifies the effectiveness of the MGWO algorithm compared to the state-of-the-art algorithms in reducing delay and Energy consumption.

**INDEX TERMS** Cloud-fog computing, delay, energy consumption, grey wolf optimizer, Internet of Things, meta-heuristic, task scheduling.

## I. INTRODUCTION

In today's world, the growth of telecommunication networks significantly impacts the Internet of Things (IoT) revolution, which is gaining popularity. Terminal devices have the essential role of detecting the environment sensing then generating data. Nevertheless, the limited capabilities of these devices lead to sending data to the cloud for storing, processing, analysing, and making a decision due to its high computation and storage. Besides the ability of cloud computing to provide ubiquitous access to its resources, it cannot satisfy the latency-sensitive IoT application [1]. Then it appears the role of fog computing is to complement the cloud, not to substitute. The cooperation of cloud-fog is to meet the various task lengths and computations. With the vast penetration of edge-cloud computing, the users' requests have increased that vary between cloud and fog nodes according to various characteristics such as input task length, the sensitivity of tasks, performance metrics like delay, makespan, cost, energy consumption, and so on. All these requests should process to meet the users' requirements [2]. However, the various features of the fog and cloud with the random users' requests and the limitation of resources make optimizing scheduling tasks more complicated and need to discuss and cannot be ignored. Task scheduling is an impact factor in improving the system performance, efficiently balancing the load to overcome the network overhead, maximize the resource utilization, and keeping the energy consumption [3]. The primary role of task scheduling is mapping tasks to the appropriate resources to guarantee to finish the execution of the task with meeting the quality of service (QoS) [4]. Even with the significant benefits of cloud-fog computing, task scheduling still faces challenges due to its dynamic nature, task configuration, and the required resources. All these factors impact the (QoS) optimization that leads to adjusting the parameters and determining the appropriate cloud and fog resources [5]. The main aim of optimization is to reduce or increase the function of the objective

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Mehmood.

during scheduling tasks. There are various optimization metrics such as delay, energy consumption, makespan, cost, etc [4]. Designing task scheduling is classified as static (offline) and dynamic (online) according to the features of the environment. Executing offline scheduling, the scheduler should consider the tasks' parameters, such as resources, optimization objectives, and QoS constraints. While the online scheduling, the task parameters, and resources auto-change in the environment [6]. Scheduling considers NP-complete. So, it used meta-heuristic algorithms to find approximate optimization solutions to the near-optimal solution [7] that is based on searching randomly. Many popular metaheuristics are used for task scheduling, such as Particle swarm (PSO), Genetic algorithm (GA), simulated annealing (SA), Ant Colony Optimization (ACO), etc [8]. Nevertheless, meta-heuristics algorithms have various search processes, randomness problems, minimum global search capabilities, and low convergence in the late iteration that make it fall in the local optimum search solution [9]. Furthermore, unbalance between global and local search [10]. From this point, the researchers go towards Grey wolf optimizer (GWO) due to its significance which leads to overcoming most meta-heuristic algorithms. The mechanism of GWO requires only the position of one vector, which means minimum memory compared to the PSO algorithm. Furthermore, GWO relies on the three best solutions to avoid falling into the local optimum solution, unlike the PSO algorithm that chooses one best solution by all particles [11]. Even more, GWO algorithm has a few parameters compared to GA algorithms which means less complexity and decreased computation time and energy consumption [11]. GWO algorithm is more dynamic than Ant Colony optimization, which does not consider the dynamic nature of computing resources [12]. GWO algorithm proposed by [13] which is a meta-heuristic approach to solving optimization problems. It considers a swarm intelligence algorithm. The GWO technique is based on the nature of grey wolves swarming in leasers and hunting. Also, it follows the social hierarchy. Recently, it has gained tremendous popularity compared to other meta-heuristics algorithms due to its significant benefits of convergence during execution, fewer parameters that lead to minimum complexity time and energy consumption, and its simplicity in the implementation [14]. GWO has been adopted in various fields to solve numerous problems such as optimization, classification, economic and power dispatch, capacitated vehicle routing, etc [15].

Lately, many researchers have been attracted to discussing MOP, which means more than one in the task scheduling, especially in cloud-fog computing, due to various characteristics of the nodes and the distance. The two main objectives that have critical impacts are delay and energy consumption. Whereas, executing tasks at the fog node reduces the delay time due to the short distance but increases the user's energy consumption of devices. On the other side, transferring tasks to the cloud decrease the user's energy consumption but increase the delay time [16]. However, sending tasks to the cloud can save energy of fog nodes from the users' sides but increase delay. In contrast, transferring tasks to the cloud saving the energy consumption of the users but increasing the transmission delay. Therefore, it is important to propose a tradeoff strategy between delay and energy. This paper concentrates on task scheduling problems in Cloud–Fog environment, a highly distributed computing platform, for processing IoT applications. The main contribution of this study is summarized as follow:

- A mathematical framework with queue theory was developed to reduce power consumption and delay via efficient workload allocation.
- Proposed a MOP approach called MGWO algorithm to solve this problem for task scheduling. The main objective of the MGWO algorithm is to reduce the delay and energy consumption tasks in cloud-fog computing.
- The simulation results demonstrate that the MGWO algorithm outperforms other related algorithms in reducing delay and energy consumption.

The rest of the article is organized as follows. In Section II, the related works are presented. Section III specifies the mathematical model for the task scheduling problem in the Cloud–Fog computing environment. Section IV presents our proposed algorithm in detail. Simulation and Experimental results are given in Section V. Finally, Section VI concludes the article, and future works are discussed.

## II. RELATED WORK

The comprehensive review of the recent task scheduling studies. First, the main algorithms and techniques adopted in the task scheduling in the cloud-fog. Second, the significant studies about applying Grey Wolf Optimizer (GWO) algorithm.

### A. TASK SCHEDULING APPROACH

This part throw light on the review of state of art algorithms and techniques in terms of scheduling tasks in the cloud-fog computing. Reference [17] discuss the balancing between makespan and cost monetary with meeting the deadline constraints during tasks scheduling by proposing a heuristic algorithm, namely Cost-Makespan aware scheduling (CMaS). Reference [18] reported the problem of dispatching and tasks and how to reduce the total response time for all tasks. The study proposed a model to generate arbitrary sort tasks and transfer tasks to the servers with both upload and download delays. They estimate the weight of how far the tasks are sensitive. Furthermore, an online tasks dispatch and scheduling, namely, OnDisc, is extendable in the speed-raising model. Reference [1] addressing the problem of large-scale task scheduling by proposing a real-time randomized algorithm that utilized the Power of Two Choices (Po2C) method to raise the QoS while reducing the cost. Reference [19] The study considers the problem of user preferences related to the fog nodes' constraints during focusing on reducing the delay and energy consumption. Thus, the study proposed a ranking-based task scheduling that collaborates with user preferences and fog nodes based on a linguistic and fuzzy quantified proposition for ranking the fog node. The study adopts two parameters called least satisfactory proportion (lsp) and greatest satisfactory proportion (gsp) to determine the similarities. Reference [20] presenting two schedulers relies on integer linear programming to decide which fog or cloud resources the tasks should schedule. The technique uses the class of service to choose the processing elements on where the tasks must execute. Reference [21] proposing the Cuckoo Optimization Algorithm (COA) based load balancing technique for ideal resources management. The algorithm detects the idle machine or less utilizing to switch them off, which saves power consumption. The primary role of COA is to map the proper tasks to Virtual Machines (VMs). Reference [22] presenting the two stages of the scheduling algorithm that depends on deep learning techniques. First stage is determining where the tasks would be executed using the clustering strategy. Second stage is scheduling the tasks according to their locations. The clustering strategy has three concepts based on the Self-Organizing Map (SOM) clustering method. Hence, the first and second concepts are the SOM and hierarchical SOM that is utilized for clustering the receiving of tasks features from IoT applications. The third concept is the extraction feature that reduces its dimensions using the Autoencoder, one of the deep learning methods. Reference [23] considering the dynamic distribution of tasks from healthcare applications among the fog and cloud devices via the mobility-aware heuristic-based scheduling and allocation method (MobMBAR). The main objective is reducing scheduling time via using the features of tasks such as the level of critical and the maximum response time of the task while the phases of ranking and reallocation. The approach facilitates the patient's mobility by an adaptive Received Signal Strength (RSS) based handoff mechanism [24] design a fog-based region architecture to offer close computing resources—proposed task scheduling for region-based cloud (FBRC) algorithm to meet the requirements of resources and sensitive latency formulated as an Integer program and solved by a heuristic algorithm. The main objective is to reduce task completion time. Reference [25] discussed the scheduling and dispatching tasks to reduce the response time by proposing a model that generates arbitrary times to minimize the weighted response time of all tasks. The proposed algorithm is called OnDisc. Reference [26] considering the task scheduling for offloading the applications of large-scale by proposing a heuristic approach for trad-off makespan and cost. Reference [27] reporting the ideal distribution of workload effectively over the processor by proposing a Heterogeneous Earliest Finish Time (HEFT) to minimize the makespan of execution tasks. Reference [6] solving the problem of accomplishing the maximum number of users' requests by meeting the deadline constraint. This problem has been solved by proposing the mixed integer programming model to reduce deadline misses. The model's mechanism considered the delay of requests when it takes a round trip. Delay contains three elements networking delay, task execution delay, transmission delay, and queuing delay. Even more, the model considers the characteristics of requests, such as request priority, deadline, and size. The model adopts the Genetic algorithm(GA) for scheduling the requests of IoT applications. Reference [28] focused on the problem of running the mobility of applications and the delay-sensitive with a less monetary cost by proposing a new Microservice container fog system (MSCFS) based framework. Furthermore, discuss the problem of cost-efficient task scheduling on heterogeneous fog servers. Even more, it presented the Cost Aware Computational Offloading and task scheduling (CACOTS)framework that tackles the problem of task scheduling into various steps such as resource matching, scheduling steps, and task sequencing. Reference [29] investigated the problem of finding the placement and resources of the optimal service in the three-tier IoT to accomplish resource efficiency and cost, higher rate of security and privacy, and higher QoS. Thus, the study proposed a cost-aware genetic-based (CAG) task scheduling algorithm to enhance cost efficiency in real-time applications with maintaining hard deadlines. Reference [30] presented a Virtual Machine (VM) scheduling method for load balancing. The techniques of the study started with the resource and load balancing model, then the heuristic VM scheduling method based on VM placement and dynamic VM scheduling leveraging the VM live migration technique. Reference [31] reported excessive delay during task offloading, which has a negative impact on the user experience. The study optimized the average response time of multi-task parallel scheduling and formulated the computation offloading and task scheduling for DNN-based applications. The proposed algorithms are genetic and greedy for solving this issue. Reference [32] investigated the traffic fluctuations, the quantity of unwanted vBBUs, and VPONs that may change periodically. Thus, the authors proposed a batch scheduling algorithm based on Integer Linear Programming (ILP) for the reconfiguration of VPONs and migration of vBBUs over processing nodes in the function of fluctuation on traffic demands. Reference [33] Employ the tabu search due to its high expansion in diverse optimization problems and memory and high-speed attributes. reported the issue of finding the optimal allocation for the highest resource usage and minimizing the response time. Thus, the study proposed a tabu search for less cost of hardware. Also, it offered a latency-aware scheduling algorithm based on VM matching utilizing meta-heuristics. Tabu search enhance based on fruit fly optimization (FOA) algorithms and approximate nearest neighbour (ANN) [5] proposed a MOP for task scheduling to reduce the makespan and cost; this work is based on a model with Discrete Non-dominated Sorting Genetic Algorithm II (DNSGA-II) for auto tasks allocated to fog or cloud devices. The model distributes the workload among cloud and fog nodes effectively. Reference [34] considering task scheduling by proposing a multi-agent-based model that served tasks according to priority, waiting time, and resource status. The goal of the study is to raise the utilization of resources effectively. Reference [35] presented the trade-off between cloud cost and makespan by proposing the BAS algorithm for sequence applications. The technique is based on gathering the applications on a timeline to guarantee their execution to reduce necessary expenditures for using the cloud resources. Reference [36] Discussing the problem of MOP in workflow scheduling using particle swarm optimization (PSO) based on Fuzzy resource utilization (FR-MOS) that aims to reduce makespan and cost with satisfying the reliability constraint. Even more, considering data transportation order and task execution location simultaneously.

### B. GWO TECHNIQUES IN TASK SCHEDULING

GWO algorithm recently many scholars has adopted duo to its capabilities. Reference [37] discussed proposed an enhanced multi- objective grey wolf optimizer (EMGWO) for multi-objective service composition formulated to get tradeoff energy consumption and QoS and optimal selection (MO-SCOS) problem. The proposed algorithm adopted the backward learning method to increase the initial population's exploration, avoid falling into the local optimum solution, and increase diversity. Also, employ the nonlinear adjustment strategy to manage the parameters and improve the global exploration of the algorithm [38] reporting the ideal utilization of cloud resources by the proposed MGWO technique for task scheduling. The study aims to reduce the energy consumption of cloud data centers for the scheduler makespan for users' requests. Reference [14] presented an energy-aware service composition and optimal selection (EA-SCOS) model to verify the Quality and reduce the energy consumption while execution task. The proposed algorithm namely, the grey wolf optimizer (GWO) [39] identifying the practical resources for scheduling an exact task on time, using the resources effectively, and minimizing the completion time for all execution tasks by proposing the grey wolf optimizer nature-inspired algorithm. Reference [40] proposed a multi-objective parallel machine scheduling method based on the oppositional grey wolf's optimization (OGWO) with adopting Opposition-based learning (OBL) to enhance the performance of the GWO algorithm during optimizing tasks and resources. Reference [12] considered the multi-objectives such as Throughput, makespan, and resource utilization by presenting a multi-objective grey wolf optimizer (TSMGWO) to find an optimal solution for scheduling tasks. Reference [41] presented a Modified Fractional Grey Wolf Optimizer for Multi-Objective Task Scheduling (MFGMTS) for the conflicting objectives communication cost, resource utilization, execution cost, execution time, communication time, and communication cost using penalty cost function and epsilon-constraint. The MFGMTS algorithm is motivated by Fractional Grey-Wolf Optimization (FGWO) collaboration with a modification in the position update.

## III. CLOUD-FOG SYSTEM ARCHITECTURE

As shown in Fig. 1, we assume that the cloud-fog system form with terminal Devices M, N fog devices, and C cloud servers. Terminal devices communicate with each other via a wireless channel. Fog nodes communicate directly with the terminal devices. Generating data from terminal devices collected from sensors are forwarded immediately to the Fog broker, which analyses, estimates, and schedules the incoming request from end users. Then, determine which appropriate devices are between fog and cloud according to the characteristics of the tasks to provide services for execution tasks. The fog broker places near fog nodes, and we can ignore the time consumption. For achieving optimal task scheduling, The MGWO algorithm is installed for the broker to find optimal task scheduling that satisfies the transmission delay and energy consumption. According to the various power and capacity of resources, the traffic model considers terminal devices as an M/M/1 queue at the end devices, M/M/C queue at the fog node, and M/M/$\infty$ queue at the cloud server.

First, the terminal device sends the request for processing (Step 1). Then, this job redirects to the fog broker (Step 2), which decomposes into a set of small and independent tasks for processing over the Cloud–Fog computing infrastructure. (Step 3). After that, the fog broker analyzed the tasks and estimated the required resources according to the tasks' characteristics (Step 4) which they are the number of tasks $t$, length of the task input $I_t$, task deadline $d_t$, flag of task execution $u_t$, and required computing unit by task $\psi_t$. Next, handling all information on tasks and nodes, the Fog broker runs an MGWO algorithm for scheduling algorithm to find the optimal task assignment (Step 5). Then, tasks send to the fog nodes and cloud server (Step 6). In this step, each node is responsible for executing the assigned tasks (Step 7). Once the executing tasks finish, the task results send to the fog broker again (Step 8). The fog broker composes the tasks again (Step 9) and then sends the job to the terminal device (Step 10). As illustrated in Fig. 2.

### A. DELAYDESCRIPTION AND ENERGYDESCRIPTION

#### 1) END USER DEVICE

We assume the service rate $\mu$ of the user device $i$ follows an exponential distribution, with an M/M/1 task queue. In addition, the generation of tasks from the end device is based on at Poisson process with an average arrival rate $\lambda$

$P_{ed}$ is the energy of end-device $i$ and $T_{ed}$ is its processing time. The power consumption of $X^i_{ed}$ for the task's execution at the end device calculated by

$$P^i_{ed} T_{ed} \times P_{ed} = \frac{X^i_{ed}}{\mu - \lambda} \times P_{ed} \tag{1}$$

We consider computing latency because tasks performed on mobile terminal devices have little communication delay. As deduced from queue theory, the delay is described as

$$D^i_{ed} \frac{\lambda}{\mu(\mu - \lambda)} \tag{2}$$

#### 2) FOG NODE

The task queue in fog node j is modelled as M/M/C. The energy consumption reflects the amount of computation,

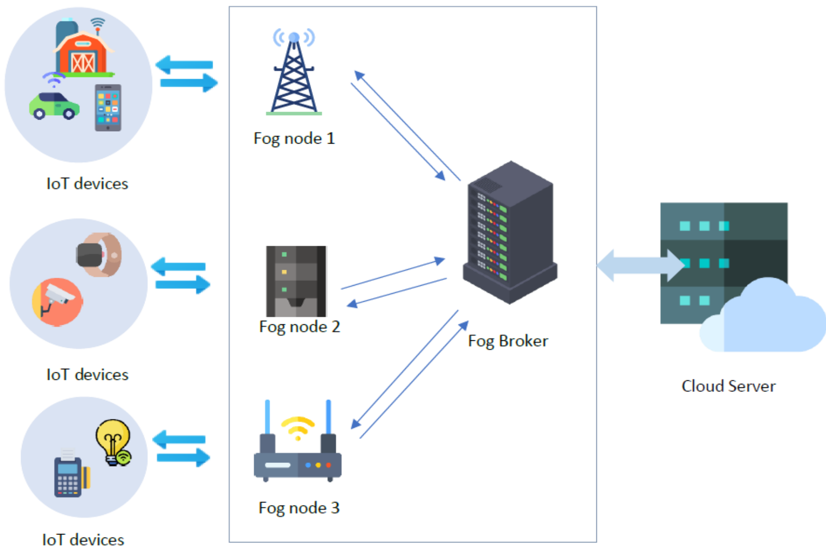



**FIGURE 1.** Cloud-fog architecture.

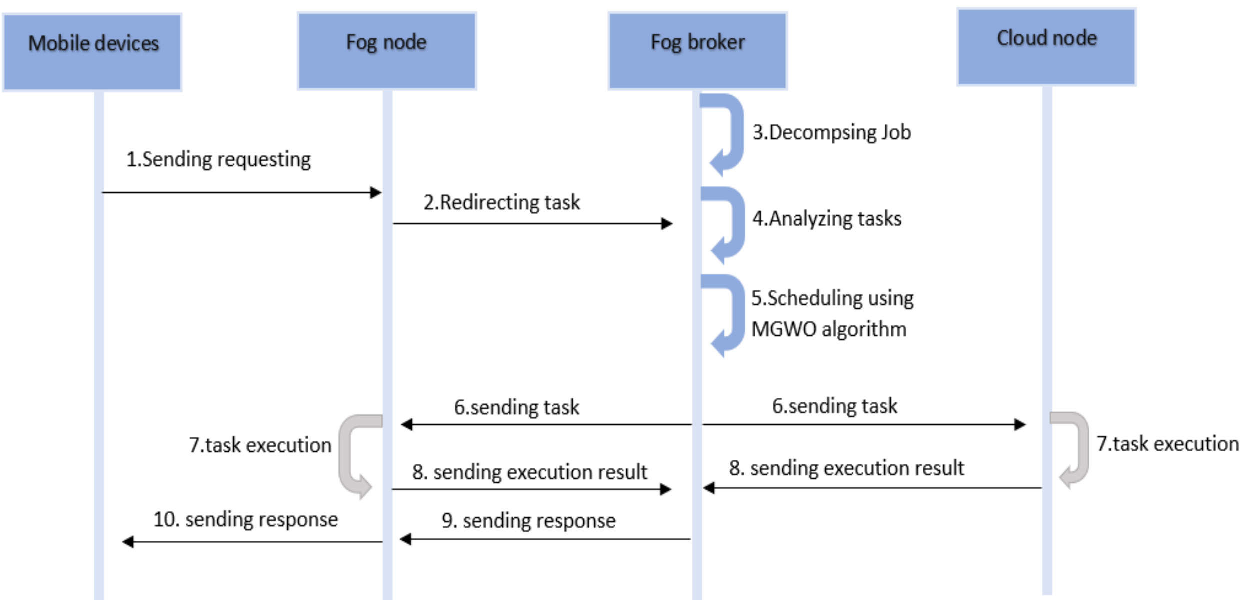



**FIGURE 2.** The cloud-fog operation.

which is a monotonically increasing and strictly convex function. Quadratic and piecewise linear functions are two alternatives to this function. Fog nodes can flexibly adapt to any function of energy consumption as long as they meet these two attributes:1) there is a direct relationship; that is, increasing energy consumption increases the computation amount. 2) The energy consumption margin increases for each fog device. The power energy expression $P^j_{fog}$ of the fog node is related to the workload $Y^j_{fog}$ as follows:

$$P^j_{fog} a Y^{j^2}_{fog} + bY^j_{fog} + c \tag{3}$$

where a > 0 and b and, c ≥ 0 are pre-determined parameters.

The fog node j consists of both communication and computing delays. The computing delay $D^{com}_{fog}$ is related to waiting time. Using queue theory, we can express the computing delay as follows:

$$D^{com}_{fog} \frac{Q_L}{\lambda} \times Y^j_{fog} \tag{4}$$

where $Y^j_{fog}$ is the workload allocated to fog node j and. $Q_L$ is the average queue length. As a result of task execution at the fog node, communication is related to the input length of the tasks. The communication delay $D^{comm}_j$ is expressed

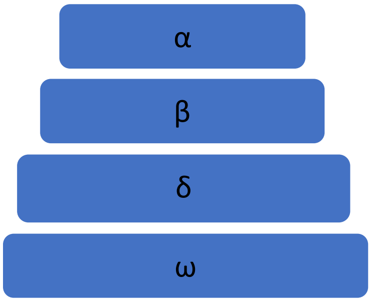



**FIGURE 3.** Hierarchy of grey wolf (dominance decreases from top down).

as follows:

$$F_{comm}\,(I_t)\begin{cases} yI_t & u_t \epsilon cloud \\ \varepsilon I_t & u_t \in fog \end{cases} \tag{5}$$

where $I_g$ is the input length of the task t ($\gamma \gg \varepsilon$). Therefore, the communication delay of the fog node is $D_{fog}^{comm} = \varepsilon I_g$. The fog node delay is composed of computing and communication delays, which can be expressed as follows:

$$D_{fog}^{j} = D_{fog}^{com} + D_{fog}^{comm} \tag{6}$$

### 3) CLOUD COMPUTING

For cloud server k, the task queue is modelled as an M/M/ ∞ queue. Assuming that every cloud server has several homogeneous computing machines and that the CPU frequency of all machines is equal; this implies that the energy consumption for all servers is the same. The approximate energy consumed by every machine on cloud server k can be obtained by utilizing the frequency of the CPU machine function. $f_k$:$A_k Z_{cloud}^{k} + B_k$, where $A_k$ and $B_k$ are positive constants. Assigning more workload to the cloud server implies more power-on. Whenever the assigned workload decreases, some cloud servers are turned off to save energy. The power consumption of the cloud server $P_{cloud}^{k}$ is related to the on/off state of the machine.

$$P_{cloud}^{k} \triangleq \sigma_k n_k (a_k Z_{cloud}^{k} + b_k), \tag{7}$$

where $a_k$ and $b_k$ are the positive constants. $\sigma_k$ indicates the on/off state of cloud server k, where 1 denotes the cloud server on and 0 indicates its off state. $n_k$ denotes the number of on-state machines on the cloud server. Owing to the heavy computational resources of cloud servers, the computational delay can be assumed to be negligible; thus, the delay is the communication delay that defines as

$$D_{cloud}^{k} \triangleq \gamma I_i \tag{8}$$

## IV. THE PROPOSED ALGORITHM MGWO

The Grey wolf belongs to the canine family. They live in groups with a clear division of labor and collaboration to survive and rely on hierarchical systems as shown in Fig. 3. The leading grey wolf is called alpha ($\alpha$) wolf, its next level is called Beta ($\beta$) wolf, the third level is called Delta ($\delta$) wolf, and the lowest level of the gray wolf is called Omega ($\omega$) wolf. Where $\alpha$ is the leader and responsible for decision-making that is related to hunting time, sleeping time, and so on, and that leads to calling alpha a dominant wolf because it order must follow by the pack. While $\beta$ indicates to the subordinate wolves that assist in alpha in decision-making and discipline the pack. Whereas, $\delta$ should submit to all alphas and betas and dominate the omega simultaneously. Hence, $\omega$ in the lower level in the hierarchy and its role submit to all the other dominant wolves [13]. However, GWO algorithm was implemented to solve the single-objective problem. In contrast, this paper discusses the MOP, which means more than one objective and conflicting. In order to perform MGOW, we cooperate the non-Pareto dominance solutions and the external archive to save non-dominated Pareto optimal solutions obtained so far. See Fig.6. However, the main stages of grey wolf hunting are as follows:1) Tracking the prey. 2) Encircling the prey until it stops moving. 3) Hunting the prey. 4) Attack towards the prey.

### A. TRACKING THE PREY

This is the first stage and the wolves randomly diverge in hunting their prey, it can be modelled mathematically by involving the computation of the distance between the Prey and the Grave wolf.

$$D_{vector} = abs(C_{Vector}.L_P\,(t) - L_w\,(t)) \tag{9}$$

$D_{vector}$ indicates the computation distance between the Prey and Grey wolf, where $C_{vector}$ referred to coefficient vector, $L_W(t)$ represents the location of the grey wolf at a given time interval t; $L_p$ indicates the location of prey at a given time interval t.

The value of $C_{vector}$ can be calculated by

$$C_{Vector} = 2.rand\,(1, 0) \tag{10}$$

Function generates random numbers between [0,1]

### B. ENCIRCLING STAGE

is the process of prey detection by alpha, beta, and delta wolves, the rest of the wolves change their positions at a time interval of *t+1* based on the equations

$$L_w\,(t+1) = L_P\,(t) - D_{vector}.A \tag{11}$$

*A* indicates as factor to adjust the abilities of exploration and exploitation during the optimization

$$A = 2.a.r_2 - a \tag{12}$$

$r_2$ is a random vector in the interval [1,0], where $a$ is a convergence factor that presents values start from 2 then decreasing linearly to 0. The current value $a$ in an iteration $t$ in the $t_{max}$ computed in the equation:

$$a = 2(1 - \frac{t}{t_{max}}) \tag{13}$$

$t$ refers to the current iteration, and $t_{max}$ expresses the maximum iteration.

### C. HUNTING STAGE

In this stage, the wolves have better knowledge about the position of the prey. Thus, the wolves change their position according to the prey position for attacking due to considering the top three dominating wolves as alpha, beta, and delta in the hierarchy as the best solution. Therefore, the alpha can attack the prey. The prey position is updated as per the position of the three dominated wolves as equation (14). Whereas omega update their locations randomly close the prey.

$$\begin{aligned} D_\alpha &= abs(C_1.X_\alpha(t).L(t)) \\ D_\beta &= abs(C_2.X_\beta(t).L(t)) \\ D_\delta &= abs(C_3.X_\delta(t).L(t)) \\ X_1 &= X_\alpha(t) - C_1.D_\alpha \\ X_2 &= X_\beta(t) - C_2.D_\beta \\ X_1 &= X_\delta(t) - C_3.D_\delta \\ X(t+1) &= \frac{(X_1+X_2+X_3)}{3} \end{aligned} \tag{14}$$

#### 1) ATTACKING PREY

The prey attack indicates the exploitation process, and as mentioned above, the grey wolves finish the hunt by attacking the prey when it stops moving. This process accomplishes by shrinking the values of $a$ linear decrease from 2 to 0 throughout iterations.

### D. FITNESS FUNCTION

MGWO is the metaheuristic algorithm, which schedules the task depending on a fitness function to find the appropriate computing resources. Fitness function is responsible to determine the solution based on the objective's requirements. So, in this study fitness function chooses the solution according on the minimum delay and energy consumption. So, the value of the fitness function has a significant impact on the proposed algorithm. It is mandatory to improve the various conflicting objectives in MOP simultaneously. Generating solutions cannot be compared to one another via the fitness function. Thus, the ideal solution is implementing the Pareto dominance to choose the optimal solution in each iteration. In contrast, optimal solutions considered as trade-off solutions instead of single solutions, namely Pareto optimal solutions. The Fitness Function of the given multi-objective task scheduling problem based on the weighted sum approach for balancing the objectives is defined as follows.

$$\begin{aligned} F &= \alpha * Total\ Delay \\ &\quad + (1-\alpha)\ Total\ energy\ consumption \end{aligned} \tag{15}$$

$\alpha$ is the Energy-delay balance factor where $\alpha$ ($\alpha \in [0, 1]$) is the balance coefficient between total Energy and total delay. $\alpha = 0.5$ means that total Energy and total Delay have same priority in optimizing. When $\alpha > 0.5$, our mechanism focuses on minimizing the Energy with higher priority than total Delay, which is the case task will be late to obtain minimum energy consumption. Inversely, when $\alpha < 0.5$, the Delay is more prioritized than Energy, i.e., the user has an instant better task delay.

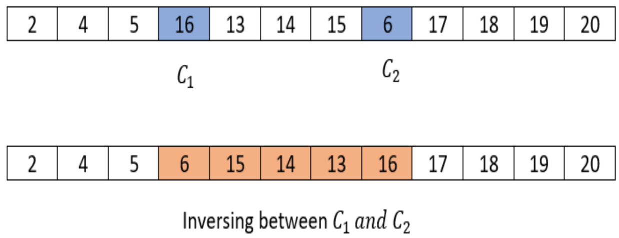



**FIGURE 4.** Inverse mutation operator.

| **Inverse mutation Operator** |
|---|
| **BEGIN** |
| Input: the solutions |
| 1. select two random points C1 and C2 at random |
| 2. Swap positions Then |
| 3. If values of (C1 > C2) Then |
| 4. Inverse the positions between C1 and C2 |
| 5. Else |
| 6. No swapping |
| 7. Output: mutate the solution |
| **End** |

**FIGURE 5.** Algorithm 1.

### E. MUTATION THE SOLUTION

The essential role of mutation is to increase the population's diversity and global search capability. In this paper, we implemented the inverse mutation operator. As evident in Fig. 4. First, select two random position points as $C_1$ *and* $C_2$ to the exchange between them to generate a solution. Then, If $C_1 > C_2$, then inverse between c and c; otherwise, no swapping. See Fig. 5.

### F. THE EXTERNAL ARCHIVE

We implemented an archive to store Non dominated solutions in MOP algorithms in every iteration as global best solution and considered the final approximated Pareto front [40]. There are a m number of new nondominated solutions S = (s1, s2,..., sm) inserted into A in each generation since the number of nondominated solutions in the external archive is finite and determined by the size of the population. The

archive must terminate because it can only hold so many non-dominated solutions before reaching its maximum size. The archive update approach is essential since it affects how well the algorithm works by Using the crowding distance as a tool to delete other solutions.

### G. CROWDING DISTANCE

It is implemented to evaluate solutions with the same rank of non-dominated by measuring the relative density of the individuals to avoid the exhaustive search for getting an

**TABLE 1.** Simulation key parameters.

| Parameters | Value |
|---|---|
| $\mu$ | 4.6 |
| $\lambda$ | 3.2 |
| $P_{ed}$ | 23 |
| $a$ | 1 |
| $b$ | 2.4 |
| $e$ | 3.5 |
| $a_k$ | [3.206 4.485 2.370] |
| $b_k$ | [68 53 70] × $10^{-20}$ |
| workload | [30 50 90 150 200] |
| MOGOW population size | 30 |
| Archive size | 50 |

optimal Pareto front. It also raises diversity in the solution selection, which prevents local optimality. The strategy of crowding distance is limiting the archive size, which solution in the archive sorting in descending order according to the crowding distance values, then determining if the solutions exceed the archive size, then deleting the non-dominated solutions beyond the size [42]. See equation (16)

$$CD = \frac{DS^J}{DS^{max} - DS^{min}} \quad \forall J = 1 : N \tag{16}$$

where $N$ indicates the number of non-dominated solutions, $DS^J$, $DS^{max}$, $DS^{min}$ are the distance between two neighbouring solutions of the $Jth$ solution, the maximum distance between two solutions, and the minimum distance between two solutions in the direction of the $Jth$ objective function, respectively.

## V. SIMULATION AND RESULT

In this section, we conduct experiments to evaluate our proposed algorithm MGWO compared to the Cloud-fog cooperation algorithm [42], NSGA-II, and MPSO algorithms regarding the objective's functions delay and energy consumption. In an Edge-Cloud environment, various IoT/mobile devices generate several applications. These applications include multiple tasks and required to be processed in the Edge-Cloud resources. We assume we have seven IoT/mobile devices connected to three fog nodes and one cloud server. Our simulation conducting with five groups of tasks, and their total workloads are 30, 50, 90, 150, and 200. The length of tasks is random due to the difficulty of predicting it in reality. The simulation tool is MATLAB R2018b on a computer with core i7 running on the windows operating system to verify the effectiveness of our proposed algorithm. Table 1 represents the key parameters of our simulations.

Fig. 7, demonstrates the transmission delay for the proposed algorithm and cloud-fog cooperation, NGSA-II, and MOPSO algorithms [30,50,90,100,150]. The purpose of the experiments was to reduce the delay for IoT applications.

**Algorithm 2 multi-objective grey wolf optimizer (MGWO)**

1. Initialize the population size n (LX)
2. random Initialize a, A, and $C_{Vector}$, $D_{vector}$
3. Initial external archive EA = [ ];
4. Defined by user the max. no. of iteration $t_{max}$
5. Set M = 0; b = 1;
6. **While** (b <= n)
7. Random initialize LX(t)
8. Calculate the objective values for each grey wolf
9. Compute the fitness value for each grey wolf
10. add non-dominated solutions to the archive
11. **End while**
12. Select the best three Top grey wolves as $X_\alpha$, $X_\beta$, $X_\delta$
13. **While** ($m < t_{max}$)
14. $\forall$ t $\in$ Lx
15. update location of each grey wolf using $X(t+1) = \frac{(X_1 + X_2 + X_3)}{3}$
16. **End for**
17. update $a = 2\left(1 - \frac{t}{t_{max}}\right)$
18. update $A = 2 \,.\, a \,.\, r_2 - a$
19. $C_{Vector} = 2 \,.\, rand\,(1,0)$
20. Mutate the solutions // See Algorithm 5.2
21. F = α ∗ Total_Delay + (1 − α) ∗ Total_Energy consumption
22. Update EA[I];
23. **If** (EA[i] is full)
24. { $CD = \frac{DS^J}{DS^{max} - DS^{min}}$ $\forall J = 1 : N$
25. ∀ i ∈ EA[ i] Sort in descending according to the values of CD
26. Delete the Min. value of CD in the EA[i] }
27. **end if**
28. t++
29. **End while**
30. Return EA[i]
31. Return the $X_\alpha$ as the best solution in the search space

**FIGURE 6.** Algorithm 2.

We can observe the apparent difference in the proposed algorithm with the related approaches in reducing the delay, that MGWO reduces delay more than comparing algorithms. Even more, the linear increase of delay with the increasing values of workloads proves the stability and effectiveness of MGWO while generating a vast number of tasks from IoT devices. In contrast, the cloud-fog cooperation algorithm provides the worst results compared to the other algorithms. On the other hand, NGSA-II performs better in reducing delay than MOPSO due to the operations operators of the NGSA-II, which avoid premature convergence, explore more in the search space, and avoid premature convergence, explore more in search space and these leads to better results than MOPSO. See Table. 2 from illustrated.

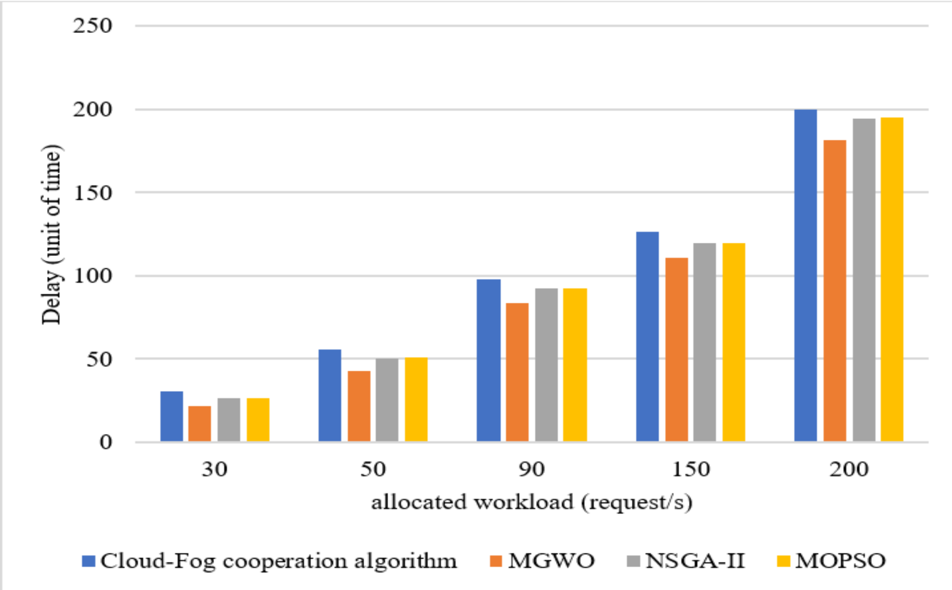



**FIGURE 7.** Delay comparison between the proposed algorithm state of art algorithms.

**TABLE 2.** Result of delay based on various workload.

| Workload size | 30 | 50 | 90 | 150 | 200 |
|---|---|---|---|---|---|
| Cloud-fog cooperation | 30.33 | 55.61 | 97.71 | 126.7 | 199.9 |
| MGWO | 22.07 | 42.54 | 71.77 | 98.05 | 151.7 |
| NSGA-II | 26.33 | 50.47 | 92.24 | 119.51 | 194.43 |
| MOPSO | 26.56 | 50.84 | 92.63 | 119.89 | 194.75 |

**TABLE 3.** Result of energy consuptio based on various workload.

| Workload size | 30 | 50 | 90 | 150 | 200 |
|---|---|---|---|---|---|
| Cloud-fog cooperation | 1.87E+05 | 3.07E+05 | 6.37E+05 | 6.69E+05 | 1.13E+06 |
| MGWO | 8.62E+04 | 2.09E+05 | 3.21E+05 | 4.21E+05 | 5.58E+05 |
| NSGA-II | 1.30E+05 | 2.77E+05 | 5.81E+05 | 5.74E+05 | 1.12E+06 |
| MOPSO | 1.28E+05 | 2.70E+05 | 5.75E+05 | 5.68E+05 | 1.11E+06 |

Fig. 8, shows the comparison between the proposed algorithm and other related algorithms in terms of energy consumption based on different workload sizes [30,50,90,100,150]. The figure shows the significant difference between the MGWO and compared algorithms that obtain the best results in reducing energy consumption. The stability of reducing energy consumption is proven while increasing the number of workloads. On the contrary, cloud-fog cooperation provides the worst result compared to the other algorithms. In addition, MOPSO provides better results in reducing energy consumption than NGSA-II. Hence, MOPSO has fewer parameters, which makes the algorithms' complexity low, meaning minimum execution time and consuming less energy than the NGSA-II algorithm. See Table 3.

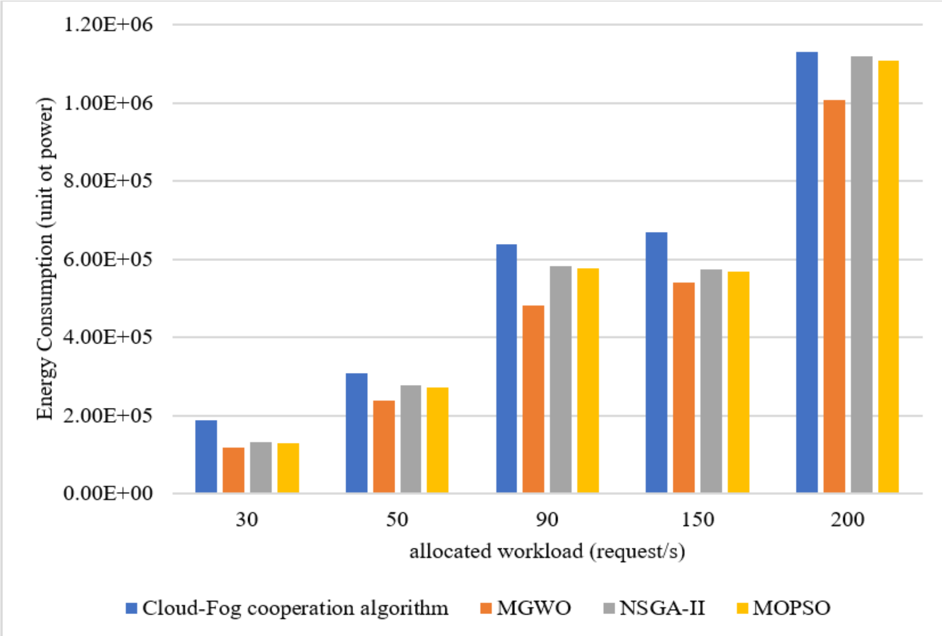



**FIGURE 8.** Energy Consumption comparison among the proposed algorithm and state of art algorithms.

Overall, many aspects lead to getting this result. First, the benefits of meta-heuristics are in providing reasonable solutions with less complexity, and that reduces the execution time of the system and the energy consumption. Also, the crowding distance technique limits the archive's size and guarantees the optimal solutions while storing the non-dominated solutions. Even more, this paper covers implementing the inverse mutation to increase the diversity and solve the problem of the meta-heuristics approach when trapped into a local optimum solution.

## VI. CONCLUSION AND FUTURE WORK

This work focused on task scheduling problems using the MOP approach in Cloud–Fog computing environment. This study considered the fog broker to analyze, estimate, and then Schelling all sending requests from terminal devices for execution in the cloud-fog system and holding the MGWO for scheduling tasks. The proposed algorithm, namely the MGWO algorithm, aims to reduce the delay and energy consumption of QoS objectives. The simulation is conducted to evaluate the performance of the MGWO algorithm compared to the state-of-the-art algorithms in reducing delay and energy consumption. The simulation results reveal that MGWO outperforms the comparison algorithms in reducing delay and energy consumption. The proposed algorithms maintain their stability and increase linearly with increasing workloads. It proves it can handle the enormous increase in generating requests from IoT devices. Investigate the heterogeneity of resources. The main limitation is that this work needs to consider the heterogeneity of resources, which may affect resource utilization and lead to an imbalance in the load. In future work, the study can extend to optimize many other objectives, such as transmission costs, computing resources, and load balancing. We can apply more algorithms to solve scheduling problems and adopt the AI approach for optimizing. Also, investigate the heterogeneity of resources.

## ACKNOWLEDGMENT

This study supported technically and financially by Universiti Putra Malaysia. Utmost appreciation and thanks to provide sufficient facilities and funding through-out this research.

## REFERENCES

[1] F. Hoseiny, S. Azizi, and S. Dabiri, "Using the power of two choices for real-time task scheduling in fog-cloud computing," in *Proc. 4th Int. Conf. Smart City, Internet Things Appl. (SCIOT)*, Sep. 2020, pp. 18–23, doi: 10.1109/SCIOT50840.2020.9250197.

[2] R. Tyagi and S. K. Gupta, "A survey on scheduling algorithms for parallel and distributed systems," in *Advances in Intelligent Systems and Computing*, vol. 718. Singapore: Springer, 2018, pp. 51–64, doi: 10.1007/978-981-10-7656-5_7.

[3] L. Chen, K. Guo, G. Fan, C. Wang, and S. Song, "Resource constrained profit optimization method for task scheduling in edge cloud," *IEEE Access*, vol. 8, pp. 118638–118652, 2020, doi: 10.1109/ACCESS.2020.3000985.

[4] P. Varshney and Y. Simmhan, "Characterizing application scheduling on edge, fog, and cloud computing resources," *Softw. Pract. Exper.*, vol. 50, no. 5, pp. 558–595, May 2020, doi: 10.1002/spe.2699.

[5] I. M. Ali, K. M. Sallam, N. Moustafa, R. Chakraborty, M. J. Ryan, and K.-K. R. Choo, "An automated task scheduling model using non-dominated sorting genetic algorithm II for fog-cloud systems," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2294–2308, Oct. 2020, doi: 10.1109/TCC.2020.3032386.

[6] R. O. Aburukba, T. Landolsi, and D. Omer, "A heuristic scheduling approach for fog-cloud computing environment with stationary IoT devices," *J. Netw. Comput. Appl.*, vol. 180, Apr. 2021, Art. no. 102994, doi: 10.1016/j.jnca.2021.102994.

[7] M. Akbari, H. Rashidi, and S. H. Alizadeh, "An enhanced genetic algorithm with new operators for task scheduling in heterogeneous computing systems," *Eng. Appl. Artif. Intell.*, vol. 61, pp. 35–46, May 2017, doi: 10.1016/j.engappai.2017.02.013.

[8] F. A. Saif, R. Latip, M. N. Derahman, and A. A. Alwan, "Hybrid meta-heuristic genetic algorithm: Differential evolution algorithms for scientific workflow scheduling in heterogeneous cloud environment," in *Proc. Future Technol. Conf.*, vol. 561, 2023, pp. 16–43, doi: 10.1007/978-3-031-18344-7_2.

[9] Z. Deng, Z. Yan, H. Huang, and H. Shen, "Energy-aware task scheduling on heterogeneous computing systems with time constraint," *IEEE Access*, vol. 8, pp. 23936–23950, 2020, doi: 10.1109/ACCESS.2020.2970166.

[10] P. Paknejad, R. Khorsand, and M. Ramezanpour, "Chaotic improved PICEA-G-based multi-objective optimization for workflow scheduling in cloud environment," *Futur. Gener. Comput. Syst.*, vol. 117, no. 2, pp. 183–195, Apr. 2021, doi: 10.1016/j.future.2020.11.002.

[11] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, 2018, doi: 10.1007/s00521-017-3272-5.

[12] D. Alsadie, "TSMGWO: Optimizing task schedule using multi-objectives grey wolf optimizer for cloud data centers," *IEEE Access*, vol. 9, pp. 37707–37725, 2021, doi: 10.1109/ACCESS.2021.3063723.

[13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[14] Y. Yang, B. Yang, S. Wang, W. Liu, and T. Jin, "An improved grey wolf optimizer algorithm for energy-aware service composition in cloud manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 105, nos. 7–8, pp. 3079–3091, Dec. 2019, doi: 10.1007/s00170-019-04449-9.

[15] T. Jiang and C. Zhang, "Application of grey wolf optimization for solving combinatorial problems: Job shop and flexible job shop scheduling cases," *IEEE Access*, vol. 6, pp. 26231–26240, 2018, doi: 10.1109/ACCESS.2018.2833552.

[16] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018, doi: 10.1109/JIOT.2017.2780236.

[17] X.-Q. Pham, N. D. Man, N. D. T. Tri, N. Q. Thai, and E.-N. Huh, "A cost- and performance-effective approach for task scheduling based on collaboration between cloud and fog computing," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 11, Nov. 2017, Art. no. 155014771774207, doi: 10.1177/1550147717742073.

[18] H. Tan, Z. Han, X.-Y. Li, and F. C. M. Lau, "Online job dispatching and scheduling in edge-clouds," in *Proc. IEEE Conf. Comput. Commun.*, May 2017, pp. 1–9, doi: 10.1109/INFOCOM.2017.8057116.

[19] M. A. Benblidia, B. Brik, L. Merghem-Boulahia, and M. Esseghir, "Ranking fog nodes for tasks scheduling in fog-cloud environments: A fuzzy logic approach," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1451–1457, doi: 10.1109/IWCMC.2019.8766437.

[20] J. C. Guevara and N. L. S. da Fonseca, "Task scheduling in cloud-fog computing systems," *Peer-Peer Netw. Appl.*, vol. 14, no. 2, pp. 962–977, Mar. 2021, doi: 10.1007/s12083-020-01051-9.

[21] S. Nazir, S. Shafiq, Z. Iqbal, M. Zeeshan, S. Tariq, and N. Javaid, "Cuckoo optimization algorithm based job scheduling using cloud and fog computing in smart grid," in *Advances in Intelligent Networking and Collaborative Systems* (Lecture Notes on Data Engineering and Communications Technologies), vol. 23. Cham, Switzerland: Springer, 2019, pp. 34–46, doi: 10.1007/978-3-319-98557-2_4.

[22] S. Shadroo, A. M. Rahmani, and A. Rezaee, "The two-phase scheduling based on deep learning in the Internet of Things," *Comput. Netw.*, vol. 185, Feb. 2021, Art. no. 107684, doi: 10.1016/j.comnet.2020.107684.

[23] R. M. Abdelmoneem, A. Benslimane, and E. Shaaban, "Mobility-aware task scheduling in cloud-fog IoT-based healthcare architectures," *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107348, doi: 10.1016/j.comnet.2020.107348.

[24] D. Hoang and T. D. Dang, "FBRC: Optimization of task scheduling in fog-based region and cloud," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 1109–1114, doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.360.

[25] Z. Han, H. Tan, X.-Y. Li, S. H.-C. Jiang, Y. Li, and F. C. M. Lau, "OnDisc: Online latency-sensitive job dispatching and scheduling in heterogeneous edge-clouds," *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2472–2485, Dec. 2019, doi: 10.1109/TNET.2019.2953806.

[26] X.-Q. Pham and E.-N. Huh, "Towards task scheduling in a cloud-fog computing system," in *Proc. 18th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Oct. 2016, pp. 1–4, doi: 10.1109/APNOMS.2016.7737240.

[27] K. Dubey, M. Kumar, and S. C. Sharma, "Modified HEFT algorithm for task scheduling in cloud environment," *Proc. Comput. Sci.*, vol. 125, pp. 725–732, Jan. 2018, doi: 10.1016/j.procs.2017.12.093.

[28] X. Zhao and C. Huang, "Microservice based computational offloading framework and cost efficient task scheduling algorithm in heterogeneous fog cloud network," *IEEE Access*, vol. 8, pp. 56680–56694, 2020, doi: 10.1109/ACCESS.2020.2981860.

[29] T. S. Nikoui, A. Balador, A. M. Rahmani, and Z. Bakhshi, "Cost-aware task scheduling in fog-cloud environment," in *Proc. CSI/CPSSI Int. Symp. Real-Time Embedded Syst. Technol. (RTEST)*, Jun. 2020, pp. 1–8, doi: 10.1109/RTEST49666.2020.9140118.

[30] X. Xu, Q. Liu, L. Qi, Y. Yuan, W. Dou, and A. X. Liu, "A heuristic virtual machine scheduling method for load balancing in fog-cloud computing," in *Proc. IEEE IEEE 4th Int. Conf. Big Data Secur. Cloud (BigDataSecurity) Int. Conf. High Perform. Smart Comput., (HPSC) IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2018, pp. 83–88, doi: 10.1109/BDS/HPSC/IDS18.2018.00030.

[31] Y. Q. J. Sun, Q. Gu, T. Zheng, P. Dong, and A. Valera, "Computation offloading and task scheduling for DNN-based applications in cloud-edge computing," *IEEE Access*, vol. 8, pp. 118638–118652, 2020

[32] R. I. Tinini, G. B. Figueiredo, and D. M. Batista, "A batch scheduling algorithm for VPON reconfiguration and BBU migration in hybrid cloud-fog RAN," in *Proc. IEEE 18th Int. Symp. Netw. Comput. Appl. (NCA)*, Sep. 2019, pp. 1–8, doi: 10.1109/NCA.2019.8935022.

[33] P. Memari, S. S. Mohammadi, F. Jolai, and R. Tavakkoli-Moghaddam, "A latency-aware task scheduling algorithm for allocating virtual machines in a cost-effective and time-sensitive fog-cloud architecture," *J. Supercomput.*, vol. 78, no. 1, pp. 93–122, Jan. 2022, doi: 10.1007/s11227-021-03868-4.

[34] F. Fellir, A. El Attar, K. Nafil, and L. Chung, "A multi-agent based model for task scheduling in cloud-fog computing platform," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 377–382, doi: 10.1109/ICIoT48696.2020.9089625.

[35] A. M. Yadav, S. C. Sharma, and K. N. Tripathi, "A two-step technique for effective scheduling in cloud-fog computing paradigm," in *Advances in Intelligent Systems and Computing*, vol. 1086. Singapore: Springer, Jan. 2021, pp. 367–379, doi: 10.1007/978-981-15-1275-9_30.

[36] M. Farid, R. Latip, M. Hussin, and N. A. W. A. Hamid, "Scheduling scientific workflow using multi-objective algorithm with fuzzy resource utilization in multi-cloud environment," *IEEE Access*, vol. 8, pp. 24309–24322, 2020, doi: 10.1109/ACCESS.2020.2970475.

[37] Y. Yang, B. Yang, S. Wang, T. Jin, and S. Li, "An enhanced multi-objective grey wolf optimizer for service composition in cloud manufacturing," *Appl. Soft Comput.*, vol. 87, Feb. 2020, Art. no. 106003, doi: 10.1016/j.asoc.2019.106003.

[38] B. V. Natesha, N. K. Sharma, S. Domanal, and R. M. R. Guddeti, "GWOTS: Grey wolf optimization based task scheduling at the green cloud data center," in *Proc. 14th Int. Conf. Semantics, Knowl. Grids (SKG)*, Sep. 2018, pp. 181–187, doi: 10.1109/SKG.2018.00034.

[39] N. Bacanin, T. Bezdan, E. Tuba, I. Strumberger, M. Tuba, and M. Zivkovic, "Task scheduling in cloud computing environment by grey wolf optimizer," in *Proc. 27th Telecommun. Forum*, Nov. 2019, pp. 1–4, doi: 10.1109/TELFOR48224.2019.8971223.

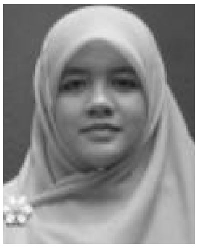
[40] G. Natesan and A. Chokkalingam, "Opposition learning-based grey wolf optimizer algorithm for parallel machine scheduling in cloud environment," *Int. J. Intell. Eng. Syst.*, vol. 10, no. 1, pp. 186–195, Feb. 2017, doi: 10.22266/ijies2017.0228.20.

[41] K. Sreenu and S. Malempati, "MFGMTS: Epsilon constraint-based modified fractional grey wolf optimizer for multi-objective task scheduling in cloud computing," *IETE J. Res.*, vol. 65, no. 2, pp. 201–215, Mar. 2019, doi: 10.1080/03772063.2017.1409087.

[42] H. Yu, Y. Gao, and J. Wang, "A multiobjective particle swarm optimization algorithm based on competition mechanism and Gaussian variation," *Complexity*, vol. 2020, pp. 1–23, Nov. 2020, doi: 10.1155/2020/5980504.


**FATEN A. SAIF** received the bachelor's degree in computer science from Sana'a University, in 2008, and the master's degree in computer science from Universiti Putra Malaysia, in 2019, where she is currently pursuing the Ph.D. degree with the Faculty of Computer Science and Information Technology. Her research interests include computer networks, workflow and task scheduling, cloud computing, the Internet of Things, optimization, workload allocation, and task offloading.

**ROHAYA LATIP** (Member, IEEE) received the bachelor's degree in computer science from Universiti Teknologi Malaysia, in 1999, and the M.Sc. degree in distributed systems and the Ph.D. degree in distributed database from Universiti Putra Malaysia. She was the Head of the HPC Section, Universiti Putra Malaysia, from 2011 to 2012. She consulted the Campus Grid Project and the Wireless for Hostel in Campus UPM Project. She is currently an Associate Professor with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. She is also the Head of the Department of Communication Technology and Networks and a Co-Researcher at the Institute for Mathematic Research (INSPEM). Her research interests include big data, cloud and grid computing, network management, and distributed database.



**ZURINA MOHD HANAPI** (Member, IEEE) received the B.Sc. degree in computer and electronic system from the University of Strathclyde, Glasgow, U.K., in 1999, the M.Sc. degree in computer and communication system engineering from Universiti Putra Malaysia (UPM), Malaysia, in 2004, and the Ph.D. degree from Universiti Kebangsaan Malaysia, in 2011. She is currently an Associate Professor with the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, UPM, where she has been a Lecturer, since 2004. She has authored over 70 articles in the cited journals and conferences in the area of security and wireless sensor networks. Her current research interests include security, routing, wireless sensor networks, wireless networks, distributed computing, and cyber-physical-systems. She is a member of the Malaysian Security Committee Research.

**KAMARUDIN SHAFINAH** received the Diploma, bachelor's, and M.Sc. degrees in computer science from Universiti Putra Malaysia, in 2000, 2003, and 2009, respectively, and the Ph.D. degree from Universiti Kebangsaan Malaysia, in 2016. She is currently a Senior Lecturer with Universiti Putra Malaysia. She has actively written book chapters and published numerous papers in journals and conferences. Her research interests include computer networks, management information systems, ICT for agriculture, and scholarship for teaching and learning (SoTL).