IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 March 2023, accepted 8 April 2023, date of publication 4 May 2023, date of current version 11 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3272901

RESEARCH ARTICLE

Cloudlet Based Computing Optimization Using Variable-Length Whale Optimization and Differential Evolution

LAYTH MUWAFAQ[1], **NOR K. NOORDIN**[1,2], **(Senior Member, IEEE)**, **MOHAMED OTHMAN**[3,4], **(Senior Member, IEEE)**, **ALYANI ISMAIL**[1], **(Member, IEEE)**, **AND FAZIRULHISYAM HASHIM**[1,2], **(Member, IEEE)**

[1]Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor D. E., Malaysia
[2]Research Centre of Excellence for Wireless and Photonics Network (WiPNET), Faculty of Engineering, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor D. E., Malaysia
[3]Department of Communication Technology and Networks, Universiti Putra Malaysia (UPM), 43400 UPM Serdang, Selangor D. E., Malaysia
[4]Laboratory of Computational Sciences and Mathematical Physics, Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia (UPM), 43400 UPM Serdang, Selangor D. E., Malaysia

Corresponding authors: Nor K. Noordin (nknordin@upm.edu.my) and Fazirulhisyam Hashim (fazirul@upm.edu.my)

This work was supported by Universiti Putra Malaysia - Research Management Centre (UPM-RMC) Ongoing Research - Adaptive Restricted Access Window (RAW) for Quality of Service (QoS) Provisioning in Industrial Internet of Things (IoT) Using Wi-Fi HaLow (UPM.RMC.800-3/3/1/GP-GPB/2021/9701500) under Grant 9701500.

ABSTRACT Cloudlet-based optimization involves deploying a set of cloudlets in an environment and assigning user tasks to optimize various metrics, including energy consumption, quality of service (QoS), and cost. Typically, approaches deal with them separately, which might cause sub-optimality. Furthermore, assuming the fixed location of the cloudlets will limit the dynamic adaptability of the problem. Enabling more optimality and adaptability to the dynamic nature of cloudlet-based computing, we propose a novel **V**ariable-**L**ength multi-objective **W**hale optimization **I**ntegrated with **D**ifferential **E**volution designated as **VL-WIDE**. Unlike the existing optimization algorithm, VL-WIDE features the capability of searching different lengths of solutions to cover the variable number of cloudlets for deployment. Furthermore, it enables a non-dominated evaluation of solutions based on four objectives using crowding distance for selection. It provides an application-oriented solutions repair operator for repairing non-valid solutions and assuring that all solutions are generated in the feasible region. The proposed algorithm enables moving the cloudlets among pre-defined locations to increase the quality of service according to the change in the user density caused by user mobility. Comparing this developed algorithm with other algorithms shows its superiority in multi-objective optimization (MOO) evaluation metrics. VL-WIDE has provided the best in a number of non-dominated solutions and delta metrics and was competitive in other metrics.

INDEX TERMS Mobile edge computing environment, cloudlet deployment, task offloading, multi-objective optimization, variable-length optimization.

I. INTRODUCTION

Cloud technology is gaining popularity globally with various applications, and its spending growth has been maintained between 26% - 38%, as shown in Figure 1. Mobile edge computing (MEC) is a well-known technique to support delay-sensitive applications at the edge of mobile networks. In recent years MEC has received significant attention from the academic and industrial communities [1]. MEC alleviates the shortcomings of traditional cloud computing by minimizing the delay of computation services for mobile devices [2]. One of MEC's critical challenges is selecting an efficient placement of the cloudlet and task offloading decision [3]. In the Mobile Edge Computing Environment (MECE), cloudlets can be collocated with the base station in the wireless metropolitan area network (WMAN) [4]. The

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

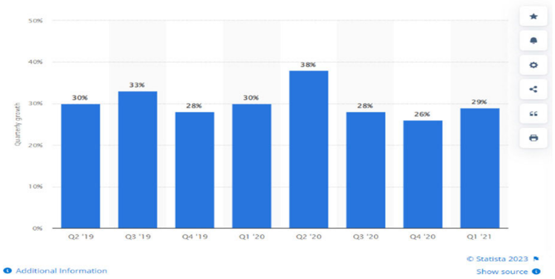

FIGURE 1. Statistical graph of the spending growth from Q2-2019, to Q1-2021 (www.statista.com).

latter is a wide area network consisting of many base stations (BSs) that allow mobile devices to access their needed services. On the one hand, the deployment of the base station is not a random process; rather, it is based on conducting certain optimization for selecting the best location for the base station to accomplish the maximum coverage [5]. On the other hand, deploying cloudlets at a certain base station should also result from an optimization algorithm aiming to maximize or minimize several factors. Hence, researchers have considered the problem of cloudlet deployment as one of the sub-problems of MEC [6]. However, a minority of studies have considered mobile cloudlets in order to enable dynamic deployment by moving cloudlets based on the temporal condition of MEC [7]. The integrated dynamic cloudlet deployment with task offloading (IDCD-TO) is a more general optimization problem compared with static cloudlet deployment separated from task offloading. Hence, IDCD-TO is an NP-hard optimization problem with combinatorial nature that has not been studied well in the literature.

Optimizing one aspect of a certain system is irregular in real-world applications due to more than one user satisfaction perspective. This has led researchers to develop the concept of Pareto-optimization, which assesses a certain decision regarding the system using a set of satisfaction metrics [8], e.g., delay, cost, energy, and quality of service. Consequently, instead of dealing with one optimal solution, we consider a set of non-dominated solutions that are provided to the decision maker or to an automated process for selecting one of them to be enabled according to certain criteria such as cost, time, and energy consumption. Some famous algorithms for multi-objective optimization are the non-dominated sorting genetic algorithm (NSGA-II) [9], NSGA-III [10], and the multi-objective evolutionary algorithm based on decomposition (MOEAD) [11].

Traditional meta-heuristic optimization algorithms consider a fixed length of the solution space, but this does not apply to many real-world problems. The reason for this is that certain values of some decision variables may generate or disable other decision variables, resulting in a variable-length nature of the solution space caused by different solution lengths [12]. Dealing with such types of problems requires special types of operators that are aware of the length variability of the solution space and capable of covering all dimensions of solutions during the search. Some of the recent algorithms that were developed with variable-length handling are the work of [13] and [14].

IDCD-TO is a multi-objective variable-length optimization problem. The multi-objective nature comes from having more than one metric to be optimized, i.e., energy consumed by cloudlet, energy consumed by the user device, delay of execution, and cost of deployment. It is observed that these objectives have a self-conflict nature. The variable-length nature comes from having more than one cloudlet to be deployed according to the demand. This article aims to propose an integrated whale optimization algorithm with differential evolution for IDCD-TO. Our proposed algorithm is designated as Variable-length multi-objective Whale optimization Integrated with Differential Evolution designated as (VL-WIDE). This article presents several contributions. We state them as follows:

- To the best of our knowledge, this is the first article that enables tackling the problem of cloudlet-based computing with an additional degree of freedom that enables not only deploying the cloudlets in optimal locations but also moving them according to the geographical demands information and integrating this with task offloading between more than one cloudlet for better load balancing.
- It presents a novel formulation of the optimization problem of cloudlet-based computing using the variable-length of solution space. This enables reserving a compact representation of the decisions in terms of the variables needed for locating the cloudlets and offloading the tasks from the user to the cloudlets.
- It provides an application-oriented solution repairing operator for fixing non-valid solutions and assuring that all solutions are generated in the feasible region.
- It incorporates variable-length searching within a hybrid framework combined with multi-objective whale optimization and differential evolution. Hence, it provides the literature with the first variable-length searching of multi-objective hybrid whale-differential evolution optimization.
- This article presents an extensive evaluation using 300 scenarios generated randomly for the variable affecting the parameters of the system. Furthermore, it compares it with state-of-the-art algorithms, including NSGA-II, NSGA-III, MOEAD, PSO, and fixed length hybrid whale optimization-differential evolution with standard MOO performance metrics, namely, hypervolume, set coverage, and delta-metrics.

The remainder of this article is organized as follows. Section II presents the related research. In Section III we provided the system model that presents problem formulation, whale optimization, differential evolution, framework, and evaluation metrics. The experimental works and analysis are provided in Section IV, and finally, the conclusion and future works are given in Section V.

II. RELATED RESEARCH

The literature is decomposed into two sub-sections. The first one is a survey about cloudlet optimization algorithms in the literature based on meta-heuristic optimization and the second one is a survey of the algorithms that were developed in the area of meta-heuristic optimization in general with the feature of variable-length despite the application nature.

A. CLOUDLET OPTIMIZATION

Cloudlet optimization involves cloudlet deployment and task offloading, and the literature is rich with numerous works for cloudlet deployments. In the work of [15], the authors have explored the edge server deployment problem in a large-scale mobile edge computing system to balance workload amongst edge servers and minimize edge server access time. Then, they described the problem as a multi-objective constraint optimization and presented a mixed integer programming (MIP) edge server deployment method to calculate the best solution. In the work of [16], the Cost Aware cloudlet PlAcement in moBiLe Edge computing (CAPABLE) technique is developed by considering both cloudlet cost and average E2E delay when placing cloudlets. A Lagrangian heuristic algorithm is developed to achieve the suboptimal solution to this problem. In addition, the creation of a workload allocation mechanism after cloudlets are installed to minimize the E2E delay between users and their cloudlets by taking user mobility into account. In the work of [17], In WMAN, concentrating on cloudlet placement for Cyber-Physical-Social Systems. Technically, a new approach is proposed based on affinity propagation with optimal preference (APOP). In the work of [18], the heterogeneous cloudlet placement problem has been tackled with a bifactor approximation approach that guarantees a bounded latency and placement cost while properly mapping user applications to appropriate cloudlets. The problem is first formulated as a multi-objective integer programming model. The bifactor approximation algorithm (ACP) has been proposed to deal with its intractability. This algorithm requires an estimate of the user demands to find the appropriate solution without the upgrade. In the work of [19], a particle swarm optimization algorithm utilizing genetic algorithm operators with the encoding library updating mode (PGEL) was used to address the problem of cloudlet deployment based on workflow applications (WAs) in wireless metropolitan area networks (WMANs). Their approach enables the cloudlet to be deployed in appropriate positions, which minimizes the execution time of WAs.

Task offloading is a method used in the MEC environment to improve the performance of mobile devices by offloading tasks to nearby cloudlets collocated with the BSs in the WMANs. In the work of [20], a new concept of Digital Twin Edge Networks (DITEN) was provided, in which edge server digital twins (DTs) estimate edge server states, and the DT of the entire MEC system provides training data for offloading decisions. In DITEN, a mobile offloading system is presented to reduce offloading delay while balancing the cost of accumulated utilized service transfer during user mobility. The Lyapunov optimization method is used to reduce the long-term migration cost constraint to a multi-objective dynamic optimization problem that is solved using Actor-Critic deep reinforcement learning. In the work of [21], In MEC, task offloading is modeled as a constrained multi-objective optimization problem (CMOP) that aims at minimizing mobile devices' energy consumption and task processing time. To solve the CMOP, an evolutionary method is proposed capable of identifying a representative sample of the optimum trade-offs between energy consumption and task processing latency, referred to as the Pareto-optimal front. The development includes initialization, selection, crossover, and mutation. The added constraints have enabled handling the two competing objectives better than the non-constraint approach. They included three constraints: the first one guarantee that all task portions are computed either locally or at a remote device. The second one prevents the ECs from offloading tasks to non-neighboring devices, and the third constrains the offloading decisions to non-negative values. The work of [22] proposes a Multi-Objective Whale Optimization Algorithm (MOWOA) based on time and energy consumption. Crowding distance was used to improve the quality of the solution set, and crowding degrees were used to sort the solutions. Also, an improved MOWOA (MOWOA2) based on the gravity reference point method was proposed to increase the number of possible solutions.

According to the specifications set by the European Telecommunication Standard Institute (ETSI)-MEC, [23] developed a method for predicting spatiotemporal dynamics of mobile users and allocating communication and computational resources at the edge of the network based on those predictions. The proposed architecture uses Software-Defined Networking to monitor user mobility and employs Convolutional Long Short-Term Memory (ConvLSTM) architecture to predict the number of users and their related service requests over various horizons, followed by Dynamic Programming to optimally allocate requests to Multi-access edge servers. In [24], the authors considered the problem of cooperative task offloading in a distributed MEC environment, where the task can be migrated between servers. To address this, they proposed a multi-agent actor-critic framework using a Variational Recurrent Neural Network (VRNN) based global state-sharing model to reduce communication overhead. Additionally, they utilized a Long Short-Term Memory (LSTM) based state estimation model to learn the spatiotemporal dynamics of tasks. They also presented a cooperative task offloading algorithm based on multi-agent deep reinforcement learning, which significantly reduces computational complexity.

Other works in the literature consider cloudlet deployment and task offloading simultaneously to address the cloudlet optimization problem. In the work of [3], the cloudlet deployment and task offloading from user to cloudlet have been modeled using $M/M/c$. Three algorithms were proposed, namely, Heaviest-AP First Placement, Density-Based

TABLE 1. Summary of existing literature for cloudlet based computing optimization and comparison with this work.

Article	Cloudlet deployment	Task offloading	Inter-Cloudlet flow	Variable-length Cloudlets	Users Mobility	Cloudlet Mobility	Delay	Cost	Energy consumption	Load balancing
[15]	√	×	×	×	×	×	√	×	×	√
[16]	√	×	×	×	√	×	√	√	×	√
[17]	√	×	×	×	√	√	√	×	×	√
[18]	√	×	×	×	×	×	√	√	×	×
[19]	√	×	√	×	×	×	√	×	×	×
[20]	×	√	√	×	√	×	√	×	×	√
[21]	×	√	×	×	×	×	√	×	√	√
[22]	×	√	×	×	×	×	√	×	√	×
[23]	×	√	√	×	√	×	√	×	×	√
[24]	×	√	√	×	√	×	√	×	×	√
[3]	√	√	×	×	×	×	√	×	×	√
[7]	√	√	×	×	×	√	√	×	×	√
[25]	√	√	×	×	×	×	√	×	√	×
[26]	√	√	√	×	×	×	√	√	√	×
[27]	√	×	×	×	×	×	√	√	×	×
This paper	√	√	√	√	√	√	√	√	√	√

Clustering Placement, and K-medians clustering algorithm. In the work of [7], a dynamic cloudlet deployment technique based on a clustering algorithm (DCDM-CA) is proposed for deploying mobile cloudlets for mobile applications. Based on the geographic location of numerous devices and the number of tasks generated by multiple devices in a unit of time, DCDM-CA calculates the cloudlet deployment location. In addition, after deploying cloudlets, task offloading is optimized to reduce system response latency. In [25], the authors first studied how to determine task completion delays and produced energy consumption models for various MEC equipment. Following that, they investigated the problem of placing cloudlets on AP nodes and allocating each user's tasks to associated cloudlets and the public cloud, intending to minimize total energy consumption while satisfying each task's delay requirement. A mixed integer linear programming and solution benders decomposition-based algorithm has been proposed to overcome the problem. In the work of [26], the issue of task offloading and combined cloudlet deployment are addressed. Its goals are to reduce user energy usage, task response time, and the number of cloudlets deployed. This optimization issue is modelled as a mixed integer nonlinear program. A modified directed population archive whale optimization technique and its time complexity are described. The methods of encoding, initializing, and restoring a whale position are designed in this algorithm. In addition, generalized and quasi-opposition-based learning, as well as mutation and crossover in differential evolution algorithm, are used to boost its optimization capacity. The work of [27] proposes a procedure combining edge server deployment and service placement, where service placement explicitly considers the structure of current edge server deployment and different service request rates and prices. Furthermore, design a joint edge server deployment and service placement model to maximize the overall profit of all edge servers under the constraints of the number of edge servers, the edge servers and base stations relationship, the computing capacity of each server, and the storage capacity. Hence, to solve the problem, propose a two-step method, the clustering algorithm and nonlinear programming.

We summarize the different criteria considered in the literature using Table 1. Our approach will consider all the criteria provided in Table 1. Some of the criteria are tackled implicitly, such as load balancing. Overall, we observe from Table 1 that non-of the existing approaches have jointly dynamically addressed the cloudlet deployment and task offloading. More specifically, the joint cloudlet deployment and task-offloading have been addressed only by [3], [25], and [26]. However, they have not considered cloudlet mobility, and by [7], where they have considered cloudlet mobility but used a fixed number of cloudlets. Furthermore, enabling joint optimization of cloudlet deployment, deactivation and activation by moving, task offloading, and inter-cloudlet flow is an optimization problem with a variable number of decision variables. Hence, this optimization is regarded as a special class of optimization algorithm that needs careful study. Furthermore, it becomes more complex when considering its multi-objective nature due to various performance criteria, including energy consumption, cost, and latency with a self-conflict nature.

To cover this issue, we dedicate another section in the literature survey to addressing the existing variable-length multi-objective optimization algorithms used in other fields.

TABLE 2. Review of existing variable-length meta-heuristic optimization algorithms.

Article	Application	Decision space	Objective space	Algorithm	Limitation	Number of objectives
[28]	Laminate stacking Wind farm Sensor coverage	Based on the problem	Based on the problem	Metameric Genetic	It is based on single objective	1
[14]	Laminate stacking problem	Vector of angles on the plies of a laminate.	The number of plies buckling load factor	Multi-objective Evolutionary algorithm	It is only tested on Mathematical problems without real world application	2 / 3
[29]	A coverage and a wind farm problem are used	Based on the problem	Based on the problem	Evolutionary algorithm	It is based on single objective	1
[30]	WSN deployment	Sensors numbers location and coverage zone	Covering percentage and cost	Genetic algorithm	It is based on single objective	1
[31]	Sensor node scheduling	Deciding the sensor that will send data	Simulator	Genetic algorithm	It is based on single objective	1
[32]	the vehicle coordination multipath problem	road traffic intersection	the set of possible routes	genetic algorithm	It is based on single objective	1
[33]	Changing the topology of CNN	Encoding neuron in the layer	Accuracy	Particle swarm optimization	It is based on single objective	1
[13]	WSN deployment	Sensors numbers location and coverage zone	Covering percentage and cost	Social class variable-length particle swarm optimization	Inter-class interaction is weak	2
[34]	IIR filters and sensor coverage	designing optimal IIR filters	Based on the problem	particle swarm optimization algorithm	It is based on single objective	1

B. VARIABLE-LENGTH MULTI-OBJECTIVE OPTIMIZATION

Several applications have found the development of variable-length multi-objective algorithms in the literature, Table 2 shows the algorithms developed in meta-heuristic optimization with the variable-length approach to solving problems in different fields. In the work of [28], a metameric genetic algorithm (MGA), which uses a segmented variable-length genome, is proposed. The development covers the recombination, mutation, and selection operators, as well as the representation of the solution in the genome. Furthermore, the algorithm has proved that even if the optimal number of components is supposed to be known a priori, the MGA outperforms the fixed-length GA on the specified problems with these adjustments. In the work of [14], the variable-length multi-objective evolutionary algorithm was proposed based on a two-level decomposition strategy (local and global) that decomposes a multi-objective optimization problem in terms of penalty boundary intersection search directions and variable dimensionality. In the work of [29], a novel metameric problem selection operator is presented: length niching selection. First, the population is divided into numerous niches based on the length of the solution. A window function determines the lengths at which a niche is generated. Local selection is done individually within each niche, resulting in a new parent population with various solution lengths. In the work of [30], the improved Dynamic Deployment Technique based on the Genetic Algorithm (IDDT-GA) has been proposed to maximize the area covered with the minimum number of nodes and undervalue overlapping area between neighbouring nodes. A two-point crossover novel is presented to confirm the notation of variable-length encoding. In the work of [31], an expanded genetic algorithm with the variable-length chromosome (VLC) and mutation and crossover operations is introduced to address the problem of sensor node scheduling. Their method may evolve the population's individuals from generation to generation, and the resulting network schedules are better optimized than those produced by algorithms using a fixed-length chromosome. In the work of [32] to solve the vehicle coordination multipath problem in crossroads, propose a genetic algorithm with variable-length chromosomes. The proposed algorithm is focused on optimizing the arrival sequencing of vehicles according to the present flow rates where the traffic flows can be asymmetric. In addition, expand one of the existent crossroad models based on fixed paths to allow multiple paths. Furthermore, each vehicle can move at the crossroad from any input point to any output point. Moreover, developed specific selection, crossover, and mutation operators and a new methodology to carry out the crossover function between different-sized individuals are adapted to the specific peculiarities of the problem.

Other algorithms were based on the swarm class of metaheuristics. The work of [33], focuses on using Particle Swarm Optimization (PSO) to find the best architecture for CNNs without having to do any manual tuning. Based on classic PSO, CNN layers were encoded using particle vectors. A disabled layer is used to learn variable-length CNN and is designed to disguise parts of the particle vector's dimensions to accomplish particles with varying lengths. In the work of [13], SC-MOPSO (Social Class Multi-objective Particle Swarm Optimization) was proposed to solve joint MOO and V-length optimization challenges. By dividing the solution space into classes based on their dimension, the technique expands the concept of social interaction from Particle Swarm Optimization. It also includes intra and inter class operators to ensure that the required dynamics of solution changes are met to reach the Pareto front. It was applied to the wireless sensor network problem, two objective problems, and the length variability comes from the change in the number of sensors. In the work of [34], the variable length particle swarm optimization algorithm with a weighted sum fitness function (WS-VLPSO) is proposed as an adaptive algorithm for designing optimal Infinite Impulse Response (IIR) filters. The algorithm is based on including the order as a discrete variable in the particle vector to minimize the order and thereby reduce the design complexity of IIR filters. In addition, the Optimum Modeling Indicator (OMI) is introduced as a measure to determine the percentage reduction of order and the success rate of the proposed Method. Furthermore, the proposed algorithm is applied to solve the sensor coverage problem as another real-world variable-length optimization application.

Overall, as shown in Table 2 from the reviewed algorithms the majority of them were developed to support a single objective except for the work of [13], which suffers from weak interclass interaction between the solutions and the work of [14] that based on an evolutionary algorithm and was applied only on a bi-objective real-world problem. Hence, the literature lacks a multi-objective swarm-based algorithm with a variable-length feature. We fill this research gap by developing a novel variable-length whale optimization algorithm with supportability of multi-objective aspects. The developed algorithm will be applied to solve the cloudlet optimization problem presented in this article.

III. SYSTEM MODEL

This section presents the developed methodology for building cloudlet-based computing optimization using a variable-length whale optimization algorithm and differential evolution. It starts with problem formulation. Following, we present an overview of the whale optimization algorithm and differential evolution. Then the development framework is given, and the evaluation metrics are provided. All the notations used in this article are presented in Table 3.

TABLE 3. List of the important notations used in the system model.

Notations or symbols with their definitions:	
BS:	Number of base stations
U:	Number of users
C:	Number of cloudlets
N_C:	Number of deployed cloudlets
N_{BS}:	Total number of base stations
bs_k:	base station k
u_i:	User i
c_j:	Cloudlet j
φ^c:	Computing capacity (CPU cycles/second) of each cloudlet
μ:	Maximum workload (CPU cycles/second) of each cloudlet
x_t:	The x position of the user at moment t
y_t:	The y position of the user at moment t
σ^2:	The variance of the user mobility random walk model
u_t:	Step of random walk at moment t
$E_{i,d}$:	The expected energy consumption for executing task on u_i
λ_i:	Average task arriving rate generated from u_i
$z_{i,j}$:	Probability of assigning the task generated from u_i to c_j
e_i^u:	Energy consumption for executing tasks generated from u_i at his device
β_i:	Exponential resource demand of tasks generated from u_i
φ_i^u:	Computing capacity of u_i
ξ_i:	Effective switching capacitance of the CPU of u_i
E_{it}:	Energy consumption for transmitting the task from u_i to c_j
$\mathrm{e}_\mathrm{i}^\mathrm{t}$:	The required energy for transmitting a task of u_i to its related BS
p_i:	Transmission power of u_i
α_i:	The input data size of the task of u_i
r_i^u:	Transmission rate of u_i
$E_{i,u}$	The expected energy consumption by u_i
$E_{i,c}$:	The expected energy consumption for executing the task at c_j
e_i^c:	Energy consumption for executing tasks generated from u_i at c_j
ξ^c:	Effective switching capacitance of the CPU of c_j
L:	The maximum number of currently operating cloudlets $\mathrm{L} \leq \mathrm{N_C}$
τ_i^c:	Average waiting time composing of the queue waiting time and the execution time at c_j

A. PROBLEM FORMULATION

Assuming that we have a mobile edge computing environment $MECE$ that contains a set of base stations $BS = \{bs_k\}, k = 1, 2, \ldots, N_{BS}$ to serve users $U = \{u_i\}, i = 1, 2, \ldots, N_u$ for offloading their tasks to movable cloudlets $C = \left\{c_j\right\}, j = 1, 2, \ldots, N_C$, where $N_C \leq N_{BS}$. Each cloudlet is equipped with the same number of servers. Denote φ^c and μ as the computing capacity (CPU cycles/second) and the maximum workload (CPU cycles/second) of each cloudlet, respectively. The set of cloudlets C are deployed at selected base stations $SBS \subseteq BS$. The cloudlets in C can move between base stations for a limited number of times per day $N_{m,max}$ for each movable cloudlet. Hence, the environment elements, base stations, and cloudlets combine two graphs, where the first one contains the second one. Another difference is that the first is static, and the second is dynamic in terms of topology. The environment can be occupied by walking users which are given by the Eqs. (1), (2), and (3). Each user carries a mobile device. Users are assumed to be walking in the environment in the random walk model.

$$x_{t+1} = x_t + u_t \tag{1}$$

$$y_{t+1} = y_t + u_t \tag{2}$$

$$u_t \sim N(0, \sigma^2) \tag{3}$$

Users generate their tasks and can serve them locally on their devices or offload them to the assigned cloudlet through the nearest base station and shortest path. The same bandwidth B is shared by all users. The model of generating the task for each user is exponential, with an average arrival rate λ_i. The input data size and computational resource demand can characterize a task of the user u_i. Both are supposed to satisfy the exponential distribution, and their average values are represented by α_i (bits) and β_i (CPU cycles). Each user has two options, executing their tasks at the local device or sending them to *MECE* for execution. The device of the user is characterized with computing capacity φ_i^u and effectively switched the capacitance of the CPU ξ_i. We assume that *MECE* is provided with an optimization algorithm that generates the decision for selecting a cloudlet for the execution of the user tasks, moving the cloudlets among base stations, and changing their number N_C in order to optimize the system from four perspectives:

- Computational energy consumption at the user device, which is given by the computational energy consumption at the user device: it indicates the consumed energy at the user device by executing the tasks instead of offloading it to the cloudlet. It is given by the Eqs. (4) and (5).

$$E_{i,d} = e_i^u \lambda_i \left(1 - \sum\nolimits_{1 \le j \le L} z_{i,j}\right) \tag{4}$$

$$e_i^u = \xi_i \left(\varphi_i^u\right)^2 \beta_i \tag{5}$$

- Communication energy consumption at the user device: it indicates the energy consumption at the user device for transmitting the task to be executed at the cloudlet

$$E_{it} = e_i^t \lambda_i \sum\nolimits_{1 \le j \le L} z_{i,j} \tag{6}$$

where

$$e_i^t = p_i \frac{\alpha_i}{r_i^u} \tag{7}$$

- The expected energy consumption at the cloudlet, which defines the energy consumption in the cloudlet that is resulted from executing the task of the user at the cloudlet

$$E_{i,c} = e_i^c \lambda_i \left(\sum\nolimits_{1 \le j \le L} z_{i,j}\right) \tag{8}$$

$$e_i^c = \xi^c \left(\varphi^c\right)^2 \beta_i \tag{9}$$

- Number of deployed cloudlets N_C.
- Delay of serving the offloaded tasks (the average of wireless delay and waiting time composing of the queue waiting for time and the execution time at cloudlet c)

$$\tau_i^c = \frac{\dfrac{\sum_{1 \le j \le U} z_{j,i}\lambda_j\beta_j}{\sum_{1 \le j \le U} z_{j,i}\lambda_j}}{\varphi^c - \sum_{1 \le j \le U} z_{j,i}\lambda_j\beta_j} \tag{10}$$

The system has several challenges:

- It presents a combinatorial NP-hard optimization problem.
- The solution space is variable-length due to the change, which implies changing the solution length. It consists of two components:
  a) x which is $1 \times N_C$ a vector that is responsible for deciding which base station is assigned to the corresponding cloudlet.
  b) z which is $1 \times (N_C \times N_u)$ vector that is responsible for deciding the probability of selecting the corresponding cloudlet for executing the user task.
  The length variability occurred regarding N_C as a decision variable that provides different solution lengths.
- The objective space is a multi-objective combination of four objectives with self-conflicting nature.

$$f = (f_1, f_2, f_3, f_4) = (E_{i,u}, E_{i,c}, \tau_i^c, N_C) \tag{11}$$

where

$$E_{i,u} = E_{i,d} + E_{it} \tag{12}$$

- The problem is dynamic due to user mobility. However, the algorithm will receive a snapshot of the users inside the map at one execution. The decision to distribute the cloudlet and the task offloading to cloudlets will be generated using this snapshot.

B. OVERVIEW OF WHALE OPTIMIZATION

The thought of the whale optimization algorithm (WOA) [35] is to imitate the behaviours of humpback whales. When humpback whales are hunting, they have three behaviours associated with a bubble net, upward spirals, double-loops, and random search. It is based on two parameters, random number $r \in [0, 1]$ and two coefficient vectors A and C given by Eqs. (13) and (14).

$$A = 2 \cdot a \cdot r - a \tag{13}$$

$$C = 2 \cdot r \tag{14}$$

where a denotes a linearly decreased from two to zero in the iterations.

The model of the bubble net is given by Eqs. (15) and (16)

$$P_i^{t+1} = D' \cdot e^{bl} \cdot \cos(2\pi l) + P_*^t \tag{15}$$

$$D = \left|P_*^t - P_i^t\right| \tag{16}$$

The model of Encircling Prey is given by Eqs. (17) and (18)

$$D = \left|C \times P_*^t - P_i^t\right| \tag{17}$$

$$P_i^{t+1} = P_i^t - A \times D \tag{18}$$

The model of random search is given by Eqs. (19) and (20)

$$D = \left|C \times P_{\text{rand}} - P_i^t\right| \tag{19}$$

$$P_i^{t+1} = P_{\text{rand}} - A \times D \tag{20}$$

Algorithm 1 Whale Optimization (WO)

```
Input
Parameters
Output
SolutionSet
Start
Initialize the whales population $P_i (i = 1, 2, \ldots, n)$
Calculate the fitness of each search agent
P$_*$= the best search agent
while (t < maximum number of iterations)
    for each search agent
    Update a, A, C, l, and w
        if1(w < 0.5)
            if2(|A| ≤ 1)
                Update the position of the current search
                agent by Eq. (18)
            else if 2 (|A|> 1)
                Select a random search agent (P$_{rand}$)
                Update the position of the current search
                agent by Eq. (20)
            end if2
        elseif1(w≥0.5)
                Update the position of the current
                search by Eq. (15)
        end if1
    end for
    Check if any search agent goes beyond
    the search space and amend it
    Calculate the fitness of each search agent
    Update P$_*$ if there is a better solution
    t=t+1
end while
returnP$_*$
End
```

C. DIFFERENTIAL EVOLUTION

More than two decades ago, the DE algorithm emerged as a highly competitive form of evolutionary computing. R. Storn and K. V. Price published the first written article on DE as a technical report in 1995 [36]. The algorithm comprises four steps, initialization, difference vector-based mutation, crossover, and selection [37]. In our article, the solutions are already generated and selected by WOA. Hence, we perform only crossover and mutation. For crossover, two mutually different solutions, S and S' or three S, S', and S'', must be randomly selected from the archive before the updating. The probabilities of selecting the two or three solutions are the same. The update is based on Eq. (21) for selecting two and Eq. (22) for selecting three.

$$S_i^{t+1}[j] = \mathbb{S}[j] + F\left(\mathbb{S}'[j] - \mathbb{S}[j]\right) \tag{21}$$

$$S_i^{t+1}[j] = \mathbb{S}[j] + F\left(\mathbb{S}'[j] - \mathbb{S}''[j]\right) \tag{22}$$

After updating the solution, we perform the mutation by changing randomly selected components from the solution.

D. FRAMEWORK OF DEVELOPMENT

The framework of cloudlets computing optimization is presented in Figure 2. As shown, the environment E which contains the users, their location, and generate tasks, and the map information that includes the positions of users, base stations, and the initial position of cloudlets, will provide this information to the optimization algorithm. The optimization consists of three sub-blocks, whale optimization, differential evolution, and solution repairing. The results are represented in the set of non-dominated solutions or Pareto front. One of the solutions will be selected to be used in the environment using uniform random selection.

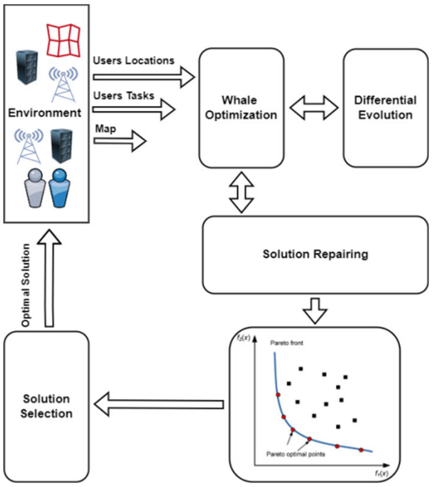

FIGURE 2. Block diagram of the framework of cloudlets computing optimization.

The algorithm is an integration of whale optimization and differential evolution. It is depicted as VL-WIDE. The general pseudo code of the algorithm is provided in Algorithm 2.

Algorithm 2 **V**ariable-**L**ength Multi-Objective **W**hale Optimization **I**ntegrated With **D**ifferential **E**volution (VL-WIDE)

```
Input:
Initial Population
Output:
Updated Population
Start:
For all solutions inside population
    r=generateRandom(0,1)
    If (r≥ 0.5)
        Use whale optimization for updating
         the position of the solution
    Else
        Use differential evolution for updating
         the position of the solution
    End
repairedSolution= Call repairing Algorithm
End
End
```

1) INITIAL POPULATION AND SOLUTION REPAIRING

The role of the solution repairing is to assure that all generated solutions are within the feasibility region. For solutions that are outside the feasibility region, an altering operation of the solution is performed in order to return the solution to the feasibility region. The pseudo-code of the initial population and solutions repairing is presented in Algorithms 3 and 4,

Algorithm 3 Initial Population

```
Input:
Population size, number_of_user, number_of_cloudlets
Output
Population
Start
Population =[ ]
For from 1 until Population_size
    Solution=Generate_random
    (1, number_of_cloudlets, number_of_cloudlets × number_of_user)
    Add Solution to Popluation
End
Return Population
End
```

Algorithm 4 Solution Repairing

```
Input:
Solution, User_total_offloads, cloudlet.max_workload
Output:
Repaired_solution
Start
[cloudlets_position,user_offload]=extract(solution)
cloudlets_position = min(cloudlet_posisions,
    number_of_base_stations)
cloudlet_position = max(cloudlet_position, -1)
    # -1 represent cloudlet would not be used.
for user_offload in total_user_offloads:
    remaining_offloads = user_offload
    cloudlet_position = shuffle(cloudlet_position)
    for cloudlet in cloudlet_position:
        if cloudlet != -1 #this cloudlet is working.
            taken_offload = min(random[0->1]
            * user_offload, remaining_offloads)
            taken_offload = min(taken_offload,
            cloudlet.max_workload - taken_offload)
            remaining_offload = remaining_offload
              - taken_offload
            cloudlet.workload + = taken_offload
            user_cloudlet_offload[user, cloudlet]
            + = taken_offload
        else  if remaining_offloads != 0:
            Distribute them to other cloudlets
        End
    End
End
Repaired_solution =solution
update Repaired_solution, user_cloudlet_offload,
    cloudlet_position
return Repaired_solution
End
```

respectively. The inputs of the pseudo-code for algorithm 3 are given by Population size, number_of_base_stations, number_of_user, and number_of_cloudlets. The output is the generated solutions that represent a population. Furthermore, the inputs of the pseudo-code for algorithm 4 are given by Solution, User_total_offloads, and cloudlet.max_workload. The output is the solution after possible repair. The algorithm starts with extracting two types of information from the solution, cloudlets positions and user-to-cloudlets offloads. The cloudlet's position is fixed by assuring their range is between -1, which indicates non-used cloudlets and number_of_base_stations, representing the maximum possible position of the cloudlet. Afterward, for each user_offload, a loop is performed to distribute the user_offload on the cloudlets with various percentages while assuring that any cloudlet will not receive a load beyond its capacity, which is given by cloudlet.max_workload.

2) SOLUTION SELECTION

The role of solution selection is to select a subset from a pool of solutions. The pool includes both the previous generation and the current generation. For solution selection, two mechanisms have been used. The first one is non-dominated sorting which is responsible for reordering the solutions in the repository, starting with the first rank, which represents the non-dominated subset of solutions, and ending with the last rank. The pseudo-code of the non-dominated sorting is presented in algorithm 5. It takes a population as input and returns the Pareto front as output. The approach is iteratively attempting to add one solution from the population to the Pareto front and assuring that it is non-dominated by its members until the end. The second one is the crowding distance which is used for solving the problem of selecting part of the last rank solutions to fit the size of the repository. The pseudo-code for calculating the crowding distance is given in algorithm 6.

Algorithm 5 Non-Dominated Sorting

```
Input:
Population
Output:
paretoFront
Start:
Initiate paretoFront
move random solution from Population to paretoFront
For each solution p in Population
Move p to paretoFront
Compare p with every solution q in paretoFront
    If(p is dominating q )
      Delete   q from paretoFront
    else
      Delete p from paretoFront
    End
End
End
```

E. EVALUATION METRICS

This section presents the evaluation metrics used to evaluate our proposed VL-WIDE and compares it with the benchmarks. It decomposes hyper-volume, delta-metric, number of non-dominated solutions, and set coverage.

1) HYPER VOLUME

The HV metric has been used widely in evolutionary multi-objective optimization to evaluate the performance of search algorithms. It computes the volume of the dominated portion of the objective space relative to the worst solution (reference point); this region is the union of the hypercube whose diagonal is the distance between the reference point and a solution x from the Pareto set P_s [38]. Higher values of this measure indicator imply desirable solutions. Hyper

Algorithm 6 Crowding Distance

```
Input:
ParetoFront, set of Objectives
Output:
CrowdingDistance
Start:
Initiate CrowdingDistance with the size of
  ParetoFront and assign zeros
For each objective $m$ of set of objectives
    Sort ParetoFront according to objective
        Set CrowdingDistance for the first and last solution to infinity
    For the remaining solutions
            Update CrowdingDistance based on the value of
            objective $m$ for the neighboring
            solutions using equation $CorwdingDistance[i] += (ParetoFront(i+1, m) - 8 : ParetoFront(i-1, m)$
        End
End
End
```

volume is given by Eq. (23):

$$HV = \text{volume}\left(\bigcup_{x \in p_s} \text{HyperCube}(x)\right) \tag{23}$$

2) DELTA METRIC

Delta metric is another variety of metric Δ. Indicates the extent to which spread is achieved among the obtained solutions. To calculate the delta metric, the function receives the non-dominated set of solutions and provides the result according to Eq. (24) [9]:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left|d_i - \bar{d}\right|}{d_f + d_l + (N-1)\bar{d}} \tag{24}$$

where the number of solutions is denoted by N, and the samples of d_f, d_l and d_i are the Euclidean distances between the boundary solutions and the extreme solutions and the average of all successive distances d_i is denoted by $\bar{d}$ for $i = 1, 2, \ldots, N-1$.

This measure should be as small as possible because this indicates a uniform distribution, which provides a variety of choices to the decision-maker.

3) NUMBER OF NON-DOMINATED SOLUTIONS

The cardinality of P_S can be used to calculate the number of non-dominated solutions (NDS) that represent the effectiveness of the optimization algorithm [39]. The higher number of non-dominated solutions means that the multi-objective optimization performs well.

$$\text{Number of non} - \text{dominated Solutions}(N) = |P_S| \tag{25}$$

4) SET COVERAGE

This metric compares Pareto sets p_{s1} and p_{s2} [40], this metric also called C-metric

$$C\left(P_{s1}, P_{s2}\right) = \frac{|\{y \in P_{s2} \mid \exists x \in P_{s1} : x \prec y\}|}{|P_{s2}|} \tag{26}$$

where the domination of x over y is indicated by $x \prec y$, and the value of C denotes the ratio of non-dominated solutions in P_{s2} that are dominated by non-dominated solutions in P_{s1} to the number of solutions in P_{s2}. When evaluating the set of P_s, X must be minimized for the value of $C\left(X, P_s\right)$ for all Pareto sets.

IV. EXPERIMENTAL WORKS AND ANALYSIS

This section presents the experimental works and analysis. It consists of experimental design in section IV-A and experimental results in section IV-B. To evaluate the performance of the proposed algorithm we first introduce the experimental design and then compare the proposed algorithm with existing benchmark algorithms, which are decomposed into two sub-sections, multi-objective metrics, and time series analysis.

A. EXPERIMENTAL DESIGN

For evaluation, WMAN was generated based on the parameters of A which denote the area dimension, N_u which denotes the number of users, N_{BS} which denotes the number of base stations, N_C which denotes the number of cloudlets, base station data rate which is assumed to be the same, $\alpha_i = \alpha$ which denotes the packet data size of user i, λ_{max} which denotes the maximum arrival rate of tasks from users, B which denotes the bandwidth. We generated more than 300 scenarios involving various parameters related to base stations, cloudlets, users, wireless communications, and algorithms settings. The algorithms are iterated for ten iterations under a fixed rate every 10 seconds. The scenarios were generated based on the range definition for each parameter involved. We present the ranges with their respective parameters in Table 4. The scenario generation is based on the uniform random distribution depicted in Table 4. The values are generated using a uniform distribution that takes the values of the min and max. For the parameters that have the same min and max, then the value is selected as the same value of min and max. Our developed VL-WIDE will be compared with the fixed-length variant of whale optimization and differential evolution algorithm named (MGW) [26]. In addition, it will be compared with two non-dominated sorting genetic optimization algorithms, NSGA-II and NSGA-III, MOEAD, and PSO. In addition, we present the parameters used for the evaluation in Table 5. The common parameters are set for each of our developed VL-WIDE and the other benchmarks as equal.

B. EXPERIMENTAL RESULTS

This section presents the evaluation results and analysis. It is decomposed into two sub-sections, the multi-objective metrics and time series analysis.

1) MULTI-OBJECTIVE METRICS

The evaluation results are depicted in Figures 3, 4, 5, and 6. Boxplot visualization is adopted to show the statistical behaviour of the algorithms. First, we present the

TABLE 4. Definition of the parameters ranges that are used for the scenarios generation.

Parameter name	Min value	Max value	Unit
Area	100 × 100	100 × 100	km^2
N_u	100	200	Users
N_{BS}	20	20	Base station
N_C	5	10	Cloudlets
BSs data rate	$3 * 10^8$	$3 * 10^8$	Bits/second
BS wireless data rate	2^{10}	2^{13}	Bits/second
α	$40 * 2^{10}$	$40 * 2^{20}$	Bits
λ_{max}	40	45	tasks
B	$50 * 2^{20}$	$50 * 2^{20}$	Hz

TABLE 5. The parameters of the algorithms used for evaluation.

Parameter name	VL-WIDE	MGW	NSGA-II	NSGA-III	MOEAD	PSO
Iterations	100 - 200	100 - 200	100 - 200	100 - 200	100 – 200	100 - 200
Population Size	100 - 200	100 - 200	100 - 200	100 - 200	100 – 200	100 - 200
Mutation rate	0.5	0.5	0.5	0.5	0.5	-
Whale			-	-	-	-
l	-1 – 1	-1 – 1				
b	2.8 – 3.2	2.8 – 3.2				
DE Weighting Factor	0.8 – 0.9	0.8 – 0.9	-	-	-	-
Binary Crossover	-	-	eta=15, prob=0.9	eta=30, prob=1	eta=20, prob=1	-
Tournament Selection	-	-			-	-
Pressure			2	2		
n-Parents			1	1		
Random Portion Size	0.15 - 0.2	0.15 - 0.2	-	-	-	-
n_neighbors	-	-	-	-	20	-
w, c1, c2	-	-	-	-	-	0.9, 2, 2
Velocity rate	-	-	-	-	-	0.2

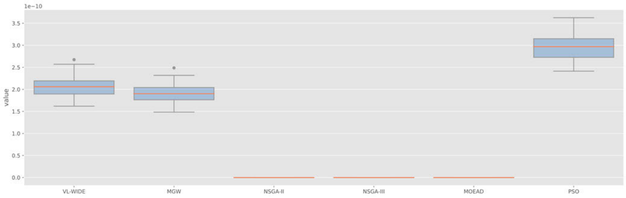

FIGURE 3. Hyper-volume of our developed VL-WIDE and other benchmarks.

hyper-volume, which provides optimization performance by providing a rich set of options to the decision-maker.

Hyper-volume is presented in Figure 3. As shown, our developed VL-WIDE has provided the highest hyper-volume in terms of all indicators of the boxplot, namely, median, min, max, Q1, and Q3 compared with MGW, NSGA-II, NSGA-III, and MOEAD while PSO has provided higher hyper-volume for its non-dominated solutions in the last generation.

TABLE 6. Descriptive statistics of hyper-volume measure for our developed VL-WIDE and its comparison with the benchmarks.

	VL-WIDE	MGW	NSGA-II	NSGA-III	MOEAD	PSO
Min	1.62E-10	1.48E-10	0	0	0	2.41E-10
Q1	1.90E-10	1.76E-10	0	0	0	2.73E-10
Median	2.06E-10	1.90E-10	0	0	0	2.97E-10
Q3	2.19E-10	2.04E-10	0	0	0	3.15E-10
Max	2.67E-10	2.49E-10	0	0	0	3.63E-10

Considering that solutions of PSO are inferior as we will observe in the set coverage, having a higher value of hyper-volume of PSO is still not an indicator of superiority. The fixed-length variant is the second one in terms of hyper-volume performance after VL-WIDE. The last observation is that the least performance in terms of hyper-volume is NSGA-II, NSGA-III, and MOEAD. It is observed that the hyper-volume value was very small for them compared with the whale optimization algorithm when integrated with differential evolution, namely, VL-WIDE and MGW. This provides that the swarming nature of the whale optimization and the mutation functionality of the differential evolution provide significant improvement in the searching capability for the optimization algorithm, which generated higher hyper-volume compared with the evolutionary type of algorithms such as NSGA-II and NSGA-III. Furthermore, the variable-length nature has contributed to an additional improvement in the performance of VL-WIDE.

For more elaboration on the quantitative performance of our developed VL-WIDE and its relationship with the benchmarks, we present the numerical values of the descriptive statistics for the MOO metrics of VL-WIDE. As shown in Table 6, VL-WIDE has accomplished the second highest values of descriptive statistics, namely, Min value or 1.62E-10, Q1 value or 1.90E-10, the value of Median or 2.06E-10., Q3 value and Max values with 2.19E-10 and 2.67E-10 respectively. This shows that VL-WIDE was superior in terms of the exploration and finding more diverse solutions in the objective space, which gives more degree of freedom to the decision maker. This is interpreted by the variable-length aspect of the VL-WIDE which has enabled searching more effectively in different dimensions to reach the optimal areas in the solution space.

The second metric used for evaluation is the delta metric, which assesses the performance in terms of how much the solutions are equally distributed in the solution space. As shown in Figure 4, it is observed that VL-WIDE has generated the least value in terms of delta metric compared with all other benchmarks. This provides that it was more capable of generating equally distributed solutions in the Pareto front. Consequently, VL-WIDE is more suitable for

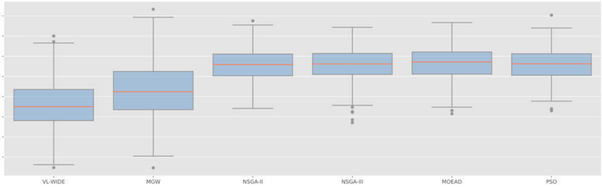

FIGURE 4. Delta Metric for our developed VL-WIDE and other benchmarks.

TABLE 7. Descriptive statistics of delta-metric measure for our developed VL-WIDE and its comparison with the benchmarks.

	VL-WIDE	MGW	NSGA-II	NSGA-III	MOEAD	PSO
Min	0.624	0.623	0.770	0.735	0.757	0.764
Q1	0.740	0.767	0.851	0.855	0.855	0.852
Median	0.774	0.812	0.879	0.880	0.885	0.881
Q3	0.817	0.862	0.905	0.906	0.910	0.906
Max	0.950	1.016	0.987	0.971	0.983	1.001

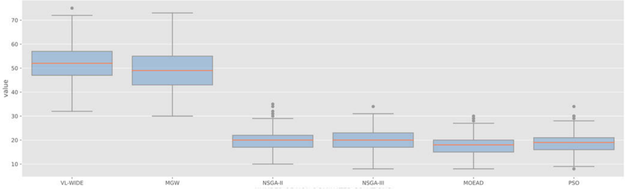

FIGURE 5. Number of Non-Dominated Solutions.

TABLE 8. Descriptive statistics of the number of non-dominated solutions measure for our developed VL-WIDE and its comparison with the benchmarks.

	VL-WIDE	MGW	NSGA-II	NSGA-III	MOEAD	PSO
Min	32	30	10	8	8	8
Q1	47	43	17	17	15	16
Median	52	49	20	20	18	19
Q3	57	55	22	23	20	21
Max	75	73	35	34	30	34

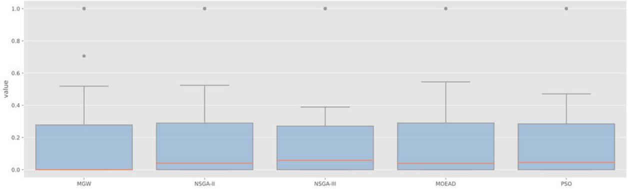

FIGURE 6. Set Coverage.

TABLE 9. Descriptive statistics of set-coverage measure for our developed VL-WIDE and its comparison with the benchmarks.

	MGW	NSGA-II	NSGA-III	MOEAD	PSO
Min	0	0	0	0	0
Q1	0	0	0	0	0
Median	0	0.040	0.057	0.038	0.044
Q3	0.277	0.289	0.270	0.289	0.284
Max	1	1	1	1	1

the decision-maker to move between solutions according to their non-dominated preference.

Similar to hyper-volume, we present the numerical values of descriptive statistics of delta metric in Table 7. It shows that VL-WIDE has accomplished the smallest values in terms of Q1, median, Q3, and maximum with values equal to 0.740, 0.774, 0.817, and 0.950 respectively. The only descriptive statistic that has generated a superiority of MGW is the minimum which was slightly lower than the VL-WIDE minimum with the value of 0.623.

The third metric that has been presented is the non-dominated solutions. The boxplot is presented in Figure 5. As observed, the highest value of a number of non-dominated solutions is accomplished by VL-WIDE in terms of median, MIN, Q1, and Q3. Furthermore, we find a similar value of MAX between each of VL-WIDE and MGW. On the other side, we find that other benchmarks, namely, NSGA2, NSGA3, MOEAD, and PSO were inferior in terms of a number of non-dominated solutions. This provides that the swarm behaviour with the integration of the local searching enabled by differential evolution was effective in generating a higher number of non-dominated solutions. Furthermore, the operators designed by VL-WIDE were effective in increasing the number of non-dominated solutions.

The numerical values of descriptive statistics of the number of non-dominated solutions are presented in Table 8. The results show that VL-WIDE has accomplished the highest values of all descriptive statistics. This is also an indicator of the strength of VL-WIDE compared with the benchmark algorithms. The number of non-dominated solutions for VL-WIDE has accomplished a value of a median equal to 52 which is higher than any of the other method's median values. Similar accomplished in witnessed for other descriptive statistics.

The last metric that is provided is the set coverage. Figure 6 shows the domination percentage of each method over the other methods. First, VL-WIDE has accomplished a domination over all other benchmarking algorithms. The domination percentage reached in the Q3 a level close to 0.3 over each MGW, NSGA-II, NSGA-III, MOEAD, and PSO. Consequently, VL-WIDE is superior in generating non-dominated solutions compared with the benchmark solutions.

In addition to the MOO metric, we present the numerical values of descriptive statistics in Table 9. The results show that VL-WIDE has accomplished maximum domination equal to 1 over all the algorithms which provide an outstanding superiority. This is justified by the fact the VL-WIDE is equipped with variable-length searching which enables exploring the solution space with different dimensions of solutions at the same time and providing sufficient interaction among them. In addition, VL-WIDE is provided with solutions repairing operator which enables preserving every generated solution inside the feasible region.

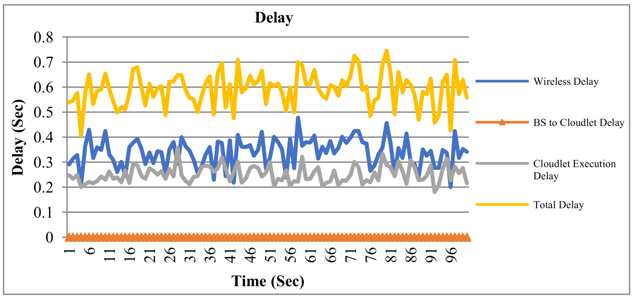

FIGURE 7. Delay of task serving with respect to time.

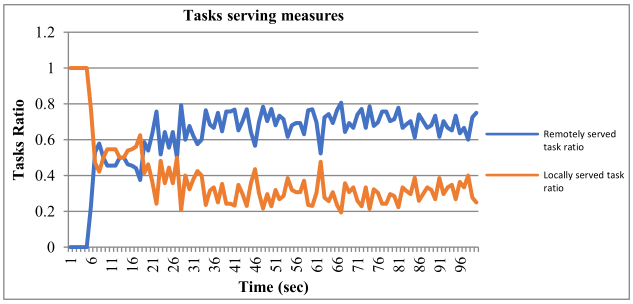

FIGURE 8. Tasks serving with respect to time.

The proposed VL-WIDE algorithm demonstrates significant positive impacts on cloudlet-based optimization tasks, offering improved solution quality, better adaptability to dynamic environments, enhanced feasibility, and more effective support for decision-makers. Its superior performance in various multi-objective optimization (MOO) evaluation metrics, including hyper-volume, delta metric, and the number of non-dominated solutions, highlights its ability to provide higher quality solutions compared to benchmark algorithms. Additionally, the variable-length search capability and the application-oriented solutions repair operator enable increased adaptability and ensure all generated solutions lie within the feasible region.

2) TIME SERIES ANALYSIS

For more elaboration, we present the time series analysis. In order to visualize the time series, we select one solution from the Pareto front, and we generate the corresponding metrics as time series. The first metric is the delay presented in Figure 7. The delay is decomposed into three components, wireless transmission, base station to cloudlet, and cloudlet execution. Hence, the total delay is the summation of the internal components. It is observed that the base station to cloudlet is the smallest delay because of the wired connection.

The second metric produced is the tasks serving measure presented in Figure 8. This metric is composed of two complementary components, namely, the remotely served task and the locally served task ratio. The summation of these two ratios is one.

The third metric that is visualized is energy consumption which is presented in Figure 9. This metric has two components, the consumed energy at the cloudlet and the consumed energy at the mobile device. The latter result from two components the execution at the mobile device and the transmission. It is observed that the significant component is the cloudlet consumed energy because the system serves the tasks.

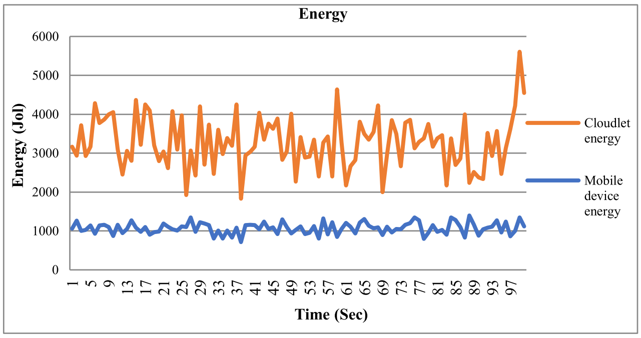

FIGURE 9. Energy consumption of task serving with respect to time.

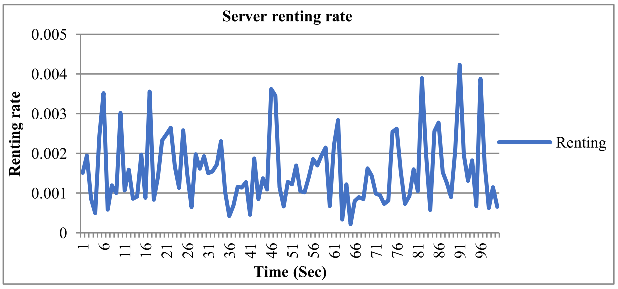

FIGURE 10. Server renting rate of task serving with respect to time.

The fourth metric is the server renting rate presented in Figure 10. It is observed that the server renting rate changes concerning time between 0.002 and 0.0043, resulting from the demand for task processing that is generated following the random process.

V. CONCLUSION AND FUTURE WORKS

This article has presented cloudlet-based computing optimization using developed Variable-length Whale Optimization and Differential Evolution. It presented the first algorithm that enables handling cloudlet-based optimization with an additional degree of freedom of moving the cloudlet according to the geographic demand. Second, the article has provided a novel formulation of the problem using variable-length for solution space. Third, the algorithm included application-oriented operators for solutions interaction. This has been accomplished based on a hybrid framework combined with multi-objective whale optimization and differential evolution. Performance analysis has been done using extensive evaluation using more than 300 scenarios generated randomly for the variable affecting parameters on the system. Furthermore, VL-WIDE compares with state-of-the-art algorithms, including fixed length hybrid whale optimization-differential evolution (MGW), NSGA-II, NSGA-III, MOEAD, and PSO with standard MOO performance metrics, namely, hyper-volume, delta-metrics, number of non-dominated solutions and set coverage. Compared to existing meta-heuristic algorithms, our algorithm can provide a non-dominated solution set with higher quality. In addition, it is concluded that the usage of variable-length searching for supporting multi-dimensional solutions space is useful in such types of combinatorial problems to reach more optimal regions. One of the limitations of our developed VL-WIDE is that it provides a Pareto front instead of a single solution which provides the need of using solution selection in order to operate in real-time as future work. Another future work is exploring the performance of reinforcement learning for solving the same problem and comparing it with our proposed algorithm. The difference between RL-based methods and a meta-heuristic searching-based method is that the former is more powerful in terms of knowledge representation and preservation. Hence, it is regarded as a strong future direction.

REFERENCES

[1] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Gener. Comput. Syst.*, vol. 97, pp. 219–235, Aug. 2019, doi: 10.1016/j.future.2019.02.050.

[2] Z. Wan and X. Dong, "Computation power maximization for mobile edge computing enabled dense network," *Comput. Netw.*, vol. 220, Jan. 2023, Art. no. 109458, doi: 10.1016/j.comnet.2022.109458.

[3] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 725–737, Oct. 2017, doi: 10.1109/TCC.2015.2449834.

[4] X. Zhao, C. Lin, and J. Zhang, "Cloudlet deployment for workflow applications in a mobile edge computing-wireless metropolitan area network," *Peer Peer Netw. Appl.*, vol. 15, no. 1, pp. 739–750, Jan. 2022, doi: 10.1007/s12083-021-01279-z.

[5] A. Mazloomi, H. Sami, J. Bentahar, H. Otrok, and A. Mourad, "Reinforcement learning framework for server placement and workload allocation in multiaccess edge computing," *IEEE Internet Things J.*, vol. 10, no. 2, pp. 1376–1390, Jan. 2023, doi: 10.1109/JIOT.2022.3205051.

[6] B. Li, P. Hou, H. Wu, and F. Hou, "Optimal edge server deployment and allocation strategy in 5G ultra-dense networking environments," *Pervas. Mobile Comput.*, vol. 72, Apr. 2021, Art. no. 101312, doi: 10.1016/j.pmcj.2020.101312.

[7] X. Jin, F. Gao, Z. Wang, and Y. Chen, "Optimal deployment of mobile cloudlets for mobile applications in edge computing," *J. Supercomput.*, vol. 78, no. 6, pp. 7888–7907, Apr. 2022, doi: 10.1007/s11227-021-04122-7.

[8] J. Zhang, Z. Ning, R. H. Ali, M. Waqas, S. Tu, and I. Ahmad, "A many-objective ensemble optimization algorithm for the edge cloud resource scheduling problem," *IEEE Trans. Mobile Comput.*, early access, Jan. 9, 2023, doi: 10.1109/TMC.2023.3235064.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[10] L. Yuan, J. Gu, J. Ma, H. Wen, and Z. Jin, "Optimal network partition and edge server placement for distributed state estimation," *J. Modern Power Syst. Clean Energy*, vol. 10, no. 6, pp. 1637–1647, 2022, doi: 10.35833/MPCE.2021.000512.

[11] J. Guo, M. Shao, S. Jiang, and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," in *Proc. Genetic Evol. Comput. Conf. Companion*, vol. 1, Jul. 2020, pp. 165–166, doi: 10.1109/TCYB.2015.2403131.

[12] M. Ryerkerk, R. Averill, K. Deb, and E. Goodman, *A Survey of Evolutionary Algorithms Using Metameric Representations*, vol. 20, no. 4. Cham, Switzerland: Springer, 2019.

[13] A. M. Jubair, R. Hassan, A. H. M. Aman, and H. Sallehudin, "Social class particle swarm optimization for variable-length wireless sensor network deployment," *Appl. Soft Comput.*, vol. 113, Dec. 2021, Art. no. 107926, doi: 10.1016/j.asoc.2021.107926.

[14] H. Li, K. Deb, and Q. Zhang, "Variable-length Pareto optimization via decomposition-based evolutionary multiobjective algorithm," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 987–999, Dec. 2019, doi: 10.1109/TEVC.2019.2898886.

[15] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 127, pp. 160–168, May 2019, doi: 10.1016/j.jpdc.2018.06.008.

[16] Q. Fan and N. Ansari, "On cost aware cloudlet placement for mobile edge computing," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 4, pp. 926–937, Jul. 2019, doi: 10.1109/JAS.2019.1911564.

[17] K. Peng, X. Qian, B. Zhao, K. Zhang, and Y. Liu, "A new cloudlet placement method based on affinity propagation for cyber-physical-social systems in wireless metropolitan area networks," *IEEE Access*, vol. 8, pp. 34313–34325, 2020, doi: 10.1109/ACCESS.2020.2974895.

[18] D. Bhatta and L. Mashayekhy, "A bifactor approximation algorithm for cloudlet placement in edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 8, pp. 1787–1798, Aug. 2022, doi: 10.1109/TPDS.2021.3126256.

[19] J. Zhang, M. Li, X. Zheng, and C. H. Hsu, "A time-driven cloudlet placement strategy for workflow applications in wireless metropolitan area networks," *Sensors*, vol. 22, no. 9, pp. 1–19, 2022, doi: 10.3390/s22093422.

[20] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12240–12251, Aug. 2020, doi: 10.1109/TVT.2020.3018817.

[21] A. Bozorgchenani, F. Mashhadi, D. Tarchi, and S. A. Salinas Monroy, "Multi-objective computation sharing in energy and delay constrained mobile edge computing environments," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 2992–3005, Oct. 2021, doi: 10.1109/TMC.2020.2994232.

[22] M. Huang, Q. Zhai, Y. Chen, S. Feng, and F. Shu, "Multi-objective whale optimization algorithm for computation offloading optimization in mobile edge computing," *Sensors*, vol. 21, no. 8, p. 2628, Apr. 2021, doi: 10.3390/s21082628.

[23] A. Rago, G. Piro, G. Boggia, and P. Dini, "Anticipatory allocation of communication and computational resources at the edge using spatio-temporal dynamics of mobile users," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4548–4562, Dec. 2021, doi: 10.1109/TNSM.2021.3099472.

[24] J. Yang, Q. Yuan, S. Chen, H. He, X. Jiang, and X. Tan, "Cooperative task offloading for mobile edge computing based on multi-agent deep reinforcement learning," *IEEE Trans. Netw. Service Manage.*, early access, Feb. 3, 2023, doi: 10.1109/TNSM.2023.3240415.

[25] S. Yang, F. Li, M. Shen, X. Chen, and X. Fu, "Cloudlet placement and task allocation in mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5853–5863, Jun. 2019, doi: 10.1109/JIOT.2019.2907605.

[26] X. Zhu and M. Zhou, "Multiobjective optimized cloudlet deployment and task offloading for mobile-edge computing," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15582–15595, Oct. 2021, doi: 10.1109/JIOT.2021.3073113.

[27] X. Zhang, Z. Li, C. Lai, and J. Zhang, "Joint edge server placement and service placement in mobile-edge computing," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11261–11274, Jul. 2022, doi: 10.1109/JIOT.2021.3125957.

[28] M. L. Ryerkerk, R. C. Averill, K. Deb, and E. D. Goodman, "Solving metameric variable-length optimization problems using genetic algorithms," *Genetic Program. Evolvable Mach.*, vol. 18, no. 2, pp. 247–277, Jun. 2017, doi: 10.1007/s10710-016-9282-8.

[29] M. Ryerkerk, R. Averill, K. Deb, and E. Goodman, "A novel selection mechanism for evolutionary algorithms with metameric variable-length representations," *Soft Comput.*, vol. 24, no. 21, pp. 16439–16452, Nov. 2020, doi: 10.1007/s00500-020-04953-1.

[30] H. ZainEldin, M. Badawy, M. Elhosseini, H. Arafat, and A. Abraham, "An improved dynamic deployment technique based-on genetic algorithm (IDDT-GA) for maximizing coverage in wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 4177–4194, Jan. 2020, doi: 10.1007/s12652-020-01698-5.

[31] V. P. Ha, T. K. Dao, N. Y. Pham, and M. H. Le, "A variable-length chromosome genetic algorithm for time-based sensor network schedule optimization," *Sensors*, vol. 21, no. 12, pp. 1–25, 2021, doi: 10.3390/s21123990.

[32] L. Cruz-Piris, I. Marsa-Maestre, and M. A. Lopez-Carmona, "A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem," *IEEE Access*, vol. 7, pp. 111968–111981, 2019, doi: 10.1109/ACCESS.2019.2935041.

[33] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8, doi: 10.1109/CEC.2018.8477735.

[34] A. Mohammadi, S. H. Zahiri, S. M. Razavi, and P. N. Suganthan, "Design and modeling of adaptive IIR filtering systems using a weighted sum–variable length particle swarm optimization," *Appl. Soft Comput.*, vol. 109, Sep. 2021, Art. no. 107529, doi: 10.1016/j.asoc.2021.107529.

[35] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: Whale optimization algorithm and its applications," *Swarm Evol. Comput.*, vol. 48, pp. 1–24, Aug. 2019, doi: 10.1016/j.swevo.2019.03.004.

[36] N. Noman and H. Iba, "Differential evolution for economic load dispatch problems," *Electr. Power Syst. Res.*, vol. 78, no. 8, pp. 1322–1331, Aug. 2008, doi: 10.1016/j.epsr.2007.11.007.

[37] M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, "Differential evolution: A review of more than two decades of research," *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103479, doi: 10.1016/j.engappai.2020.103479.

[38] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006, doi: 10.1109/TEVC.2005.851275.

[39] G. Valkanas, A. N. Papadopoulos, and D. Gunopulos, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proc. CEUR Workshop*, vol. 1133, 2000, pp. 182–187, doi: 10.1007/3-540-45356-3_83.

[40] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999, doi: 10.1109/4235.797969.

LAYTH MUWAFAQ received the B.Eng. degree from Al-Rafidain University College, Iraq, in 2007, and the M.Eng. degree in communication and computer engineering from Universiti Kebangsaan Malaysia, Malaysia, in 2016. He is currently pursuing the Ph.D. degree with Universiti Putra Malaysia. His research interests include mobile cloud computing and mobile edge computing.

NOR K. NOORDIN (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from The University of Alabama, Tuscaloosa, AL, USA, in 1987, the M.Eng. degree from Universiti Teknologi Malaysia, and the Ph.D. degree from Universiti Putra Malaysia. She is currently a Professor with the Department of Computer and Communication System Engineering, Universiti Putra Malaysia. She has published more than 300 journals, book chapters, and conference papers. She has led many research projects. Her research interests include wireless communications and network systems.

MOHAMED OTHMAN (Senior Member, IEEE) received the Ph.D. degree (Hons.) from the National University of Malaysia. He was the Deputy Director of the Information Development and Communication Centre, where he was in charge of the UMPNet Network Campus, uSport Wireless Communication Project, and the UPM Data Centre. He is currently a Professor in computer science with the Department of Communication Technology and Networks, Universiti Putra Malaysia, where he is also an Associate Researcher and a Coordinator in high-speed machines with the Laboratory of Computational Science and Mathematical Physics, Institute for Mathematical Science Research. He has led six Malaysian, one Japanese, one South Korean, and three U.S. patents. He has published more than 300 international scientific journal articles. His research interests include computer networks, parallel and distributed computing, high-speed interconnection networks, network design and management (network security, and wireless and traffic monitoring), consensus in the IoT, and mathematical models in scientific computing.

ALYANI ISMAIL (Member, IEEE) received the B.Eng. degree (Hons.) in electronic and information engineering from the University of Huddersfield, U.K., in 2000, and the M.Sc. degree in communication and computer and human-centered systems engineering (communication) and the Ph.D. degree in electronics engineering from the University of Birmingham, U.K., in 2002 and 2006, respectively. Her Ph.D. thesis was "Design of Microwave Waveguides and Filters for Micromachining." She is currently a Lecturer with the Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia, Malaysia. She is a member of the International Association of Engineers.

FAZIRULHISYAM HASHIM (Member, IEEE) received the M.Sc. degree from Universiti Sains Malaysia and the Ph.D. degree in wireless communication and network engineering from The University of Sydney, Australia. He is currently an Associate Professor with the Department of Computer and Communication Systems Engineering, Universiti Putra Malaysia. His research interests include heterogeneous and future wireless communication systems, and network security.