

A Survey on Cloudlet Computation Optimization in the Mobile Edge Computing Environment

Layth Muwafaq¹, Nor K. Noordin², Mohamed Othman³, Alyani Ismail⁴, Fazirulhisyam Hashim⁵

Department of Computer and Communication Systems Engineering-Faculty of Engineering,
Universiti Putra Malaysia, Serdang 43400 Malaysia^{1, 2, 4, 5}

Research Centre of Excellence for Wireless and Photonics Network (WiPNET)-Faculty of Engineering,
Universiti Putra Malaysia, Serdang 43400 Malaysia^{2, 5}

Department of Communication Technology and Networks, Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia³
Laboratory of Computational Sciences and Mathematical Physics-Institute for Mathematical Research (INSPEM), Universiti
Putra Malaysia (UPM), Serdang 43400, Malaysia³

Abstract—Mobile Edge Computing (MEC) uses to perform computation operations at the edge of a network for mobile devices. This allows the deployment of more powerful and efficient computing resources in a cost-effective, lightweight and scalable manner. MEC can optimize mobile device performance, enhance security and privacy, improve battery life, provide increased bandwidth, and reduce latency across wireless networks. Cloudlets are a new concept of computations that can perform at the edge of the networks. The service provider can deploy cloudlets services in a MEC environment with the ability for mobile devices to offload their tasks to cloudlets. In the MEC environment, the offloading problem depends on cloudlets' availability of computation resources. Also, the deployment method of cloudlets in the environment will affect the task offloading. This paper investigates the approach to the cloudlet deployment and task offloading problem in the MEC environment. First demonstrate that the problem has to be considered a Multi-objective optimization problem since it needs more than one objective to be optimized. Then prove that the problem is NP-completeness, give an overview of existing solutions using the meta-heuristic algorithms, and suggest future solutions for this problem. Finally, explain the advantages of using Variable-length of solution space with meta-heuristic algorithms for this problem.

Keywords—Mobile edge computing, cloudlet deployment; task offloading; mobile device; multi-objective optimization; meta-heuristics; variable-length

I. INTRODUCTION

Mobile Edge Computing (MEC) is a technology that enables computation and storage capabilities at the edge of a mobile network, closer to the end-users [1]. It extends cloud computing to the network's edge, allowing for delivering low-latency and high-bandwidth services to mobile users [2]. The end-users typically execute the applications on their resource-constrained mobile devices for current internet-enabled applications requiring fast processing and less response time [3]. MEC is designed to address the challenges of providing low-latency and high-bandwidth services to mobile users, such as those required for augmented reality, virtual reality, and Internet of Things (IoT) applications [4]. Furthermore, the mobile devices at the edge of the network offload their computation tasks to the (edge-server, edge-cloud, sometimes

referred to as Cloudlet) instead of the remote cloud, which will decrease the response time for offloaded tasks (low latency) and reduce overcrowding in the back-haul networks [5].

Cloudlet is a new computing paradigm introduced to the Mobile Edge Computing (MEC) service framework. It allows computing resources to be closer to mobile devices. The cloudlets location is essential to the delay tolerance of mobile devices, primarily in a large-scale Wireless Metropolitan Area Network (WMAN) that consists of hundred Base Stations (BSs) [6], where mobile devices can access the cloudlets. The capacity of cloudlet is much smaller than cloud computing as edge computing is supplied with one or a few servers due to the limitation of space and cooling requirements [7]. MEC has been adopted to allow mobile devices to offload their tasks to the cloudlets because of the limited processing, small storage and low computational capabilities of mobile devices [8]. The high quality of service (QoS) requirements for the highly interactive applications, which include low latency and high throughput, are computationally demanding [9]. Mobile computing has recently experienced a paradigm change from mobile central cloud computing to MEC, fueled by hopes for 5G and 6G IoT connectivity [10]. In order to support latency-intensive computing applications and critical latency for mobile devices with limited resources, MEC's main purpose is to push mobile computing, network control and storage to the network's edges [11]. In the MEC environment, cloudlets are usually deployed collocated with cellular base stations so that the mobile devices can offload tasks to the nearby cloudlets.

Offloading is a technique used in the MEC environment to increase the effectiveness of mobile device applications by moving resource-intensive activities to nearby cloudlets [12]. Offloading in MEC mostly refers to running resource-intensive applications on behalf of local mobile devices to minimize workloads, overhead, and processing costs compared to local computing. To perform compute offloading, mobile devices and cloudlets must operate offloading frameworks [13]. Many technical articles view the topic as incredibly important to provide new ways of reaching the objectives in the offloading criteria. Most of the strategies presented in these technical publications were based on mathematics, model-based, machine learning, game theory,

heuristic-based, meta-heuristic, or a combination of the abovementioned techniques. In the MEC context, computation offloading problems are a very difficult challenge [14]. The primary drawback of offloading work to a remote cloud is the latency, which disrupts user experiences in interactive applications like mobile gaming [15]. Cloudlets get around this problem by giving users low-latency access to network-edge computing resources, which significantly boosts the efficiency of mobile applications [16]. The primary issue with WMAN is the deployment of fewer cloudlets with good services to end user. From the perspective of network management, it is costly to place a cloudlet at each BS to service end users [17].

Multi-objective optimization techniques are an excellent approach in this situation. In multi-objective optimization, as opposed to single-objective optimization, the search is for a collection of non-dominated solutions known as the Pareto optimal set rather than a single optimal solution, which must be optimized [18]. The non-dominated objective solutions are the ones that provide the best potential compromises between the many objectives of the problem (i.e., these solutions cannot enhance one objective without affecting another). The decision-makers, in this case, the service providers, are given access to such non-dominated solutions so that they may choose the one that caters to their specific demands and requirements in the most effective manner. The computational methods that are currently available to solve multi-objective optimization problems include meta-heuristics and high-level strategies governing underlying techniques. Additionally, in contrast to the conventional mathematical programming approaches used to solve multi-objective optimization problems, meta-heuristics can create many members of the Pareto optimal set in a single iteration [19]. Most researchers consider the problem as a Single Objective Optimization in the related works, while a few deal with it as a Multi-Objective Optimization problem.

A meta-heuristic optimization has shown its effectiveness in tackling several NP-hard problems. Computer science, networking, communication, robotics, and manufacturing, are just a few areas where this mechanism has been used in the real world. The literature on meta-heuristic optimization has yet to adequately address the multi-objective optimization component, despite its prevalence in many problems. The meta-heuristic algorithms have a substantial impact on solving cloudlet deployment and task offloading optimization issues. Consequently, this paper provides a survey of published articles about cloudlet deployment and task offloading to summarize the problems that need solving for future research. First discuss the challenges addressed partially or entirely for the problem and suggest several promising directions for future research to minimize time, reduce cost, and save energy. Second, demonstrate that the problem is multi-objective optimization since it deals with more than one objective to be optimized and proves it is an NP-completeness. Third, explain the diverse current, well-known traditional heuristic and meta-heuristic algorithms for cloudlet

deployment and task offloading problems in the MEC environment, emphasizing meta-heuristics. Fourth present the capabilities of using the variable-length searching approach within meta-heuristic optimization algorithms for solving this type of problem, especially with the constant change in the required number of cloudlets to deploy in MEC based on computing requirements. This work can be helpful for academia and companies regarding service provisioning in the mobile edge computing environment.

The rest of this paper is organized as follows: Section II presents a background and an overview of the relevant literature, which helps to understand the rest of the article. Sections III introduce the optimization technique. Section IV demonstrates the multi-objective optimization and the meta-heuristics algorithms. Section V Provides the Variable-length approach for solution space and its benefits for future research, and finally, the conclusion are provided in Section VI.

II. BACKGROUND

This section presents an overview of the relevant works, which assists in understanding the rest of the article. Cloudlet deployment and task-offloading (for simple CDTO) have been fertile research ground for several years. Thus, many papers on solutions related to this area are in the literature.

A. Mobile Edge Computing

One of the essential fields on the technology scene today is Cloud Computing, which is one of the elements preparing for the future. Thanks to advancements in wireless communication and mobile computing, in the last decade, new smartphone device services have prospered in diverse fields such as transportation, mobile payment, and social media [20]. The spread of mobile devices and their constant presence in daily life has generated massive traffic between end users and remote clouds [21]. To plan for the increasing data traffic in the following years and the need for low-latency computation resources near the users, network service providers are gradually turning to "Mobile Edge Computing" to bring cloud computing capabilities to the edge of the network [22]. Users at the edge of the network offload their computation tasks to the (edge-server, edge-cloud, sometimes referred to as Cloudlet) instead of remote clouds, which will decrease the response time for offloaded tasks (low latency) and reduce overcrowding in the back-haul network [20].

Due to the limitations of the cloud environment (i.e., long-distance from mobile devices and limited geographical distribution), mobile devices with hungry applications for computation resources need to offload tasks to the nearby clouds closer to the end-devices through newer computing paradigms, such as cloudlets in Mobile Edge Computing environment (MECE) [23]. Cloudlets are a distributed decentralized infrastructure with nearby mobile devices that can leverage the computing and storage resources of the cloudlets. Table I presents the comparative summary of Cloud and Cloudlet.

TABLE I. COMPARISON BETWEEN CLOUD AND CLOUDLET

Feature	Cloud	Cloudlet
Cost	High	Low
End-to-end latency	High	Low
Infrastructure	Centralized	Decentralized
Deployment Environment	Large Data Center	Can be deployed anywhere
Bandwidth for end-user	Low	High
Offline Availability	Not available	Available
Computing Power	High	Low
Resource Elasticity	High	High
Availability	High	High
Access to resources	through core network	typically via 1-hop wireless gateway
Resources at individual locations	Many	Few
Geo-distribution of computing resources	locally clustered	widespread
Offloading granularity	mostly entire applications	computationally intensive and latency-critical parts
User Experience	Satisfactory QoE	Excellent QoE
Network resource sharing at individual locations	Large number of users	Limited number of users

B. Wireless Metropolitan Area Networks

The wireless metropolitan area networks (WMANs) have emerged in recent years as a public network that enables mobile devices to easily access the abundant computing resources in urban cities as the number of mobile devices continues to grow [24]. In addition, wireless broadband connectivity requirements are growing (i.e., 5G and 6G), and the technology continues to evolve to meet these requirements [25]. It is currently evolving as a public network to expand mobile device performance, covering the metropolis and enabling mobile devices to access abundant computing resources [26]. Compared to Wireless Local Area Networks (WLANs), Wireless Metropolitan Area Networks (WMANs) offer wireless communications over significantly broader geographic regions. The problem of CDTO gets significantly worse when considering the deployment of cloudlets in WMANs because of the size of the WMANs [27]. Due to the high population density in urban regions, many users will have access to cloudlets. As a result, cloudlets will be more cost-effective since they will be less likely to stay idle. Furthermore, due to the population in WMANs, service providers can take advantage of economies by deploying a small number of cloudlets to provide services through the WMANs [20] and making cloudlets accessible to the general public.

C. Cloudlet-deployment and Task-offloading (CDTO)

Cloudlets deployment is the first step in implementing mobile edge computing. Service providers always hope to provide services to as many users as possible under limited funds. The idea of treating cloudlets as a separate “data center in a box” must be scrapped. There are obvious benefits of connecting several cloudlets to form a network of cloudlets that can be distributed in the wireless metropolitan area networks, collocated with the base stations [20]. Cloudlet can be a single server or a cluster of servers, collocated with base stations, and mobile devices can access cloudlets close to the

range [28]. Cloudlets can also be considered as offloading mobile device destinations, aiming to save energy [29] and reduce the latency between the remote clouds and mobile devices [30]. The close physical proximity between cloudlets and mobile devices is a significant benefit of cloudlets over the remote cloud. So that allows for low latency in communication, thus improving the QoS and the user experience of interactive applications [31]. To overcome the restrictions of mobile devices in the offloading process technique, mobile devices offload their tasks to the MECE [32]. Task-offloading improves the computing efficiency of mobile devices while reducing latency and overall spent energy if adequately designed and planned [5]. There are four types of delays in task-offloading from mobile devices to cloudlets: uploading, queuing, execution, and downloading the result. A typical process flow of task-offloading in WMAN is illustrated in Fig. 1.

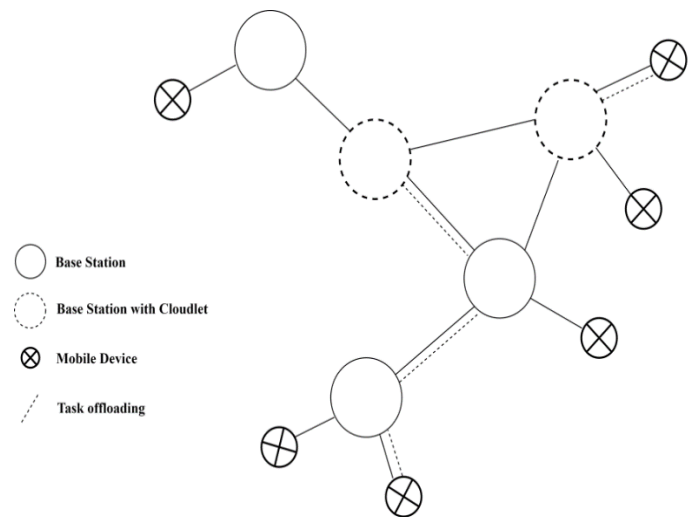


Fig. 1. Process flow of the task-offloading in WMAN.

In related works, there are mainly three metrics expressing system requirements that can be optimized, the task response time [33], the cost of service for service provider and user side [34], and the energy consumption (the energy consumed by the device while task-offloading and the power needed by cloudlet to process tasks) [35]. In addition, some issues have been partially addressed in previous works, which are reliability, load balancing, cache content, task migration, user mobility, and cloudlet mobility. The exciting issues for related works that dealt with the problem as a single objective optimization are shown in Table II.

1) Response Time

The QoS metric in the design of offloading methods is the response time, which is the time between offloading the task from a mobile device to a cloudlet, processing it, and receiving the result back [36]. Furthermore, the response time is also defined as the delay [37], or latency [38] in some related works. Many aspects influence the response time of a task as followings:

- The communication delay between the mobile device and the base station while offloading the task to the cloudlet.
- The communication delay between the base station and the attached cloudlet, which is always minimal while using fiber optics cables.
- The communication delay while routing the offloaded task from overloaded to under-loaded cloudlet.
- The queuing time before processing in the cloudlet.
- The processing time on the cloudlet.
- The communication delay between the base station and mobile device while receiving the result.

Fig. 2 shows the task lifetime, from the mobile device to cloudlet, which is an uplink delay, queuing and execution at cloudlet, which is a processing delay, and from cloudlet to mobile device, which is a downlink delay.

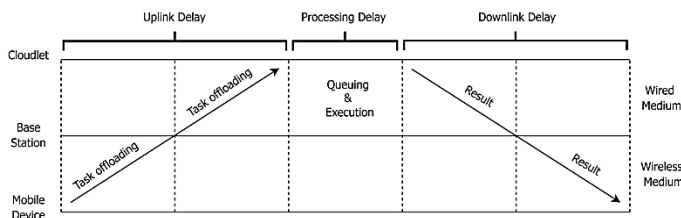


Fig. 2. Task service delay.

2) Cost

The cost can be categorized into two sides: service providers and users. For the service provider, the cost of deploying cloudlets is related to two factors [21], which are site rentals and computations needs. The former means that the more locations are selected to deploy cloudlets, the higher the cost. The latter aspect tells us that the greater the computation needs, the greater the number of cloudlets, resulting in higher costs. On the other hand, users look for the best service at a reasonable price. The service price (the

amount user have to pay) mainly specifies the willingness of users to pay for services [39].

3) Energy

Most existing works on MECE mainly focus on saving energy for mobile devices by offloading tasks to the cloudlets. Due to the battery life limitation for various mobile devices, the energy consumed by the mobile device processing the tasks is one of the most significant issues [40]. The energy is mainly consumed by mobile device data processing and transmission. Due to the limitation of processing in a mobile device [39], it is better to offload tasks from mobile devices to nearby cloudlets for energy-saving. The offloading of a task can decrease the processing energy by reducing the processed data size but increases the transmission energy by increasing the size of transmitted data [35]. On the other hand, service providers are looking for energy savings for cloudlets, reducing power costs.

4) Reliability

Reliability is a significant component that influences user QoS and service provider profit [41]. The significant causes of reliability decreases are failures or errors in the mobile edge computing environments. Failures and errors in the system can occur during task-offloading, processing, routing tasks from one cloudlet to the other, and receiving the result back [42]. Improving the reliability of task-offloading under the resource capabilities of the cloudlets and the dynamic network is an important challenge in the MECE.

5) Load Balancing

The cloudlets contain single or multiple servers to process the offloaded tasks. The load inequality in these cloudlets leads to different response times for offloaded tasks. Therefore, the load balancing between cloudlets for task-offloading becomes essential [22]. Increasing the number of servers in the cloudlet makes balancing user workload less critical as the cloudlets can handle higher loads [43]. On the other hand, increasing the number of users will increase the task waiting time. Once the cloudlet reaches its maximum workload, it must route its overflow users' requests to the other cloudlets for better performance [44]. This means that a better strategy is to assign user tasks to under-loaded cloudlets rather than keeping them in an overloaded cloudlet [22]. Hence, it is clear that balancing the workload between cloudlets is necessary by mapping the tasks of users to different cloudlets.

6) Cache Content

When a task is offloaded to a cloudlet, the critical impact on task execution performance is data transmission delay. One of the most successful solutions is to cache data for the offloaded task in advance. Furthermore, because user data access patterns in WMANs are challenging to forecast [45], creating an adequate data caching technique with high access hit ratio is challenging. The number of content cached at cloudlet increases with the number of offloaded tasks increasing [46]. Content cache performance significantly impacts delay and bandwidth [47]. For example, the popular cached content in cloudlet can reduce the delay and the bandwidth usage of subsequent access to the same content.

7) Task Migration

In the MECE, the inter-flow between cloudlets is essential to make a load-balancing and support user mobility. In the network of cloudlets, to make load-balancing, tasks have to be forwarded from the overloaded cloudlet to the under-loaded cloudlet, so there will be no idle cloudlet in the system [48]. Hence, as shown in Fig. 3, to support user mobility in the system, tasks must route from one cloudlet to another cloudlet so that the task will be close to the mobile user for result download [49].

8) User Mobility

To support user mobility, edge computing requires the flexible and scalable deployment of cloudlets for the inherent dynamism of the operating environment and various applications, some of which require real-time response [50]. In related works, WMAN's user mobility pattern has attracted very little attention. A mobile user may move around the area covered by WMAN with specific transmission patterns, but different movement patterns may change the traffic size of the network. The cloudlet placement and task offloading strategy should reflect the mobility pattern better to meet the requirements of mobile devices to access network resources [51]. The mobility pattern is reflected by the connection possibilities between mobile devices and BS. Furthermore, the mobility pattern of mobile devices is highly correlated with service access delay. Connection failures may happen during user mobility with the poor quality of the wireless connection between mobile devices and BSs [52]. However, the next generation of cellular communication has overcome this issue. The excellent service would necessitate frequent wireless handover between multiple BSs to guarantee user QoS during

movement. For constant QoS, mobile devices may be serviced by many cloudlets. Because of user mobility, wireless handover and service migration may be performed frequently, putting a heavy load on the other network entities [53].

To support user mobility, task migrations have to perform when a mobile user moves from the service region of one cloudlet to another. As mentioned four types of task service delay; there will also be a delay in task migration during user movement. Fig. 3 illustrates the task migration during user movement. To provide QoS for users, mobility management should make a wireless handover decision to select optimum BSs and cloudlets [54]. The primary goal of mobility management is to provide mobile computing services to continuous and uninterrupted customers during user movements.

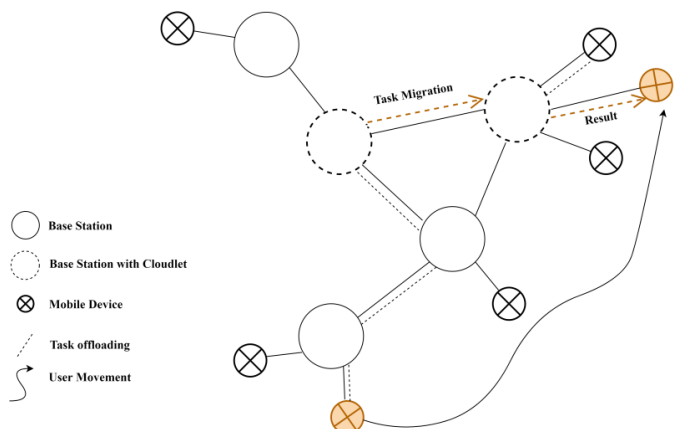


Fig. 3. Task migration while user movement.

TABLE II. RELATED WORKS CONSIDERING A SINGLE OBJECTIVE OPTIMIZATION

Authors	Cloudlet Deployment	Task offloading	Time	Cost	Energy	Reliability	Load balancing	Cache content	Task Migration	User Mobility	Cloudlet Mobility
[20]	√	√	√	×	×	×	√	×	×	×	×
[55]	√	×	√	×	×	×	×	×	×	×	×
[37]	√	×	√	×	×	√	√	×	×	×	×
[56]	×	√	√	×	√	×	√	×	×	×	×
[31]	√	×	√	√	×	×	×	×	×	×	×
[49]	×	√	√	√	×	×	×	×	√	√	×
[7]	√	×	√	√	×	×	×	×	√	√	×
[57]	√	×	√	×	×	×	×	×	√	√	×
[58]	√	×	√	√	√	×	×	×	×	×	×
[59]	√	×	√	×	√	×	×	×	×	×	×
[60]	√	√	√	×	√	×	×	×	×	×	×
[41]	√	×	√	√	×	√	×	×	×	×	×
[22]	×	√	√	×	×	×	√	√	√	×	×
[61]	√	×	√	×	×	×	√	×	×	√	√
[62]	×	√	√	×	√	×	×	√	×	×	×
[63]	×	√	√	√	√	×	×	×	×	×	×
[42]	×	√	√	×	×	√	×	×	×	×	×
[44]	√	√	√	×	×	×	√	×	×	×	×
[64]	√	√	√	√	×	×	×	×	×	×	×
[19]	√	√	√	×	×	×	×	×	×	×	×
[65]	√	×	√	√	×	×	×	×	×	×	×
[66]	√	×	√	√	√	×	×	×	×	×	×

9) Cloudlet Mobility

The cloudlet in the MECE can be static or mobile [12]. In mobile cloudlet networks, it is crucial to figure out how to make load-balancing between all mobile cloudlets so that all resources are utilized and tasks can be processed concurrently and sustainably by different cloudlets, therefore, reducing the average task response time and energy [61]. Cloudlet mobility is essential in MECE to provide good services for users, especially when the population moves (e.g. festivals, sporting events, etc.).

In Table II, all researchers consider the problem as a single-objective optimization. Most of them take the problem on one side (cloudlet-deployment or task-offloading), while very few consider both themes. Furthermore, most existing works mainly focus on minimizing time, reducing cost, saving energy, and load balancing, while the others take little attention. In Table II, various algorithms and methods were employed to address the issues, including greedy algorithms, clustering algorithms, heuristics, meta-heuristics, reinforcement learning, and mathematical programming. Comparisons were made between meta-heuristic and heuristic algorithms, meta-heuristic and greedy algorithms, reinforcement learning and mathematical programming. In addition, research studies such as [44] have compared meta-heuristic and unsupervised learning algorithms, while [65] has compared meta-heuristic and mathematical programming, with results consistently demonstrating the superiority of meta-heuristic algorithms.

D. Related Survey Works

CDTO has been a fertile study area in recent years. As a result, many surveys and studies in the literature on solutions related to this area. In the work of [67], the problems of mobile cloud computing are presented and provide the most recent mobile cloudlet architecture. Furthermore, propose a hierarchical taxonomy to classify the most recent cloudlet solutions and discover the cloudlet application areas. In addition, it presents the aspects of cloudlet management, like cloudlet discovery, resource management, data security, mobility, and application offloading. The work of [68], studied the fundamental concepts of cloud and edge computing. Presented the application domain classified the state-of-the-art edge computing (Mobile Edge Computing, Cloudlet, and Fog) and the application domain area services such as real-time applications, resource management, data analytics, and security. Furthermore, state the essential requirements that must be achieved for edge computing to be enabled. The work of [69] proposes a taxonomy of task-offloading in edge-cloud environments to examine and categorize related research papers and outline the challenges that still need to be studied before using edge-cloud computing to improve services. To identify the modern processes, the work of [70] provides a review of the machine learning-based computation offloading strategies in the MECE in classical taxonomy. Furthermore, it investigated various strategies and novel approaches related to machine learning-based offloading mechanisms in the MECE ecosystem. Also, taxonomy for classifying various principles of machine learning-based offloading mechanisms was proposed.

Finally, there are many other surveys, each considering a particular aspect of Mobile Edge Computing, which also lacks considering the Multi-objective optimization approach for the problem of CDTO.

III. OPTIMIZATION

Optimization is a method of finding and comparing appropriate solutions until cannot find any better solutions. Optimization often involves minimizing or maximizing the objective functions. Optimization, in other terms, refers to a collection of methods that may be applied to a mathematical model of the problem. The optimization algorithm provides systematic and efficient methods for producing and comparing new solutions to achieve the optimal solution [71]. Optimization is experimenting with various input-output combinations to find the resulting outputs. Its methods are far more advanced than those used in the computation. Optimization theory has custom-made algorithms to identify the best solution with little processing by utilizing model information. Optimum selection evaluations, suitable trade-offs, and non-intuitive analysis of optimization techniques are utilized to develop a better and faster optimal design [72]. The role of modeling in optimization involves testing out many different combinations of inputs to determine what the output (i.e., the number you are trying to minimize or maximize) will be under each of those circumstances, with many inputs, it is usually not practical to guess and check on an existing system [73], optimization requires a mathematical model of the system, which is just the math that relates the inputs to the outputs. By using a system model to iterate, there is no limit to the number of combinations you can try (maybe just computational limitations). Optimization algorithms are much more sophisticated than guessing and checking a range of varieties (a process known as enumeration) [74]. Fig. 4 outlines the steps usually involved in an optimal design formulation.

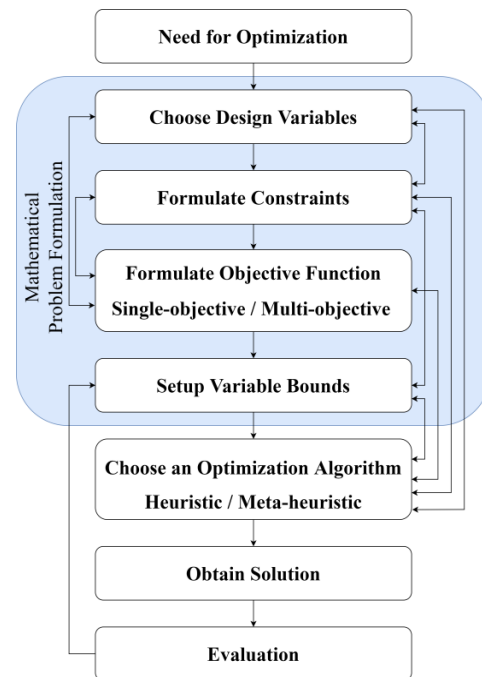


Fig. 4. Steps of problem optimization.

In the Mobile Edge Computing environment, users and the service provider have various requirements. The deployment of a small number of cloudlets in the WMANs to serve many users is significant for service providers. In addition, user satisfaction is one of the most important quality-of-experience (QoE) evaluation measures, especially affecting the service provider's profit due to the penalty when there is any infringement of the Service Level Agreement (SLA), which establishes an agreement between service providers and their users [75]. It significantly influences users' willingness to pay for services in the future. Thus, when offering services, user requirements must be considered and satisfied [40]. Furthermore, in the problem of CDTO, most existing works consider single-objective optimization, while very few consider it a Multi-objective optimization. The relationship "better than" between two solutions is insignificant when dealing with single-objective optimization problems. Whether a function has to be maximized or minimized, the one with the lower or higher fitness value is preferred. The relationship "better than" has to be redefined while dealing with the Multi-objective optimizations. Furthermore, when two non-dominated solutions are compared, there is only one way to rate the best if enforcing specific user preferences. As a result, a new kind of fitness measurement is required [74].

IV. MULTI-OBJECTIVE OPTIMIZATION

Problems involve more than one objective in the real world, and conflict naturally leads to trade-off solutions. It is hard for the mobile edge computing service provider to determine the constraints. A feasible solution cannot be found if the constraints are not appropriate. Only one objective is confirmed, and only one solution may be generated if a constrained single-objective optimization problem is considered rather than a multi-objective optimization problem which may not result in a satisfactory trade-off between all objectives [76]. Most researchers in related works dealt with the problem as a single-objective optimization. In contrast, few researchers are dealt with the problem as multi-objective optimization, as shown in Table III.

For the Multi-objective optimization problem arising from Cloudlet-deployment and Task-offloading (for simple MO-CDTO) in MECE, not only the minimization of task completion time and the energy consumption is required, but also the minimization of cloudlet-deployment cost is demanded [60]. In Table III, researchers consider the problem

as multi-objective optimization. Most of them take the problem on one side, which is task offloading. Furthermore, time, cost, energy, and task migration have more extensive attention than others. The algorithms and methods were employed to address the multi-objective optimization problems, including meta-heuristic algorithms, a hybrid of meta-heuristic algorithms, machine learning and reinforcement learning. The results consistently demonstrate the superiority of the hybrid of meta-heuristic algorithms to address the multi-objective optimization problems.

A. The NP-Completeness of MO-CDTO Problem

The NP-complete problem is the hardest to solve. It is a problem that if there is an algorithm that can solve any of these problems, then it can solve all of them. Nobody has been able to prove that NP-complete problems are intractable. To prove that the MO-CDTO is an NP-completeness, first must explain that the problem belongs to the NP class and then show it is NP-hardness. When presenting a solution for the problem of MO-CDTO, the problem takes $O(ML)$ time to evaluate if the total task-offloading probability for each user is no more than one and the maximum workload restriction for each cloudlet. Furthermore, $O(L)$ time is required to verify that the number of deployed cloudlets at each base station is not greater than one. As a result, verifying a solution for MO-CDTO takes polynomial time. Therefore, MO-CDTO is under the class of NP problems. Based on the explanation of the p-median problem in [81]. When $\forall 1 \leq i \leq$ the number of users, the computing capacity of user i is zero, the computing capacity of each cloudlet is infinity, the cost of each cloudlet is equal to zero, the energy consumption of users is constant, the cloudlet deployment cost can be neglected, and the maximum number of cloudlets that may be deployed must be deployed to reduce user task response time. Therefore, the problem of MO-CDTO becomes selecting BSs to deploy the number of cloudlets and assigning each user to a cloudlet so that the sum of the connection delay is minimized from all users to the cloudlets they are assigned. To that end, each user must be assigned to a cloudlet with the smallest communication delay so that the MO-CDTO problem is NP-hard. Furthermore, based on the definition of NP-complete in [82] and [83] "If the problem $X \in$ NP class and X is NP-hard so X is NP-complete", while the problem of MO-CDTO belongs to the NP class and it is NP-hard, so it is NP-complete.

TABLE III. RELATED WORKS CONSIDERING A MULTI-OBJECTIVE OPTIMIZATION

Authors	Cloudlet Deployment	Task offloading	Time	Cost	Energy	Reliability	Load balancing	Cache content	Task Migration	User Mobility	Cloudlet Mobility
[39]	×	√	√	√	√	×	×	×	√	×	×
[29]	×	√	√	×	√	×	×	×	√	×	×
[26]	×	√	√	×	√	×	√	×	√	×	×
[77]	×	√	√	×	√	×	×	×	×	×	×
[78]	×	√	√	√	√	×	×	×	×	×	×
[79]	×	√	√	×	√	×	×	×	√	×	×
[5]	×	√	√	√	√	×	×	×	×	×	×
[76]	√	√	√	√	√	×	×	×	√	×	×
[80]	√	×	√	×	×	×	√	×	×	×	√

B. Meta-heuristic Algorithms

The meta-heuristic algorithms are a class of local search algorithms used to solve multi-objective optimization problems. The main objective of these algorithms is to find the optimal solution for the given set of problems based on a given set of constraints. The main idea behind this method is that it considers all the constraints and variables that affect the solution. Users can specify these constraints or automatically generate them by the algorithm. In mathematical programming, heuristic algorithms are used to design solutions to problems as quickly as possible. It may not produce the best solution, but it will give a near-optimal solution in a short time [71]. Some trade-off conditions give an initial idea of whether a heuristic algorithm is a good fit or not for a given problem. One of these conditions is the

completeness of the problem. If several solutions exist for a given problem, it is better to use a meta-heuristic algorithm [74]. A heuristic algorithm generally provides one solution that may not be the best among all the available solutions. Furthermore, meta-heuristic-based optimization has proved effective in solving many NP-hard problems. This method has been used in various real-world applications, including computer science, networking, communication, robotics, and manufacturing [84]. The main difference is that heuristic is the problem-specific method, while meta-heuristic is the problem-independent method that can be applied to many problems. Table IV classifies the heuristic and meta-heuristic algorithms. Meta-heuristic algorithms are divided into nature-inspired and non-nature-inspired.

TABLE IV. CLASSIFICATION OF HEURISTICS AND META-HEURISTICS ALGORITHMS

Heuristics	Meta-heuristics	
	Nature-inspired	Non-nature-inspired
<ul style="list-style-type: none"> • Greedy Algorithms • Best-First Search • Brute-force Search • A* Search • Hill Climbing • Bidirectional Search • Beam Search 	<ul style="list-style-type: none"> • Swarm Intelligence <ul style="list-style-type: none"> • Whale Optimization (WO) • Cuckoo search algorithm (CSA) • Particle Swarm Optimization (PSO) • Ant Colony Optimization (ACO) • Estimation of Distribution Algorithm (EDA) • Simulated annealing (SA) • Evolutionary Computation (EC) <ul style="list-style-type: none"> • Differential Evolution (DE) • Evolution Strategy (ES) • Genetic Algorithm (GA) • Genetic Programming (GP) 	<ul style="list-style-type: none"> • Guided Local Search (GLS) • Greedy Randomized Adaptive Search Procedure (GRASP) • Iterated Local Search (ILS) • Path Relinking (PR) • Scatter Search (SS) • Tabu Search (TS) • Variable Neighborhood Search (VNS)

Meta-heuristic algorithms are generated based on the mathematical models of different biological processes [3] and activities [85] that appear in nature, modified and applied according to a specific problem so that the result statistical data can be analyzed. Meta-heuristic algorithms can be used for CDTO performance metrics, such as cost, time and energy consumption, to reach the optimal solutions for the problem of MO-CDTO. The frameworks and procedures for the different evolutionary, swarm and hybrid algorithms are comparable [86]. However, the methods for population initiating, evaluating the initial fitness (the quality of the solution), the strategies for coming up with new solutions, and the iterative procedures are often different [3]. Because of their success in solving complex and important computing problems, meta-heuristic algorithms have achieved widespread during the last two decades [87]. Due to their applicable independence in problem-solving, meta-heuristic algorithms are useful and well respected for solving problems in different fields with highly acceptable performance. Meta-heuristic algorithms are frequently exploited as an efficient way to address NP-hard optimization problems [88]. The most well-known and extensively used under meta-heuristics are evolutionary algorithms (EAs). Individuals, or potential solutions in EAs, are made up of a chromosome (a representation of the problem's variables) and fitness (the quality of solutions) [89]. Individuals are grouped to form populations. Every time a new solution is developed, it is examined to determine its fitness value. A generation occurs when a new one replaces the

present population [90]. Evolution is iterating through consecutive generations that end when a termination condition is met. The most meta-heuristic algorithms used in the related works are the following:

1) Genetic Algorithm (GA)

Genetic Algorithms are natural search algorithms inspired by Darwin's theory of evolution. JH Holland first introduced it in 1973. Genetic algorithms can create high-quality solutions for various problems, including search and optimization, by imitating the processes of natural selection, reproduction, and mutation [91]. The population is updated using the traditional GA update approach, which includes binary tournament selection, a two-point crossover operator, and a mutation operator. Furthermore, the elite individuals (those with the highest fitness values in the population) are entirely replicated in the next generation using the elitist preservation mechanism of this algorithm [8].

2) Simulated Annealing (SA)

Simulated annealing is a technique for addressing unconstrained and bound-constrained optimization problems. It was first introduced by [92] in 1983. Simulated annealing investigates alternative configurations to produce better solutions to the massive combinatorial problem. The configurations are updated and compared at each iteration, allowing the best configuration to be chosen [93]. Furthermore, a simulated annealing algorithm initially allows other random movements in the neighboring search space, but

the activities are reduced over time. The algorithm chooses neighboring states with a cost lower than the current state or the exact cost [94].

3) Whale Optimization (WO)

Whale Optimization is a new nature-inspired meta-heuristic optimization algorithm that simulates humpback whale social behavior. The bubble-net hunting approach inspired the algorithm. Mirjalili and Lewis, two Australian researchers, introduced the Whale Optimization Algorithm in 2016 [95]. The WOA offers a lot of potential benefits. It is unaffected by early solutions, which can considerably impact some traditional algorithms. It also contains adaptive mechanisms to balance explorative and exploitative behaviors appropriately. It is widely used in various domains, including engineering optimization, feature selection, and parameter extraction [96].

4) Particle Swarm Optimization (PSO)

The particle swarm optimization algorithm is based on the study of bird predation behavior. Its straightforward premise is to discover the best solution through collaboration and sharing information between individuals in the swarm. The PSO is one of the meta-heuristic algorithms proposed by [97] in 1995 based on the social swarm evolving notion. According to experimental studies and applications, the PSO is highly competitive in optimization. The essential idea of particle swarms is the social interaction that results in collective intelligent behavior. Both the set of entities and the swarm of birds move with two velocity components. The first one attracts them to the best global entity, and the second attracts them to the best local entity [84].

5) Ant Colony Optimization (ACO)

In the ACO technique, natural ant behavior is crucial in determining the optimum route between colonies and food sources. This concept was first presented as the "ant system" [98] in 1992. As they go forward, the ants expel the pheromones. Pheromones form the shortest paths over time, and the intensity of the pheromone helps to identify the quickest route to the food source. Indeed, the ACO was inspired by the ant behavior to determine the shortest path between anthill and the location of the food supply [99].

6) Cuckoo Search Algorithm (CSA)

Cuckoo Search is a meta-heuristic optimization algorithm used to solve optimization problems. Yang and Deb invented it in 2009 [100]. The CSA is a nature-inspired meta-heuristic algorithm based on cuckoo brood parasitism, and Levy flights random walks. The CSA is based on the parasitism of the offspring of a bird species known as the cuckoo. The CSA algorithm implements the Lévy flights [100]. Usually, the parameters of the cuckoo search are kept constant for a particular duration; this results in a decrease in the algorithm's efficiency.

7) Differential Evolution (DE)

Differential evolution (DE) is one of the most well-known generation EAs, developed by Storn and Price [101] in 1997 and effective for constrained optimization problems in nonlinear and multimodal environments. Differential Evolution is a generic Nature-inspired population-based

global-search meta-heuristic optimization algorithm. Therefore, it is largely suited for numerical optimization problems based on vector differences.

8) Tabu Search (TS)

Tabu search (TS) is a heuristic method proposed initially by Glover in 1986 [102]. Tabu search is a potential tool to discover a feasible optimal solution from a limited set of solutions. In other words, it is an optimization approach that uses a guided local search procedure that avoids local optimum and denies moves to points already visited in the search space using the so-called tabu list [102].

The common trend of EAs is hybrids of different evolutionary algorithms. These hybrids are relatively easy since most EAs can use the same population table and gene presentation. Hybrids are mainly accomplished to avoid the shortcomings of one algorithm; for example, using a hybrid of two algorithms, one for a local search and the other for a global search.

V. VARIABLE-LENGTH

While there is no proper definition for a vector of a variable-length searching of solution space, therefore, most optimization theories depend on the fixed-length assumption to represent solutions [73]. There are several cases of variable-length approaches in which the number of variables is not fixed. One of these cases is CDTO. Standard optimization procedures may be used by considering a fixed number of variables. However, a sub-optimal length will result in a sub-optimal solution [86]. The algorithms can be executed iteratively and change the given length until an optimal solution is found. However, this is inefficient and impractical if there is an extensive range of possible lengths. For other fields of study, Table V shows the algorithms developed in meta-heuristic optimization with the variable length features. Variable-length algorithms are better since their solution vectors can vary in length [103].

Overall, as shown in Table V of the reviewed algorithms, the majority of them were developed to support a single objective except for the work of [89], which is based on an evolutionary algorithm and was applied only to a Bi-objectives problem without investigating the proposed algorithm and the work of [84], which suffers from weak interaction between solutions.

The traditional meta-heuristic optimization algorithms consider a fixed length of solution space, but these solutions do not apply to many real-world problems. The specific values of some decision variables might generate another or disable others, which cause the variable-length nature of solution space because of the different lengths of solutions. Multi-objective nature comes from having more than one objective to be optimized, e.g., delay, cost, and energy. Hence, MO-CDTO is a multi-objective variable-length optimization problem; the variable-length nature comes from the different number of cloudlets to deploy according to the computing requirements. Furthermore, the related works for the problem of MO-CDTO lack a single or multi-objective optimization algorithm with the variable-length feature.

TABLE V. THE PRESENT VARIABLE-LENGTH OPTIMIZATION FOR DIFFERENT FIELDS

Article	Application	Decision space	Algorithm	Limitation	Number Of objectives
[104]	Path planning	Path points	Genetic algorithm	Single objective	1
[73]	Laminate stacking Wind farm Sensor coverage	Based on the problem	Metameric Genetic	Single objective	1
[105]	Changing the topology of Convolutional Neural Networks	Encoding neuron in the layer	Particle swarm optimization	Single objective	1
[89]	Laminate stacking problem / angle-based transformation	Vector of angles on the plies of a laminate / general mathematical form	Multi-objective evolutionary algorithm	the application of the proposed algorithm is not investigated	2 / 3
[86]	A coverage and a wind farm problem	Based on the problem	Evolutionary algorithm	Single objective	1
[106]	Wireless Sensor Network deployment	Sensors location and coverage area	Genetic algorithm	Single objective	1
[84]	Wireless Sensor Network deployment	Sensors location and coverage area	Particle swarm optimization	Weakness of Inter-class interaction	2
[107]	Sensor node scheduling	Deciding the sensor that will send data	Genetic algorithm	Single objective	1

VI. CONCLUSION

Mobile edge computing has become an important technology to overcome some of the inherent constraints of mobile devices. Service providers are looking to provide excellent and cheap service to the MECE end-users. Furthermore, researchers focus on three main things in the MECE, which are minimizing time, reducing cost, and saving energy while solving different issues. Hence, the issues are related to each other. In contrast, the reliability and load-balancing affect time and energy, the cached content affects time, and the task migration and cloudlet mobility support the user mobility for good service and reducing time. The problem of CDTO has become an exciting research area. It has to be considered a multi-objective optimization problem since it needs more than one objective to be optimized. The problem of MO-CDTO is NP-complete because it belongs to the NP class and is NP-hard. The Meta-Heuristic algorithms have to be used for this type of problem as it is practical to solve MOO problems. For feature works, variable length of solution space is an appropriate approach for the problem of MO-CDTO because of the variation in the number of cloudlets to be deployed. A hybrid of two meta-heuristic algorithms with the variable-length aspect may generate powerful solutions. Many issues still need to address in the MECE, like reliability, user mobility, and load-balancing between cloudlets. This study is excellent material for future researchers to have an overview of the problem and take the research forward to resolve the unaddressed issues.

REFERENCES

- [1] X. Xu, X. Liu, X. Yin, S. Wang, Q. Qi, and L. Qi, "Privacy-aware offloading for training tasks of generative adversarial network in edge computing," *Information Sciences*, vol. 532, pp. 1–15, 2020, doi: 10.1016/j.ins.2020.04.026.
- [2] M. T. Kabir and C. Masouros, "A Scalable Energy vs. Latency Trade-Off in Full-Duplex Mobile Edge Computing Systems," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5848–5861, 2019, doi: 10.1109/tcomm.2019.2915833.
- [3] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task Scheduling in Cloud Computing based on Meta-heuristics: Review, Taxonomy, Open Challenges, and Future Trends," *Swarm and Evolutionary Computation*, vol. 62, no. October 2020, 2021, doi: 10.1016/j.swevo.2021.100841.
- [4] X. Zhao, Y. Shi, and S. Chen, "MAESP: Mobility aware edge service placement in mobile edge networks," *Computer Networks*, vol. 182, no. April, p. 107435, 2020, doi: 10.1016/j.comnet.2020.107435.
- [5] P. Wang, K. Li, B. Xiao, and K. Li, "Multi-objective Optimization for Joint Task Offloading, Power Assignment, and Resource Allocation in Mobile Edge Computing," *IEEE Internet of Things Journal*, vol. 4662, no. c, pp. 1–12, 2021, doi: 10.1109/JIOT.2021.3132080.
- [6] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Capacitated cloudlet placements in Wireless Metropolitan Area Networks," in *Proceedings - Conference on Local Computer Networks, LCN, 2017*, vol. 26-29-Octo, pp. 570–578, doi: 10.1109/LCN.2015.7366372.
- [7] X. Guan, X. Wan, T. Wang, and Y. Li, "A long-term cost-oriented cloudlet planning method in wireless metropolitan area networks," *Electronics (Switzerland)*, vol. 8, no. 11, p. 1216, 2019, doi: 10.3390/electronics8111213.
- [8] X. Zhao, C. Lin, and J. Zhang, "Cloudlet deployment for workflow applications in a mobile edge computing-wireless metropolitan area network," *Peer-to-Peer Networking and Applications*, vol. 15, no. 1, pp. 739–750, 2022, doi: 10.1007/s12083-021-01279-z.
- [9] M. Hui, J. Chen, Y. Zhou, B. He, K. Wu, and L. Yang, "Server Deployment and Load Balancing in Stochastic Mobile Edge Computing Networks," *IEEE Communications Letters*, vol. 26, no. 5, pp. 1194–1198, 2022, doi: 10.1109/LCOMM.2022.3151467.
- [10] S. Lai, R. Zhao, S. Tang, J. Xia, F. Zhou, and L. Fan, "Intelligent secure mobile edge computing for beyond 5G wireless networks," *Physical Communication*, vol. 45, p. 101283, 2021, doi: 10.1016/j.phycom.2021.101283.
- [11] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017, doi: 10.1109/COMST.2017.2745201.
- [12] X. Jin, F. Gao, Z. Wang, and Y. Chen, "Optimal deployment of mobile cloudlets for mobile applications in edge computing," *Journal of Supercomputing*, no. 0123456789, 2022, doi: 10.1007/s11227-021-04122-7.
- [13] H. Wu et al., "Resolving Multitask Competition for Constrained Resources in Dispersed Computing: A Bilateral Matching Game," *IEEE*

- Internet of Things Journal, vol. 8, no. 23, pp. 16972–16983, 2021, doi: 10.1109/JIOT.2021.3075673.
- [14] R. Singh, S. Armour, A. Khan, M. Sooriyabandara, and G. Oikonomou, “Towards Multi-Criteria Heuristic Optimization for Computational Offloading in Multi-Access Edge Computing,” IEEE International Conference on High Performance Switching and Routing, HPSR, vol. 2021-June, 2021, doi: 10.1109/HPSR52026.2021.9481852.
- [15] S. S. G. Chalapathi, V. Chamola, W. Johal, J. Aryal, and R. Buyya, “Energy and latency aware mobile task assignment for green cloudlets,” Simulation Modelling Practice and Theory, vol. 118, p. 102531, 2022, doi: 10.1016/j.simpat.2022.102531.
- [16] H. Ye, F. Huang, and W. Hao, “On Cost-Aware Heterogeneous Cloudlet Deployment for Mobile Edge Computing,” International Journal of Information Technology and Web Engineering, vol. 17, no. 1, pp. 1–23, 2022, doi: 10.4018/ijitwe.297968.
- [17] C. He, R. Wang, D. Wu, H. Zhang, and Z. Tan, “QoS-aware hybrid cloudlet placement over joint fiber and wireless backhaul access network,” Optical Switching and Networking, vol. 45, no. April, p. 100678, 2022, doi: 10.1016/j.osn.2022.100678.
- [18] J. Blank and K. Deb, “A Running Performance Metric and Termination Criterion for Evaluating Evolutionary Multi- And Many-objective Optimization Algorithms,” 2020 IEEE Congress on Evolutionary Computation, CEC 2020 - Conference Proceedings, pp. 3–10, 2020, doi: 10.1109/CEC48606.2020.9185546.
- [19] H. Song, B. Gu, K. Son, and W. Choi, “Joint Optimization of Edge Computing Server Deployment and User Offloading Associations in Wireless Edge Network via a Genetic Algorithm,” IEEE Transactions on Network Science and Engineering, vol. 9, no. 4, pp. 2535–2548, 2022, doi: 10.1109/TNSE.2022.3165372.
- [20] M. Jia, J. Cao, and W. Liang, “Optimal Cloudlet Placement and User to Cloudlet Allocation in Wireless Metropolitan Area Networks,” IEEE Transactions on Cloud Computing, vol. 5, no. 4, pp. 725–737, 2017, doi: 10.1109/tcc.2015.2449834.
- [21] F. Zeng, Y. Ren, X. Deng, and W. Li, “Cost-effective edge server placement in wireless metropolitan area networks,” Sensors (Switzerland), vol. 19, no. 1, pp. 1–21, 2019, doi: 10.3390/s19010032.
- [22] M. Jia, W. Liang, Z. Xu, M. Huang, and Y. Ma, “QoS-Aware Cloudlet Load Balancing in Wireless Metropolitan Area Networks,” IEEE Transactions on Cloud Computing, vol. 8, no. 2, pp. 623–634, 2020, doi: 10.1109/TCC.2017.2786738.
- [23] N. Hassan, K. L. A. Yau, and C. Wu, “Edge computing in 5G: A review,” IEEE Access, vol. 7, pp. 127276–127289, 2019, doi: 10.1109/ACCESS.2019.2938534.
- [24] S. B. M. Baskaran and G. Raja, “Blind key distribution mechanism to secure wireless metropolitan area network,” CSI Transactions on ICT, vol. 4, no. 2–4, pp. 157–163, 2016, doi: 10.1007/s40012-016-0110-3.
- [25] M. L. Attiah, A. A. M. Isa, Z. Zakaria, M. K. Abdulhameed, M. K. Mohsen, and A. M. Dinar, “Independence and Fairness Analysis of 5G mmWave Operators Utilizing Spectrum Sharing Approach,” Mobile Information Systems, vol. 2019, 2019, doi: 10.1155/2019/4370847.
- [26] X. Xu et al., “An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks,” Journal of Network and Computer Applications, vol. 133, no. September 2018, pp. 75–85, 2019, doi: 10.1016/j.jnca.2019.02.008.
- [27] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C. H. Hsu, “Edge server placement in mobile edge computing,” Journal of Parallel and Distributed Computing, vol. 127, pp. 160–168, 2019, doi: 10.1016/j.jpdc.2018.06.008.
- [28] F. Guo, B. Tang, and J. Zhang, “Mobile edge server placement based on meta-heuristic algorithm,” Journal of Intelligent and Fuzzy Systems, vol. 40, no. 5, pp. 8883–8897, 2021, doi: 10.3233/JIFS-200933.
- [29] X. Xu et al., “A computation offloading method over big data for IoT-enabled cloud-edge computing,” Future Generation Computer Systems, vol. 95, pp. 522–533, 2019, doi: 10.1016/j.future.2018.12.055.
- [30] B. Li, P. Hou, H. Wu, R. Qian, and H. Ding, “Placement of edge server based on task overhead in mobile edge computing environment,” Transactions on Emerging Telecommunications Technologies, vol. November, pp. 1–19, 2020, doi: 10.1002/ett.4196.
- [31] L. Chen, J. Wu, G. Zhou, and L. Ma, “QUICK: QoS-guaranteed efficient cloudlet placement in wireless metropolitan area networks,” Journal of Supercomputing, vol. 74, no. 8, pp. 4037–4059, 2018, doi: 10.1007/s11227-018-2412-8.
- [32] Z. Qin, F. Xu, Y. Xie, Z. Zhang, and G. Li, “An improved Top-K algorithm for edge servers deployment in smart city,” Transactions on Emerging Telecommunications Technologies, vol. 32, no. 8, pp. 1–20, 2021, doi: 10.1002/ett.4249.
- [33] K. Cao, L. Li, Y. Cui, T. Wei, and S. Hu, “Exploring Placement of Heterogeneous Edge Servers for Response Time Minimization in Mobile Edge-Cloud Computing,” IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 494–503, 2021, doi: 10.1109/TII.2020.2975897.
- [34] Q. Fan and N. Ansari, “On cost aware cloudlet placement for mobile edge computing,” IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 4, pp. 926–937, 2019, doi: 10.1109/JAS.2019.1911564.
- [35] L. Cui et al., “Joint optimization of energy consumption and latency in mobile edge computing for internet of things,” IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4791–4803, 2019, doi: 10.1109/JIOT.2018.2869226.
- [36] X. Li, F. Zeng, G. Fang, Y. Huang, and X. Tao, “Load balancing edge server placement method with QoS requirements in wireless metropolitan area networks,” no. iii, 2021, doi: 10.1049/iet-com.2020.0651.
- [37] L. Zhao, W. Sun, Y. Shi, and J. Liu, “Optimal Placement of Cloudlets for Access Delay Minimization in SDN-Based Internet of Things Networks,” IEEE Internet of Things Journal, vol. 5, no. 2, pp. 1334–1344, 2018, doi: 10.1109/JIOT.2018.2811808.
- [38] S. Mondal, G. Das, and E. Wong, “Efficient cost-optimization frameworks for hybrid cloudlet placement over fiber-wireless networks,” Journal of Optical Communications and Networking, vol. 11, no. 8, pp. 437–451, 2019, doi: 10.1364/JOCN.11.000437.
- [39] K. Peng et al., “An energy- and cost-aware computation offloading method for workflow applications in mobile edge computing,” Eurasip Journal on Wireless Communications and Networking, vol. 2019, no. 1, 2019, doi: 10.1186/s13638-019-1526-x.
- [40] Liqing Liu, Zheng Chang, Xijuan Guo, and T. Ristaniemi, “Multi-objective optimization for computation offloading in mobile-edge computing,” in 2017 IEEE Symposium on Computers and Communications (ISCC), Jul. 2017, vol. 5, no. 1, pp. 832–837, doi: 10.1109/ISCC.2017.8024630.
- [41] Z. Wang, F. Gao, and X. Jin, “Optimal deployment of cloudlets based on cost and latency in Internet of Things networks,” Wireless Networks, vol. 26, no. 8, pp. 6077–6093, 2020, doi: 10.1007/s11276-020-02418-9.
- [42] J. Liu et al., “Reliability-Enhanced Task Offloading in Mobile Edge Computing Environments,” IEEE Internet of Things Journal, vol. PP, no. c, pp. 1–1, 2021, doi: 10.1109/JIOT.2021.3115807.
- [43] F. Guo, B. Tang, L. Kang, and L. Zhang, “Mobile Edge Server Placement Based on Bionic Swarm Intelligent Optimization Algorithm,” in Lecture Notes of the Institute for Computer Sciences, Social- Informatics and Telecommunications Engineering, LNICST, 2021, vol. 350, pp. 95–111, doi: 10.1007/978-3-030-67540-0_6.
- [44] X. Zhao, Y. Zeng, H. Ding, B. Li, and Z. Yang, “Optimize the placement of edge server between workload balancing and system delay in smart city,” Peer-to-Peer Networking and Applications, vol. 14, no. 6, pp. 3778–3792, Nov. 2021, doi: 10.1007/s12083-021-01208-0.
- [45] H. Wei, H. Luo, Y. Sun, and M. S. Obaidat, “Value-driven Cache Replacement Strategy in Mobile Edge Computing,” 2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings, vol. 2020-Janua, pp. 19–24, 2020, doi: 10.1109/GLOBECOM42002.2020.9348106.
- [46] Z. Liu, J. Zhang, and J. Wu, “Joint Optimization of Server Placement and Content Caching in Mobile Edge Computing Networks,” ACM International Conference Proceeding Series, pp. 149–153, 2019, doi: 10.1145/3375998.3376024.
- [47] T. L. Chin, Y. S. Chen, and K. Y. Lyu, “Queueing Model Based Edge Placement for Work Offloading in Mobile Cloud Networks,” IEEE Access, vol. 8, pp. 47295–47303, 2020, doi: 10.1109/ACCESS.2020.2979479.

- [48] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: Blockchain-Enabled Computation Offloading for IoT in Mobile Edge Computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020, doi: 10.1109/TII.2019.2936869.
- [49] C. Zhang and Z. Zheng, "Task migration for mobile edge computing using deep reinforcement learning," *Future Generation Computer Systems*, vol. 96, pp. 111–118, 2019, doi: 10.1016/j.future.2019.01.059.
- [50] L. Loven et al., "Scaling up an Edge Server Deployment," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2020, pp. 1–7, doi: 10.1109/PerComWorkshops48775.2020.9156204.
- [51] H. Yao, C. Bai, M. Xiong, D. Zeng, and Z. Fu, "Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing," *Concurrency Computation*, vol. 29, no. 16, pp. 1–9, 2017, doi: 10.1002/cpe.3975.
- [52] 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.
- [53] H. Zhang, R. Wang, W. Sun, and H. Zhao, "Mobility Management for Blockchain-based Ultra-dense Edge Computing: A Deep Reinforcement Learning Approach," *IEEE Transactions on Wireless Communications*, vol. 1276, no. c, pp. 1–14, 2021, doi: 10.1109/TWC.2021.3082986.
- [54] Y. Miao, G. Wu, M. Li, A. Ghoneim, and M. Al-rakhami, "Intelligent task prediction and computation offloading based on mobile-edge cloud computing," *Future Generation Computer Systems*, vol. 102, pp. 925–931, 2020, doi: 10.1016/j.future.2019.09.035.
- [55] 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.
- [56] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient Algorithms for Capacitated Cloudlet Placements," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2866–2880, 2016, doi: 10.1109/TPDS.2015.2510638.
- [57] Z. Xu et al., "An IoT-oriented offloading method with privacy preservation for cloudlet-enabled wireless metropolitan area networks," *Sensors (Switzerland)*, vol. 18, no. 9, pp. 1–18, 2018, doi: 10.3390/s18093030.
- [58] I. Hadžić, Y. Abe, and H. C. Woithe, "Server Placement and Selection for Edge Computing in the ePC," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 671–684, 2019, doi: 10.1109/TSC.2018.2850327.
- [59] S. Mondal, G. Das, and E. Wong, "Cost-optimal cloudlet placement frameworks over fiber-wireless access networks for low-latency applications," *Journal of Network and Computer Applications*, vol. 138, no. March, pp. 27–38, 2019, doi: 10.1016/j.jnca.2019.04.014.
- [60] T. K. Rodrigues, K. Suto, and N. Kato, "Edge Cloud Server Deployment with Transmission Power Control through Machine Learning for 6G Internet of Things," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 4, pp. 2099–2108, 2019, doi: 10.1109/ETC.2019.2963091.
- [61] S. Yang, F. Li, M. Shen, X. Chen, X. Fu, and Y. Wang, "Cloudlet placement and task allocation in mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5853–5863, 2019, doi: 10.1109/JIOT.2019.2907605.
- [62] S. Bi, L. Huang, and Y. J. A. Zhang, "Joint optimization of service caching placement and computation offloading in mobile edge computing systems," *arXiv*, vol. 19, no. 7, pp. 4947–4963, 2019, doi: 10.1109/TWC.2020.2988386.
- [63] N. Shan, Y. Li, and X. Cui, "A Multilevel Optimization Framework for Computation Offloading in Mobile Edge Computing," *Mathematical Problems in Engineering*, vol. 2020, 2020, doi: 10.1155/2020/4124791.
- [64] B. Li, P. Hou, H. Wu, and F. Hou, "Optimal edge server deployment and allocation strategy in 5G ultra-dense networking environments," vol. 72, 2021, doi: 10.1016/j.pmcj.2020.101312.
- [65] D. Bhatta and L. Mashayekhy, "A Bifactor Approximation Algorithm for Cloudlet Placement in Edge Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1787–1798, 2022, doi: 10.1109/TPDS.2021.3126256.
- [66] Y. Li, A. Zhou, X. Ma, and S. Wang, "Profit-Aware Edge Server Placement," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 55–67, Jan. 2022, doi: 10.1109/JIOT.2021.3082898.
- [67] U. Shaukat, E. Ahmed, Z. Anwar, and F. Xia, "Cloudlet deployment in local wireless networks: Motivation, architectures, applications, and open challenges," *Journal of Network and Computer Applications*, vol. 62, no. DECEMBER, pp. 18–40, 2016, doi: 10.1016/j.jnca.2015.11.009.
- [68] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019, doi: 10.1016/j.future.2019.02.050.
- [69] B. Wang, C. Wang, W. Huang, Y. Song, and X. Qin, "A survey and taxonomy on task offloading for edge-cloud computing," *IEEE Access*, vol. 8, pp. 186080–186101, 2020, doi: 10.1109/ACCESS.2020.3029649.
- [70] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective," *Computer Networks*, vol. 182, no. March, 2020, doi: 10.1016/j.comnet.2020.107496.
- [71] V. Pardo-Castello and F. R. Tiant, "Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction," *Journal of the American Medical Association*, vol. 121, no. 16, pp. 1264–1269, 2011, doi: 10.1007/978-0-85729-652-8_1.
- [72] A. Rahbari, M. M. Nasiri, F. Werner, M. M. Musavi, and F. Jolai, "The vehicle routing and scheduling problem with cross-docking for perishable products under uncertainty: Two robust bi-objective models," *Applied Mathematical Modelling*, vol. 70, pp. 605–625, 2019, doi: 10.1016/j.apm.2019.01.047.
- [73] M. L. Ryerkerk, R. C. Averill, K. Deb, and E. D. Goodman, "Solving metameric variable-length optimization problems using genetic algorithms," *Genetic Programming and Evolvable Machines*, vol. 18, no. 2, pp. 247–277, 2017, doi: 10.1007/s10710-016-9282-8.
- [74] G. R. Zavala, A. J. Nebro, F. Luna, and C. A. Coello Coello, "A survey of multi-objective metaheuristics applied to structural optimization," *Structural and Multidisciplinary Optimization*, vol. 49, no. 4, pp. 537–558, 2014, doi: 10.1007/s00158-013-0996-4.
- [75] A. M. Maia, Y. Ghamri-doudane, D. Vieira, and M. F. De, "An Improved Multi-Objective Genetic Algorithm With Heuristic Initialization for service placement and Load Distribution in Edge Computing," *Computer Networks*, p. 108146, 2021, doi: 10.1016/j.comnet.2021.108146.
- [76] X. Zhu and M. C. Zhou, "Multi-Objective Optimized Cloudlet Deployment and Task Offloading for Mobile Edge Computing," *IEEE Internet of Things Journal*, vol. 4662, no. c, pp. 1–1, 2021, doi: 10.1109/jiot.2021.3073113.
- [77] K. Peng, H. Huang, S. Wan, and V. C. M. Leung, "End-edge-cloud collaborative computation offloading for multiple mobile users in heterogeneous edge-server environment," *Wireless Networks*, vol. 1, 2020, doi: 10.1007/s11276-020-02385-1.
- [78] F. Sufyan and A. Banerjee, "Computation Offloading for Distributed Mobile Edge Computing Network: A Multiobjective Approach," *IEEE Access*, vol. 8, no. Mcc, pp. 149915–149930, 2020, doi: 10.1109/ACCESS.2020.3016046.
- [79] F. Song, H. Xing, S. Luo, D. Zhan, P. Dai, and R. Qu, "A Multiobjective Computation Offloading Algorithm for Mobile-Edge Computing," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8780–8799, 2020, doi: 10.1109/JIOT.2020.2996762.
- [80] F. Luo, S. Zheng, W. Ding, J. Fuentes, and Y. Li, "An Edge Server Placement Method Based on Reinforcement Learning," *Entropy*, vol. 24, no. 3, pp. 1–14, 2022, doi: 10.3390/e24030317.
- [81] Zhilei Ren, He Jiang, Jifeng Xuan, and Zhongxuan Luo, "An Accelerated-Limit-Crossing-Based Multilevel Algorithm for the \$P\$-\$S\$-Median Problem," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1187–1202, Aug. 2012, doi: 10.1109/TSMCB.2012.2188100.
- [82] Michael R. Garey and D. S. Johnson, *A Guide to the Theory of NP-Completeness*. 1979.
- [83] O. Kariv and S. L. Hakimi, "An Algorithmic Approach to Network Location Problems. I: The p -Centers," *SIAM Journal on Applied*

- Mathematics, vol. 37, no. 3, pp. 513–538, Dec. 1979, doi: 10.1137/0137040.
- [84] A. M. Jubair, R. Hassan, A. H. M. Aman, and H. Sallehudin, “Social class particle swarm optimization for variable-length Wireless Sensor Network Deployment,” *Applied Soft Computing*, vol. 113, p. 107926, Dec. 2021, doi: 10.1016/j.asoc.2021.107926.
- [85] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, “A survey on new generation metaheuristic algorithms,” *Computers & Industrial Engineering*, vol. 137, no. September, p. 106040, Nov. 2019, doi: 10.1016/j.cie.2019.106040.
- [86] M. Ryerkerk, R. Averill, K. Deb, and E. Goodman, “A novel selection mechanism for evolutionary algorithms with metameric variable-length representations,” *Soft Computing*, vol. 24, no. 21, pp. 16439–16452, 2020, doi: 10.1007/s00500-020-04953-1.
- [87] P. Lu, L. Ye, Y. Zhao, B. Dai, M. Pei, and Y. Tang, “Review of metaheuristic algorithms for wind power prediction: Methodologies, applications and challenges,” *Applied Energy*, vol. 301, no. June, p. 117446, Nov. 2021, doi: 10.1016/j.apenergy.2021.117446.
- [88] N. Razmjoo, V. V. Estrela, R. Padilha, and A. C. B. Monteiro, “Metaheuristics and Optimization in Computer and Electrical Engineering,” in *Lecture Notes in Electrical Engineering*, vol. 696, 2021, pp. 25–47.
- [89] H. Li, K. Deb, and Q. Zhang, “Variable-length pareto optimization via decomposition-based evolutionary multiobjective algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 987–999, 2019, doi: 10.1109/TEVC.2019.2898886.
- [90] R. Cheng, C. He, Y. Jin, and X. Yao, “Model-based evolutionary algorithms: a short survey,” *Complex & Intelligent Systems*, vol. 4, no. 4, pp. 283–292, 2018, doi: 10.1007/s40747-018-0080-1.
- [91] R. Lahoz-Beltra, “Quantum Genetic Algorithms for Computer Scientists,” *Computers*, vol. 5, no. 4, p. 24, Oct. 2016, doi: 10.3390/computers5040024.
- [92] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.
- [93] P. C. Huang, T. L. Chin, and T. Y. Chuang, “Server Placement and Task Allocation for Load Balancing in Edge-Computing Networks,” *IEEE Access*, vol. 9, pp. 138200–138208, 2021, doi: 10.1109/ACCESS.2021.3117870.
- [94] S. K. Kasi et al., “Heuristic Edge Server Placement in Industrial Internet of Things and Cellular Networks,” *IEEE Internet of Things Journal*, vol. 4662, no. c, pp. 1–8, 2020, doi: 10.1109/JIOT.2020.3041805.
- [95] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.
- [96] 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, pp. 1–24, 2021, doi: 10.3390/s21082628.
- [97] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.
- [98] M. Dorigo, V. Maniezzo, and A. Colomi, “Ant system: optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, Feb. 1996, doi: 10.1109/3477.484436.
- [99] Y. Dai, Y. Lou, and X. Lu, “A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization Algorithm with Multi-QoS Constraints in Cloud Computing,” in *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Aug. 2015, vol. 2, no. 1, pp. 428–431, doi: 10.1109/IHMSC.2015.186.
- [100] X. Yang and Suash Deb, “Cuckoo Search via Levy flights,” in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 2009, pp. 210–214, doi: 10.1109/NABIC.2009.5393690.
- [101] H. Li, P. G. H. Nichols, S. Han, K. J. Foster, K. Sivasithamparam, and M. J. Barbetti, “Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *Australasian Plant Pathology*, vol. 38, no. 3, p. 284, 1997, doi: 10.1023/A:1008202821328.
- [102] M. Gendreau and J.-Y. Potvin, “Tabu Search,” in *Search Methodologies*, vol. 1–2, Boston, MA: Springer US, 2005, pp. 165–186.
- [103] M. Ryerkerk, R. Averill, K. Deb, and E. Goodman, A survey of evolutionary algorithms using metameric representations, vol. 20, no. 4. Springer US, 2019.
- [104] Z. Qiongbing and D. Lixin, “A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems,” *Expert Systems with Applications*, vol. 60, pp. 183–189, 2016, doi: 10.1016/j.eswa.2016.04.005.
- [105] B. Wang, Y. Sun, B. Xue, and M. Zhang, “Evolving Deep Convolutional Neural Networks by Variable-Length Particle Swarm Optimization for Image Classification,” *2018 IEEE Congress on Evolutionary Computation, CEC 2018 - Proceedings*, pp. 1–8, 2018, doi: 10.1109/CEC.2018.8477735.
- [106] 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,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 10, pp. 4177–4194, 2020, doi: 10.1007/s12652-020-01698-5.
- [107] 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.