Edge Server Placement and Allocation Optimization: A Tradeoff for Enhanced Performance

Ardalan Ghasemzadeh
University of Tabriz
Hadi S. Aghdasi (aghdasi@tabrizu.ac.ir)
University of Tabriz
Saeed Saeedvand
National Taiwan Normal University

Research Article

Keywords: Multi-objective optimization, Edge server placement and allocation, Latency, Workload balance

Posted Date: November 16th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3597093/v1

Additional Declarations: No competing interests reported.

Edge Server Placement and Allocation Optimization: A Tradeoff for Enhanced Performance

Ardalan Ghasemzadeh[a], Hadi S. Aghdasi[b,*], Saeed Saeedvand[c]

[a,b]Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran,
a.ghasemzadeh@tabrizu.ac.ir, aghdasi@tabrizu.ac.ir
[c]Department of Electrical Engineering, National Taiwan Normal University, Taipei, Taiwan
saeedvand@ntnu.edu.tw

*Corresponding author

Abstract

Considering the expansion of the Internet of Things (IoT) and the volume of data and user requests, Mobile Edge Computing (MEC) is considered a novel and efficient solution that puts decentralized servers at the network's edge. This has the effect of lowering bandwidth demand and transmission latency. Optimal edge server placement and allocation, as the first stage of MEC, can improve end-user service quality, edge computing system utility, and cost and energy consumption. The majority of previous edge server placement studies have employed only one objective or developed a fitness function by the weighted sum method for optimization. Usually, using a single optimization objective without considering other objectives cannot yield the desired results for a problem with a multi-objective design. On the other hand, assigning weights to objectives can lead to losing optimal points in non-convex problems and selecting improper weights. Therefore, in this paper, we propose a multi-objective solution for the positioning and allocation of edge servers for MEC services based on the NSGA-II algorithm. In this regard, we identify two workload variance and latency reduction objectives with extensive evaluations. The experimental evaluation of the results using real-world data reveals that solutions based on the NSGA-II yield superior convergence and diversity of Pareto front points compared to Multi-Objective Particle Swarm Optimization (MOPSO), Multi-Objective Biogeography Based Optimization (MOBBO), and Adaptive Weighted Sum Method (AWSM). Additionally, it effectively mitigates workload variance on servers and exhibits an average latency reduction of 8.79% in comparison to the adaptive weighted-sum approach, 9.19% in comparison to MOPSO, and 0.28% in comparison to MOBBO.

Keywords: Multi-objective optimization, Edge server placement and allocation, Latency, Workload balance.

1. Introduction

Due to the accelerating growth of the Internet of Things and fifth-generation (5G) mobile networks, along with the exponential production of big data, cloud computing is no longer sufficiently efficient enough as a solution for lower latency, greater influence, and high security [1, 2]. These requirements are forcing mobile operators to look for a decentralized solution to enhance the level of service quality for customers. Mobile Edge Computing (MEC) is a novel approach that involves the placement of decentralized computing and storage resources at the network's edge [3, 4]. This solution has been shown to effectively reduce latency in responding to user requests, as compared to the centralized cloud system [5, 6]. It reduces the traffic in the main network. MEC is a nascent paradigm that aims to enhance the responsiveness of user requests by deploying edge servers equipped with storage and computing capabilities on base stations within the network. This approach is designed to mitigate transmission latency and enhance real-time data analysis [7-11]. To reduce the latency caused by the considerable distance between the cloud computing center and end-users, in MEC networks, compute servers are installed on base stations at the network's edge [9, 12]. There are many challenges associated with implementing MEC networks. Many services and requests run on edge servers, such as the Internet of Things (IoT), virtual reality, and the Internet of Vehicles (IoV), which are sensitive to latency. They require intensive computation and the resources of edge servers, and they are very limited compared to large data centers [9, 13]. One of the first phases in implementing a mobile edge computing network is edge server placement that plays a vital role in mobile edge computing to enable low-latency and high-throughput services by deploying edge servers at suitable geographical locations [14]. Given the budget constraints of MEC service providers, optimal server placement can be an appropriate solution to improve the quality of response to user requests with the lowest latency and cost [15]. The effectiveness of the MEC network may be improved by allocating computing and storage resources fairly to all users, which is achieved via the strategic deployment of edge servers. Improper placement is a well-known problem that can result in uneven distribution of workload across servers, causing some servers to be idle while other servers are heavily utilized [16, 17]. While strategic positioning of edge servers within the MEC network has many benefits, it also has challenges and constraints, such as deciding how many edge servers to deploy. This directly relates to cost, determining the location of a limited quantity of edge servers to be installed

at the location of a significant quantity of base stations, and assigning a lot of base stations to the servers after the servers have been deployed [18].

In optimal edge server placement, important parameters should be considered. One of the most important parameters is reducing latency. Latency in responding to user requests can cause irreparable damage. In the IoV, for example, it can lead to accidents. Proportional use of a limited number of servers can reduce costs while ensuring high quality of service for users. The use of a large number of servers causes an increase in costs and a waste of energy due to the inactivity of the servers. Using a small number of servers causes a sharp decrease in the quality of service and sends more requests to the cloud computing server, resulting in an increase in latency [19]. Edge server deployment is an extensive problem. The process involves the identification of prospective locations among a vast array of base stations, followed by the deployment of edge servers in said locations. The extensive quantity of base stations results in a search space that exhibits a significantly elevated time complexity. Therefore, the task of identifying the most suitable location for servers is categorized as an NP-hard problem and can be addressed by utilizing metaheuristic optimization methodologies [20].

Another challenge in this optimization problem is to find a tradeoff between the objectives. Using a large number of edge servers to reduce latency results in a significant cost rise, higher energy consumption, and wasted hardware resources. Using a small number of edge servers does not guarantee a reduction in latency when responding to user requests. In the realm of cloud computing, servers that are not in use are powered down in order to mitigate energy wastage. However, in mobile edge computing, since space on base stations is limited and only one edge server can be deployed on each base station, shutting down one edge server results in a communication gap in the entire network. Therefore, a multi-objective optimization method should be used, which can show the tradeoff between the objectives [20, 21].

Placing the edge servers on all base stations is deemed unfeasible in light of the exorbitant expenses involved. The server placement optimization problem aims to minimize costs while maximizing service quality for users, all while working within the constraints of a limited number of servers. [22]. Many studies have been carried out on the subject of user offloading issues. However, there exists a limited quantity of research in the domain of server placement. Numerous investigations on this topic have examined the problem of optimizing server placement by looking

at one or two objectives independently, and these studies have ignored the tradeoff between the objectives [23].

In this paper, we reduce the latency in responding to user requests and balance the workload across a limited number of servers by defining an optimization model. To achieve this, we propose a strategy to optimally deploy edge servers and allocate base stations to those servers. In addition to defining the effective latency reduction and workload balancing objectives, the optimization effect of each objective was studied at the same time. We employed the Genetic Algorithm (GA) to investigate the effects of individual objectives on problem-solving efficacy. The tradeoff between these two objectives is investigated based on the NSGA-II algorithm. Examination of the results shows that this metaheuristic optimization algorithm has not only good convergence compared to the methods studied but also produces good diversity for the Pareto front points. The present study examines the edge servers allocation and placement problem, focusing on the viewpoints of both edge providers and end users. The subsequent paragraphs provide concise synopses of the principal contributions:

1) A multi-objective algorithm that employs NSGA-II is proposed to tackle the issue of edge server allocation and placement. The algorithm was devised with the aim of optimizing two objectives that are both desirable but inherently contradictory. Each objective's optimization effect on the problem solution was investigated individually.

2) A comprehensive comparison between our approach and Multi-Objective Particle Swarm Optimization (MOPSO), Multi-Objective Biogeography Based Optimization (MOBBO), and Adaptive Weighted Sum Method (AWSM) has been conducted, and results demonstrated by well-known metrics, Inverted Generational Distance (IGD), and Hypervolume (HV).

3) In this study, we examined the impact of the tradeoff between objectives on problem resolution. Our findings indicate that our approach outperforms existing methods in terms of problem convergence and the diversity of Pareto front points.

4) In this study, we conducted an analysis of real-world scenarios based on an authentic telecom dataset from Shanghai. Our objective was to address the issue of edge server placement and allocation in MEC, taking into consideration the perspectives of both edge providers and end users.

The remainder of the paper is structured as follows. Section 2 provides a short overview of relevant work. The system model is explained in Section 3. In Section 4, the simulation results are illustrated and analyzed and the results are assessed , and this work is concluded in Section 5.

2. Related Work

Research interest in MEC has increased significantly due to its optimistic outlook [24]. There is a significant amount of study on offloading in MEC in the literature. This issue could be broken down into a number of smaller issues, such as what to be offloaded, whether to offload, how the tasks should be offloaded, etc. But a few investigations have been performed on the issue of edge server placement recently [16, 20, 22, 25, 26]. The primary focus of most studies was on the extent of edge server coverage, which was assessed based on the number of mobile consumers or the coverage area. The objective was to optimize the placement advantages or minimize the associated costs and energy consumption. Edge server placement-related issues can be divided into two categories. First, attempting to position servers with the goal of maximizing the quality of experience for mobile users [27-29], and second, seeking to identify a set of placement sites with the goal of maximizing the performance of servers [30-32].

Despite the recent focus on MEC research, the placement of edge servers has been covered in prior literature. The problem of edge placement is getting more and more attention [9]. Furthermore, once the number of base stations in a given area is established, we must first choose the ideal quantity and precise location for the servers, assign the base stations to the servers, and then tweak the placement scheme to achieve the ideal one while taking into account the access delay and workload balance of edge servers [15].

Recently, a few initiatives have been made to effectively deploy edge servers while adhering to various optimization objectives and financial constraints. In fact, it is not practical to deploy enough edge servers everywhere since edge computing service providers' budgets are constantly constrained. The research in [33] looked at ways to deploy edge servers efficiently and affordably by limiting the number of edge servers while maintaining some Quality of Service (QoS) standards.

According to [34], the placement plan significantly affects the efficiency of edge resources. They offer a set of criteria to assess where edge servers should be located in the upcoming 5G

scenario. The goal of the study in [22] is to reduce the overall number of edge servers. To overcome this problem, they employ a greedy-based method and a simulated annealing-based strategy. In [19], the authors use information from social networks to position edge servers, relieving bandwidth demand and lowering access delay. The network robustness of the distributed MEC environment is taken into account in [35]. The authors suggest using integer programming to deploy edge servers and enhance user experience. An issue is formulated by [25] in order to decrease overall energy consumption. The matter of selecting an edge server is conceptualized by reference [17] through the utilization of metrics such as energy consumption and time latency. The edge server placement problem is investigated by [36] and [20] in order to balance workload distribution and access latency. The coverage of edge servers inside the specific geographic area is maximized by [35]. The authors do not, however, account for the placement expense. In order to achieve the lowest cost, the authors of [16] developed the problem of placing edge servers without taking the crucial QoS into account. The objective of reducing the quantity of edge servers has been formulated by the authors in reference [22].

In particular, The authors of [22] turned the cost-effective placement of edge servers into a problem in graph theory involving the minimal dominating set. The research conducted by [33] established the issue utilizing integer linear programming and employed a greedy algorithm to address it. The deployment expenses and the geographical expanse serviced by edge servers were taken into account in the study [37], and the placement solution was then determined using a dynamic programming approach and geometric image technique. The paper [25] presented an energy-aware placement of edge servers utilizing particle swarm optimization, whereas LESP [38] proposed a placement method based on load for peripheral servers and developed a placement strategy based on a tree. The authors of [20] and [39] suggested a deployment strategy based on mixed integer programming for edge servers in order to balance their workloads and reduce access delay. In [40], the authors characterize the placement of edge servers as a location-allocation issue with the capacity to move servers closer to Wi-Fi access points.

In [41], the authors suggested a latency-aware heuristic placement approach to effectively manage numerous applications within mobile edge networks. The SPAC [42] employed a local search algorithm to minimize the combined cost of opening edge servers and providing services. In work [43], it was suggested that the expansion of edge server deployment could be achieved by strategically determining the appropriate quantity of new edge servers and effectively

redistributing access points among both the pre-existing and newly added edge servers. The study conducted in reference [26] increased operational efficiency and decreased costs associated with edge provisioning through the identification of suitable and unanticipated edge locations. To define the fitness function and achieve the best server placement, the authors applied the two delay and efficiency objectives [44]. Using the Genetic Algorithm (GA), they have provided the optimal location for placing servers based on 'Telecom's data from Shanghai, China. Latency and energy consumption objectives are utilized to define a fitness function in [9], and the PSO algorithm is then used to optimize this fitness function. In another work, the authors have optimized the server placement problem using the Biogeography Based Optimization (BBO) algorithm. They defined a fitness function using the time delays and costs spent in implementing the system [15]. The workload balance and server deployment cost are used to define the fitness function [45]. By applying the gray wolf algorithm, time and work are saved.

3. Proposed placement and allocation method for edge servers

MEC is a three-layer network. The top layer is a central cloud computing system with powerful computing and storage resources. The middle layer, or edge layer, contains base stations that receive user requests, and the bottom layer is connected to user devices. In this study, we are focusing on the middle layer. We plan to deploy k edge servers on m base stations to implement the decentralized MEC system optimally. Each of these m base stations can potentially host an edge server. Since the base stations have limited space for computing and storage resources, only one edge server can be installed on each base station.

Many base stations are geographically dispersed throughout the major cities, and these base stations are served by each of the servers positioned at the network's edge. The connectivity between an edge server and a base station is singular, and user requests are transmitted to the corresponding server by means of these base stations.

3.1. Dataset

Through simulations using data from the actual world, we leverage Shanghai Telecom's base station locations and data request datasets. The dataset includes the users' access records as well as the geographic data of base stations, such as longitude and latitude. The datasets comprise approximately 7 million instances of calls and data requests that were transmitted via 3042 base

stations from 9481 edge devices [16, 31, 32] in order to make the suggested environment design feasible. The data contained in each call and data record is represented by a tuple of requests that have been transmitted by a device to a base station at distinct points in time. Due to its high population, the Shanghai dataset is a proper dataset to be used for implementing a mobile edge server placement solution. Figure 1 depicts the location of each base station in Shanghai, China. Each dot's color represents the number of incoming calls and data requests originating from edge devices. The red region represents a densely packed dispersion of base stations.

Figure 1. Base station locations from the Shanghai Telecom dataset [9, 46, 47].

The visual representation of the locations of base stations holds significance in comprehending that a plausible resolution for the placement of mobile edge servers would entail their relocation to alternative base station sites.

Theorem. The optimal edge server placement problem is NP-Hard.

Proof. We illustrate the NP-hardness of the optimum edge server placement problem via a reduction from the set cover problem. The objective of the set cover problem is to determine a collection P that satisfies $|P| < K$ and $\bigcup_{i \subseteq P} Z_i = S$. The set cover problem aims to identify a collection P that satisfies a given universe set $S = \{S_1, S_2, ..., S_n\}$, a collection of subsets $\{Z_1, Z_2, ..., Z_n\}$ of S, and a size constraint K.

We create a one-to-one mapping that links each S_i in S to a base station b_i. Similarly, we create a one-to-one mapping that associates an edge server's coverage area with the subset Z_j of

S. This coverage region only has one edge server and the edge server there will be given the relevant b_i for every u_i in Z_j. The total number of base stations $|B|$ is the collection size constraint K. We observe that a possible cover of the universe set S is also a solution to the edge server placement problem. The optimal edge server placement problem is NP-hard because the set cover problem is NP-complete [48]. The placement of the ideal edge servers is an NP-hard problem.

3.2. Problem formulation

The connectivity between base stations and edge servers can be represented by an undirected graph. Suppose we represent this graph as $G = (E, N)$, where $N = \bigcup(S, B)$, $E = (E_1, E_2, ..., E_n)$ shows the set of connections between each server and the corresponding base stations. $S = (S_1, S_2, ..., S_k)$ and $B = (B_1, B_2, ..., B_n)$ represent the set of servers and the set of base stations, respectively.

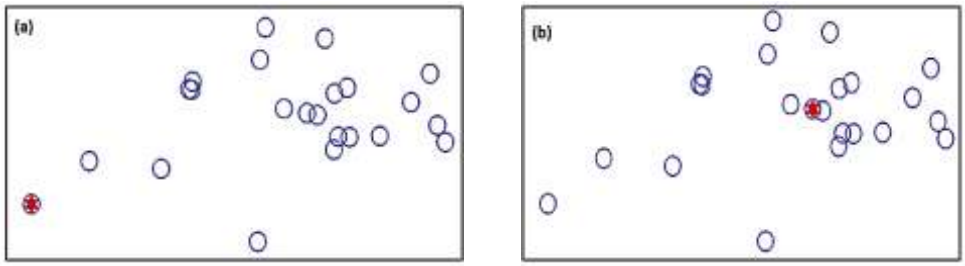
Each $E_i = \{e_{ij} \mid i = 1, 2, ..., n\ j = 1, 2, ..., k\ e_{ij} \in \{0, 1\}\}$ represents the connection of the base stations i to the edge server j. If $e_{ij} = 1$, it denotes that the connection between base station i and edge server j exists, while the absence of such a connection is implied otherwise. The representation of the deployment of individual edge servers on base stations can be denoted as $A = \{a_{ij} \mid i = 1, 2, ..., n\ j = 1, 2, ..., k\ a_{ij} \in \{0, 1\}\}$, if $a_{ij} = 1$, it means that server j is located on base station i. Table 1 presents a comprehensive list of all symbols.

Table 1. Definitions and notations.

Notation	Definition
K	Edge servers, $(K = 1, 2, ...)$
N	Base stations, $(N = 1, 2, ...)$
s_j	Edge server j, $(j = 1, 2, ..., K)$
b_i	Base station i, $(i = 1, 2, ..., N)$
S	Edge servers, $S = (s_1, s_2, ..., s_k)$
B	Base stations, $B = (b_1, b_2, ..., b_n)$
e_{ij}	edge server j and base station i connection, $e_{ij} \in \{0,1\}$
a_{ij}	Allocating edge server j on the base station i, $a_{ij} \in \{0,1\}$
W_i	The workload of the edge server i
$\overline{W}$	The average workload of edge servers
$d(b_i, b_j)$	Euclidean distance between base stations b_i and b_j

3.2.1. Latency

Latency is computed by factoring in the distance between the edge servers and the associated base stations. When a group of base stations is chosen for connection to an edge server, each of these base stations may be considered a potential candidate for an edge server deployment. Deploying the edge server on a base station with the minimum sum of distances to other base stations within the corresponding network segment would yield the minimum latency. In this paper, we will use this idea, and the server of each subnet will be placed on a base station that has the least distance and latency. Figure 2 shows an example of this server deployment on appropriate base stations.



(a) (b)

Figure 2. Edge server placement on allocated base stations. (a) Server placement without minimum distance, (b) Server Placement with minimum distance.

In Figure 2(a), the idea of minimum distance is not considered, while in Figure 2(b), the minimum distance of each base station to related base stations is considered. As can be seen in the figures, Figure 2(b) provides a better distribution for the server. To minimize the latency in the whole MEC network, we use the Equation 1 as an objective.

$$L = \min(\sum_{j=1}^{k}(\min \sum_{i=1}^{n} d(b_i, b_j))) \tag{1}$$

The variable d represents the Euclidean distance between two base stations. k denotes the number of edge servers and n represents the number of base stations.

3.2.2. Workload Balance

One of the most important objectives in optimizing the placement of edge servers is to balance the workload across the provisioned servers. If latency reduction is taken into account, but workload balancing is ignored, this may result in some servers remaining idle and other servers being heavily loaded. In this case, not only does the latency of responding to user requests increase, but the efficiency of the system decreases, causing higher power and cost consumption. When the workload is balanced, the quality of service for users increases.

In previous studies, the total workload or the average workload on the servers was used as an objective, but in this paper, minimizing the variance of the workload was considered as an objective. As the objective value decreases, the workload disparity among servers decreases, resulting in an improved workload distribution. The Equation 2 illustrates this objective.

$$WV = \frac{1}{N+1}(\sum_{i=1}^{N}(W_i + \overline{W})^2) \tag{2}$$

where the N denotes the number of servers, W_i refers to the workload of each individual server and $\overline{W}$ represents the average workload of all servers.

3.2.3. The statement of the problem

Mobile edge computing involves the deployment of resources by edge providers at the periphery of the network, with the aim of providing services to end-users. Reducing latency is one of the most important objectives in the edge server placement problem, which should be considered when implementing MEC. On the other hand, balancing the workload of servers can not only increase the quality of services for users but also reduce costs and energy consumption. Consequently, the implementation of an optimization problem that satisfies two desirable yet incompatible objectives can offer a viable solution for the deployment of MEC. In consideration of the above description, the multi-objective problem of edge server placement can be summarized by the following equations.

$$Minimize \quad F = (L, WV) \tag{3}$$

Subject to:

$$\sum_{j=1}^{k} x_{ij} = 1 \quad (\forall i \in B), x_{ij} \in \{0,1\}, (i \in B, j \in ES) \tag{4}$$

$$\sum_{i=1}^{n} y_{ji} = 1 \quad (\forall j \in S), y_{ji} \in \{0,1\}, (i \in B, j \in ES) \tag{5}$$

$$\{B_1 \bigcup B_2 \bigcup ... \bigcup B_n\} = G \tag{6}$$

$$B_1 \bigcap B_2 \bigcap ... \bigcup B_n = \varnothing \tag{7}$$

$$\lambda_i(c) < \lambda_{\max} \quad 1 \le i \le |C| \tag{8}$$

$$\sum_{i=1}^{k} w_i \le W \tag{9}$$

The objective function is represented by Equation 3, which needs to be subjected to some constraints. Constraint 4 states that an edge server can only be installed on one base station, and constraint 5 limits base stations to one server. According to the constraint In the Equation 6, edge servers are responsible for processing all network users' requests and according to constraint 7. There is no crossover between base stations and users in any two subnets. Based on the constraint 8, the total number of users arriving at any given edge server is less than the maximum number of users that the edge server can support. According to constraint 9, the workload assigned to each edge server is within its capacity limit.

3.3. Base station assignment encoding/decoding

In solving this problem, a $k \times n$ matrix is defined, where the variable n denotes the number of base stations and k is the count of edge servers. If $P = \{ p_{ij} \,|\, 1 \le i \le k, 1 \le j \le n, p_{ij} \in \{0,1\} \}$ is the desired matrix. $p_{ij} = 1$ signifies that the edge server i is allocated to the base station j, and if $p_{ij} = 0$, it means that the base station j is not assigned to the edge server i. Since each base station may be assigned to only one edge server, each column of this matrix will have only one entry with a value of 1, and the other entries will be zero. After the base stations of each edge server are determined, the latency objective calculation section assigns the edge server to the base station that incurs the lowest latency relative to the other connected base stations. Since metaheuristic methods based on NSGA-II were used to solve this optimization problem, and each solution is in the form of a $k \times n$ matrix, this matrix cannot be used as a chromosome. To apply the mutation and crossover operators, the desired matrix is first encoded into a linear vector of numerical values between 1 and k, and after applying the mutation and crossover operators, the new chromosomes are decoded into $k \times n$ matrices to calculate the objectives and select the optimal generation. In Algorithm 1, we demonstrated the algorithm of the proposed method, which is based on the NSGA-II.

ALGORITHM 1: NSGA-II BASED MOO FOR EDGE SERVER PLACEMENT AND ALLOCATION

Input: *Base Stations, Edge Servers, Population Size(N), Crossover Percentage(P_c), Mutation Percentage(P_m)*

Output: *Pareto Front*

Initialization

Pop ← Create N individuals randomly based on base stations and edge servers

Evaluate each individual based on objectives

Pop ← Sort Pop using the non-dominated sorting algorithm

While not stop criteria do

Crossover

Popc ← Select P_c individuals using binary tournament selection

Encode placement and allocation matrix and do crossover for each individual in Popc

Decode placement and allocation matrix for each individual in Popc

Evaluate Popc based on objectives

Mutation

Popm ←Select P_m individuals using binary tournament selection

Encode placement and allocation matrix and do mutation for each individual in Popm

Decode placement and allocation matrix for each individual in Popm

Evaluate Popm based on objectives

Pop ← Union(Pop, Popc, Popm)

Pop ← Sort Pop using the non-dominated sorting algorithm

Pop ← Calculate crowding distance between Pop individuals

Pop ← Sort Pop based on crowding distance

F1 ← The first frontier members as Pareto frontier

Pop ←The first N individuals of Pop

Return F1 as Pareto Front

4. Experiments Evaluations

To demonstrate the efficiency of the NSGA-II, we evaluated both the latency and workload variance objectives. Initially, a comprehensive assessment was conducted on each of the objectives separately. Afterward, we conducted an examination to determine the necessity of addressing the issue through a multi-objective approach. The subsequent section outlines the aforementioned assessments and their corresponding outcomes.

4.1. Multi-Objective Evaluation

In real optimization problems, optimizing one objective without considering other objectives may not lead to the expected results. Optimizing latency in the edge server placement problem without taking server load balancing into account may leave some servers idle and other servers overloaded. Uneven load balancing on servers eventually results in more requests being sent to the cloud server and higher latency. This degrades the quality of customer service and increases customer dissatisfaction. A more practical and appealing approach might be offered by multi-objective optimization, in which all objectives are equally optimized. A significant portion of the articles we reviewed defined a fitness function by weighting objectives and then made an effort to optimize the defined fitness function using optimization algorithms. Since the fitness function is defined linearly in these methods, many solution points cannot be reached by these optimization methods. Another problem with these methods is that no special criterion is taken into account in the weighting of the objectives.

This study implements a multi-objective optimization method based on NSGA-II. Based on the base stations assigned to the servers, latency objective optimization leads to a severely unbalanced distribution. On the other hand, because workload variance objective optimization ignores latency and attempts to equalize the load on the servers, it results in a very balanced distribution of base stations on the servers. This distribution is ideal but far from reality. Since the NSGA-II based algorithm takes into account the tradeoff between the objectives, it leads to a more acceptable solution than the other two methods. These results are depicted in Figure 3. The figure illustrates that the NSGA-II-Based method creates a middle ground between latency optimization and workload variance optimization that is closer to real-world realities.

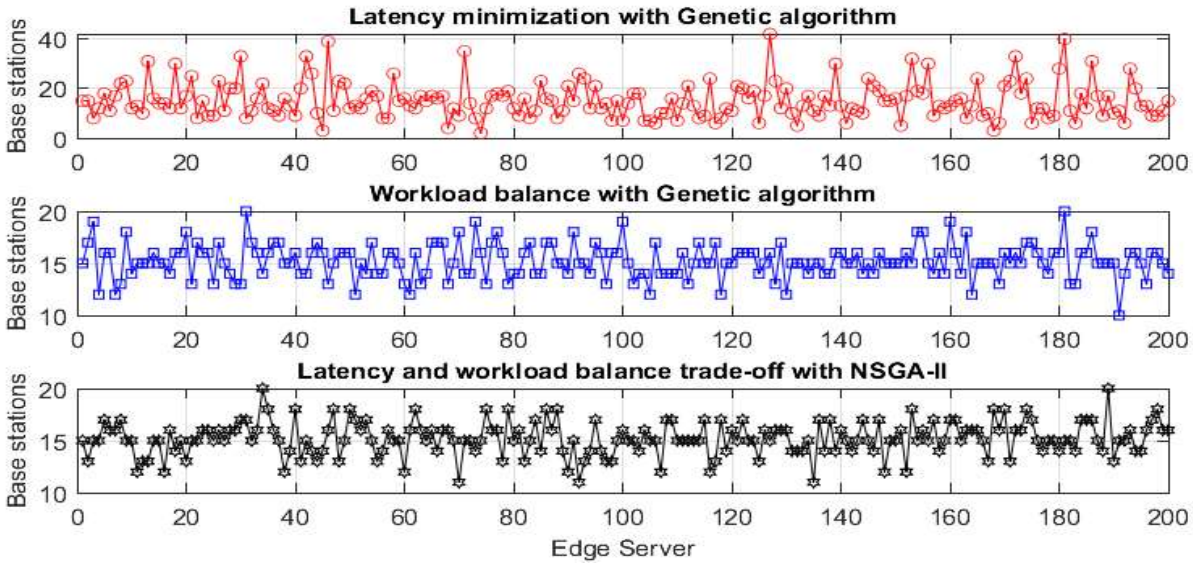


Figure 3. Base station distribution between edge servers.

As observed in the figure, during the optimization of the latency objective, certain servers are given fewer than five base stations. As a result, some servers remain idle while others are heavily loaded. The workload balance optimization entails a range of 10 to 20 base stations, and a very normal distribution is observed. This technique is also undesirable because it can result in long latency in responding to user queries and lowering QoS. In the tradeoff between the two objectives of latency and workload balance, it is clear that, in addition to minimizing latency, base station assignment to servers was done fairly. These investigations indicate that the problem of placing and allocating edge servers in the realm of MEC necessitates the implementation of multi-objective optimization techniques.

The NSGA-II based algorithm was used in 5000 iterations for 100, 200, and 300 servers to evaluate the tradeoff between the objectives, and the Pareto front points were obtained, as shown in Figure 4.

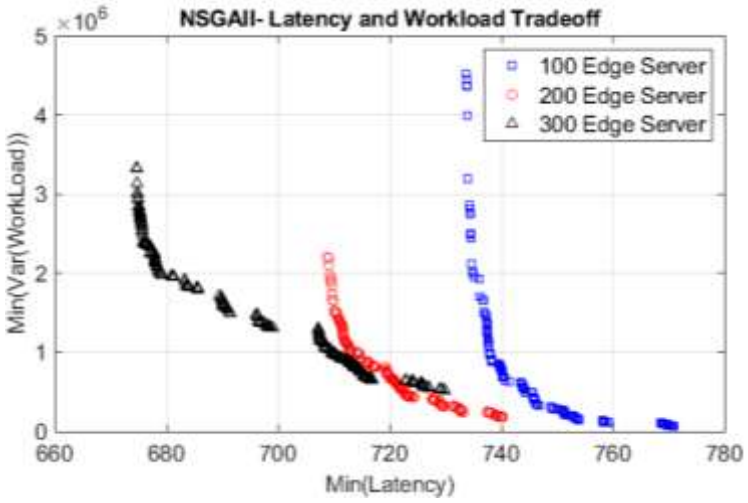


Figure 4. Pareto fronts in a tradeoff between objectives with NSGA-II based algorithm for 100, 200, and 300 edge servers.

As depicted in the figure, the number of servers has an inverse relationship with latency, where an increase in the first results in a decrease in the second. The Pareto front points for 300, 200, and 100 edge servers are represented by the black, red, and blue points, respectively. The convergence of the problem and the diversity of Pareto front points are depicted in Figure 4.

4.2. Multi-objective Performance Comparisons

In order to evaluate the effectiveness of the NSGA-II algorithm for the given problem, a comparison was conducted with other multi-objective algorithms. Running these algorithms on the same data reveals that the NSGA-II based algorithm is more efficient in terms of convergence and diversity of Pareto Front points. The establishment of parameters is determined through the analysis of multiple tests and the identification of optimal algorithmic performance. As such, the resulting parameters are as follows.

Table 2. Multi-objective algorithms parameter setting.

Parameter	Value	Description
Iter	5000	Number of Iterations
nPop	350	Initial population size
nRep	100	Number of repositories in MOPSO
$\alpha 1$	0.1	Inflation lower rate in MOPSO
β	2	Leader selection pressure in MOPSO
γ	2	Deletion selection pressure in MOPSO
$\alpha 2$	0.9	Inflation upper rate in MOPSO
μ	linespace(1, 0, nPop)	Emigration rate in BBO
λ	$1-\mu$	Immigration rate in BBO
KRate	0.2	Keep the rate of habitats in BBO
PC	0.8	Probability of Crossover
PM	0.08	Probability of Mutation

The results of the NSGA-II based algorithm, MOBBO, MOPSO, and AWSM algorithms with parameters according to the table are depicted in Figure 5.

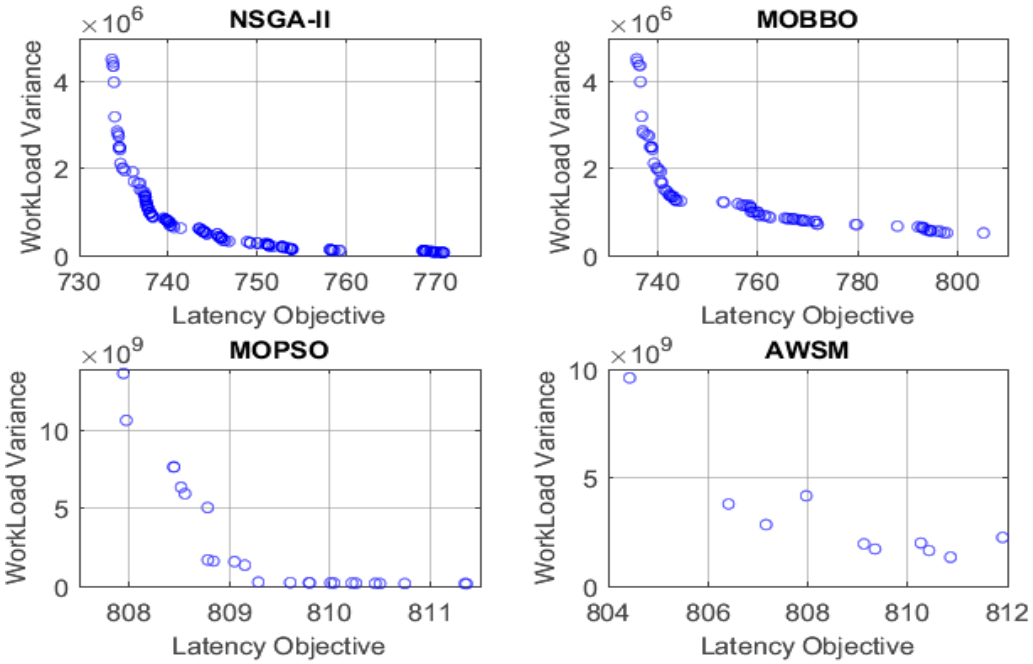


Figure 5. The results of NSGA-II based algorithm, MOPSO, MOBBO, and Adaptive-Weighted-Sum-Method.

As shown in Figure 5, the algorithm based on NSGA-II exhibited superior performance in comparison to the other algorithms. In the NSGA-II based algorithm, the Pareto front points have lower values for both objectives. The diversity of Pareto points in the MOBBO algorithm is comparable to that of the NSGA-II based algorithm and better than the diversity of Pareto points in the other two algorithms. This demonstrates that the NSGA-II based and MOBBO algorithms outperform in exploration. The NSGA-II based algorithm has better Pareto point convergence than the other algorithms, implying that it performs better in exploitation. The MOPSO algorithm performed better than the AWSM algorithm in latency objective but not in terms of workload variance. The AWSM algorithm has lower diversity of Pareto front points and lower convergence than the other algorithms.

4.3. Convergence and Diversity Evaluation

In addition to comparative results, we also evaluate our performance with well-known metrics. We used two metrics, Inverted Generational Distance (IGD) and Hypervolume (HV), to assess the diversity and convergence of the algorithms.

The IGD metric is used to evaluate the algorithm's convergence. If $\Lambda = \{y_1, y_2, ..., y_r\}$ is an approximation to the true Pareto front and $P = \{p_1, p_2, ..., p_k\}$ is the result of the multi-objective optimization algorithm, then IGD is defined by Equation 10. d represents the Euclidean distance between Λ and P in this equation. This metric determines how far the solution to the problem is from the true Pareto front. A lower IGD value indicates that the algorithm is more convergent.

$$IGD(\Lambda, P) = \frac{1}{r}(\sum_{i=1}^{r} d(y_i, P)^2)^{\frac{1}{2}} \tag{10}$$

Another important metric for evaluating multi-objective optimization algorithms is the HV metric. This metric computes the volume of objective space covered by members of nondominated solution sets obtained by a multi-objective optimization algorithm with all objectives minimized. The algorithm performs better when the HV value is high. The HV metric can be calculated as Equation 11.

$$HV = Volume(\bigcup_{i=1}^{|P_A|} V_i) \tag{11}$$

The IGD and HV values for the MOPSO, MOBBO, AWSM, and NSGA-II based algorithms are shown in Table 3. The initial population for all of these algorithms is 350, and the optimization takes 5000 iterations.

Table 3. Multi-objective optimization algorithms results.

Optimization Algorithm	**NSGA-II based**	**MOPSO**	**MOBBO**	**AWSM**
Max Workload Variance	$4.5193 * 10^6$	$1.3728 * 10^{10}$	$4.5313 * 10^6$	$9.6009 * 10^9$
Min Workload Variance	$6.5886 * 10^4$	$1.5159 * 10^8$	$5.1731 * 10^5$	$2.2476 * 10^9$
Max Latency	770.9193	811.3639	805.2257	811.9054
Min Latency	733.6813	807.9429	735.7398	804.4314
IGD	**65.89**	1515.90	517.31	13343.43
HV	**0.1779**	0.0932	0.1756	0.0883

According to the data presented in Table 3, the algorithm based on NSGA-II exhibited superior performance compared to the other three methods in the tradeoff between objectives, obtaining a lower value for both objectives. The NSGA-II based algorithm exhibits a significantly lower IGD value compared to the other three algorithms, which indicates that the algorithm based on NSGA-II exhibits better convergence. For the algorithm based on NSGA-II, the HV metric is larger. A higher HV value indicates that the algorithm is more effective and that the Pareto front point diversity is greater. The AWSM algorithm has lower convergence and diversity than the other three algorithms, and the values obtained for the objectives are higher in this algorithm.

5. Conclusion

The majority of studies on edge server placement and allocation have predominantly employed either a single objective or developed a fitness function for the optimization problem through the application of a weighted sum method. Consequently, such approaches inadequately encapsulate the inherent tradeoffs between the objectives comprehensively. In this study, we attempted to ascertain a more pragmatic resolution to the problem of placing and allocating edge servers by focusing on the application of each objective and then studying the tradeoff between them. According to the results, tackling the problem with a multi-objective optimization method can provide a more realistic solution from the perspective of MEC providers and end users. Based

on the results of the numerous tests, it seems that each of the objectives identified may have a substantial influence on improving the placement and allocation of edge servers. An NSGA-II based algorithm was used to examine the issue in a multi-objective method to highlight the tradeoff between the objectives, and it was discovered that this technique generates not only a wider variety of Pareto front points but also a higher convergence than the MOBBO, MOPSO, and weighting methods. The suggested technique greatly decreases workload variance on the servers and has an average of 8.79% smaller latency than the weighted-sum method. It is possible to fulfill the demands of the users and the desired objectives of the service providers in the edge server placement and allocation problem by studying and examining objectives such as energy consumption reduction, cost reduction, profit increase, and quality of service optimization, and using these objectives in a many-objective optimization problem.

References

[1] B. Shen, X. Xu, L. Qi, X. Zhang, and G. Srivastava, "Dynamic server placement in edge computing toward internet of vehicles," *Computer Communications,* vol. 178, pp. 114-123, 2021.

[2] B. Bahrami, M. R. Khayyambashi, and S. Mirjalili, "Edge server placement problem in multi-access edge computing environment: models, techniques, and applications," *Cluster Computing,* pp. 1-26, 2023.

[3] E. Del-Pozo-Puñal, F. García-Carballeira, and D. Camarmas-Alonso, "A scalable simulator for cloud, fog and edge computing platforms with mobility support," *Future Generation Computer Systems,* vol. 144, pp. 117-130, 2023.

[4] C. Jian, L. Bao, and M. Zhang, "A high-efficiency learning model for virtual machine placement in mobile edge computing," *Cluster Computing,* vol. 25, no. 5, pp. 3051-3066, 2022.

[5] C. Ding, A. Zhou, Y. Liu, R. N. Chang, C.-H. Hsu, and S. Wang, "A cloud-edge collaboration framework for cognitive service," *IEEE Internet of Things Journal,* vol. 10, no. 3, pp. 1489-1499, 2020.

[6] C. Ding, A. Zhou, X. Ma, and S. Wang, "Cognitive service in mobile edge computing," in *2020 IEEE International Conference on Web Services (ICWS)*, 2020, pp. 181-188: IEEE.

[7] S. Deng *et al.*, "Optimal application deployment in resource constrained distributed edges," *IEEE transactions on mobile computing,* vol. 20, no. 5, pp. 1907-1923, 2020.

[8] S. Deng, C. Zhang, C. Li, J. Yin, S. Dustdar, and A. Y. Zomaya, "Burst load evacuation based on dispatching and scheduling in distributed edge networks," *IEEE Transactions on Parallel Distributed Systems,* vol. 32, no. 8, pp. 1918-1932, 2021.

[9] 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, 2021.

[10] H. Zhao, S. Deng, Z. Liu, J. Yin, and S. Dustdar, "Distributed redundant placement for microservice-based applications at the edge," *IEEE Transactions on Services Computing,* vol. 15, no. 3, pp. 1732-1745, 1 May-June 2022 2019.

[11] H. Mehmood, A. Khalid, P. Kostakos, E. Gilman, and S. Pirttikangas, "A novel Edge architecture and solution for detecting concept drift in smart environments," *Future Generation Computer Systems,* 2023.

[12] W. Wei, H. Li, and W. Yang, "Cost-effective stochastic resource placement in edge clouds with horizontal and vertical sharing," *Future Generation Computer Systems,* vol. 138, pp. 213-225, 2023.
[13] J. Lu, J. Jiang, V. Balasubramanian, M. R. Khosravi, and X. Xu, "Deep reinforcement learning-based multi-objective edge server placement in Internet of Vehicles," *Computer Communications,* vol. 187, pp. 172-180, 2022.
[14] Y. Chen, D. Wang, N. Wu, and Z. Xiang, "Mobility-aware edge server placement for mobile edge computing," *Computer Communications,* vol. 208, pp. 136-146, 2023.
[15] Q. Zhang, S. Wang, A. Zhou, and X. Ma, "Cost-aware edge server placement," *International Journal of Web Grid Services,* vol. 18, no. 1, pp. 83-98, 2022.
[16] K. Xiao, Z. Gao, Q. Wang, and Y. Yang, "A heuristic algorithm based on resource requirements forecasting for server placement in edge computing," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018, pp. 354-355: IEEE.
[17] Y.-w. Zhang, W.-m. Zhang, K. Peng, D.-c. Yan, and Q.-l. Wu, "A novel edge server selection method based on combined genetic algorithm and simulated annealing algorithm," *Automatika,* vol. 62, no. 1, pp. 32-43, 2021.
[18] 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 Applications,* vol. 14, pp. 3778-3792, 2021.
[19] G. Manasvi, A. Chakraborty, and B. Manoj, "Social network aware dynamic edge server placement for next-generation cellular networks," in *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2020, pp. 499-502: IEEE.
[20] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel Distributed Computing,* vol. 127, pp. 160-168, 2019.
[21] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Transactions on Mobile Computing,* vol. 18, no. 2, pp. 319-333, 2018.
[22] F. Zeng, Y. Ren, X. Deng, and W. Li, "Cost-effective edge server placement in wireless metropolitan area networks," *Sensors,* vol. 19, no. 1, p. 32, 2018.
[23] Q. Li, S. Wang, A. Zhou, X. Ma, F. Yang, and A. X. Liu, "QoS driven task offloading with statistical guarantee in mobile edge computing," *IEEE Transactions on Mobile Computing,* vol. 21, no. 1, pp. 278-290, 2020.
[24] M. Satyanarayanan, "The emergence of edge computing," *Computer,* vol. 50, no. 1, pp. 30-39, 2017.
[25] Y. Li and S. Wang, "An energy-aware edge server placement algorithm in mobile edge computing," in *2018 IEEE International conference on edge computing (EDGE)*, 2018, pp. 66-73: IEEE.
[26] H. Yin *et al.*, "Edge provisioning with flexible server placement," *IEEE Transactions on Parallel Distributed Systems,* vol. 28, no. 4, pp. 1031-1045, 2016.
[27] S. Jamin, C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt, "Constrained mirror placement on the Internet," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, 2001, vol. 1, pp. 31-40: IEEE.
[28] B. Li, M. J. Golin, G. F. Italiano, X. Deng, and K. Sohraby, "On the optimal placement of web proxies in the internet," in *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, 1999, vol. 3, pp. 1282-1290: IEEE.
[29] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, "On the placement of web server replicas," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, 2001, vol. 3, pp. 1587-1596: IEEE.

[30] C. Huang, A. Wang, J. Li, and K. W. Ross, "Measuring and evaluating large-scale CDNs," in *ACM IMC*, 2008, vol. 8, pp. 15-29.
[31] B. Krishnamurthy, C. Wills, and Y. Zhang, "On the use and performance of content distribution networks," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, 2001, pp. 169-182.
[32] Y. Zhang, D. Li, and M. Tatipamula, "The freshman handbook: A hint for server placement in online social network services," in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, 2012, pp. 588-595: IEEE.
[33] Y. Ren, F. Zeng, W. Li, and L. Meng, "A low-cost edge server placement strategy in wireless metropolitan area networks," in *2018 27Th International conference on computer communication and networks (ICCCN)*, 2018, pp. 1-6: IEEE.
[34] A. Santoyo-González and C. Cervelló-Pastor, "Edge nodes infrastructure placement parameters for 5G networks," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2018, pp. 1-6: IEEE.
[35] G. Cui, Q. He, X. Xia, F. Chen, H. Jin, and Y. Yang, "Robustness-oriented k edge server placement," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, 2020, pp. 81-90: IEEE.
[36] X. Ma, S. Wang, S. Zhang, P. Yang, C. Lin, and X. J. I. T. o. C. C. Shen, "Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing," vol. 9, no. 3, pp. 968-980, 2019.
[37] F. Wang, X. Huang, H. Nian, Q. He, Y. Yang, and C. Zhang, "Cost-effective edge server placement in edge computing," in *Proceedings of the 2019 5th international conference on systems, control and Communications*, 2019, pp. 6-10.
[38] X. Xu *et al.*, "Load-aware edge server placement for mobile edge computing in 5G networks," in *Service-Oriented Computing: 17th International Conference, ICSOC 2019, Toulouse, France, October 28–31, 2019, Proceedings 17*, 2019, pp. 494-507: Springer.
[39] S. K. Kasi *et al.*, "Heuristic edge server placement in industrial internet of things and cellular networks," vol. 8, no. 13, pp. 10308-10317, 2020.
[40] T. Lähderanta *et al.*, "Edge computing server placement with capacitated location allocation," vol. 153, pp. 130-149, 2021.
[41] L. Zhao and J. J. I. T. o. V. T. Liu, "Optimal placement of virtual machines for supporting multiple applications in mobile edge networks," vol. 67, no. 7, pp. 6533-6545, 2018.
[42] J. Meng, C. Zeng, H. Tan, Z. Li, B. Li, and X.-Y. Li, "Joint heterogeneous server placement and application configuration in edge computing," in *2019 IEEE 25Th International conference on parallel and distributed systems (ICPADS)*, 2019, pp. 488-497: IEEE.
[43] L. Lovén *et al.*, "Scaling up an edge server deployment," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1-7: IEEE.
[44] Z. Hu, X. Xu, and J. Chen, "An Edge Server Placement Algorithm based on Genetic Algorithm," in *ACM Turing Award Celebration Conference-China (ACM TURC 2021)*, 2021, pp. 92-97.
[45] Z. Wang, W. Zhang, X. Jin, Y. Huang, and C. Lu, "An optimal edge server placement approach for cost reduction and load balancing in intelligent manufacturing," *The Journal of Supercomputing,* vol. 78, no. 3, pp. 4032-4056, 2022.
[46] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C. H. Hsu, "User allocation-aware edge cloud placement in mobile edge computing," *Software: Practice Experience,* vol. 50, no. 5, pp. 489-502, 2020.
[47] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. J. I. T. o. M. C. Shen, "Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach," vol. 20, no. 3, pp. 939-951, 2019.
[48] R. M. Karp, *Reducibility among combinatorial problems*. Springer, 2010.