Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Metaheuristics Method for Computation Offloading in Mobile Edge Computing: Survey

Safaa Alali[1,*], Abdulkarim Assalem[2]

1 Communication Engineering, Faculty of Electrical and Mechanical Engineering, Al-Baath University, Homs, Syria
2 Department of Electronic and Communication Engineering, Faculty of Electrical and Mechanical Engineering, Al-Baath University, Homs, Syria

ARTICLE INFO

Article history:
Received 7 November 2023
Received in revised form 16 December 2023
Accepted 17 December 2023
Available online 25 December 2023

Keywords:
Computation offloading; Mobile edge computing; Genetic algorithm; Particle swarm optimization

ABSTRACT

In recent years, edge computing has emerged as a computing paradigm to support the computationally intensive and latency-critical applications for resource limited devices. The main feature of edge computing is to push computation, networking, and storage facilities closer to the network edge. This enables user equipment (UE) to profit from the edge computing paradigm by mainly offloading their intensive computation tasks to edge resources. Because of some rising problems such as inherent software and hardware heterogeneity, restrictions, dynamism, and stochastic behaviour of the ecosystem, the computation offloading issues consider as the essential challenging problems in the MEC environment. However, to the best of the author's knowledge, in spite of its significance, in Metaheuristic -based computation offloading mechanisms, there is not any systematic, comprehensive, and detailed survey in the MEC environment. In this paper, we provide a review on the Metaheuristic -based computation offloading mechanisms in the MEC environment in the form of a classical taxonomy to identify the contemporary mechanisms on this crucial topic and to offer open issues as well. The proposed taxonomy is classified into three main fields: genetic algorithm (GA)-based mechanisms, Partical Swarm Optimization (PSO)-based mechanisms, and Hybrid GA with PSO -based mechanisms. Next, these classes are compared with each other based on the essential features such as performance metrics, case studies, utilized techniques, and evaluation tools, and their advantages and weaknesses are discussed, as well. Finally, open issues and uncovered or inadequately covered future research challenges are argued, and the survey is concluded.

1. Introduction

Edge offloading is a strategy to transfer computations from the resource-limited mobile device to resource-rich cloud nodes to run resource-hungry applications which demand low latency and high data rates, capabilities of the far remote cloud has been changing with closer innovative technologies such as Mobile Edge Computing (MEC) with minor restrictions for specific demanding applications [1].

* *Corresponding author.*
E-mail address: En.safaa.alali@gmail.com

https://doi.org/10.37934/araset.36.1.4373

MEC is established at the possible closest location to the mobile devices, with moderate server's capabilities placed at the edge of the network to achieve necessary user-centric requirements and his applications such as: Internet of things (IOT) applications [64], and Augmented Reality (AR) applications which are getting more widely applied in various fields such as education, art, manufacturing field and entertainment [65,69], and e-learning system which is a helpful tool in a learning process [66] and etc.

Due to some inherent limitations of MEC, including storage, bandwidth, and CPU, practical, well-organized resource management is necessary to make edge computation a real useful mechanism [3].

Mainly, the idea of offloading in MEC is executing resource-demanding applications, on behalf of local mobile devices, aimed to alleviate the burden of the work and decrease the computation overhead and costs compared with local execution. Both mobile devices and MEC servers have to necessarily operate offloading frameworks to fulfil computation offloading [53]. Many technical papers consider the subject intensely to recommend new methods of achieving the goals in the offloading criteria. These technical papers mostly proposed their methods based on game theory, 0-1 integer linear programming problem as seen in [7,8], K-dimensional bin parking as seen in [9], a Markov decision process as seen in [10,11], and Lyapunov optimization as seen [12], machine learning, heuristic-based, metaheuristics based, or a hybrid form of mentioned techniques. In this study, we focus on metaheuristics-based techniques, which are suitable for the dynamic behaviour of Mobile Equipment (MEs).

Finally, to the best of the authors' knowledge, in spite of its significance, there is not any survey and review in terms of offloading covering the metaheuristics -based mechanisms, identifying the requirement of researchers to fulfil the work in the mentioned field. Therefore, this survey aims at reviewing the present researches in the MEC paradigms, which depend on metaheuristics -based approach.

Briefly, the main contributions of this review are as follows:

i. Reviewing some survey articles related to offloading mechanisms in MEC and presenting the advantages and the weaknesses for each one;
ii. Exploring the latest metaheuristics -based approaches in the field of offloading in MEC;
iii. Providing a comprehensive systematic review of current approaches and proposing a comprehensive taxonomy;
iv. Discussing future research challenges to improve computation offloading mechanisms in the MEC environment;

In this paper, the most utilized performance metrics are energy consumption and delay, we didn't consider some issues which are very important specially for internet of things system such as methods of threats and intrusion that could cause security breaches [67], and the cybercrime that is probable to effect on network response, recovery, and management [68], and Availability which is important property of information systems, especially in critical infrastructure and revenue-generating systems [64].

The rest of this paper is organized as follows: Section 2 provides the necessary background of offloading issues in mobile edge computing and metaheuristics methods. In Section 3, we review some essential related works. The research methodology is provided in Section 4. Section 5 discusses metaheuristics -based offloading approaches in the mobile edge computing and classifies them, also provides the taxonomy and comparison of the discussed techniques. The comparison and a discussion of the reviewed techniques are presented in Section 6. Also, Section 7 discusses some critical open issues as future work. Finally, we present the conclusions in Section 8.

2. Background

2.1 Computational Offloading in MEC

The offloading process responsibility is divided among three main agents: mobile devices, communication links, and EC servers. Specifically, mobile devices are responsible for determining how an application is partitioned, which parts should be executed locally or remotely, and the offloading scheme. The communication link is influenced by fluctuation of bandwidth, connectivity, and device mobility. EC servers handle the balance of the server load to achieve maximum service rates and system throughput [57]. The MEC structure has become one of the most attractive architectures in the literature of computation. The European Telecommunications Standards Institute (ETSI) had presented the first standard for Mobile Edge Computing, and later it has been changed to Multi-access Edge Computing [55]. The standard structure of the Mobile Edge Computing system is illustrated in Figure 1. As it is illustrated, this structure has two main layers:

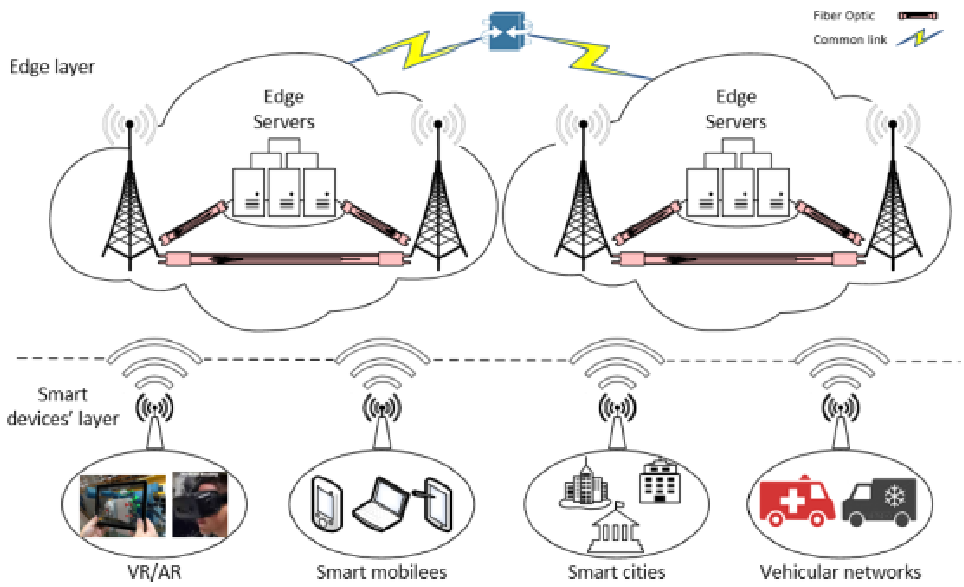

Fig. 1. Standard structure of the Mobile Edge Computing system

i. At the smart devices' layer, there are heterogeneous mobile devices concerning storage, processing, and interface capabilities. These mobile devices can exchange information with the other nearby smart devices and the adjacent access points of the edge environment. The first place to decide whether to offload the particular tasks to the remote edge environments can be fulfilled in this layer.
ii. At the edge layer, the APs and the edge servers with moderately small data centres are located. These servers are typically accessed via high data rate powerful communication links by a colony of geographically scattered APs [56]. The APs themselves are usually interconnected via fibre optic.

2.2 Metaheuristics Methods

metaheuristics methods are generally inspired by nature. The main idea of these approaches is improving the result in a reasonable time through an iterative process of searching for better solutions while trying to avoid getting stuck in local optima, unlike heuristic approaches that are prone to this problem. A number of metaheuristics techniques have been proposed in the literature, such as genetic algorithms (GAs) [2], and particle swarm optimization (PSO) [4]. These algorithms are typically based on the idea of population (solution) evolution, in which the best solutions for a given objective are usually preserved for the next evolutionary step of obtaining a new generation of solutions with a hope of getting a fitter population [5].

2.2.1 Particle Swarm Optimization (PSO)

The term "Swarm Intelligence'' is used to describe algorithms inspired by the collective behaviour of colonies and other animal societies [59]. The particle swarm optimization (PSO) is then an algorithm based on self-organizing systems 'collective and decentralized behaviour. Thus, the PSO simulates a flock of birds or a shoal of fish in search of food. It treats each solution to the optimization problem as a bird that flies at a certain speed in the search space, and its speed is dynamically adjusted [4]. So, a swarm solution is a particle in a multidimensional search space.

Consider that a solution is a vector of D positions, where each position represents a request and its value is its destination, that is, the address of the MEC or Cloud server. Each particle has a flight acceleration that determines its direction and speed, so they move within the search space at speed adjusted to each iteration according to cognitive and social factors. the population is initialized at random through a linear distribution. The solution, once created, is evaluated using the 1 equation. After each particle has a fitness level, the particles move in the search space, looking for better solutions. With each movement that is performed by the particles, their fitness level is updated. The particle's motion is described by the Eq. (1).

$$Particle_i(t + 1) = Particle_i(t) + V_i(t + 1) \tag{1}$$

$Particle_i$is an element of a set of solutions. $Particle_i$is the same as$C_i V_i$is the speed at which the particle moves in the search space, at time t. V is presented in the Eq. (2).

$$V_i(t+1) = W(t) + V_i(t) + C_1(t) \times rand_1 \times \left(Pbest(t) - Particle_i(t)\right) + C_2(t) \times rand_2 \times \left(Gbest(t) - Particle_i(t)\right) \tag{2}$$

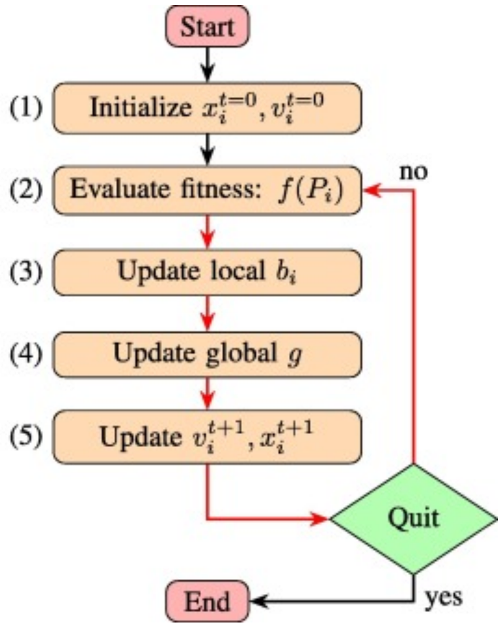

Fig. 2. Flow chart of (PSO) algorithm

Wt is inertia at time t, that is, the tendency of the particle to continue in the same direction. C_1 is the cognitive factor, the tendency of the particle to move from past learning. C_2 is the social factor, the tendency of the particle to move from the learning obtained with the neighboring particles.

$rand_1$ and $rand_2$ assume a random value between 0 and 1. The first step of the update is to add the velocity vector to the particle vector, updating each of the i elements within the search space. After that, the elements that are outside the sample space undergo a correction to make the solution valid. After the solutions are updated, they are evaluated again until the stopping criterion is satisfied, either the number of moves or a determined fitness level [60].

2.2.2 Genetic algorithm (GA)

A Genetic Algorithm is a meta-heuristic for solving optimization problems inspired by the process of natural selection. In the algorithm, a population of possible solutions, called chromosomes, evolves within the optimization problem's domains towards an optimal solution [58]. Each individual corresponds to a chromosome, which is the coded representation of the solution. Each chromosome can be represented by a vector, with values representing the problem domain. The GA is an iterative process where a set of processes, called generation, creates a new population through the random recombination and mutation of selected individuals from the current population to generate the next population. Individuals are stochastically selected, with those with better fitness being favoured over those with lower fitness. Generally, the evolution process ends up reaching a certain number of generations or finding a satisfactory fitness level. The first step of GA is initializing of the population, which is a set of vectors with random values. Each vector generated represents a chromosome. The population is randomly initialized using a linear distribution.

After initializing generations, the objective function evaluates each individual in the population. Each of the subsequent generations, called daughter generations, is created by natural selection methods, whereas the previous generation, called the parent generation, uses the selection, crossover, and mutation operators. This process is carried out until the predetermined stopping criterion is satisfied. The parents must be chosen from the current population to generate the child population in each generation. The proposed algorithm uses the roulette method, where individuals from a generation are chosen through a roulette drawing. In this method, each population member is represented on the roulette wheel according to their fitness level. Thus, individuals with high fitness are given a more significant portion of the roulette wheel. In contrast, those with lower fitness are given a relatively minor portion of the roulette wheel. Finally, the roulette wheel is spun a certain number of times, depending on the size of the population, and those drawn on the roulette wheel are chosen as parent individuals.

Once the parents are selected, the crossover process can be performed with a probability rate called the crossover rate. A crossing point is chosen, and from this point, the genetic information of the parents will be exchanged. Information prior to this point in one parent is linked to information after this point in the other parent, generating a new individual. After crossover, the mutation occurs. The mutation operator is necessary for the introduction and maintenance of the genetic diversity of the population, arbitrarily altering one of the genes of the chosen individual, thus providing means for introducing new elements into the population. In this way, the mutation ensures that the probability of reaching any point in the search space will never be zero, in addition to circumventing the problem of local minima since, with this mechanism, the direction of the search is slightly altered. The mutation operator is applied to individuals with a probability named mutation rate.

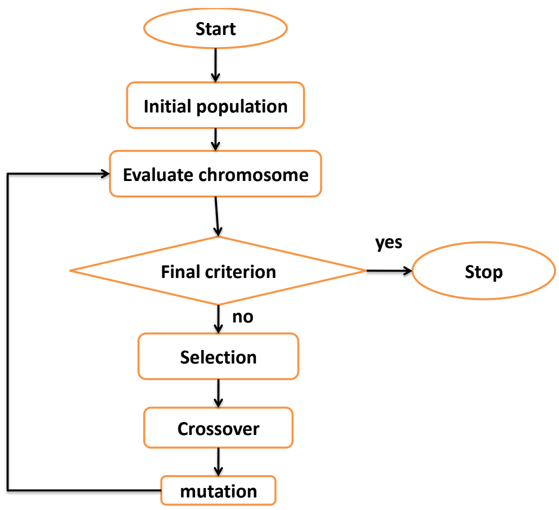

Fig. 3. Flow chart of (GA) algorithm

2.2.3 Integrating (GA) with (PSO)

The hybrid method is combining two heuristic optimization techniques, PSO and GA. The proposed algorithm integrates the concept of evolving individuals originally modelled by GA with the concept of self-improvement of PSO, where, the algorithm initialized by a set of a individuals which travel in the search space using the PSO. During this travel we implement GA to evolve these individuals. Also, in order to keep the feasibility of the particles, an additional parameter is introduced, where the algorithm co-evolves the population of infeasible individuals until they become feasible.

W.F. Abd-El-Wahed *et al.,* [61] presented a result of various experimental studies using a suite of multimodal test functions taken from the literature which demonstrated the superiority of the hybrid approach to finding the global optimal solution.

2.2.4 Metaheuristics based offloading

Because of the inherent complexities of wireless communication and computation technologies incurred by the dynamism of such technologies, decision-making and resource management problems over these technologies for improving the efficiency of the system and meeting the user requirements are becoming more complicated. Noteworthy, incorrect offloading decisions can degrade the efficiency of the system. Since metaheuristics are strategies that guide the search process, complex decision-making problems of offloading can be more efficient by using such approaches. Therefore, to address such problems and related challenges appropriately, metaheuristics methods are utilized in the field of offloading [24].

3. Related works

In this section, recent papers reviewed on computation offloading in MEC will be surveyed. Next, the main advantages and disadvantages of each survey paper will be given. Then we would go more in-depth to review some studied papers in the literature. Obviously, because of the professional relationship between MEC and fog computing, some technical points have the same functionality and meaning. Still, this matter doesn't cause loss of generality and integrity of either of the two technologies.

P. J. Escamilla-Ambrosio *et al.,* [1] surveyed the more developed paradigms aimed to bring computational, storage and control capabilities closer to where data is generated in the IoT: fog and edge computing, contrasted with the cloud computing paradigm. Also, an overview of some practical use cases was presented to exemplify each of these paradigms and their main differences.

Ali Shakarami *et al.,* [3] surveyed the paper concerning the stochastic-based offloading approaches in various computation environments such as Mobile Cloud Computing (MCC), Mobile Edge Computing (MEC), and Fog Computing (FC) in which to identify new mechanisms. The proposed taxonomy was classified into three main fields: Markov chain, Markov process, and Hidden Markov Models.

SAEIK Firdose *et al.,* [53] provides a detailed survey of how the Edge and/or Cloud can be combined together to facilitate the task offloading problem, with emphasizing the mathematical, artificial intelligence and control theory optimization approaches that can be used to satisfy the various objectives, constraints and dynamic conditions of this end-to-end application execution approach.

JIANYU WANG *et al.,* [57] surveyed the key issues, methods, and various state-of-the-art efforts related to the offloading problem in the edge cloud framework. He adopted a new characterizing model to study the whole process of offloading from mobile devices to the edge cloud, which consists of the basic categorizing criteria of offloading destination, load balance, mobility, partitioning, and granularity. The factors of algorithms such as environment constraints, cost models, user configuration, and mathematical principles were discussed in detail. This survey was introduced an integrated offloading system of an edge as balanced combination of these five perspectives to achieve low latency and better energy efficiency at each step of computation offloading. However, this survey suffers from the lack of recently published articles in the related field (i.e., about 7% published in 2018, 4 out of 54). It is also neglected some essential factors such as fault tolerance and security issues in the system that, in their turn, directly affect the overall system efficiency. This degradation can mislead the user's points of view, which are QoS, QoE. The paper does not have a systematic format to select papers, too.

Also, Mach *et al.,* [63] described some use cases, functionality, standardization, and computation offloading in MEC environments. As an advantage, the survey is constructed by reviewing plenty of MEC related researches and articles, of which 23% are newly published at the time of acceptance (i.e., published in 2016 – 29 out of 124). However, the article is not well organized in some respects, such as the proposed granularity. Also, some essential subjects and techniques are not covered in the literature. As another drawback, some aspects are simplified that lead to loss of generality. The paper does not have a systematic format to select papers, too. It is also worth mentioning that future directions are not well covered in this paper.

In another survey, Boukerche *et al.,* [30] studied energy-awareness in Mobile Cloud Computing (MCC) and reviewed protocols, architecture, scheduling and balancing algorithms in the field of MCC and Green Cloud Computing strategies. Next, the advantages and disadvantages of those researches in terms of offloading process and resource management types are compared and categorized. The strength of this article is its professional review on the subject that has been well organized addressing the energy-aware issues. However, because of the scarcity of standards in the subject, making strict borderlines between emerging technologies such as MCC and MEC and giving the exact meaning and definition for each of them is a tough job, and this is precisely one of the weak points of this review. As another instance, the review doesn't cover related subjects entirely. Also, in spite of considering huge numbers of reviewed articles, this survey suffers from a lack of recently published articles in the reviewed field (about 5% published in 2018 – 7 out of 141-and about 8% published in 2017– 11 out of 141).

Some offloading schemes in the field of computation paradigms, including Edge, Fog, Cloud, and also IoT have been reviewed by Aazam *et al.,* [62]. Next, they present a taxonomy for these paradigms. An enabling offloading technology as middleware and related factors are also discussed. As the strength, the review has been well categorized in the predefined criterion, with appropriate examples for each described criterion. It is also included a reasonable percentage of newly published articles (about 24% published in 2017 - 12 out of 51) in the subject. However, it doesn't review those researches with some essential factors such as granularity and mobility in the literature. Also, the paper does not have a systematic format to select papers.

In addition, K. Penget *et al.,* [26] surveyed some articles related to MEC in terms of architecture, service adoption, and provision. For service adoption, it is categorized computation offloading and data offloading as two essential aspects of the MEC paradigm. For service provision, Edge Server (ES) service provision and its technical indicator, ES deployment, and resource allocation are reviewed. Some other issues, like MEC applications, are also investigated in this survey. As a strength, the review is provided with reasonable newly published articles (about 30% published in 2017- 37 out of 123). However, the survey is not technically well organized. The survey also suffers from describing the literature fluently with insufficient technical explanations in each category and the predefined fields.

In [50], the authors presented a comprehensive review of game-theoretic offloading approaches in the MEC environment. They also compared various essential aspects of the literature in the form of tables and charts. As a strength, their paper is the only system antiliterature review covering the subject in a well-organized method. The survey has also included appropriate recent published papers of the related field. As a drawback, the paper is not related to machine learning approaches, which is not off course its goal.

Likewise, Bin Cao *et al.,* [48] proposed a survey on fundamental concepts in MEC with a focus on Machine Learning-based approaches and the leading applications. As a strength, the survey focuses professionally on a particular subject that does not have any background in the literature. However, this survey suffers from a lack of sufficient persuasive articles to cover the subject appropriately. As another drawback, the paper does not have a systematic format to select papers, too. Also, future directions are not enough powerful and well covered in this paper. Briefly, the previous review papers suffer from some weak points as follows:

i. These papers don't contain newly published articles in the field of metaheuristics-based offloading mechanisms in the MEC environment, specifically in 2020, 2021, 2022 and 2023.
ii. These papers do not provide helpful future directions of approaches in metaheuristics-based offloading in MEC environments.
iii. Some papers do not have a systematic format to select papers.
iv. Some papers explore the offloading approaches in MEC environments by other mechanisms except metaheuristics-based methods.

The mentioned reasons motivated us to prepare a survey paper on metaheuristics-based computation offloading mechanisms in the MEC paradigm to overcome all of these lacks.

4. Methodology

In this section, an instruction to explore appropriate papers in the MEC Offloading is described. For constructing a survey more knowledge-rich, searching, gathering, classifying, and investigating applicable papers is necessary.

4.1 Question Formalization

This survey aims at exploring significant features and methods applied in the articles in an identified time accompanied by the main problems and challenges in the Metaheuristics-based offloading methods. Since covering the complete study of MEC offloading and presenting related open issues is a significant objective of the current survey, some important research questions have to be replied to address related concerns.

i. TQ1: What classification is utilized in Metaheuristics-based offloading approaches?
ii. TQ2: What performance metrics are usually utilized in Metaheuristics-based offloading approaches?
iii. TQ3: What case studies are applied in Metaheuristics-based offloading approaches?
iv. TQ4: What evaluation tools are utilized for assessing the Metaheuristics-based approaches?
v. TQ5: What are the similarities and differences between two offloading approaches?
vi. TQ6: What is the benefit of combining the two algorithms together in the offloading scheme?
vii. TQ7: What are the future research directions of Metaheuristics-based offloading approaches?

4.2 Data Exploring and Article Selection

Suitable papers in the MEC have been explored in the accessible academic databases that result in taxonomy to categorize the subjects of the topic better. The principles of selecting articles in the process of exploring are summarized as follows:

i. Published papers between 2019 and 2023
ii. Published papers in the MEC
iii. Technical quality selection to choose appropriate papers in the MEC.

In the famous databases, appropriate keywords such as "mobile edge", "computing", "offloading", "MEC", "metaheuristics", "genetic algorithm", "particle swarm optimization " and "hybrid GA with PSO", have been used in the exploring process. The exploration is taken place in July 2023, by limiting the time boundaries between 2019 and 2023. Since the topic of offloading covers various models in the literature, including stochastic and non-stochastic models with extensive approaches such as game theory, machine learning, queuing theory, and pure mathematic models, the result of the exploration was extremely high in numbers. Therefore, the results of the search were above1100 articles. By evaluating some critical parts, including Abstract, Contributions, and Conclusion, for the first stage, 800irrelevant papers have been removed. Next, by evaluating the organization of the remaining papers, because of quality, 237 papers have been removed as low quality. Finally, the remaining 41 papers related to the Metaheuristics are included in the current study. Additionally, the distribution of the mentioned articles has been compared by the number of Papers for each Year, as it is illustrated in Figure 4 in 2018and from 2019 to 2023.

According to the name of Publishers, IEEE has gained the highest points compared with the other publishers up to July 2023.

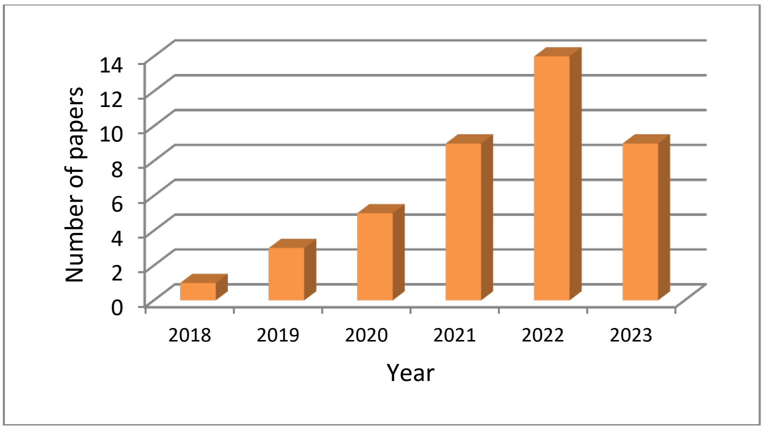

Fig. 4. Number of used articles in our survey

5. Metaheuristics-Based Offloading Mechanisms in Mobile Edge Computing

This section is to classify and review the offloading approaches in the MEC environment for selected papers by the proposed taxonomy. Based on the reviewed papers of the current study in the literature of Metaheuristics criterion and the definitions of subsection 2.2, reviewed Metaheuristics algorithms are classified into three main categories: "genetic algorithm", "particle swarm optimization", and "hybrid genetic algorithm with particle swarm optimization". In the following, we describe these approaches, and in each approach, related articles will be briefly reviewed.

5.1 (PSO) Based Computational Offloading

In this subsection, we will describe the PSO-based offloading mechanisms in the MEC environment. Then, the different approaches will be reviewed and summarized at the end of this subsection.

5.1.1 Overview of PSO-based offloading mechanisms

Mohamed A. Alqarni *et al.,* [6] proposed a smart metaheuristic optimization model to address the problem of low service quality due to vehicle movements and limited edge coverage. Then, the proposed model was used to characterize the overall latency of vehicle task offloading by considering resource utilization, workload at edge servers and vehicle movement characteristics. Furthermore, an intelligent placement metaheuristic was proposed based on the PSO metaheuristic. By using Compute Unified Device Architecture (CUDA), the proposed PSO metaheuristic is modified to match the architecture of current graphics processing unit (GPU) to enhance the search for offloading placements.

Yousef Alhaizaey *et al.,* [14] proposed an optimization technique for heterogeneous task allocation in edge compute micro clusters using particle swarm optimization (PSO) metaheuristic. The proposed approach aims to maximize the overall performance of the edge computing system by optimizing the allocation of tasks to different nodes in the cluster. The authors considered a scenario where multiple tasks with different requirements and priorities need to be executed in a shared edge computing infrastructure. The proposed approach used PSO to search for the optimal solution that minimizes the makespan (i.e., the total processing time of all tasks) subject to constraints on the available resources and task deadlines. The authors tested their approach using a set of benchmark datasets and showed that it outperforms traditional optimization techniques in terms of computational efficiency and solution quality.

The article [13] proposed a new method for task offloading in edge computing for industrial internet of things (IIoT) applications. The method used a particle swarm optimization (PSO) algorithm to optimize task offloading decisions and reduce computation complexity and communication overhead. The proposed method, called PSO-based Task Offloading (PSOTO), considered various factors such as task priority, resource availability, and network conditions when making task offloading decisions. It also takes into account the heterogeneity of edge devices and their dynamic changing status.

Luan N.T. *et al.,* [15] proposed a novel approach for efficient computation offloading in multi-tier multi-access edge computing systems using particle swarm optimization (PSO) aiming to minimize the overall latency and maximize the throughput of the system by optimizing the allocation of computational resources among different tiers of edge devices. They formulated the computation offloading problem as a mixed-integer nonlinear programming (MINLP) problem, taking into account various constraints such as resource availability, communication overhead, and quality-of-service (QoS) requirements. Then they proposed a PSO-based algorithm that utilizes a population of particles to represent potential solutions. Each particle represents a particular allocation of computational resources among the different tiers, and its fitness is evaluated based on the objective function that captures the trade-off between latency and throughput. The PSO algorithm iteratively updates the positions of the particles based on their velocities and applies a cognitive coefficient to control the diversity of the population.

Shi Dong *et al.,* [16] proposed a novel approach to task offloading in mobile edge computing using quantum particle swarm optimization. to address the challenge of efficiently allocating tasks to edge servers in mobile edge computing environments, where there are multiple edge servers with different computing capabilities and the tasks have varying requirements. They proposed a quantum-inspired algorithm that leverages the advantages of particle swarm optimization to optimize task offloading. The proposed algorithm, called Quantum Particle Swarm Optimization (QPSO), combines the concepts of quantum computing and particle swarm optimization to search for the optimal solution to the task offloading problem.

Yi Zhang *et al.,* [18] proposed a slow-movement particle swarm optimization (PSO) algorithm for scheduling security-critical tasks in resource-limited mobile edge computing (MEC) environments. They highlighted the challenges of scheduling security-critical tasks in MEC environments, where computational resources are limited and tasks have strict deadlines. They proposed a slow-movement PSO algorithm that incorporates a velocity reduction factor to control the movement speed of particles, which helps to avoid premature convergence and improve the algorithm's ability to explore the solution space.

DeGan Zhang *et al.,* [19] proposed a new approach to offload computation-intensive tasks from mobile devices to edge servers in a mobile edge computing (MEC) environment. The proposed approach uses a chaotic quantum particle swarm optimization (QPSO) algorithm to optimize the offloading process. They first introduced the concept of MEC and the challenges associated with offloading tasks in such environments. They then presented a brief overview of QPSO and its advantages over traditional optimization methods. The proposed offloading approach consists of three stages: task clustering, edge server selection, and task allocation. In the task clustering stage, the authors use a hierarchical clustering algorithm to group tasks based on their computational requirements. In the edge server selection stage, the authors use a chaotic QPSO algorithm to select the most suitable edge server for each task cluster. Finally, in the task allocation stage, the authors use a round-robin scheduling algorithm to allocate tasks to the selected edge servers.

Taha Alfakih *et al.,* [20] proposed a novel resource allocation method for mobile edge computing (MEC) networks. The proposed method combined multi-objective accelerated particle swarm optimization (MOPSO) with dynamic programming technique (DPT) to optimize the resource allocation problem in MEC networks. They first highlighted the challenges of resource allocation in MEC networks, where the objective is to minimize the total cost of resources while meeting the service level agreements (SLAs) of various applications. They then proposed the MOPSO-DPT method, which was designed to search for the optimal solution that simultaneously minimizes the cost and meets the SLAs.

Wenqi Zhou *et al.,* [21] discussed a novel offloading strategy for mobile edge computing (MEC) based on a cache mechanism. The authors use a particle swarm optimization (PSO) algorithm to optimize the offloading decision-making process. They first introduced the concept of MEC and the importance of offloading in improving the performance of mobile devices. They then discussed the challenges of offloading in MEC, including the large amount of data generated by mobile devices and the limited computing resources of edge servers. To address these challenges, they proposed a cache-based offloading strategy that utilizes a PSO algorithm to optimize the offloading decision-making process. The cache mechanism was used to store frequently accessed data and reduce the amount of data that needs to be transmitted between the mobile device and the edge server.

Rui Ma *et al.,* [44] proposed an intelligent education evaluation mechanism that leverages 5G technology and a particle swarm optimization (PSO) algorithm to improve the efficiency and accuracy of ideology and politics education evaluation. The proposed mechanism utilizes edge computing to enable real-time evaluation and feedback, thus enhancing the learning experience for students. They described the proposed intelligent education evaluation mechanism, which consists of four components: data collection, data processing, evaluation models, and feedback. The data collection component involves gathering student performance data from various sources, such as online assignments, quizzes, and exams. The data processing component cleans and preprocesses the data to ensure its quality and relevance. The evaluation model's component employed a PSO algorithm to optimize the evaluation process. The algorithm iteratively evolves the best solutions based on their fitness levels, which are determined by factors such as accuracy, completeness, and consistency. Finally, the feedback component provided real-time feedback to students based on the evaluation results. The feedback includes suggestions for improvement and reinforcement of key concepts, helping students adjust their learning strategies accordingly.

Shun Li *et al.,* [45] proposed a computation offloading strategy for improving the performance of particle swarm optimization (PSO) algorithms in mobile edge computing (MEC) environments. They proposed a computation offloading strategy that delegates parts of the PSO calculation to the edge server. The strategy was designed to minimize the communication overhead between the device and the edge server while maintaining the accuracy of the PSO algorithm. The proposed strategy consists of two main components:

i. a task division scheme, which divides the PSO calculation into smaller tasks that can be processed locally on the device and remotely on the edge server
ii. a communication-efficient method, which minimizes the communication overhead between the device and the edge server.

Yu Chen *et al.,* [46] proposed a novel resource allocation strategy for Multi-Access Edge Computing (MEC) networks in 5G communication networks. The proposed strategy is based on an improved version of the Particle Swarm Optimization (PSO) algorithm, that takes into account the specific characteristics of MEC networks. The proposed algorithm employed a novel particle representation scheme that integrates both spatial and temporal information about the MEC network. Additionally, the algorithm incorporated a cognitive factor to enhance the diversity of particles and prevent premature convergence.

Nebojsa Bacanin *et al.,* [47] proposed an energy-efficient offloading mechanism for 5G-enabled edge nodes using particle swarm optimization (PSO). They aim to minimize the energy consumption of edge nodes while maintaining the required level of computing performance. They proposed a novel approach that combines PSO with a hierarchical clustering algorithm to optimize the offloading process. The proposed mechanism consists of three stages:

i. cluster formation
ii. task assignment
iii. offloading decision

In the first stage, they used a hierarchical clustering algorithm to group tasks based on their compute intensity and create clusters of similar tasks. In the second stage, they assigned tasks to the nearest available edge node based on a distance metric. Finally, in the third stage, they used PSO to optimize the offloading decision, where each particle represents a task and the swarm searches for the optimal solution that minimizes energy consumption while meeting the required performance criteria.

Ali Almashhadani *et al.,* [49] presented three algorithms, namely the heuristic Bald Eagle Search Optimisation [BESO] algorithm, Particle Swarm Optimization algorithm [PSO], and Genetic Algorithm [GA], to carry out heuristic offloading of computational tasks with a view to improving the latency and performance of MEC. Then, they attempted to find an algorithm that is most appropriate for

MEC. To achieve this. the three algorithms were tested in the Long-Term Evolution [LTE] based Orthogonal frequency-division multiplexing [OFDM] network during a period when the edge nodes had no adequate resources. The performance and efficiency of the three algorithms, BESO, PSO and GA, were determined and compared. In terms of offloading the computational tasks, the BESO algorithm was discovered to perform better, with greater energy efficiency and lower latency, than the other two algorithms.

Table 1 shows some of recent studies that used PSO algorithm to solve computational offloading in several use case such as vehicles, Internet of Things (IoT), industrial internet of things (IIoT) and mobile devices (MDs).

Table 1
Recent studies using PSO algorithms to solve computational offloading

Ref.	Case Study	Performance metric	Evaluation Tools	Advantages	Weaknesses
[6]	vehicles	Qos time and precision.	using an NVIDIA CUDA Maxwell architecture	improves the delay by 60% compared to the randomly offloaded method.	does not consider the dependency of the user tasks
[13]	industrial internet of things (IIoT)	QoE	simulation	takes into account task priority, resource availability, and network conditions.	Lack of real-world experiments
[14]	(IoT) applications	Minimizes processing time of all tasks	simulation	heterogeneous task allocation in edge compute micro clusters	do not consider energy efficiency No discussion of scalability Limited consideration of QoS
[15]	Mobile Devices (MDs)	minimize latency and maximize throughput	simulation	perform sensitivity analysis to investigate the impact of key parameters on the performance of the PSO algorithm.	Lack of Real-World Testing and No Comparison with Other Optimization Algorithms such GA
[16]	Mobile Devices (MDs)	computation time	simulation	QPSO achieves a higher success rate and lower computation time than the compared algorithms.	Computational Complexity
[18]	Mobile Devices (MDs)	reduce the processing time and memory usage of tasks	conduct experiments using a real-world application scenario	scheduling security-critical tasks in resource-limited MEC environments.	it does not adapt to changing conditions and do not provide a detailed analysis of the security risks
[19]	applications in smart-city	reduce delay and energy consumption of Real-time video analysis	simulation	significantly reduces the total processing time and energy consumption of mobile devices	computationally complex and difficult to implement in practice.

[20]	cell phone networks	minimizing the computing time and service cost	simulation environment	improve the efficiency and effectiveness of resource allocation in MEC networks	The authors do not provide sufficient details regarding the practical aspects of implementing their proposed method in real-world MEC networks.
[21]	Internet of Things (IoT)	reduce the system latency and energy consumption	simulations	proposed strategy achieves better performance in terms of processing time, transmission overhead, and task completion ratio.	It does not provide information on how to implement the strategy in existing MEC systems
[44]	5G technology which support online education	minimizing the worst-case energy consumption of users to ensure the fairness of task processing	simulations and experiments	The results demonstrate that the PSO-driven edge computing approach significantly improves the accuracy and efficiency of education evaluation compared to traditional methods.	proposed mechanism relies on a simplified scenario and requires further testing in more complex real-world environments.
[45]	5G communication scenario	delay minimization under energy consumption constraints	simulations and experiments using a real-world dataset.	achieves a 30% reduction in computation time and a 25% increase in optimization accuracy compared to local computing,	The paper does not address security concerns related to data privacy and confidentiality
[46]	5G communication networks	realize low-latency and high-speed information exchange	extensive simulations and comparisons	the proposed algorithm achieves a 25% improvement in resource utilization and a 15% increase in user satisfaction compared to traditional methods.	High computational complexity
[47]	5G communication networks	minimizes energy consumption	simulated testbed	reduced energy consumption while maintaining the required computing performance.	involves a combination of clustering and particle swarm optimization (PSO), which can increase the computational complexity.
[49]	IoT application in MEC	improving the latency and performance of MEC	LTE-based OFDM and MATLAB Simulink	The results for the best throughput, performance and energy efficiency were obtained	The authors do not discuss how their proposed scheme scales to larger numbers of MEC nodes

5.2 (GA) Based Computational Offloading

In the context of mobile edge computing, using a genetic algorithm metaheuristic can be an effective way to solve complex optimization problems related to computational offloading. To apply this approach, it could start by defining the problem which need to optimize it: e.g., finding the optimal set of resources (computational power, memory, etc.) at the edge servers to execute a given application with minimum latency and maximum efficiency. Then, it would need to define the search space of potential solutions, which might involve variables like resource allocation, task scheduling, and network configuration [54]. Once determining these elements, the genetic algorithm library or framework can be implemented the optimization process. This typically involves creating a population of initial solutions, running each solution through a series of iterations, and updating the population based on the results of each iteration. Table 2 shows some of recent studies that used GA algorithm to solve computational offloading in edge servers in several use cases.

5.2.1 Overview of (GA)-based offloading mechanisms

Sumit Singh *et al.,* [22] proposed a novel approach to solving the issue of profit optimization in Mobile Edge Computing (MEC) environments using Genetic Algorithms (GAs). GAs is a type of evolutionary algorithm inspired by the process of natural selection, which can effectively search complex solution spaces to find optimal solutions. They tested their approach using a simulation environment that models a realistic MEC scenario. They compare the results obtained using their GA-based approach with those obtained using traditional optimization techniques like Greedy Randomized Adaptive Search Procedure (GRASP) and Simulated Annealing (SA). The results show that the GA-based approach outperforms GRASP and SA in terms of solution quality and computational efficiency.

Hong Wang *et al.,* [23] proposed a collaborative task offloading strategy for Unmanned Aerial Vehicle (UAV) clusters in Mobile Edge Computing (MEC) environments. The strategy used an improved Genetic Algorithm (IGA) to optimize the task offloading process by considering various factors such as task priority, computing resources, and communication overhead. The IGA uses a population of candidate solutions, each representing a potential task offloading strategy. The fitness function evaluates the quality of each solution based on the above factors, and the best solutions are selected to produce the next generation of offspring.

Ali Shahidinejad *et al.,* [24] proposed a metaheuristic-based approach for computation offloading in edge-cloud environments. They aimed to optimize the offloading process by minimizing the processing time and reducing the energy consumption of edge devices. They proposed a hybrid approach that combines the advantages of both genetic algorithms and simulated annealing to solve the offloading problem. The proposed approach consists of three stages:

i. task clustering
ii. resource allocation
iii. task scheduling.

In the first stage, the authors use a clustering algorithm to group tasks based on their characteristics and requirements. In the second stage, they allocate resources to each cluster using a genetic algorithm. Finally, in the third stage, they schedule the tasks using a simulated annealing algorithm, taking into account the allocated resources and the dependencies between tasks.

Sheuli Chakraborty *et al.,* [25] proposed a sustainable task offloading decision framework using genetic algorithms in sensor mobile edge computing (SMGE). They aimed to minimize the energy consumption and computational latency of SMGE systems while ensuring task deadlines are met. They consider a scenario where multiple sensors generate data that need to be processed in real-time, and the tasks are offloaded to nearby edge servers for processing. The proposed framework consists of two stages:

i. task classification
ii. task offloading decision

In the first stage, the authors classify tasks based on their priority levels using a fuzzy c-means clustering algorithm. In the second stage, they use a genetic algorithm to determine the optimal offloading decisions for each task. The genetic algorithm considers various factors such as task priority, computing resources, communication overhead, and energy consumption.

Banghua Wu *et al.,* [26] proposed a metaheuristic-based multi-objective approach for optimally deploying Internet of Things (IoT) services on fog computing platforms in a way that maximizes resource utilization, minimizes latency, and ensures QoS (Quality of Service) requirements are met. So, they proposed a multi-objective optimization approach that considers various conflicting objectives, including deployment cost, latency, reliability, and security. The proposed approach used a metaheuristic algorithm, specifically a variant of the particle swarm optimization (PSO) algorithm, to search for the optimal solution that balances the competing objectives. The algorithm iteratively evaluates candidate solutions and updates the swarm of particles representing the potential solutions. The authors also employ a technique called crowding distance to diversify the population of particles and avoid converging to a single solution.

Ahmed A. Al-habob *et al.,* [27] considered sequential task offloading to multiple mobile-edge computing servers to providing ultra-reliable low- latency mobile edge computing. They aimed to minimize both latency and offloading failure probability by scheduling sub-tasks to servers. So, they formulated an optimization problem with constraints over binary scheduling decision variables. Then genetic algorithm was devised to solve the formulated optimization problems.

Zhuofan Liao *et al.,* [28] presented a multi-user-to-multi-servers (MUMS) edge computing problem in ultra-dense cellular networks. The MUMS problem is divided and conquered by two phases, which are server selection and offloading decision. For the server selection phases, mobile users are grouped to one BS considering both physical distance and workload. After the grouping, the original problem is divided into parallel multi-user-to-one-server offloading decision sub problems. To get fast and near-optimal solutions for these sub problems, they designed a distributed offloading strategy based on a binary-coded genetic algorithm to get an adaptive offloading decision. The extensive simulations show that the proposed strategy significantly reduces the average latency and energy consumption of mobile devices.

Heekang Song *et al.,* [29] studied the joint design of computing server deployment and user offloading associations in wireless edge networks with wireless backhaul, enabling broadband transmission at a lower cost than the existing wired backhaul. Leveraging the evolutionary concept of a genetic algorithm, they devise a novel algorithm to solve the problem and minimize the average service delay while satisfying the delay requirements of individual users.

Hao Liu *et al.,* [31] proposed a computing resource allocation strategy for 5G communication in the Internet of Things (IoT) environment by applying UAV-assisted edge computing. First, they constructed a system model with the UAV deployed with mobile edge computing (MEC) servers to provide assisted computing services for multiple users on the ground. Based on the optimization of the UAV trajectory, communication scheduling, and the energy consumption model of the UAV, they formulated a problem of the total computational cost minimization. Then, they improved a genetic algorithm by introducing a penalty function to solve this problem, in which selection, crossover, and mutation operations are iterated to obtain the optimal allocation strategy for computational resources.

ZHOU Tianqing *et al.,* [32] introduced a frequency spectrum partitioning mechanism to tackle serious network interference caused by ultra-dense deployment of base stations, and they introduced Non-Orthogonal Multiple Access (NOMA) technology to improve the uplink frequency spectrum efficiency. They Consider that the optimization problem as nonlinear mixed-integer form, and used an effective Adaptive Genetic Algorithm with Diversity-Guided Mutation (AGADGM) for cooperative computation offloading and resource allocation is designed.

Benjamin Kwapong Osibo *et al.,* [39] proposed a novel Context-aware Computation Offloading (CaCO) architecture, particularly considering the execution time and battery consumption of SMDs when running resource-intensive tasks before proposing offloads. Secondly, they presented Efficient Genetic Algorithm (EGA) to obtain the optimized solution for the formulated task allocation NP-hard problem in accessible time complexity.

Shihong Hu *et al.,* [40] formulated the transmitting power allocation (PA) problem for mobile users to minimize energy consumption in ultra dense network (UDN). Using the quasiconvexity technique, they addressed the PA problem and presented a noncooperative game model based on sub gradient (NCGG). Then, they formulated the problem of joint request offloading and resource scheduling (JRORS) as a mixed-integer nonlinear program to minimize the response delay of requests. The JRORS problem could be divided into two problems, namely, the request offloading (RO) problem and the computing resource scheduling (RS) problem. Therefore, they analysed the JRORS problem as a double decision-making problem and proposed a multiple-objective optimization algorithm based on i-NSGA-II, referred to as MO-NSGA.

Shuang Fu *et al.,* [41] proposed an optimal offloading and scheduling scheme for workflow tasks to minimize the total energy consumption in the MEC network with multiple users and multiple virtual machines (VMs), based on an improved genetic algorithm. Then, they formulated problem of how to determine the optimal offloading and scheduling scheme of workflow to minimize the total energy consumption of the system while meeting the deadline constraint. To solve this problem, they adopted improved genetic algorithm to obtain the optimal offloading strategy and scheduling.

Zhi Li *et al.,* [42] proposed joint optimization method based on the Genetic Algorithm (GA) for task offloading proportion, channel bandwidth, and mobile edge servers' (MES) computing resources in the scenario where some computing tasks can be partly offloaded to the MES. Under the limitation of wireless transmission resources and MESs' processing resources, GA was used to solve the optimization problem of minimizing user task completion time, and the optimal offloading task strategy and resource allocation scheme were obtained. The simulation results showed that the proposed algorithm can effectively reduce the task completion time and ensure the fairness of users' completion times.

Arash Bozorgchenani *et al.,* [43] modelled a task offloading in MEC as a constrained multi-objective optimization problem (CMOP) that minimizes both the energy consumption and task processing delay of the mobile devices. To solve the CMOP, they designed an evolutionary algorithm that can efficiently find a representative sample of the best trade-offs between energy consumption and task processing delay, i.e., the Pareto-optimal front. Compared to existing approaches for task offloading in MEC, this approach finds offloading decisions with lower energy consumption and task processing delay.

Amina LAMMARIa *et al.,* [52] proposed an efficient hybrid genetic algorithm based on a genetic algorithm and a VNS variable neighbourhood search taking into account the problem specificity. In order to deduce the best combination that provides the solution with the lowest cost, several possible hybridization schemes are proposed: a sequential hybridization, and two memetic algorithms where the local search is integrated into the genetic evolutionary process to overcome the shortcomings of the genetic algorithm, improving the exploration and exploitation capabilities of the algorithm. Both genetic operators (mutation and crossover) and VNS are adapted to this problem, making them more effective in finding the offloading strategy as quickly as possible. Several experiments conducted on generated instances of different sizes proved the effectiveness of the newly proposed approach, and corroborated by the comparison they made with other works in the literature dealing with the same problem.

Table 2
Recent studies using GA algorithms to solve computational offloading

Ref.	Case Study	Performance metric	Evaluation Tools	Advantages	Weaknesses
[22]	Mobile Devices (MDs)	examine the profitability of computation offloading from the perspective of a network operator	Matlab simulation	GA-based approach achieves a 27% increase in profit compared to GRASP and a 14% increase compared to SA.	Computational Complexity
[23]	unmanned aerial vehicle (UAV)	task completion time, energy consumption, and success rate	simulations	the IGA-based approach achieves a 30% reduction in task completion time and a 25% reduction in energy consumption compared to traditional methods.	the author assumes that all UAVs have the same computing resources and communication capabilities, which may not be the case in reality
[24]	Mobile Devices (MDs)	reduce processing time and energy consumption	simulations	it reduces the processing time by up to 30% and energy consumption by up to 20%.	One limitation of the paper is that it assumes that all edge devices have the same computing capacity and communication bandwidth, which may not be the case in practice.
[25]	(MDs) connected with Sensor Mobile Edge Computing (SMEC)	reducing energy consumption and computational latency	simulation experiments	it reduces energy consumption by up to 27%, computational latency by up to 33%, and increases the task completion ratio by up to 17%.	the title of the paper suggests a focus on sustainability, the proposed approach primarily concentrates on minimizing energy consumption and computational latency.
[27]	multiple mobile-edge computing servers	minimize both latency and offloading failure probability	Simulation	The proposed approach has the potential to enable more efficient and effective task execution in various MEC applications	Computational Complexity especially when applied to large and complex problems

[28]	next generation cellular networks	low latency and energy cost.	Simulation	reduces the average delay by 56% and total energy consumption by 14% in the ultra-dense cellular networks.	The paper assumes a static network environment, which may not accurately reflect real-world scenarios where network conditions change frequently
[29]	Mobile Devices (MDs)	low delay with a limited number of servers	simulation	proposed algorithm outperforms the conventional random search or heuristic algorithms	they assume that the network topology is fixed and that there is no interference between nodes. This can limit the accuracy of the results obtained from the proposed approach.
[31]	(IoT) environment	energy consumption model of the UAV	simulation	the total cost and total time of the proposed strategy are better than other comparison strategies	The paper does not consider energy consumption, which is a critical concern for UAVs.
[32]	ultra-dense heterogeneous edge computing networks	energy consumption	simulation	achieved lower system energy consumption than other existing algorithms under strict constraints of users' delay	The paper focuses specifically on cooperative computation offloading and resource management in NOMA-MEC systems, which limits its applicability to other areas of edge computing.
[39]	Smart Mobile Devices	execution time and battery consumption of SMDs	experiments conducted with real Android SMDs and simulation results	the proposed algorithm is superior in performance and could effectively reduce energy consumption and task completion latency.	Limited attention to data privacy and security, and No consideration of edge server placement and deployment

5.3 Integrating (PSO) with (GA)

Integrating genetic algorithms (GA) with particle swarm optimization (PSO) can provide several benefits for computational offloading in mobile edge computing, such Improved convergence rate, Increased exploration ability, robustness against noise and ability to handle multi-objective problems. Table 3 shows some of recent studies that used hybrid PSO with GA algorithm to solve computational offloading in edge servers.

5.3.1 Overview of integrating (PSO) with (GA)

Jing Bi *et al.,* [17] proposed a genetic particle swarm optimization (GPSO) algorithm to solve this problem. GPSO combines the strengths of genetic algorithms and particle swarm optimization. They tested their GPSO algorithm on a real-world dataset collected from a smart home system. The results showed that GPSO significantly reduces energy consumption compared to traditional task offloading approaches, while still meeting task deadlines and resource constraints. In addition, they performed sensitivity analysis to investigate the impact of various parameters on the performance of GPSO. They found that the population size, crossover probability, and mutation probability have a significant influence on the algorithm's performance.

R. Ezhilarasie *et al.,* [37] proposed an approach that employs Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) is used to determine the near optimal solution for scheduling off loadable components in an application, with the intent of significantly reducing the execution time of an application and energy consumption of the smart devices. With a new inertial weight equation, they proposed an Adaptive Genetic Algorithm–Particle Swarm Optimization (AGA-PSO) algorithm which uses GA's ability in exploration and PSO's ability in exploitation to make the offloading optimized without violating the deadline constraint of an application.

Noha El Menbawy *et al.,* [36] proposed a model which utilized to determine the optimal way of task offloading for Internet of Robotic Things (IoRT) devices for reducing the amount of energy consumed in IoRT environment and achieving the task deadline constraints. The approach was implemented based on fog computing to reduce the communication overhead between edge devices and the cloud. To validate the efficacy of the proposed schema, an extensive statistical simulation was conducted and compared to other related works. The proposed schema is evaluated against the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Artificial Bee Colony (ABC), Ant Lion Optimizer (ALO), Grey Wolf Optimizer (GWO), and Salp Swarm Algorithm to confirm its effectiveness. After 200 iterations, this proposed schema was found to be the most effective in reducing energy, achieving a reduction of 22.85%. This was followed closely by GA and ABC, which achieved reductions of 21.5%. ALO, WOA, PSO, and GWO were found to be less effective, achieving energy reductions of 19.94%, 17.21%, 16.35%, and 11.71%, respectively.

Fengxian Guo *et al.,* [35] studied the energy-efficient computation offloading management scheme in the MEC system with small cell networks (SCNs). To minimize the energy consumption of all UEs via jointly optimizing computation offloading decision making, spectrum, power, and computation resource allocation. Specially, the UEs need not only to decide whether to offload but also to determine where to offload. So, they First presented the computation offloading model and formulate this problem as a mix integer non-linear programming problem, which is NP-hard. Taking advantages of genetic algorithm (GA) and particle swarm optimization (PSO), they design a suboptimal algorithm named as hierarchical GA and PSO-based computation algorithm to solve this problem.

Truong Van Truong *et al.,* [34] investigated a performance and optimization of MEC surveillance systems using non-orthogonal multiple access (NOMA) technology. Specifically, two camera units (CUs) perform the monitoring task to be accomplished by the MEC access point (AP) through Rayleigh fading wireless links. They then proposed the four-phase protocol for this system. Accordingly, they derive the closed-form exact expressions of the successful computation probability (SCP), and study the impact of the network parameters on the system performance. Furthermore, they proposed and compared three meta-heuristic-based algorithms, namely MSCP-GA, MSCP-PSO, and MSCP-HGAPSO, to find the optimal parameters set to help the proposed system achieve the maximum SCP.

Jing Bi *et al.,* [33] proposed a partial computation offloading method to minimize the total energy consumed by Smart mobile devices (SMDs)and edge servers by jointly optimizing the offloading ratio of tasks, CPU speeds of SMDs, allocated bandwidth of available channels, and transmission power of each SMD in each time slot. It jointly considers the execution time of tasks performed in SMDs and edge servers, and transmission time of data. It also jointly considers latency limits, CPU speeds, transmission power limits, available energy of SMDs, and the maximum number of CPU cycles and memories in edge servers. Considering these factors, a nonlinear constrained optimization problem was formulated and solved by a novel hybrid metaheuristic algorithm named genetic simulated annealing-based particle swarm optimization (GSP) to produce a close-to-optimal solution. GSP achieves joint optimization of computation offloading between a cloud data centre and the edge, and resource allocation in the data centre.

Zheyi Chen *et al.,* [38] designed a multi-unmanned aerial vehicles (UAVs)-enabled MEC system model to further enhance the Quality-of-Service (QoS) of MEC systems. Here, UAVs are regarded as edge servers to offer computing services for MDs. So, they proposed a two-layer joint optimization method (PSO-GA-G) to minimize the average task response time by jointly optimizing UAV deployment and computation offloading. First, the outer layer utilized a Particle Swarm Optimization algorithm combined with Genetic Algorithm operators (PSO-GA) to optimize UAV deployment. Next, the inner layer adopted a greedy algorithm to optimize computation offloading.

Md Muzakkir Hussain *et al.,* [51] proposed a new computational model called Vehicular Fog Computing (VFC) and offloaded the computation workload from passenger devices (PDs) to transportation infrastructures such as roadside units (RSUs) and base stations (BSs), called static fog nodes. It can also exploit the underutilized computation resources of nearby vehicles that can act as vehicular fog nodes (VFNs) and provide delay- and energy-aware computing services. However, the capacity planning and dimensioning of VFC, which come under a class of facility location problems (FLPs), is a challenging issue. The complexity arises from the spatio-temporal dynamics of vehicular traffic, varying resource demand from PD applications, and the mobility of VFNs. So, this paper proposed a multi-objective optimization model to investigate the facility location in VFC networks. The solutions to this model generated optimal VFC topologies pertaining to an optimized trade-off (Pareto front) between the service delay and energy consumption. Thus, to solve this model, they proposed a hybrid Evolutionary Multi-Objective (EMO) algorithm called Swarm Optimized Non-dominated sorting Genetic algorithm (SONG) which combines the convergence and search efficiency of two popular EMO algorithms: the Non-dominated Sorting Genetic Algorithm (NSGA-II) and Speed-constrained Particle Swarm Optimization (SMPSO). First, we solve an example problem using the SONG algorithm to illustrate the delay–energy solution frontiers and plotted the corresponding layout topology. Subsequently, we evaluate the evolutionary performance of the SONG algorithm on real-world vehicular traces against three quality indicators: Hyper-Volume (HV), Inverted Generational Distance (IGD) and CPU delay gap.

Table 3
Some recent studies using PSO algorithms

Ref.	Case Study	Performance metric	Evaluation Tools	Advantages	Weaknesses
[33]	Smart mobile devices	minimize the total energy consumed by SMDs and edge servers	Real-life data-based experiment	achieves lower energy consumption in less convergence time than its three typical peers	High computational cost, it means the proposed algorithm requires significant computational resources, especially when dealing with complex tasks or large datasets.
[34]	MEC surveillance systems using NOMA	improve the system's performance by using NOMA	simulation	improve the system's performance by 40% higher than when the optimal algorithm	

[35]	integrating MEC into small cell networks (SCNs)	minimize the energy consumption	simulation	improving the efficiency of computation offloading in densely deployed small cell networks,	high complexity of computation offloading decision-making
[37]	IoT (Internet of Things) and CPS (Cyber Physical Systems)	reducing the execution time of an application and energy consumption	evaluate their algorithm using a testbed comprising of IoT devices and a cloud server	adaptive GA-PSO algorithm outperforms all other algorithms in terms of computational latency, energy consumption, and system performance.	algorithm assumes a fixed availability of computing resources at the edge server,
[38]	multi-UAV-enabled MEC system	minimize the average task response time by jointly optimizing UAV deployment and computation offloading	simulations	proposed algorithm significantly reduces the total cost of the system compared to the benchmark schemes	assumes that the UAVs have infinite battery life, and incorporating battery constraints could further enhance the algorithm's practicality.
[17]	smart home system	energy consumption	simulation	The proposed algorithm is tested on a real-world dataset and shown to significantly reduce energy consumption compared to traditional approaches	Unrealistic assumptions such as assuming that the computing resources of edge servers and cloud servers are perfectly utilized

6. Discussion and Comparison

This section illustrates an analytical examination and discussion of the existing metaheuristics -based offloading mechanisms in mobile edge computing. The analytical examination and reports are based on the mentioned TQs in Section 4 that are explained in the following subsections.

6.1 Classifications

This subsection tries to answer the following technical question:

TQ1: What classification is utilized in metaheuristics -based offloading approaches?

According to the proposed taxonomy, a statistical comparison among metaheuristics -based approaches in the MEC environment is depicted in Figure 9. Based on the taxonomy, three mechanisms are considered in metaheuristics, genetic algorithm, Particle Swarm Optimization, and hybrid GA with PSO. As it is shown in Figure 6, most of the research area of the selected articles belongs to a kind of genetic algorithm with a percentage coverage of 42%, then Particle Swarm Optimization approach with a percentage coverage of 36%. The other rank belongs to hybrid GA with PSO, with 22% coverage.

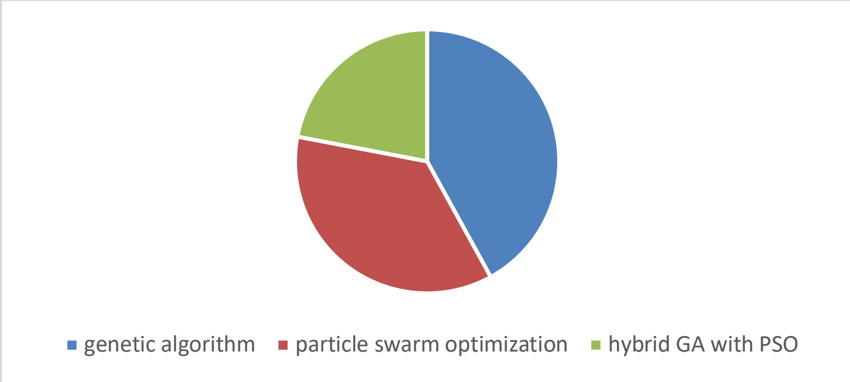

Fig. 6. Classification of Metaheuristics-based approaches

6.2 Performance metrics

This subsection tries to answer the following technical question:

TQ2: What performance metrics are usually utilized in metaheuristics -based offloading approaches?

As it is depicted in Figure 7, some specifications are analysed and compared to the performance metrics for metaheuristics-based offloading methods. It is worth mentioning that since some reviewed articles were multi-objective, some of the mentioned metrics might be considered in more than one article. The investigation of these specifications shows that the energy has the most usage in the metaheuristics-based offloading methods with the percentage of 41; delay and processing time with 33% and 27% respectively; cost, throughput and QoS catch the least rank that is represented as open challenges in the metaheuristics -based offloading methods.

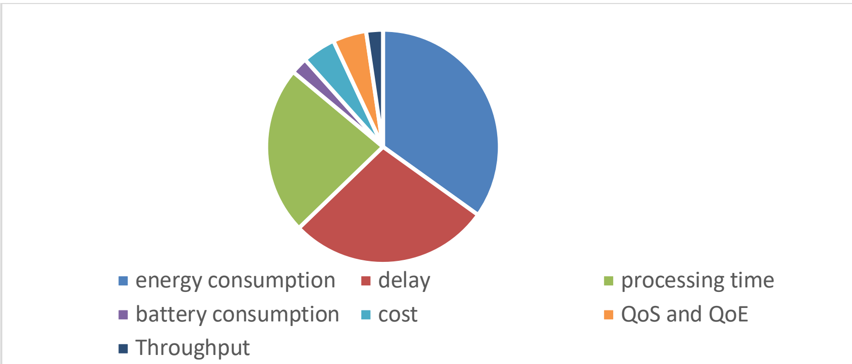

Fig. 7. Performance metrics of Metaheuristics-based approaches

6.3 Case studies

This subsection tries to answer the following technical question:

TQ3: What case studies are applied in Metaheuristics-based offloading approaches?

The utilized case studies of the Metaheuristics-based offloading mechanisms are illustrated in Figure 8. Noteworthy, because some studied papers were suitable for more than one technology, some case studies might be presented in more than one article. As it is observed in Figure 7, Mobile

Devices (MDs) has the highest number of usages, and in the second place, IOT. Also, 5g network, Vehicles, ultra dense network, smart city, and Unmanned Aerial Vehicles (UAU) take the least attraction with only 1 article each.

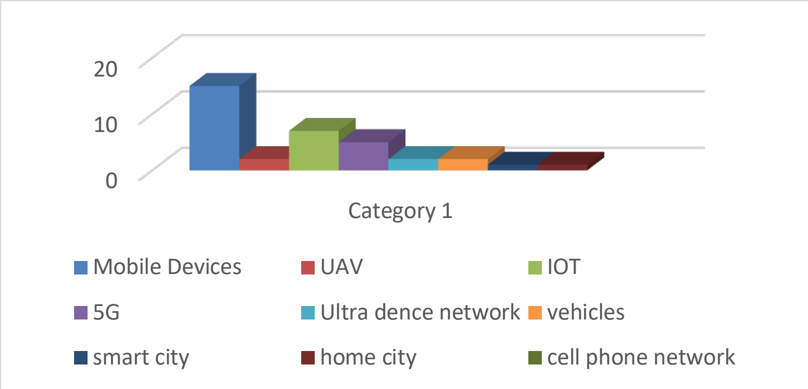

Fig. 8. Number of applied case studies in Metaheuristics-based approaches

6.4 Evaluation tools

This subsection tries to answer the following technical question:

TQ4: What evaluation tools are utilized for assessing the Metaheuristics-based approaches?

As it is illustrated in Figure 9, 45% of the research papers have not specified evaluation tools for their proposed model. Besides, 30% and 20% of papers used Python and MATLAB, respectively, in the literature to implement their model. In addition, 5% of the research papers used CloudSim, iFogSim, and NS2/3 tools separately to assess and analyse the existing case studies.

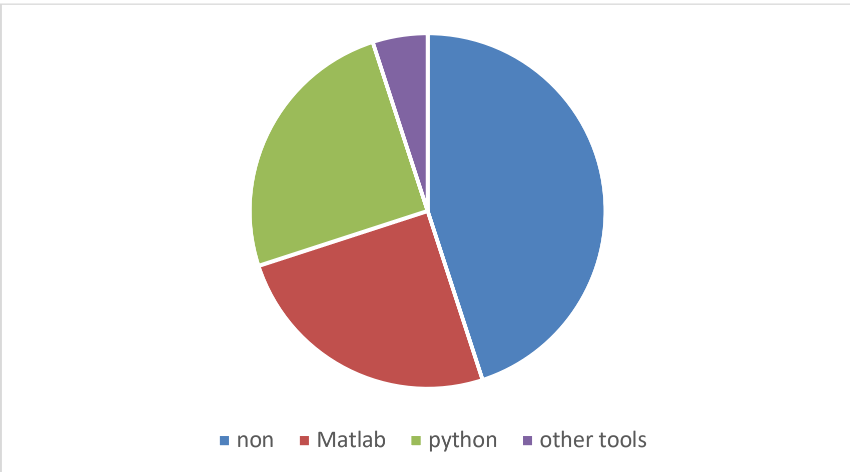

Fig. 9. Evaluation tools comparison in Metaheuristics-based approaches

6.5 Similarities and Differences

This subsection tries to answer the following technical question:

TQ5: What are the similarities *and* differences between two offloading approaches?

Table 4 shows similarities and differences between GA algorithm and PSO algorithm.

Table 4
Similarities and differences between GA and PSO

Similarities	Differences
-they work with a set of candidate solutions (population) rather than an individual solution.	-GAs represent solutions as binary strings (genes), whereas PSO represents them as particles with position and velocity variables. This difference affects how each algorithm handles complexity and diversity within the population.
- selecting the fittest individuals based on their fitness values.	-The search space explored by GAs is more structured and deterministic due to the gene representation. In contrast, PSO has a more flexible and probabilistic approach, allowing it to explore different parts of the search space more efficiently.
-Both algorithms can handle nonlinear problems and complex search spaces	-GAs employ crossover and mutation operators to combine and modify existing genes, respectively. PSO uses a velocity vector to determine the direction of movement for each particle, which allows it to adapt faster but may lead to less diverse solutions.
-Both algorithms do not require explicit knowledge of the objective function's gradient	-GAs tend to converge slower than PSO because of their more structured search process. However, GAs often achieves higher accuracy levels once convergence is reached
-Both methods can be parallelized easily, but GAs may require more computational resources since they operate on larger populations while PSO can benefit from distributed computing architectures, making it well-suited for large-scale optimization problems.	-PSO is generally considered more robust than GA since it doesn't rely solely on the quality of the initial population. Instead, its ability to adapt to changing conditions through the velocity vector helps maintain diversity throughout the optimization process.
- GAs have been successfully applied to various optimization problems in mobile edge computing, such as network resource allocation, scheduling, and routing. PSO has been mainly employed in optimizing energy consumption, task assignment, and scheduling in mobile devices.	-Computational cost: The computational cost of GAs can increase rapidly with the size of the population, while PSO remains relatively constant regardless of the number of particles.

6.6 Benefit of combining the two algorithms

This subsection tries to answer the following technical question:

TQ6: What are the benefits of combining the two algorithms together in the offloading scheme?

Integrating genetic algorithms (GA) with particle swarm optimization (PSO) can provide several benefits for computational offloading in mobile edge computing:

i. Improved convergence rate: GA has been shown to have a faster convergence rate than PSO when optimizing complex functions. By combining both techniques, the improved convergence rate of GA can be leveraged to optimize the placement of computation-intensive tasks on the network.
ii. Increased exploration ability: PSO is known for its ability to explore the search space effectively, but it may get stuck in local optima. On the other hand, GA uses crossover and mutation operators to introduce new solutions into the population, which can lead to

more diverse and effective exploration of the search space. Combining these two techniques can result in a better balance between exploitation and exploration.

iii. Robustness against noise: Mobile edge computing environments are often characterized by high levels of noise and variability due to factors such as changing network conditions, device mobility, and interference from other wireless systems. Both GA and PSO are robust against noisy data, but GA has been shown to perform better in this regard. By integrating GA with PSO, the overall performance can be further enhanced.

iv. Ability to handle multi-objective problems: Computational offloading in mobile edge computing often involves trade-offs between different objectives, such as minimizing latency while also maximizing throughput. GA and PSO can both be used to solve multi-objective optimization problems, but GA has been shown to be more effective at handling complex objective functions.

6.7 Future Research Direction

This subsection tries to answer the following technical question:

TQ7: What is the future research direction of Metaheuristics-based offloading approaches?

In future research, we must investigate metaheuristics-based offloading approaches in many issues:

i. Proposing new scheduling-related metaheuristics: Because choosing and scheduling an appropriate server to optimize essential metrics in a multi-server-based system is a pivotal point to be considered.

ii. Because of the high dynamic behaviour of the MEC environment with its high data rates in one hand and heterogeneity in other hands, to fulfil offloading successfully, it is necessary to work on new methods of Metaheuristics -based to address high dynamic behaviour.

iii. The mobility is a severe challenge in some research areas such as Unmanned Aerial Vehicles (UAV), Intelligent Transportation Systems (ITS), and Vehicular Ad hoc networks (VANETs), Despite its importance, mobility issues have been weakly or incompletely cover in the studied works in the literature of MEC environments. So, applying newer metaheuristics methods combined with other mathematical models is more suitable.

iv. There must be applicable approaches for forecasting, preventing, protecting, and recovering the system from catastrophic situations. Because of the stochastic behaviour of such threats, different kinds of Metaheuristics or hybrid-based methods could be offered as ideal methods to deal with all the problems mentioned above.

v. Since the nature of high demanding data rates of some environments such as vehicular networks or health-care is life-critical, related applications need real-time execution in the MEC environment. On the other hand, there is not any prior knowledge in these environments. Therefore, these kinds of problems must use new Metaheuristics -based or hybrid methods to offload the required applications without faults.

7. Conclusions

In this article, we have reviewed the most important recent studies related to the use of three metaheuristic algorithms in mobile edge computing. These algorithms have been adopted in multiple use cases such as mobile devices, Internet of Things devices, smart cities, smart vehicles, and others. In all previous cases, the use of these Algorithms reduces power consumption, reduce time delays for completed tasks, and improve overall system performance.

Both GA and PSO are powerful optimization techniques, they differ in several ways including their approach to solving optimization problems, convergence rates, complexity, and applicability to specific domains. Better optimization outcomes can be achieved by choosing the appropriate method depending on the particular application, available resources in mobile edge computing, specific problem and the characteristics of the search space.

In future research, we will work on developing a new algorithm that combines the genetic algorithm with the particle swarm algorithm, and search for the best parameters for it, in order to be compatible with the problem of computational offloading.

Acknowledgement
This research was not funded by any grant.

References

[1] Escamilla-Ambrosio, P. J., A. Rodríguez-Mota, E. Aguirre-Anaya, R. Acosta-Bermejo, and M. Salinas-Rosales. "Distributing computing in the internet of things: cloud, fog and edge computing overview." In *NEO 2016: Results of the Numerical and Evolutionary Optimization Workshop NEO 2016 and the NEO Cities 2016 Workshop held on September 20-24, 2016 in Tlalnepantla, Mexico*, pp. 87-115. Springer International Publishing, 2018. https://doi.org/10.1007/978-3-319-64063-1_4

[2] Abdel-Basset, M., L. Abdel-Fatah, and A. K. Sangaiah. "Chapter 10-Metaheuristic Algorithms: A Comprehensive Review In Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications." (2018). https://doi.org/10.1016/B978-0-12-813314-9.00010-4

[3] Shakarami, Ali, Mostafa Ghobaei-Arani, Mohammad Masdari, and Mehdi Hosseinzadeh. "A survey on the computation offloading approaches in mobile edge/cloud computing environment: a stochastic-based perspective." *Journal of Grid Computing* 18 (2020): 639-671. https://doi.org/10.1007/s10723-020-09530-2

[4] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." In *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942-1948. IEEE, 1995.

[5] Salaht, Farah Ait, Frédéric Desprez, and Adrien Lebre. "An overview of service placement problem in fog and edge computing." *ACM Computing Surveys (CSUR)* 53, no. 3 (2020): 1-35. https://doi.org/10.1145/3391196

[6] Alqarni, Mohamed A., Mohamed H. Mousa, and Mohamed K. Hussein. "Task offloading using GPU-based particle swarm optimization for high-performance vehicular edge computing." *Journal of King Saud University-Computer and Information Sciences* 34, no. 10 (2022): 10356-10364. https://doi.org/10.1016/j.jksuci.2022.10.026

[7] Chen, Liuhua, Haiying Shen, and Karan Sapra. "RIAL: Resource intensity aware load balancing in clouds." In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 1294-1302. IEEE, 2014. https://doi.org/10.1109/INFOCOM.2014.6848062

[8] Sardellitti, Stefania, Gesualdo Scutari, and Sergio Barbarossa. "Joint optimization of radio and computational resources for multicell mobile-edge computing." *IEEE Transactions on Signal and Information Processing over Networks* 1, no. 2 (2015): 89-103. https://doi.org/10.1109/TSIPN.2015.2448520

[9] Wu, Ziyu, Lin Gui, Jiacheng Chen, Haibo Zhou, and Fen Hou. "Mobile cloudlet assisted computation offloading in heterogeneous mobile cloud." In *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*, pp. 1-6. IEEE, 2016. https://doi.org/10.1109/WCSP.2016.7752655

[10] Kosta, Sokol, Andrius Aucinas, Pan Hui, Richard Mortier, and Xinwen Zhang. "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading." In *2012 Proceedings IEEE Infocom*, pp. 945-953. IEEE, 2012. https://doi.org/10.1109/INFCOM.2012.6195845

[11] Verbelen, Tim, Pieter Simoens, Filip De Turck, and Bart Dhoedt. "Leveraging cloudlets for immersive collaborative applications." *IEEE Pervasive Computing* 12, no. 4 (2013): 30-38. https://doi.org/10.1109/MPRV.2013.66

[12] Hao, Fang, Murali Kodialam, T. V. Lakshman, and Sarit Mukherjee. "Online allocation of virtual machines in a distributed cloud." *IEEE/ACM Transactions on Networking* 25, no. 1 (2016): 238-249. https://doi.org/10.1109/TNET.2016.2575779
[13] You, Qian, and Bing Tang. "Efficient task offloading using particle swarm optimization algorithm in edge computing for industrial internet of things." *Journal of Cloud Computing* 10 (2021): 1-11. https://doi.org/10.1186/s13677-021-00256-4
[14] Alhaizaey, Yousef, Jeremy Singer, and Anna Lito Michala. "Optimizing Heterogeneous Task Allocation for Edge Compute Micro Clusters Using PSO Metaheuristic." In *2022 Seventh International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 1-8. IEEE, 2022. https://doi.org/10.1109/FMEC57183.2022.10062755
[15] Huynh, Luan NT, Quoc-Viet Pham, Xuan-Qui Pham, Tri DT Nguyen, Md Delowar Hossain, and Eui-Nam Huh. "Efficient computation offloading in multi-tier multi-access edge computing systems: A particle swarm optimization approach." *Applied Sciences* 10, no. 1 (2019): 203. https://doi.org/10.3390/app10010203
[16] Dong, Shi, Yuanjun Xia, and Joarder Kamruzzaman. "Quantum Particle Swarm Optimization for Task Offloading in Mobile Edge Computing." *IEEE Transactions on Industrial Informatics* (2022). https://doi.org/10.1109/TII.2022.3225313
[17] Bi, Jing, Kaiyi Zhang, Haitao Yuan, and Qinglong Hu. "Energy-aware task offloading with genetic particle swarm optimization in hybrid edge computing." In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3194-3199. IEEE, 2021. https://doi.org/10.1109/SMC52423.2021.9658678
[18] Zhang, Yi, Yu Liu, Junlong Zhou, Jin Sun, and Keqin Li. "Slow-movement particle swarm optimization algorithms for scheduling security-critical tasks in resource-limited mobile edge computing." *Future Generation Computer Systems* 112 (2020): 148-161. https://doi.org/10.1016/j.future.2020.05.025
[19] Zhang, DeGan, GuiXiang Sun, Jie Zhang, Ting Zhang, and Peng Yang. "Offloading approach for mobile edge computing based on chaotic quantum particle swarm optimization strategy." *Journal of Ambient Intelligence and Humanized Computing* 14, no. 10 (2023): 14333-14347. https://doi.org/10.1007/s12652-023-04672-z
[20] Alfakih, Taha, Mohammad Mehedi Hassan, and Muna Al-Razgan. "Multi-objective accelerated particle swarm optimization with dynamic programing technique for resource allocation in mobile edge computing." *IEEE Access* 9 (2021): 167503-167520. https://doi.org/10.1109/ACCESS.2021.3134941
[21] Zhou, Wenqi, Lunyuan Chen, Shunpu Tang, Lijia Lai, Junjuan Xia, Fasheng Zhou, and Liseng Fan. "Offloading strategy with PSO for mobile edge computing based on cache mechanism." *Cluster Computing* (2022): 1-13.
[22] Singh, Sumit, and Dong Ho Kim. "Profit optimization for mobile edge computing using genetic algorithm." In *2021 IEEE Region 10 Symposium (TENSYMP)*, pp. 1-6. IEEE, 2021. https://doi.org/10.1109/TENSYMP52854.2021.9550947
[23] Wang, Hong. "Collaborative task offloading strategy of UAV cluster using improved genetic algorithm in mobile edge computing." *Journal of Robotics* 2021 (2021): 1-9. https://doi.org/10.1155/2021/3965689
[24] Shahidinejad, Ali, and Mostafa Ghobaei-Arani. "A metaheuristic-based computation offloading in edge-cloud environment." *Journal of Ambient Intelligence and Humanized Computing* 13, no. 5 (2022): 2785-2794. https://doi.org/10.1007/s12652-021-03561-7
[25] Chakraborty, Sheuli, and Kaushik Mazumdar. "Sustainable task offloading decision using genetic algorithm in sensor mobile edge computing." *Journal of King Saud University-Computer and Information Sciences* 34, no. 4 (2022): 1552-1568. https://doi.org/10.1016/j.jksuci.2022.02.014
[26] Peng, Kai, Victor Leung, Xiaolong Xu, Lixin Zheng, Jiabin Wang, and Qingjia Huang. "A survey on mobile edge computing: Focusing on service adoption and provision." *Wireless Communications and Mobile Computing* 2018 (2018). https://doi.org/10.1155/2018/8267838
[27] Al-Habob, Ahmed A., Octavia A. Dobre, and Ana Garcia Armada. "Sequential task scheduling for mobile edge computing using genetic algorithm." In *2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1-6. IEEE, 2019. https://doi.org/10.1109/GCWkshps45667.2019.9024374
[28] Liao, Zhuofan, Jingsheng Peng, Bing Xiong, and Jiawei Huang. "Adaptive offloading in mobile-edge computing for ultra-dense cellular networks based on genetic algorithm." *Journal of Cloud Computing* 10, no. 1 (2021): 1-16. https://doi.org/10.1186/s13677-021-00232-y
[29] Song, Heekang, Bonjun Gu, Kyungrak Son, and Wan 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* 9, no. 4 (2022): 2535-2548. https://doi.org/10.1109/TNSE.2022.3165372
[30] Boukerche, Azzedine, Shichao Guan, and Robson E. De Grande. "Sustainable offloading in mobile cloud computing: algorithmic design and implementation." *ACM Computing Surveys (CSUR)* 52, no. 1 (2019): 1-37. https://doi.org/10.1145/3286688
[31] Liu, Hao. "An UAV-assisted edge computing resource allocation strategy for 5G communication in IoT environment." *Journal of Robotics* 2022 (2022). https://doi.org/10.1155/2022/9397783

[32] ZHOU, Tianqing, Haiqin HU, and Xinliang ZENG. "Cooperative Computation Offloading and Resource Management Based on Improved Genetic Algorithm in NOMA-MEC Systems." *电子与信息学报* 44, no. 9 (2022): 3014-3023.

[33] Bi, Jing, Haitao Yuan, Shuaifei Duanmu, MengChu Zhou, and Abdullah Abusorrah. "Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization." *IEEE Internet of Things Journal* 8, no. 5 (2020): 3774-3785. https://doi.org/10.1109/JIOT.2020.3024223

[34] Van Truong, Truong, and Anand Nayyar. "System performance and optimization in NOMA mobile edge computing surveillance network using GA and PSO." *Computer Networks* 223 (2023): 109575. https://doi.org/10.1016/j.comnet.2023.109575

[35] Guo, Fengxian, Heli Zhang, Hong Ji, Xi Li, and Victor CM Leung. "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing." *IEEE/ACM Transactions on Networking* 26, no. 6 (2018): 2651-2664. https://doi.org/10.1109/TNET.2018.2873002

[36] El Menbawy, Noha, Hesham A. Ali, Mohamed S. Saraya, Amr MT Ali-Eldin, and Mohamed M. Abdelsalam. "Energy-efficient computation offloading using hybrid GA with PSO in internet of robotic things environment." *The Journal of Supercomputing* (2023): 1-40. https://doi.org/10.1007/s11227-023-05387-w

[37] Ezhilarasie, R., Mandi Sushmanth Reddy, and A. Umamakeswari. "A new hybrid adaptive GA-PSO computation offloading algorithm for IoT and CPS context application." *Journal of Intelligent & Fuzzy Systems* 36, no. 5 (2019): 4105-4113. https://doi.org/10.3233/JIFS-169970

[38] Chen, Zheyi, Hongqiang Zheng, Jianshan Zhang, Xianghan Zheng, and Chunming Rong. "Joint computation offloading and deployment optimization in multi-UAV-enabled MEC systems." *Peer-to-Peer Networking and Applications* (2022): 1-12. https://doi.org/10.1007/s12083-021-01245-9

[39] Osibo, Benjamin Kwapong, Zilong Jin, Tinghuai Ma, Bockarie Daniel Marah, Chengbo Zhang, and Yuanfeng Jin. "An edge computational offloading architecture for ultra-low latency in smart mobile devices." *Wireless Networks* 28, no. 5 (2022): 2061-2075. https://doi.org/10.1007/s11276-022-02956-4

[40] Hu, Shihong, and Guanghui Li. "Dynamic request scheduling optimization in mobile edge computing for IoT applications." *IEEE Internet of Things Journal* 7, no. 2 (2019): 1426-1437. https://doi.org/10.1109/JIOT.2019.2955311

[41] Fu, Shuang, Chenyang Ding, and Peng Jiang. "Computational Offloading of Service Workflow in Mobile Edge Computing." *Information* 13, no. 7 (2022): 348. https://doi.org/10.3390/info13070348

[42] Li, Zhi, and Qi Zhu. "Genetic algorithm-based optimization of offloading and resource allocation in mobile-edge computing." *Information* 11, no. 2 (2020): 83. https://doi.org/10.3390/info11020083

[43] Bozorgchenani, Arash, Farshad Mashhadi, Daniele Tarchi, and Sergio A. Salinas Monroy. "Multi-objective computation sharing in energy and delay constrained mobile edge computing environments." *IEEE Transactions on Mobile Computing* 20, no. 10 (2020): 2992-3005. https://doi.org/10.1109/TMC.2020.2994232

[44] Ma, Rui, and Xuefeng Chen. "Intelligent education evaluation mechanism on ideology and politics with 5G: PSO-driven edge computing approach." *Wireless Networks* 29, no. 2 (2023): 685-696. https://doi.org/10.1007/s11276-022-03155-x

[45] Li, Shun, Haibo Ge, Xutao Chen, Linhuan Liu, Haiwen Gong, and Rui Tang. "Computation offloading strategy for improved particle swarm optimization in mobile edge computing." In *2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pp. 375-381. IEEE, 2021. https://doi.org/10.1109/ICCCBDA51879.2021.9442609

[46] Chen, Yu. "MEC Network Resource Allocation Strategy Based on Improved PSO in 5G Communication Network." *International Journal on Semantic Web and Information Systems (IJSWIS)* 19, no. 1 (2023): 1-17. https://doi.org/10.4018/IJSWIS.328526

[47] Bacanin, Nebojsa, Milos Antonijevic, Timea Bezdan, Miodrag Zivkovic, K. Venkatachalam, and Sharaf Malebary. "Energy efficient offloading mechanism using particle swarm optimization in 5G enabled edge nodes." *Cluster Computing* 26, no. 1 (2023): 587-598. https://doi.org/10.1007/s10586-022-03609-z

[48] Cao, Bin, Long Zhang, Yun Li, Daquan Feng, and Wei Cao. "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework." *IEEE Communications Magazine* 57, no. 3 (2019): 56-62. https://doi.org/10.1109/MCOM.2019.1800608

[49] Almashhadani, Hasnain Ali, Xiaoheng Deng, Suhaib Najeh Abdul Latif, Mohammed Mohsin Ibrahim, and Osama H. Ridha AL-hwaidi. "Deploying an efficient and reliable scheduling for mobile edge computing for IoT applications." *Materials Today: Proceedings* 80 (2023): 2850-2857. https://doi.org/10.1016/j.matpr.2021.07.050

[50] Shakarami, Ali, Ali Shahidinejad, and Mostafa Ghobaei-Arani. "A review on the computation offloading approaches in mobile edge computing: A g ame-theoretic perspective." *Software: Practice and Experience* 50, no. 9 (2020): 1719-1759. https://doi.org/10.1002/spe.2839

[51] Hussain, Md Muzakkir, Ahmad Taher Azar, Rafeeq Ahmed, Syed Umar Amin, Basit Qureshi, V. Dinesh Reddy, Irfan Alam, and Zafar Iqbal Khan. "SONG: a multi-objective evolutionary algorithm for delay and energy aware facility location in vehicular fog networks." *Sensors* 23, no. 2 (2023): 667. https://doi.org/10.3390/s23020667
[52] Lammari, Amina, and Malika BESSEDIK. "Contribution of Three-Neighborhood Local Search on the Design of a Genetic Algorithm Applied to the Optimization of Computational Offloading in Mec."
[53] Saeik, Firdose, Marios Avgeris, Dimitrios Spatharakis, Nina Santi, Dimitrios Dechouniotis, John Violos, Aris Leivadeas, Nikolaos Athanasopoulos, Nathalie Mitton, and Symeon Papavassiliou. "Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions." *Computer Networks* 195 (2021): 108177. https://doi.org/10.1016/j.comnet.2021.108177
[54] Koza, John R. "Survey of genetic algorithms and genetic programming." In *Wescon conference record*, pp. 589-594. Western Periodicals Company, 1995.
[55] European Telecommunications Standards Institute (ETSI). (2020). https://www.etsi.org/technologies/multi-access-edge-computing.
[56] Huang, Liang, Xu Feng, Anqi Feng, Yupin Huang, and Li Ping Qian. "Distributed deep learning-based offloading for mobile edge computing networks." *Mobile networks and applications* (2018): 1-8. https://doi.org/10.1007/s11036-018-1177-x
[57] Wang, Jianyu, Jianli Pan, Flavio Esposito, Prasad Calyam, Zhicheng Yang, and Prasant Mohapatra. "Edge cloud offloading algorithms: Issues, methods, and perspectives." *ACM Computing Surveys (CSUR)* 52, no. 1 (2019): 1-23. https://doi.org/10.1145/3284387
[58] D. E. Goldberg and J. H. Holland, ``Genetic algorithms and machine learning,'' in *Proc. Annu. Conf. Comput. Learn. Theory*. (1988).
[59] Kennedy, James. "Swarm intelligence." In *Handbook of nature-inspired and innovative computing: integrating classical models with emerging technologies*, pp. 187-219. Boston, MA: Springer US, 2006. https://doi.org/10.1007/0-387-27705-6_6
[60] Vieira, Rafael F., Daniel Da Silva Souza, Marcelino Silva Da Silva, and Diego Lisboa Cardoso. "A heuristic for load distribution on data center hierarchy: A MEC approach." *IEEE Access* 10 (2022): 69462-69471. https://doi.org/10.1109/ACCESS.2022.3185992
[61] Abd-El-Wahed, Waiel F., A. A. Mousa, and Mohammed A. El-Shorbagy. "Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems." *Journal of Computational and Applied Mathematics* 235, no. 5 (2011): 1446-1453. https://doi.org/10.1016/j.cam.2010.08.030
[62] Aazam, Mohammad, Sherali Zeadally, and Khaled A. Harras. "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities." *Future Generation Computer Systems* 87 (2018): 278-289. https://doi.org/10.1016/j.future.2018.04.057
[63] Mach, Pavel, and Zdenek Becvar. "Mobile edge computing: A survey on architecture and computation offloading." *IEEE communications surveys & tutorials* 19, no. 3 (2017): 1628-1656. https://doi.org/10.1109/COMST.2017.2682318
[64] Alhazmi, Omar. "A survivable internet of things scheme." *J Adv Res Comput Appl* 13, no. 1 (2018): 19-26.
[65] Haekal, Mohammad Ghifari. "An Observation on Social Acceptance of Location-Based Augmented Reality Games in Open Space."
[66] Umar, Norazah, Nurhafizah Ahmad, Jamal Othman, and Rozita Kadar. "Investigating students' acceptance of e-learning effectiveness: A case study in higher education." *Journal of Advanced Research in Computing and Applications* (2023): 1-10.
[67] A. S. Thiab, A. S. Shibghatullah,"Hypervisor Security Issues in Cloud Computing:The Need to Mitigate the Risks." *Advanced Research in Computing and Applications.* (2015).
[68] F. N. Bhatti, R. B. Ahmad, H. M. Bhatti, M. N. M. Warip. "A Review and Survey of Vehicular Network's Cybercrime and Security Approaches. " *Journal of Advanced Research in Computing and Applications.* (2016).
[69] Hesham Ahmed Abdul Mutaleb Abas. "Review of augmented reality applications in manufacturing engineering." *Journal of Advanced Research in Computing and Applications.* (2016).