SEMARAK ILMU PUBLISHING

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

New Methods for Optimal Power Allocation and Joint Resource Scheduling in 5G Network which Use Mobile Edge Computing

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

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

ARTICLE INFO

Article history:
Received 11 June 2023
Received in revised form 20 November 2023
Accepted 5 April 2024
Available online 25 June 2024

Keywords:
Computation offloading; mobile edge computing; multi-objective optimization; 5G network task offloading; genetic algorithm; particle swarm optimizationalgorithm

ABSTRACT

Mobile Edge Computing (MEC) is considered one of the enabling and promising technologies in 5G networks, especially with the massive data movement of various devices and the increased demand for computing. Here, computational offloading of tasks to edge clouds provides an effective, flexible, low-latency solution for mobile users in the network. However, the limited computing resources in edge clouds and the dynamic demands of mobile users make it difficult to schedule computing requests to appropriate edge clouds, and make the offloading process energetically expensive for devices. Therefore, it is very important to design an energy-efficient offloading strategy. To this end, we first formulate the transmission power allocation (PA) problem for mobile phone users to minimize power consumption. Using a quasi-convex technique, we address the (PA) problem using a (Hybrid GAPSO) algorithm resulting from combining the Genetic Algorithm (GA) with the Particle Swarm Optimization (PSO) algorithm. Next, we model the joint request offloading and resource scheduling (JRORS) problem as a mixed-nonlinear program to minimize the response delay of requests. The (JRORS) problem can be divided into two problems, namely the request offloading (RO) problem and the computing resource scheduling (RS) problem. Therefore, we analyze the JRORS problem as a dual decision problem and propose a multi-objective particle swarm optimization algorithm referred as (MO-PSO). The simulation results show that (HGAPSO) can save transmission power consumption and has good convergence property, and (MO-PSO) outperforms existing methods in terms of response rate and can maintain good performance in a dynamic network**.**

1. Introduction

5G is the latest generation of mobile networks deployed to facilitate emerging applications and services. This technology provides enhanced Mobile BroadBand (eMBB), massive machine-to-machine (mMTC) communications, and Ultra reliable low-latency communications (URLLC), meeting the requirements of many applications such as Autonomous Vehicle (AV) [1], Internet of Thing (IOT) technology which enable a range of applications in various fields such as smart cities, industrial automation, healthcare, transportation, agriculture [2-3], and Augmented Reality (AR) applications

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

https://doi.org/10.37934/araset.47.2.237265

which are getting more widely applied in various fields such as education, art, manufacturing field and entertainment [4-5], and E-learning system which is a helpful tool in a learning process [6].

Many of these applications have real-time service needs that have bad consequences if the response delay exceeds the tolerable latency. Due to limited resources, such as low battery capacity and limited processing power, these devices will not be able to work for a long time as mention by Heidari *et al.,* [7]. Therefore, enhancing the efficiency of real-time processing of massive amounts of data while reducing energy consumption is a major challenge. Energy efficiency is considered absolutely essential from the perspective of future Information and Communications Technology (ICT) [8].

Recently, (MEC) technology has emerged as a new paradigm to facilitate access to advanced cloud computing capabilities at the edge of the network near end devices, thus enabling a rich set of latency-sensitive services required by various devices. Recent studies showed that integrating 5G networks with (MEC) technology led to expanding the capacity of devices using offloading methods [9]. This integration overcomes the limitations of cloud computing and extends its services to the edge of the network, which may reduce the load on the main network, reduce device power consumption, and enhance scalability [10].

The task's offloading mechanism is an effective way to reduce energy consumption [11]. The energy consumed in the offloading process is divided into two parts: the energy consumed for sending data and the energy consumed for processing requests. However, the offloading process will be useless if there is a lack of homogeneity in the requirements of the tasks that have been offloaded (different computing requirements, access time, etc.) and capabilities (MEC) limited. The limited computing resources of the edge cloud may consume more energy and lead to additional delays, especially if the offloaded task contains a large amount of data or a high workload [12]. Therefore, determining the optimal offloading strategy is a complex task due to competing objectives such as minimizing energy consumption, minimizing latency, and optimizing resource allocation.

Metaheuristics and multi-objective optimization methods are powerful tools that can effectively address various challenges faced by 5G networks using edge computing. One of the main problems that can be solved using these methods is the problems associated with the computational offloading process. Metaheuristic algorithms are models of computational intelligence that are used to address complex optimization problems due to their robustness and ability to manage nonlinearity and discontinuity. In addition, multi-objective optimization algorithms intelligently balance multiple objectives and alleviate problems related to task scheduling, workload distribution, and resource management simultaneously. Therefore, adopting these technologies in 5G networks leads to more efficient and reliable systems, ensures seamless connectivity and meets the demands of the ever-evolving era [13].

In this research, first a mechanism was designed to solve the problem of uplink transmission power allocation when applying the Non-Orthogonal Multiple Access (NOMA) scheme by combining the genetic algorithm (GA) and the particle swarm optimization (PSO) algorithm. Secondly, Multi-Objective Particle Swarm Optimization (MO-PSO) was used to solve jointly task offloading and resource scheduling. Our utilized performance metrics are power consumption, response rate and system welfare, 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 [14], and the cybercrime that is probable to effect on network response, recovery, and management [15]. The contributions of this paper can be summarized as follows:

i. A review of the latest studies related to the research topic.
ii. Propose a hybrid optimization technique by combining (GA) with PSO called (HGAPSO) to achieve optimal power allocation in the offloading process.
iii. Propose a multi-objective optimization technique to jointly solve task offloading and resource scheduling.
iv. Implement a simulation process to evaluate the performance of the proposed algorithms and compare them with other techniques.

This research is organized as follows: Section 2 provides basic information related to the research. Section 3 shows relevant studies related to the research. Section 4 discusses the system model and problem formulation. The proposed algorithms are presented separately in Section 5. Section 6 presents the simulation results. In Section 7 the conclusion and future work are presented.

2. Background

2.1 Computational Offloading in MEC

The offloading process responsibility is divided among three main agents: mobile devices, communication links, and Edge Clouds (EC). 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) handle the balance of the server load to achieve maximum service rates and system throughput [16]. The structure of the (MEC) system is illustrated in Figure 1. As it is illustrated, this structure has three main layers:

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 centers are located. These servers are typically accessed via high data rate powerful communication links by a colony of geographically scattered Aps [17]. The APs themselves are usually interconnected via fiber optic.
iii. Center cloud.

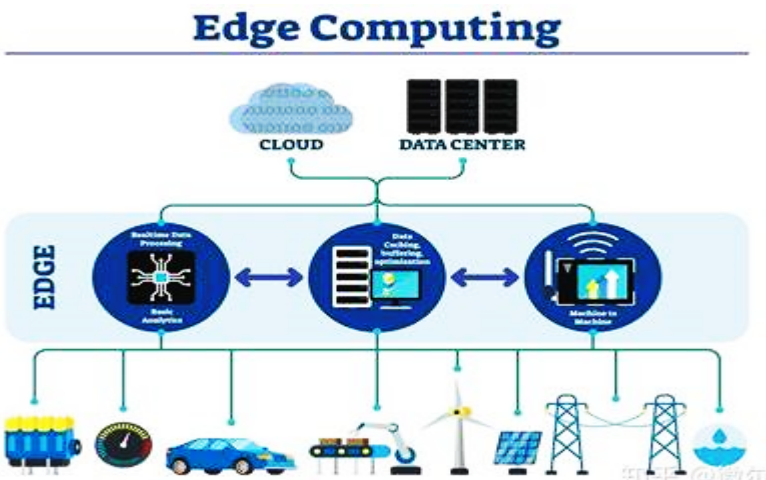


Fig.1. Structure of the Mobile Edge Computing

2.2 Metaheuristics Methods

Metaheuristic methods are generally inspired by nature. The main idea of these methods is to improve the result in a reasonable time through an iterative search process for better solutions while trying to avoid falling into local optimal solutions, unlike heuristic methods which suffer from this problem. A number of metaheuristic techniques have been proposed in the literature, such as genetic algorithms GA [18], single-objective particle swarm optimization (PSO) [19], and multi-objective particle swarm optimization (MO-PSO) [20]. These algorithms are usually based on the idea of population (solution) evolution, where the best solutions for a given goal are kept for the next evolutionary step to obtain a new generation of solutions [21].

Due to the complexities caused by the dynamics of wireless communications and computing technologies, The process of decision making and resource management to improve the efficiency of these systems and meet user requirements is becoming more complex. Especially, the incorrect offloading decisions can reduce the efficiency of the system. Since metaheuristics are strategies that guide the search process, these methods are considered very suitable for addressing the power allocation and resource scheduling problem related to the computational offloading process [13].

2.3 Ultra-Dense Networks (UDN) with Multiple Base Stations and Collaborative Services

To handle dense connections and huge data traffic between devices in Ultra Dense Network (UDN), the network operator deploys a large number of (micro-BSs) and (macro-BSs) together to provide services to mobile devices in those networks. Thus, a cooperative service scenario will occur between multiple micro-BSs and macro-BSs [22]. All micro-BS is connected to the EC via a local network, and macro-BS is connected to the resource rich deep cloud over the Internet. Mobile devices in UDN can specify that computation requests are offloaded to macro-BSs when micro-BSs are not able to process all offloaded requests. A major limitation of UDN is that all single edge cloud in micro-BSs that provide mobile access services are more computationally intensive than database macro-BSs. Therefore, in the event of a large number of offloading requests, the limited computing resources in the EC will lead to increased response time to the requests and increased power consumption. Therefore, an appropriate offloading scheme is an important issue to be resolved. In this study, the NOMA protocol was considered as a multiple access system between users and base stations within an ultra-dense 5G network, consisting of a macro-BS, many micro-BSs, and a large number of mobile users. There are many challenges related to offloading tasks in this system, such as transmission power allocation and resource scheduling issues, so it is necessary to address them to take full advantage of the offloading process.

3. Related works

In this section, recent papers on computation offloading in EC will be reviewed. They can be classified into two parts according to used algorithms:

3.1 Using Hybrid Genetic Algorithm with Particle Swarm Optimization

Bi *et al.*, [23] proposed a genetic particle swarm optimization (GPSO) algorithm to solve total energy consumption minimization 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.

Ezhilarasie *et al.,* [24] 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.

Menbawy *et al.,* [25] 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 was 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.

Guo *et al*., [26] 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 and particle swarm optimization, they design a suboptimal algorithm named as hierarchical GA and PSO-based computation algorithm to solve this problem.

Truong *et al.,* [27] investigated in a performance of MEC surveillance systems using 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. Then they proposed a 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.

Bi *et al.,* [28] 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 center and the edge, and resource allocation in the data center.

Chen *et al.,* [29] 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.

3.2 Using Multi-Objective Optimization

Hussain *et al.,* [30] 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, they solve an example problem using the SONG algorithm to illustrate the delay–energy solution frontiers and plotted the corresponding layout topology. Subsequently, they 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. The empirical results showed that SONG exhibits improved solution quality over the NSGA-II and SMPSO algorithms and hence can be utilized as a potential tool by the service providers for the planning and design of VFC networks.

Almasri *et al*., [31] proposed a multi-objective optimization solution to assign different application tasks to different edge devices while minimizing the energy consumption of edge devices and the computation time of tasks. Task dependencies and data distribution were considered within a new and more general MEC model. Multi-objective evolutionary algorithm (MOEA) framework was used to solve the optimization problem subject to deadline and power consumption constraints. Results showed that the proposed multi-objective approach achieved better performance in terms of energy and computation time when compared to a single objective approach.

Peng *et al*., [32] studied the optimization problem of the UAV from a multi-objective viewpoint by considering the UAV’s flight safety which provide offloading services for nearby devices. A constrained multi-objective optimization problem (CMOP) involving two objective functions about the energy-efficient offloading and safe path planning was formulated for the UAV. To solve the formulated CMOP, they presented a constrained decomposition-based multi-objective evolution algorithm. To further improve the algorithm, they particularly utilized the infeasible individuals with great objective values, which provided useful information for improving the optimized objective values during the evolution process. experimental results demonstrated that, this scheme is beneficial to simultaneously reduce energy consumption and ensure safe flight for the UAV.

The response time of the computing tasks, the energy consumption of the mobile terminal device and the load balance of the server were regarded as three optimization objectives by Zhu *et al*., [33]. Here a multi-objective optimization model was set up, and an offloading decision scheme based on multi-objective optimization immune algorithm was proposed. A large number of comparative experiments were done to verify the effectiveness of proposed scheme. Experimental results showed that the proposed scheme can make the whole server system achieve a better load balancing state while meeting the requirements of response time and energy consumption.

Jinglei *et al.*, [34] investigated in a multi objective task scheduling problem in MEC-aided 6G network where computation-intensive applications that were commonly modelled as Directed Acyclic Graphs (DAG) can be performed locally and offloaded to MEC servers to enhance execution efficiency. Then, an improved multi objective cuckoo search (IMOCS) algorithm was proposed to deal with a DAG-based task scheduling problem, which aims to reduce the execution latency and energy consumption of UE. Particularly, the proposed IMOCS algorithm was based on the single-objective cuckoo search algorithm and Pareto dominance. An external archive was used to record nondominated solutions, whose update strategy improved the quality of solutions by the aid of fast nondominated sorting and crowding distance sorting. Simulation results demonstrated that IMOCS algorithm outperforms other four benchmark algorithms, which can provide optimal task scheduling policy for MEC severs in 6G networks.

Asghari *et al.,* [35] investigated that the proper placement of mobile cloud resources has an important impact on their efficiency and energy consumption. The appropriate resource placement model can reduce latency and improve energy consumption. Because of the large number of mobile servers, finding the best geographical placement of all resources is an NP-Hard problem, so researchers have introduced a novel multi-objective edge server placement algorithm using the trees social relations optimization algorithm (TSR) and the DVFS (dynamic voltage and frequency scaling) technique (MSP-TD), has been introduced for optimal placement of edge servers to extend the network coverage. Parallelization methods can improve the scalability of the resource placement problem and reduce the time complexity of finding the optimal solution. The simulation results showed that our proposed model leads to less latency and energy consumption reduction than some state-of-the-art and similar algorithms.

Zhu *et al.,* [36] investigated in task offloading decisions when the system's mobile and service device count rises. which become a significant difficult. So, they modelled the problem of response time and energy consumption of the system as a multi-objective optimization problem, and they designed an improved evolutionary algorithm based on immune algorithm, which can effectively obtain a set of solutions between response time and energy consumption. The simulation results showed that this scheme can meet the response time requirements and obtain a lower energy consumption strategy when compared to the offloading scheme in the existing literature.

A new multi-objective strategy based on the biogeography-based optimization (BBO) algorithm was proposed for MEC offloading to satisfied users' multiple requirements (the execution time, energy consumption and cost) by Li *et al.*, [37]. In this strategy, a time-energy consumption model and a cost model were constructed for task offloading firstly. Based on these models, the BBO algorithm was introduced into task offloading for MEC to solve the problem of multi-objective optimization. Compared with the traditional strategies, the offloading strategy based on BBO decreased the average task completion time by an average of 25.03%, and compared with the technique for order preference by similarity to an ideal solution (TOPSIS) strategy, the BBO offloading strategy proposed in this paper reduced energy consumption 75% and cost by 36.9%. The proposed strategy can well solve the problem of multi-objective optimization in the task offloading for MEC.

4. System Model

We consider an 5G network consisting of a set of mobile users *U*, a set of micro-BSs (micro-BSs are abbreviated as BSs in the following) with edge clouds *N*, and a macro-BS with a deep cloud C. As shown in Figure 2. The system problems are formulated as shown in Figure 2, but we will solve those problems by using different algorithms.

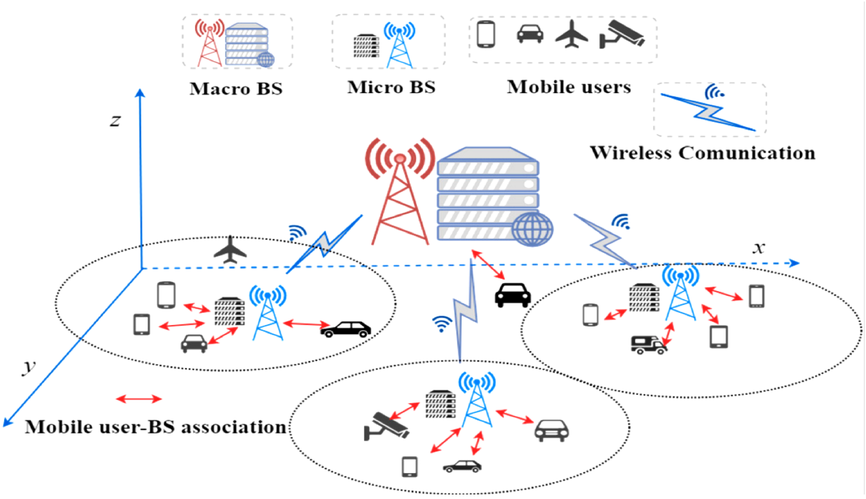


Fig. 2. System model [38]

It is assumed that each BS covers a local area called a zone, and a mobile user should be associated with only one zone. Edge server may be a physical server or a virtual machine with computing capacities, and we assume that its associated BS is interconnected by backhaul links, allowing a mobile user to be served by a nonlocal BS. Each mobile user can offload computing request to a BS in its zone. we assume that the macro-BS is used as the central controller, which is responsible for collecting task information, computing resource information of edge clouds in BSs, and the network status. Specially, the set of mobile users and BSs are denoted by $U = \{1, 2, \dots, u\}$ and $N = \{1, 2, \dots, n\}$, respectively. We assume each mobile user $u \in U$ generate one computing request at a time, given as $q_u = \langle w_q, s_q, pr_q, Tg_q, Tb_q \rangle$. Here, w_q denotes the workload of request *q*, i.e., the required computing to accomplish the request, and s_q denotes the request input data size. We use pr_q to denote the request priority representing the importance of different requests. Tg_q and Tb_q are ideal delay and tolerable delay thresholds, respectively [38].

Considering the position of mobile user varies over time, we use $p_u^t = (x_u, y_u, 0)$ to denote the location of mobile user *u* at time *t*. All BSs are fixed and the location of BS *n* is given as $p_n^t = (x_n, y_n, H)$ with the same attitude *h*.

4.1 Delay Model

In this paper, Non-Orthogonal Multiple Access (NOMA) scheme was applied as the communication scheme between mobile users and BSs. Therefore, mobile users in the same zone can transmit data to BS simultaneously at the expense of the interference. In this case, the interference may cause performance degradation, i.e., the decrease of uplink rate. Suppose that the location of each mobile user is unchanged during the time interval, The uplink rate $v_{un}(t)$ from mobile user *u* to BS *n* can be formulated as follows [38]:

$$v_{un}(t) = Blog_2\left(1 + \frac{p_{un}(t)g_{un}(t)}{\sigma_0^2 + \sum_{u'}^{U_n} p_{u'n}(t)g_{u'n}(t)}\right) \tag{1}$$

where p_{un} denotes the transmitting power from mobile user *u* to BS *n*, *B* and σ_0^2 represent the bandwidth of the uplink system and background white Gaussian noise power respectively. The channel power gain between mobile user *u* to BS *n* is defined as follows [39]:

$$g_{un}(t) = \frac{g_0}{\left(x_u - x_n(t)\right)^2 + \left(y_u - y_n(t)\right)^2 + H^2}, u \in U, n \in N, t \in T \tag{2}$$

where g_0 represents the channel power gain at the reference distance $d_0 = 1$ m and the transmitting power is 1W. Suppose that x_{qn} is a binary variable, in which $x_{qn} = 1$ indicates that request *q* is offloaded to BS *n*, and $x_{qn} = 0$ indicates the request *q* is offloaded to macro-BS. Thus, the time taken to transmit data I_q from mobile user *u* for offloading is given as:

$$t_{up}^q = \begin{cases} \frac{I_q}{v_{un}(t)}, & x_{qn} = 1 \\ \frac{I_q}{v_{un}(t)}, & x_{qn} = 0 \end{cases} \tag{3}$$

Suppose R_{qn} denotes the amount of computing resource that BS *n* schedules to request *q*. Thus, the execution time of request *q* at BS or macro-BS is given as:

$$t_{pro}^q = \begin{cases} \frac{I_q}{R_{qn}}, & x_{qn} = 1 \\ \frac{I_q}{R_C}, & x_{qn} = 0 \end{cases} \tag{4}$$

where R_C is the computing capacity of macro-BS. Therefore, the total delay for offloading request *q* is assigned:

$$t_q = t_{up}^q + t_{pro}^q \tag{5}$$

4.2 Energy Model

The energy consumption for offloading requests includes the energy consumed for transmitting the data and the energy consumption of processing requests. Thus, the transmitting energy consumption for data offloading from mobile user *u* to BS *n* at time *t* is defined as:

$$E_u^{tra}(t) = p_{un}(t)\, t_{up}^q \tag{6}$$

Given the average power consumption of BS and macro-BS, the energy consumed by executing request *q* is defined as:

$$E_u^{pro}(t) = \begin{cases} p_{BS}\ t_{pro}^q\ , & x_{qn} = 1 \\ p_C\ t_{pro,c}^q\ , & x_{qn} = 0 \end{cases} \tag{7}$$

where p_{BS} and p_C are the average power consumption of BS and macro-BS, respectively.

4.3 Problem Formulation

Assume that every mobile user aims to reduce its power consumption allocated to data transmission. Since the protocol applied in this study is NOMA as a communication scheme, i.e. mobile users can send data simultaneously using the bandwidth of the entire system. In this case the transmission delay can be reduced for mobile users who use more transmission power, but this may result in more interference and power consumption. Therefore, to reduce the power consumption needed to transmit data of the entire system at time t, we formulate the power allocation (PA) problem as in Reference [38]:

$$P1\text{: } \min_P = \sum_n^N \sum_u^U E_u^{tra} \tag{8a}$$

$$s.t\quad 0 \leq P_{un}(t) \leq P_{max}\ ،\ \forall\, n \in N ،\ u \in U \tag{8b}$$

The objective in Eq. (8a) minimizes the power consumption of data transmission using $E_u^{tra}(t)$ given in Eq. (6). While constraint in Eq. (8b) ensures that the transmission power for each mobile user is less than P_{max} and greater than (0). Mobile users in the same zone compete for the computing resources of the same BS to complete the requests within the ideal delay. Referring to [40], we define the edge system utility for processing request q as:

$$k_p = \begin{cases} 1 & t_q \leq Tg_q \\ 1 - \frac{1}{1+e^{\alpha(T_{avg}-t_q)/(T_{avg}-Tg_q)}} & Tg_q < t_q \leq T_{avg} \\ \frac{1}{1+e^{\alpha(t_q-T_{avg})/(Tb_q-T_{avg})}} & T_{avg} < t_q \leq Tb_q \\ 0 & t_q > Tb_q \end{cases} \tag{9}$$

where

$$T_{avg} = \frac{Tg_q + Tb_q}{2} \tag{10}$$

and the edge system cost for processing request q is defined as [38]:

$$c_q = \alpha \int_{E_0 - E_r^t}^{E_0 - E_r^t + E_u^{pro}} e^{x/10}\, dx \tag{11}$$

where α is a user-defined constant to ensure that c_q is in the range [0, 1], E_0 and E_r^t are the initial energy and residual energy at time t of BS. With the increase of power consumption of executing requests, the energy cost c_q of the edge server is increased. Given the fixed computing resources, the BS may not be able to process all requests in a timely manner.

Therefore, mobile users can choose to send the request to the macro-BS for processing, and the edge system should pay for this work. The extra cost for offloading to macro-BS is defined as:

$$e_q = \varepsilon k_p + (1-\varepsilon)E_q^{pro} \tag{12}$$

where ε is a constant implying the relative importance of total delay and executing power consumption. Thus, we define the total system welfare as:

$$W = \sum_n^N \sum_q^Q \left[x_{qn}\left(k_p - c_p\right) - \left(1 - x_{qn}\right)e_q\right] \tag{13}$$

The joint request offloading and computing resource scheduling problem is formulated as a system welfare maximization problem:

$$P_2\colon \max_{X,Y} W \tag{14}$$

$$s.t\colon \ \sum_{n\in N} x_{qn} \ \leq 1 \ ,\forall \ q \in Q \tag{14a}$$

$$x_{qn} \in \{0,1\} \ \forall \ q \in Q \ , n \in N \tag{14b}$$

$$\sum_{q\in Q} R_{qn} \leq R_n \ \ ,\forall \ n \in N \tag{14c}$$

$$R_{qn} > 0, \forall \ q \in Q \ , n \in N \tag{14d}$$

Constraint in Eq. (14a) and constraint in Eq. (14b) imply that each request generated by mobile user can be either offloaded to only one BS or macro-BS. Given the fixed computing resources, the BS may not be able to process all requests in a timely manner. Therefore, mobile users can choose to send the request to the cloud center for processing. Constraint in Eq. (14c) ensures that the total computing resources scheduled to requests should not exceed the BS's computing capacity. Constraint in Eq. (14d) ensures that BS must schedule a positive computing resource to each request that offloaded to it.

5. Efficient Algorithms

5.1 Power Allocation (PA)

The (PA) problem can be expressed as follows:

$$\frac{p_{un}(t)I_u}{v_{un}(t)} = \min_p E \tag{15}$$

$$= \sum_n^N \sum_u^U \sum_n^N \sum_u^U \frac{p_{un}(t)I_u}{B\log_2\left(1+\frac{p_{un}(t)g_{un}(t)}{\sigma_0^2+\sum_{\acute{u}\neq u}^{Un} p_{\acute{u}n}(t)g_{\acute{u}n}(t)}\right)} \tag{15a}$$

$$s.t \quad 0 \leq P_{un}(t) \leq P_{max} \text{،} \ \forall \ n \in N \text{،} \ u \in U \tag{15b}$$

Problem in Eq. (15) is difficult to solve because the objective function in Eq. (15a) is nonlinear, and the term $v_{un}(t)$ depends on the transmission power $p_{un}(t)$ and on $g_{un}(t)$ associated with the location of other mobile users in the same area, Assume that each base station calculates the power allocation (PA) to its associated mobile users $(\boldsymbol{P_n})$ independently to minimize the power consumption $(\boldsymbol{E_n})$ every time (t). Then the (PA) problem can be solved by solving a set of sub-problems as follows [38]:

$$\min_{p_n} E_n = \sum_u^{U_n} \phi(p_{un}) = \sum_u^{U_n} \frac{p_{un} I_u}{B \log_2(1+\gamma.p_{un})} \tag{16a}$$

$$s.t \quad 0 \le P_{un}(t) \le P_{max} \text{، } \forall\, n \in N \text{، } u \in U \tag{16b}$$

where

$$\gamma = \frac{g_{un}(t)}{\sigma_0^2 + \sum_{\acute{u} \neq u}^{U_n} p_{\acute{u}n}(t) g_{\acute{u}n}(t)} \tag{17}$$

The second-order derivative of the objective in Eq. (16a) with regards $(\boldsymbol{p_{un}})$ is not always positive, so the problem in Eq. (16a) is non-convex. Therefore, it cannot be solved with standard techniques such as the Lagrange multiplier, Lyapunov stochastic method, or successive convex approximation techniques as it is a non-convex problem. This problem is well suited for the use of metaheuristic algorithms, because these algorithms are one of the models of computational intelligence that are used to address complex optimization problems due to their ability to manage nonlinearity and discontinuity. There are many types of metaheuristic algorithms such as GA, PSO, and others. We presented a previous survey on these two algorithms and their hybrid algorithm [41]. We compared these algorithms based on different performance metrics, use cases, and evaluation tools, as well as discussing the strengths and weaknesses of each algorithm. However, each algorithm has its advantages and disadvantages. For example, PSO convergence is fast, but can be limited to locally optimal solutions when used to solve complex problems with high-dimensional solution spaces. In addition, GA has great ability to search comprehensively and provide diverse solutions, but their convergence process takes a long time. Therefore, in this study, we propose a hybrid optimization technique based on GA with PSO called (HGAPSO) to solve the power allocation (PA) problem.

Figure 3 shows the main steps of the HGAPSO algorithm. It will be implemented in two stages:

i. **The first stage:** HGAPSO begins by implementing the GA algorithm up to a specified number of iterations, where the solutions are developed until reaching an appropriate standard through successive iterations, i.e. reaching the pre-determined fitness value or reaching the maximum number of generations.
ii. **The second stage:** The PSO algorithm then starts. Once the optimized solutions for the GA are available, the particles will be initialized, where the position of the particles reflects the solutions of the GA algorithm and the speed of each particle is initially initialized.

The HGAPSO is formulated as shown in Table 1. Here GA was used as the basis for the algorithm while PSO is used to enhance the solution provided by GA and improve its performance.

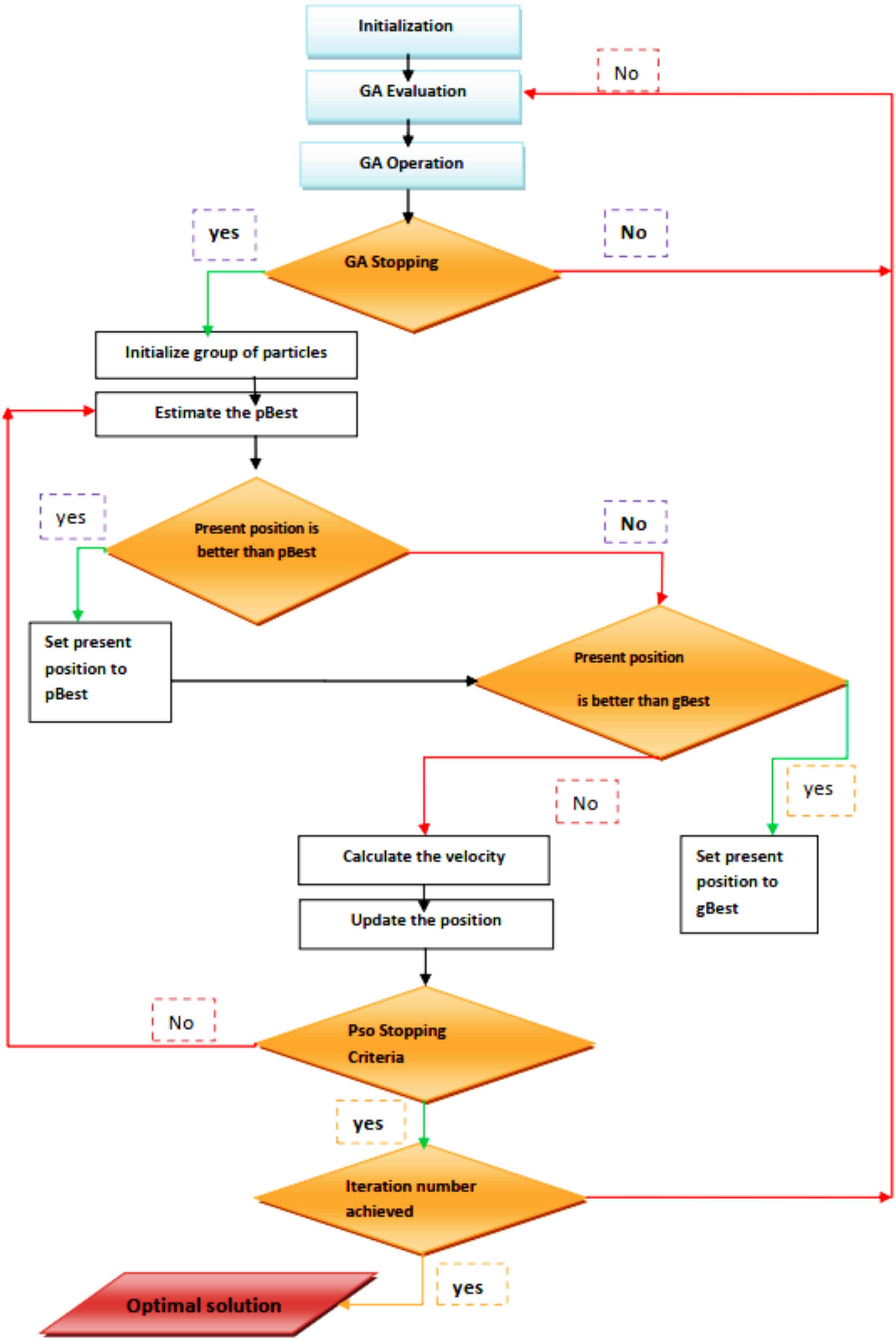


Fig. 3. Flow chart of the HGAPSO algorithm used in power allocation

Table 1
Hybrid Optimization Method based on GA with PSO

Algorithm 1: Hybrid Optimization Method based on GA with PSO
Input: the size of population S، the number of variables N، THE Max iteration of HGAPSO I، the parameters of GA P_m ، P_C، I_1، tournament size t_s، the parameter of PSO w، C_1 ،C_2، I_2
Output: the optimal values of v_{un}، x_{qn}، and E
Begin
Randomly generate individuals in initial population P and assign number of generations to 0 $(i = 0)$
While termination criteria are not satisfied do
$perform\ GA\ operation\ in\ Algorithm\ 2$
$perform\ PSO\ operation\ in\ Algorithm\ 3$
$i = i + 1$
End
Obtaining the optimal values of v_{un}، x_{qn}
The Optimal power allocation can be obtained by equation (10) with optimal values from in the previous step
END

5.1.1. Genetic algorithm (GA) scheme

Algorithm 1 starts by randomly initializing the population. Individuals are then processed through GA operations in Algorithm 2 (Table 2) up to $(\mathbf{I_1})$number of iterations. Here, GA uses a random search strategy that mimics biological evolution, which uses the idea of survival of the fittest as an evolutionary concept. Selection, crossover and mutation are the three basic factors in the GA model.

Table 2
Genetic Algorithm operations

Algorithm 2: Genetic Algorithm operations
Input: the population P ، P_m ، P_C، I_1، tournament size t_s، MAX iteration I_1
Output: the S solutions after I_1 iterations
Begin
Calculate the fitness of individuals and assign number of generations to 0 $(i = 0)$
While termination criteria are not satisfied do
$select\ parents\ by\ the\ tourmanet\ selection$
$Apply\ crossover\ by\ probability\ P_c$
$Apply\ mutation\ by\ probability\ P_m$
$Evaluate\ the\ new\ candidates$
$i_1 = i_1 + 1$
End

Chromosome structure and GA processes:

i. **Initial stage:** Each individual is represented by a chromosome, which is a string (usually binary or decimal) that encodes the solution [42]. Figure 4 shows the method of chromosome coding. Since a population matrix consists of a number of individuals, each individual has a set of genes $\{x_{qn}, \mathbf{v}_{\mathbf{un}}\}$. In each iteration, the individual with the lowest value of the allocated power is chosen as the optimal individual.

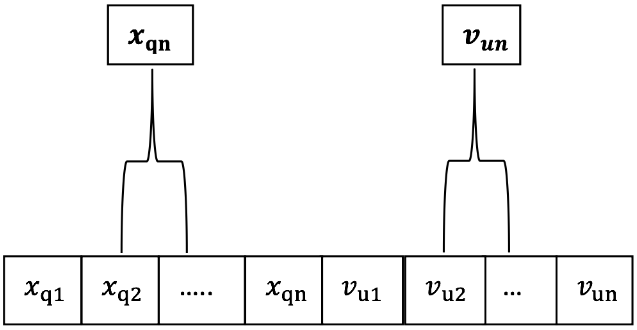


Fig. 4. Coding of individuals

ii. **Selection Process:** There are two widely used methods for the selection process, namely roulette wheel and tournament [42]. This study adopts the tournament method because the roulette wheel selection technique is more suitable for maximization problems. In tournament selection, N individuals are randomly selected and the fittest is chosen to become a parent. A similar process is then performed to select the new parent.

iii. **Crossing Process:** Parents use crossing procedures to generate new offspring to enhance diversity and provide better solutions to this problem. Crossing helps in improving and promoting convergence [42]. Crossing over is performed when parents are exchanged by choosing a random point on the chromosome. After that, hybridization occurs with the formation of new offspring according to the chosen point of intersection with certain parts of the parents. Figure (5) shows an example of a single-point intersection process. Whether or not a crossover will occur is decided based on the crossover probability. The results of practical and theoretical research indicate a much greater probability of intersection in the field {Pc=0.95-0.6}.

iv. **Mutation process:** It is described as a small random modification of chromosomes in order to find a new solution. It is used in order to maintain population diversity and create new adaptive individuals that prevent local optima. It is generally used with low probabilities around {Pm=0.001-0.05}. For an integer representation, a gene chosen at random from a list of possible values is assigned a random value. Figure 6 shows the exact procedure of the operation. Note that a new random value is assigned to the mutation if the modified gene is higher than the restriction limit.

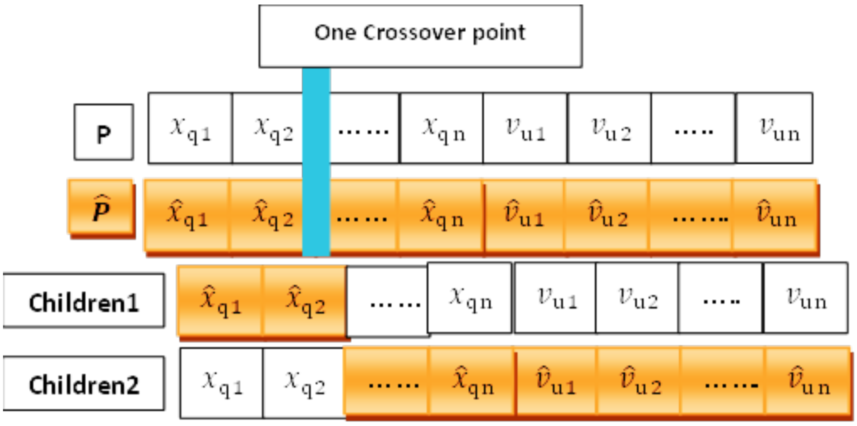


Fig.5. Example of one intersection process

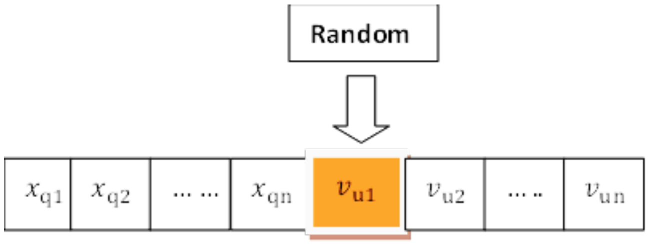


Fig.6. Mutation process

5.1.2 *Particle swarm optimization (PSO) scheme*

PSO particles are then initialized once the optimized solutions of GA from Algorithm 2 are provided and implemented PSO operations in Algorithm 3 (Table 3) up to ($\mathbf{I_2}$) number of iterations. This approach searches for the best solution using agents called particles. A collection of moving particles is called a swarm. Particles have only two properties, namely position and velocity, as each particle adjusts its position in the search space in response to the motion experiences of the surrounding particles in order to achieve the best suitable position [43]. Note that the position of the particle in PSO represents the solution in the GA population, while the speed indicates the rate of change in the position of the particle. Each particle has a record that recalls its previous best position called ($\mathbf{p}_{\boldsymbol{Best}}$). The particle with the highest fitness value is referred to as the best group position ($\mathbf{g}_{\mathbf{Best}}$).

Assuming that the search space has dimensions S, then the position of particle (i) in the swarm is expressed by the vector: ($\boldsymbol{X_i} = \boldsymbol{X_{i،1}}،\ \boldsymbol{X_{i،2}}، \ldots ،\ \boldsymbol{X_{i،s}}$)، Another dimensional vector (S) is used to describe the speed of this particle: ($\boldsymbol{V_i} = \boldsymbol{V_{i،1}}،\ \boldsymbol{V_{i،2}}، \ldots ،\ \boldsymbol{V_{i،s}}$)، each particle adjusts its position according to the new velocity with each iteration. The particle's velocity and position are updated as follows:

$$V_{i،d}^{t+1} = \omega V_{(i،d)}^{t} + r_1.c_1\left(pBest_{(i،d)}^{t} - X_{(i،d)}^{t}\right) + r_2.c_2\left(gBest_{(i،d)}^{t} - X_{(i،d)}^{t}\right) \tag{18}$$

$$X_{i،d}^{t+1} = X_{i،d}^{t} + V_{i،d}^{t+1} \tag{19}$$

where (t) represents the repetition number, and $\mathbf{d} = \{\mathbf{1}،\mathbf{2}،\ \mathbf{3}،\ \ldots ،\ \mathbf{S}\}$ represents the number of dimensions; And $\boldsymbol{i} = \{\mathbf{1}،\ \mathbf{2}،\ \ldots ،\ \boldsymbol{M}\}$, where (M) is the swarm size, and (ω) is the inertia, which means the weight of the particle at its previous speed. ($\mathbf{c_1}$) and ($\mathbf{c_2}$) are acceleration constants. ($\boldsymbol{r_2}$) and ($\boldsymbol{r_1}$) are two random parameters that take their value within (1-0) to increase randomness in the search. The particle is directed to its optimal position through a set of acceleration parameters represented by the constants $(\mathbf{c_1})$ and $(\mathbf{c_2})$. If the values of $(\mathbf{c_1})$ and $(\mathbf{c_2})$ are large, a rapid search will occur, and thus the ideal solutions may be neglected. If the values of $(\mathbf{c_1})$ and $(\mathbf{c_2})$ are low, the search time will be slow [43], and the local optimal solution can be found. So, we will assume that $(\mathbf{c_1})$ = 2 and $(\mathbf{c_2})$ = 2. Individuals are iteratively improved using GA and PSO until convergence or the maximum number of generations is reached.

Table 3
Particle Swarm Optimization operations

Algorithm 3: Particle Swarm Optimization operations
Input: The best solution (particles) produced by algorithm 2, inertia weight (w),C_1 , C_2, MAX iteration I_2
Output: The position of S particles after I_2 iterations
Begin
Initialize the velocity of S particles and assigning number of generations to 0 $(i_2 = 0)$
While termination criteria are not satisfied do
$\quad$ $Calculate\ the\ fitness\ function\ of\ particles$
$\quad$ $Update\ the\ P_{best}\ and\ g_{best}\ for\ each\ particles$
$\quad$ $Update\ the\ velocity\ of\ the\ particles$
$\quad$ $Update\ the\ position\ of\ the\ particles$
$\quad$ $i_2 = i_2 + 1$
End

5.2 *Joint Request Offloading and Resource Scheduling*

After allocating transmission power for mobile users, Delay-sensitive requests from users must be offloaded to BS or Macro-BS. In particular, the problem can be expressed as Eq. (14), We note that constraints (a) and (b) for offloading policy X, and constraints (c) and (d) for offloading policy Y are separate from each other. Problem in Eq. (14) can be divided into two problems, namely the request offloading (RO) problem and the computational resource scheduling (RS) problem. Thus, the RO problem of minimizing the extra cost of the edge system can be expressed as:

$$\min_{X} M = \sum_{n}^{N} \sum_{q}^{Q} \left(1 - x_{qn}\right)\left(\varepsilon k_p + (1-\varepsilon) E_q^{pro}\right) \tag{20}$$

$$s.t{:}\ \ \sum_{n \in N} x_{qn} \le 1 , \forall\ q \in Q \tag{2a}$$

$$x_{qn} \in \{0,1\}\ \forall\ q \in Q , n \in N \tag{2b}$$

The RS problem of maximizing the welfare of the edge system can be expressed as:

$$\max_{Y} W = \sum_{n}^{N} \sum_{q}^{Q} \left[x_{qn}\left(k_p - c_p\right)\right] \tag{21}$$

$$\sum_{q \in Q} R_{qn} \le R_n \ \ , \forall\, n \in N \tag{21a}$$

$$R_{qn} > 0, \forall\ q \in Q , n \in N \tag{21b}$$

Therefore, this problem is a dual decision-making problem which is very complex and involves a trade-off between two conflicting objectives. In this paper, we propose a multi-objective optimization algorithm based on Particle Swarm Optimization, referred to as (MOPSO), to solve the problem which is divided into Problem (20) and Problem (21).

Multi-objective particle swarm optimization (MOPSO), which arises from simulating the behavior of bird flocks, is one of the most promising stochastic research methodologies due to its ease of implementation and high convergence speed. The MOPSO algorithm intelligently sifts the large amount of information contained within each particle representing a filter solution and exchanges information to increase the overall quality of particles in the swarm. In the multi-objective optimization problem, there is one best solution for each objective. However, this solution may not achieve all other goals. One of the main differences between single objective (SO) and Multi-objective (MO) optimization is that MO problems form a multi-dimensional objective space. This leads to three possible states of the MO problem, depending on whether the goals are completely conflicting, not conflicting, or partially conflicting [20]:

i. **First category:** The conflicting nature of the objectives are such that no improvements can be made without violating any constraints. This result in an interesting situation where all feasible solutions are also optimal.
ii. **Second category:** A nonconflicting MO problem if the various objectives are correlated and the optimization of any arbitrary objective leads to the subsequent improvement of the other objectives. This class of MO problem can be treated as a SO problem by optimizing the problem along an arbitrarily selected objective or by aggregating the different objectives into a scalar function. Intuitively, a single optimal solution exists for such a MO problem.
iii. **Third category:** A partially conflicting objectives which more often than not, real world problems are instantiations of this type and this is the class of MO problems that we are interested in. One serious implication is that a set of solutions representing the tradeoffs between the different objectives rather than a unique optimal solution. Assuming that the two objectives are indeed partially conflicting, this presents at least two possible extreme solutions, one for lowest cost and one for highest performance. The other solutions, if any, represent the varying degree of optimality with respect to these two objectives [20].

5.2.1 Basic MOPSO

The general MOPSO framework can be represented as in the algorithm 4 (Table 4). There are many similarities between SOPSO and MOPSO with both techniques involving an iterative adaptation of a set of solutions until a pre-specified optimization goal/stopping criterion is met [44]. What sets these two techniques apart is the manner in which solution assessment and gbest selection are performed. in addition to incorporation of elitism which consider one of the distinct features that characterizes the MOPSO algorithms. The different MOPSO algorithms can be distinguished by the way in which the mechanisms of elitism and diversity preservation are implemented. Elitism in MOPSO involves two closely related process, 1) the archiving of good solutions and 2) the selection of gbest for each particle from these solutions. Before presenting the MOPSO algorithm, we provide an explanation of the components of this algorithm.

Table 4
MO-PSO

Algorithm 4: MO-PSO
P ← Particle initialization
A ← Create archive while (stopping criteria not satisfied)
P ← Evaluate (P)
A ← Update (A)
P ← Select pbest (P)
P ← Select gbest (g)
P ← Update (P)
END While

5.2.2 MOPSO Components

The framework presented in the previous section serves to highlight the primary components of MOPSO, elements without which the algorithm is unable to fulfil its basic function of finding Pareto optimal front satisfactorily. The main components of this algorithm are**:**

i. **Fitness Assignment**: Based on the literature, it is possible to identify two different classes of fitness assignment: 1) Pareto based assignment, 2) aggregation-based assignment. At this point, it seems that Pareto based fitness are more effective in low dimensional MO problems while aggregation-based fitness has an edge with increasing number of objectives. Naturally, some researchers have attempted to marry both methods together. For example, the reference study in [45] proposed a hybrid MO fitness assignment method which assigns a nondominated rank that is normalized by niche count and an aggregation of weighted objective values. Since the example problem in this paper contains only two objectives, we will rely on Pareto-based fitness assignment.
ii. **Diversity Preservation:** A basic component of diversity preservation strategies is density assessment. Density assessment evaluates the density at different sub-divisions in a feature space, which may be in the parameter or objective domain. Depending on the manner in which solution density is measured, the different density assessment techniques can be broadly categorized under 1) Distance-based, 2) Grid-based, and 3) Distribution-based.
iii. **Elitism**: The use of the elitist strategy is conceptualized by De Jong in reference [46] to preserve the best individuals found during the searching process. The first issue to be considered in Elitism is Archiving which storage elitist solutions. Archiving usually involves an external repository and this process is much more complex than in (SO) since we are now dealing with a set of Pareto optimal solutions instead of a single solution. The fact that Pareto front is an infinite set raises the natural question of what should be maintained? Without any restriction on the archive size, the number of nondominated solutions can grow exceedingly large. Therefore, in the face of limited computing and memory resources in implementation, it is sometimes unwise to store all the nondominated solutions found. So, it is only natural to truncate the archive based on some form of density assessment discussed earlier when the number of nondominated solutions exceeds the upper bound.
iv. **Selection of gbest**: The next issue to be considered is the introduction of elitist solutions into the gbest selection process. Contrary to SO optimization, the gbest for MO optimization exist in the form of a set of nondominated solutions which inevitably leads to the issue of gbest selection for each particle. One problem faced is the "exploration-exploitation" dilemma. A higher degree of exploitation attained through the selection of

gbest according to domination relationship leads to the loss of diversity which led to fail to span the entire Pareto front uniformly and, in the worst case, premature convergence to local optimal solutions. While too much exploration through selection of least crowded nondominated solution as gbest may lead to slow convergence speed. gbest selection schemes that sought to balance the tradeoff between exploration and exploitation have been proposed. For example, the paper [47] presented a general framework for MOPSO) which allows designers to control the balance between exploration of diversity and exploitation of proximity. Figure 7 shows the flow chart of the used multi-objective particle swarm optimization algorithm.

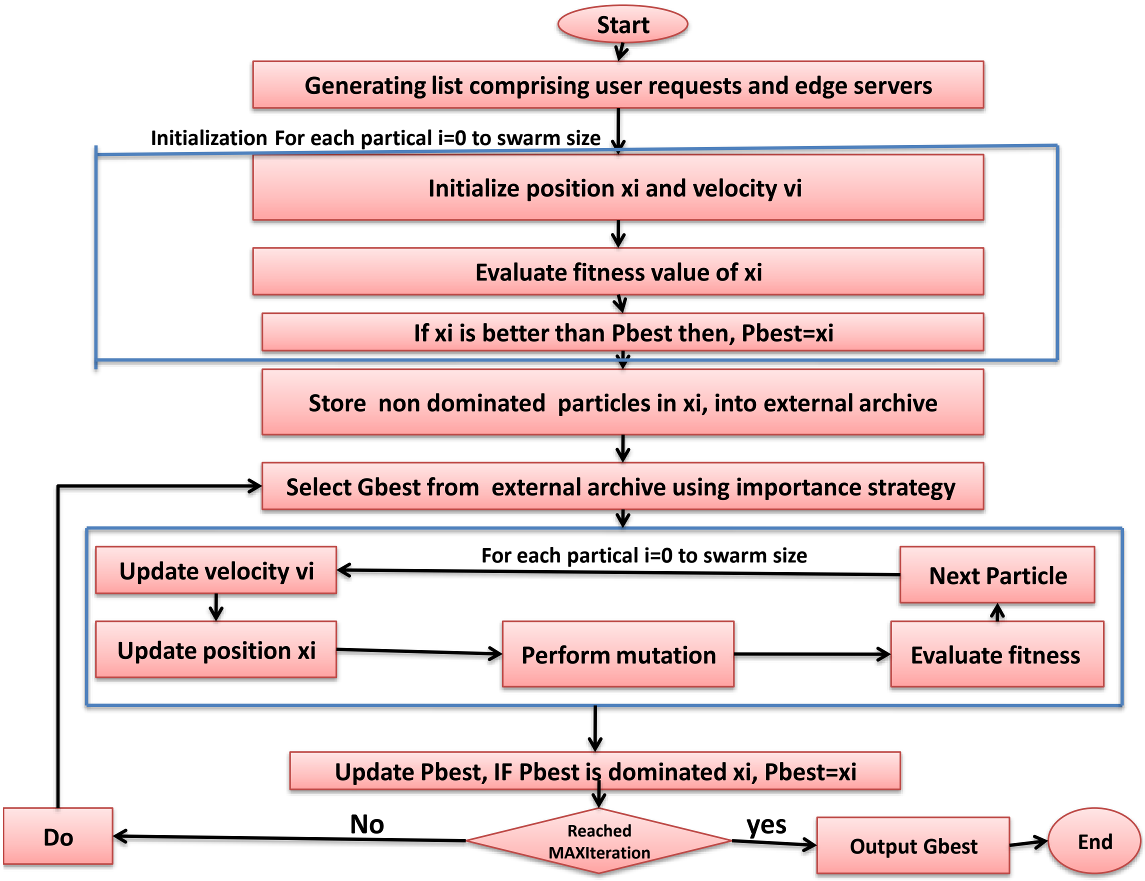


Fig. 7. flow chart of the multi-objective particle swarm optimization algorithm

Therefore, the PA problem can be addressed using the quasi-convex technique and solved using the HGAPSO algorithm shown in Algorithm 1. In addition, we formulate the joint request offloading and resource scheduling problem as a dual decision-making problem that can be solved using the MOPSO algorithm shown in Algorithm 4.

6. Performance Evaluation

6.1 Simulation Setup

To evaluate the effectiveness of the proposed algorithms, we implemented the HGAPSO and MOPSO algorithms using (MATLAB 2021a). The simulations were conducted on an Intel core i7 laptop, with 16GB RAM. Assuming that the area is equipped with a number of base stations (BSs) with a computing ability of (60GHz), and a (macro-BS) with a computing ability of (120GHz). Assuming that the bandwidth used is (Bw = 20 MHZ) and the white noise energy is (σ_0^2 = -100 (dBm)) in a communication environment. We quantize a mobile user into a zone associated with the BS based on the location of the mobile user and the area covered by the BS. The parameters of the simulation are shown in Table 5.

Table 5
Average node degree vs. time

Parameter	Value
Number of mobile users *U*	{12,20,32,40,52,60,72,80,92,100}
Number of BSs *N*	{3,5,8,10,13,15,18,20,23,25}
Workload of request w*q*	1000-2000 (MHz)
Input data of request *Iq*	600-1000 (KB)
Priority of request *q prq*	(0, 1)
Ideal delay of request *q Tgq*	[0.4, 0.6] (s)
Tolerable delay of request *q Tbq*	*Tgq* + [0.1, 0.15] (s)
Computing capacity of BS *n Rn*	{60,70, 80} (GHz)
Computing capacity of macro-BS *Rc*	120 (GHz)
The fixed bandwidth B	20 (MHz)
The fixed altitude of BS H	10 (m)
Noise powerσ_0^2	-100 (dBm)
The maximum transmitting power of mobile user	{4, 5, 6} (w)

The performance of MOPSO algorithm and HGAPSO algorithm are compared with the following methods:

i. The performance of the HGAPSO algorithm was compared with the GA and PSO algorithms alone, and the noncooperative game model based on sub-gradient (NCGG) algorithm which proposed in reference [38]. We apply HGAPSO to solve the PA problem defined in Section 5.
ii. The performance of the MOPSO algorithm was compared with the single-objective PSO algorithm and the multiple-objective optimization algorithm based on i-NSGA-II (MO-NSGA) algorithm which proposed in reference [38]. We apply MOPSO to solve the joint request offloading and resource scheduling problem defined in section 5.

The evaluation of performance by using both the system welfare and the response rate which means ratio of the number of completed calculation to the total number of requests within the tolerant delay of the request.

6.2 HGAPSO Performance

Each user in the network has a unique task that must be calculated. The workload for each task is randomly determined from (1000) to (2000) MB for each user. The simulation parameters are summarized in Table 1. Finally, the simulation was run to evaluate power consumption when the number of mobile users increases.

i. Power consumption versus number of mobile nodes: In this case the maximum power for mobile users is set to (5w). Figure 8 shows the performance of the proposed algorithm compared to (GA) and (PSO) alone. It is noticeable that the total transmission power consumption increases when the number of mobile phone users increases. It is also noted that the power consumption of the hybrid algorithm is always smaller than the power consumption of both algorithms individually under different numbers of mobile users, which means that the hybrid algorithm gave a better result for power allocation compared to the two mentioned algorithms. It is also noted that as the number of user devices decreases, the difference in power consumption between the compared algorithms will become almost non-existent. While when the number of devices increases, the difference in energy consumption between the compared algorithms becomes clear and the proposed approach becomes more efficient than other methods. This is because GA and PSO are more likely to fall into the local optimal solution as the number of devices present increases.

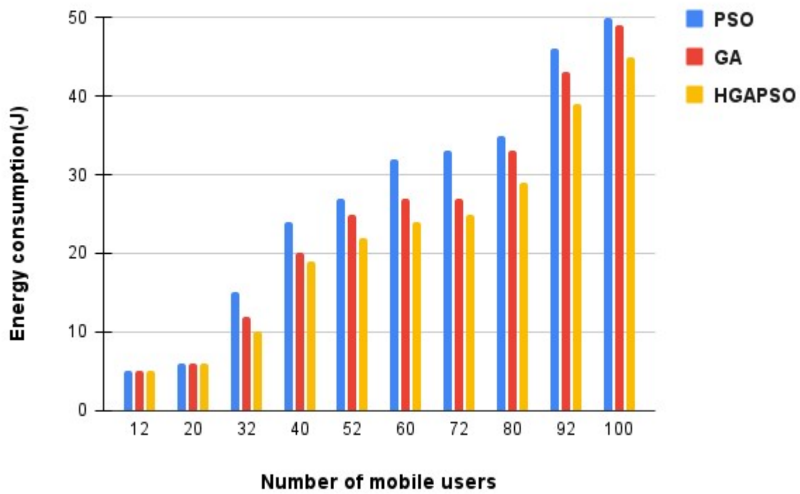


Fig.8. Performance of the proposed algorithm compared to (GA) and (PSO) alone

The performance of the proposed hybrid algorithm was compared with the performance of the NCGG algorithm proposed in the reference study [38] to solve the transmission power allocation problem. The results showed that the proposed algorithm reduced power consumption by an average of (9%) compared to the NCGG algorithm. The Figure 9 shows the difference in power consumption between the two algorithms when the number of users changes.

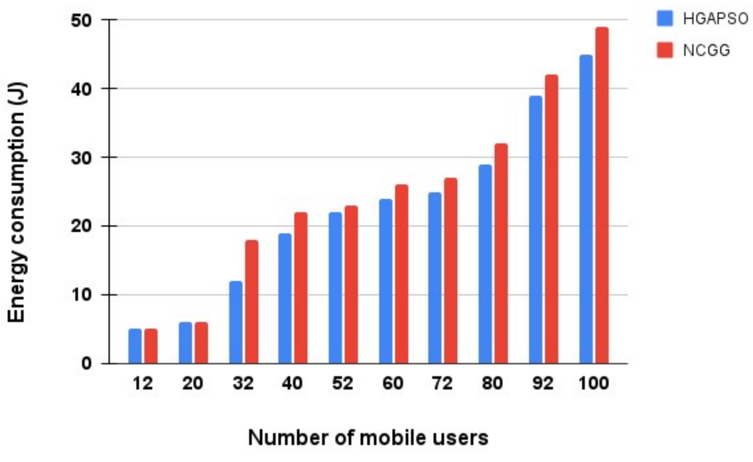


Fig.9. The difference in energy consumption between the two algorithms

ii. **Power consumption versus maximum power (p_{max}):** As shown in Figure 10, the power consumption was evaluated under different number of mobile phone users and at different values of maximum power ($p_{max} = 4w، 5w، 6w$). It is noted that the power consumption is related to the maximum power, that is, the higher the maximum power for mobile users, the higher the power consumption. This is because the average transmission power for mobile users is higher under the maximum power value.

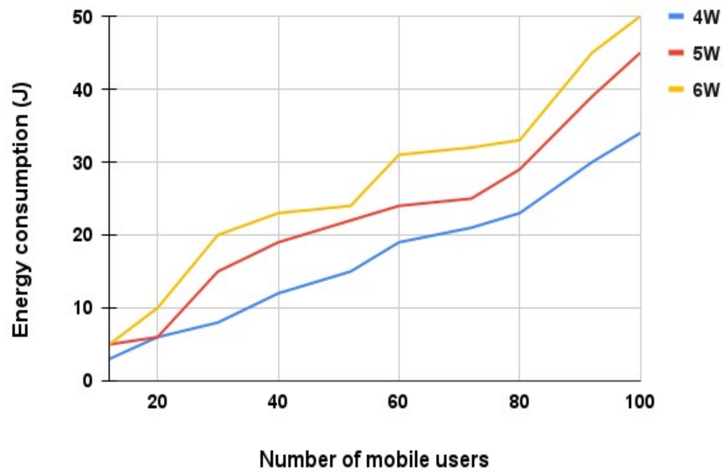


Fig. 10. Power consumption versus maximum power (p_{max})

iii. **Convergence property of (HGAPSO)**: The system was run to evaluate the effect of the number of iterations on the power consumption of the devices when **(p_{max} = 5w)**. First, the cost of mobile users decreased rapidly. In the following rounds, the rate of decline increased with the increasing repetitions. From Figure 11, it can be seen that the total energy consumption of the proposed algorithm tends to converge after (70) iterations when the number of mobile users is (32). That is, increasing the number of repetitions from one to seventy led to a decrease in energy consumption by (83.3%) from (62J to (10J). After that, the rate of decline stabilized when the number of generations increased above seventy. While the PSO algorithm required only fifty iterations, meaning that increasing the number of iterations from one to fifty led to a reduction in energy from (62J to (17J), i.e., by (72.6%). As for the genetic algorithm, it required (295) iterations to reduce the energy from (62J to (13J), i.e., by (79%). Table (6) summarizes the convergence iterations of the three algorithms under a different number of mobile users. It can be seen from the table that it takes, for example, (115) iterations to converge (HGAPSO) when the number of mobile users is (100), which indicates that our proposed (HGAPSO) has good convergence property.

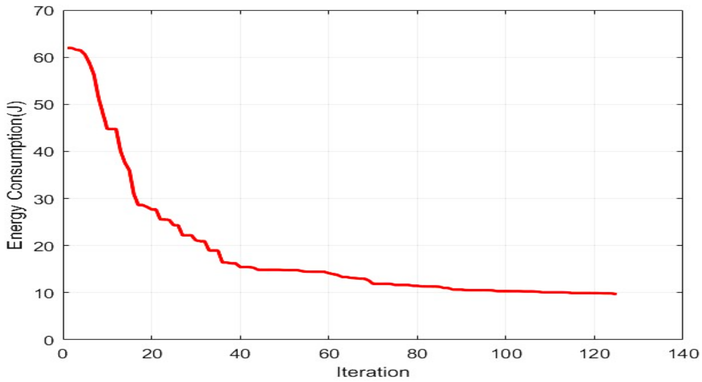


Fig. 11. convergence Property (HGAPSO) when the number of users is 32

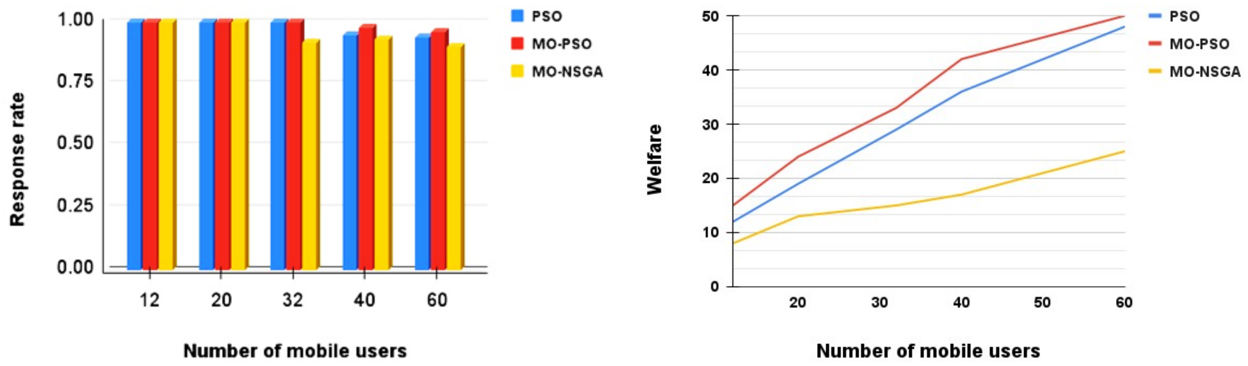
Table 2
Convergence iterations under a different number for mobile users

users number	Iterations of the (HGAPSO) algorithm	Iterations of the (PSO) algorithm	Iterations of the (GA) algorithm
12	21	18	98
20	29	24	195
32	70	50	295
40	76	65	403
52	84	73	550
60	91	82	700
72	97	90	810
80	102	97	915
90	109	101	1050
100	115	108	1230

When comparing the convergence of the HGAPSO algorithm with the convergence of the genetic algorithm [42] and the particle swarm optimization algorithm [43], according to the convergence perspective, PSO will be the best choice, followed by HGAPSO and finally GA. Although (HGAPSO is the most energy efficient, but PSO is the most convergently efficient, while GA's performance is somewhere in the middle between HGAPSO and PSO. This is due to the fact that HGAPSO combines the benefits of GA and PSO. Where GA is better at searching the global domain and PSO is faster at convergence.

6.3 (MO-PSO) performance
6.3.1 Effect of number of mobile users

In this case, the computing capacity of all BSs are the same, i.e., Rn = 70 GHz, and all mobile users offload the same profile request with wq = 1500 (Magacycles), Iq = 700 (KB), Tgq = 0.5 (s) and Tbq = 0.65 (s). As shown in Figure 12, we evaluate the performance including system welfare and response rate of MOPSO, compared to the other two algorithms against different number of mobile users. From Figure 12(a), all algorithms have same responses rate when the number of users was low. It should be noted that MOPSO can achieve a high response rate even in the case of a large number of mobile users, which also reflects the extensibility of MOPSO. From Figure 12(b), It can be seen that with the increasing number of mobile users, the system welfare increases and MOPSO can achieve best compared to both algorithms.

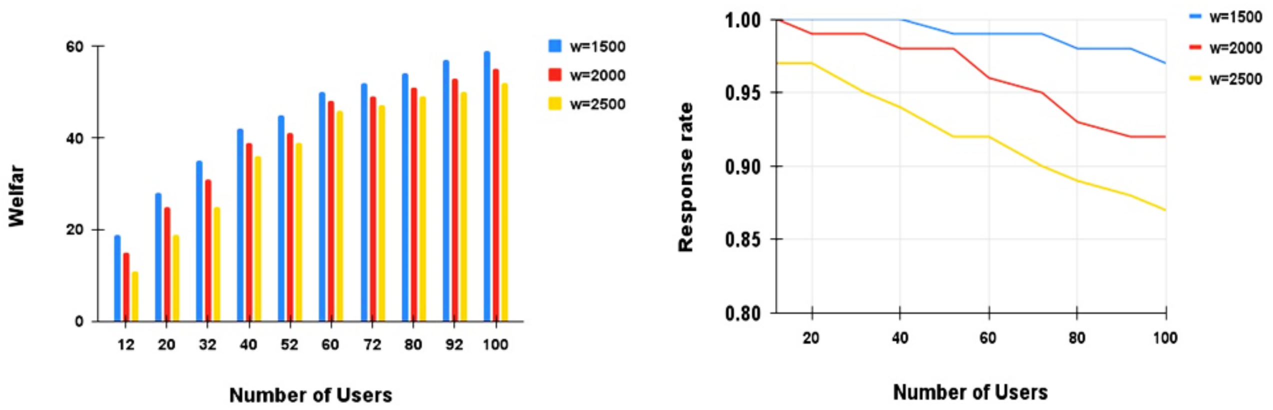


(a) (b)

Fig. 12. Performance vs different number of mobile uses: (a) Welfare, (b) Response rate

6.3.2 Effect of request workload

Here, we evaluate the performance of MO-PSO under different request workload, w_q = 1500, 2000, 2500 and we assume the number of users is 60. As shown in Figure 13, we observe that with the decrease of request workload, both the system welfare and response rate increases. In particular, when the request workload exceeds 2000, the response rate decreases. This is because, the computing resources of BS are not sufficient to be scheduled for offloading requests with more workloads, thus degrading the response rate and the system welfare.

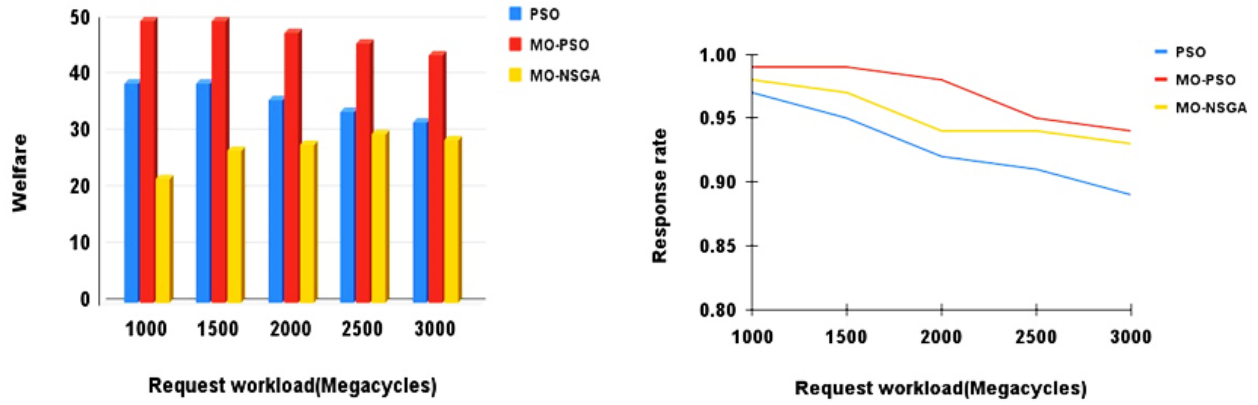


(a) (b)

Fig. 13. Performance under different request workload: (a) Welfare, (b) Response rate

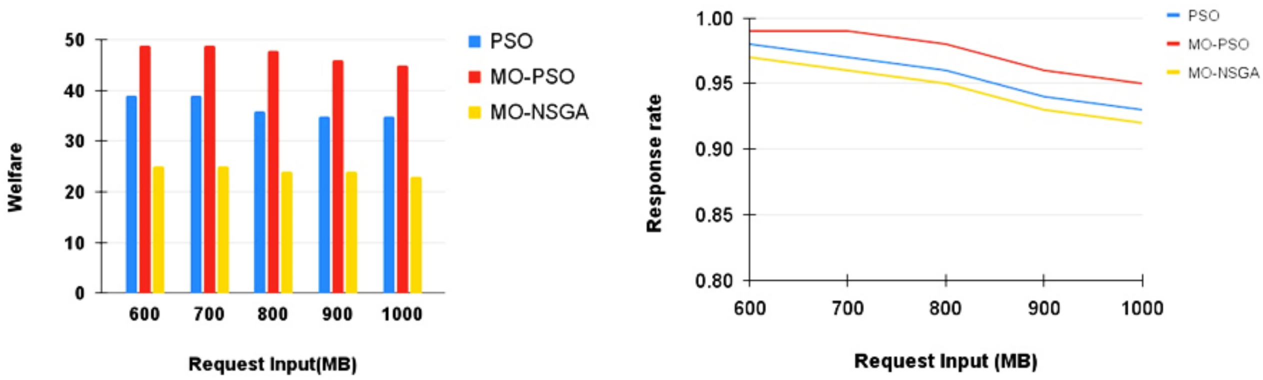
6.3.3 Effect of request profile

In this case, different request profiles in terms of request workload wq and request input size lq are configured to evaluate the performance of MOPSO, compared with other approaches. The system welfare and response rate are plotted in Figure 14(a, b) under different values of wq, we observe that MOPSO always outperforms PSO and MONSGA in response rate and system welfare. whereas MONSGA sacrifices part of the system welfare to maximize the response rate in the optimization process. From Figure 14(a), we observe that when wq increases, the response rate decreases. It is evidently because the computing overhead of BS becomes higher as wq increases, leading to more and more requests being unable to response in same time. Similarly, as shown in Figure 15(a, b). It can be seen that with the increase of lq, the response rate also decreases, which is because a large amount of input data increases the transmitting delay. Even though, the MOPSO has the best performance in terms of response rate, even under different request profiles**.**



(a) (b)

Fig. 14. Performance vs different request workload, with U = 60, Iq = 700 KB: (a) Welfare, (b) Response rate



(a) (b)

Fig. 15. Performance vs different request input, with U = 60, wq = 1500 Magacycles, (a) Welfare, (b) Response rate

7. Conclusions

In this paper, we studied the problems of power allocation for data transfer and joint tasks offloading and resource scheduling in the edge computing network in 5G networks. We consider a network consisting of a Macro-BS, many Micro-BS units, and a large number of mobile users within a 5G network. In particular, we consider the interference between mobile users and base stations under the NOMA protocol. The power allocation (PA) problem is formulated as a non-convex problem and a hybrid algorithm is proposed to solve this problem. The joint tasks offloading and resource scheduling problem is formulated as a nonlinear mixed integer program problem. It is analyzed as a dual decision problem, and then we proposed a multi-objective optimization algorithm based on MO-PSO to address it.

The simulation results showed that the HGAPSO algorithm is able to outperform the two methods (GA and PSO) alone in terms of its reduction in the power consumed in data transfer and its acceptable convergence. The results also show that our algorithm (MO-PSO) outperforms existing methods in terms of response rate which about average 98% and maintains good performance in a dynamic MEC system. However, the proposed algorithms have not been implemented in real-world applications. In future studies, the computation offloading model will be improved by applying it to realistic settings. Additional methods to improve task offloading in a dynamic mobile environment will be tested.

Acknowledgement
This research was not funded by any grant

References

[1] Hakak, Saqib, Thippa Reddy Gadekallu, Praveen Kumar Reddy Maddikunta, Swarna Priya Ramu, M. Parimala, Chamitha De Alwis, and Madhusanka Liyanage. "Autonomous Vehicles in 5G and beyond: A Survey." *Vehicular Communications* 39 (2023): 100551. https://doi.org/10.1016/j.vehcom.2022.100551

[2] Pons, Mario, Estuardo Valenzuela, Brandon Rodríguez, Juan Arturo Nolazco-Flores, and Carolina Del-Valle-Soto. "Utilization of 5G technologies in IoT applications: Current limitations by interference and network optimization difficulties—A review." *Sensors* 23, no. 8 (2023): 3876. https://doi.org/10.3390/s23083876

[3] Alhazmi, Omar. "A survivable internet of things scheme." *Journal of Advanced Research in Computing and Applications* 13, no. 1 (2018): 19-26.

[4] Haekal, Mohammad Ghifari. "An Observation on Social Acceptance of Location-Based Augmented Reality Games in Open Space." *Journal of Advanced Research in Computing and Applications* 13, no. 1 (2018).

[5] Abas, Hesham Ahmed Abdul Mutaleb, Faieza Binti Abdul Aziz, and Rosaliza Hasan. "Review of augmented reality applications in manufacturing engineering." *Journal of Advanced Research in Computing and Applications* 5 (2016): 11-16.

[6] 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* 18, no. 1 (2020): 1-10.

[7] Fowdur, T. P., and K. M. S. Soyjaudah. "Robust JPEG image transmission using unequal error protection and code combining." *International journal of communication systems* 21, no. 1 (2008): 1-24. https://doi.org/10.1002/dac.875

[8] Shabani, Zahra Dehghan, and Rouhollah Shahnazi. "Energy consumption, carbon dioxide emissions, information and communications technology, and gross domestic product in Iranian economic sectors: A panel causality analysis." *Energy* 169 (2019): 1064-1078. https://doi.org/10.1016/j.energy.2018.11.062

[9] Chen, Wuhui, Yuichi Yaguchi, Keitaro Naruse, Yutaka Watanobe, Keita Nakamura, and Jun Ogawa. "A study of robotic cooperation in cloud robotics: Architecture and challenges." *IEEE Access* 6 (2018): 36662-36682. https://doi.org/10.1109/ACCESS.2018.2852295

[10] 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

[11] Zhang, Ke, Yuming Mao, Supeng Leng, Quanxin Zhao, Longjiang Li, Xin Peng, Li Pan, Sabita Maharjan, and Yan Zhang. "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks." *IEEE access* 4 (2016): 5896-5907. https://doi.org/10.1109/ACCESS.2016.2597169

[12] Alameddine, Hyame Assem, Sanaa Sharafeddine, Samir Sebbah, Sara Ayoubi, and Chadi Assi. "Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing." *IEEE Journal on Selected Areas in Communications* 37, no. 3 (2019): 668-682. https://doi.org/10.1109/JSAC.2019.2894306

[13] 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

[14] Thiab, A. S., and A. S. Shibghatullah. "Hypervisor security issues in cloud computing: the need to mitigate the risks." *Advanced Research in Computing and Applications* 1 (2015): 1-5.

[15] Bhatti, F. N., R. B. Ahmad, H. M. Bhatti, M. N. M. Warip, and S. J. Elias. "A Review and Survey of Vehicular Network's Cybercrime and Security Approaches." *Journal of Advanced Research in Computing and Applications* (2016).

[16] 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

[17] 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* 27, no. 3 (2022): 1123-1130. https://doi.org/10.1007/s11036-018-1177-x

[18] Abdel-Basset, Mohamed, Laila Abdel-Fatah, and Arun Kumar Sangaiah. "Metaheuristic algorithms: A comprehensive review." *Computational intelligence for multimedia big data on the cloud with engineering applications* (2018): 185-231. https://doi.org/10.1016/B978-0-12-813314-9.00010-4
[19] 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. https://doi.org/10.1109/ICNN.1995.488968
[20] Tan, Kay Chen, Eik Fun Khor, and Tong Heng Lee. *Multiobjective evolutionary algorithms and applications*. Springer Science & Business Media, 2005.
[21] 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
[22] Ge, Xiaohu, Song Tu, Guoqiang Mao, Cheng-Xiang Wang, and Tao Han. "5G ultra-dense cellular networks." *IEEE wireless communications* 23, no. 1 (2016): 72-79. https://doi.org/10.1109/MWC.2016.7422408
[23] 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
[24] 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
[25] 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* 79, no. 17 (2023): 20076-20115. https://doi.org/10.1007/s11227-023-05387-w
[26] 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
[27] 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
[28] 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
[29] 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
[30] 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
[31] Almasri, Sanaa, Moath Jarrah, and Basheer Al-Duwairi. "Multi-objective optimization of task assignment in distributed mobile edge computing." *Journal of Reliable Intelligent Environments* 8, no. 1 (2022): 21-33. https://doi.org/10.1007/s40860-021-00162-1
[32] Peng, Chaoda, Xumin Huang, Yuan Wu, and Jiawen Kang. "Constrained multi-objective optimization for UAV-enabled mobile edge computing: Offloading optimization and path planning." *IEEE Wireless Communications Letters* 11, no. 4 (2022): 861-865. https://doi.org/10.1109/LWC.2022.3149007
[33] Zhu, Si-feng, Jiang-hao Cai, and En-lin Sun. "Mobile edge computing offloading scheme based on improved multi-objective immune cloning algorithm." *Wireless Networks* 29, no. 4 (2023): 1737-1750. https://doi.org/10.1007/s11276-022-03157-9
[34] Li, Jinglei, Ying Shang, Meng Qin, Qinghai Yang, Nan Cheng, Wen Gao, and Kyung Sup Kwak. "Multiobjective oriented task scheduling in heterogeneous mobile edge computing networks." *IEEE Transactions on Vehicular Technology* 71, no. 8 (2022): 8955-8966. https://doi.org/10.1109/TVT.2022.3174906
[35] Asghari, Ali, Hossein Azgom, and Ali Abbas Zoraghchian. "Multi objectives Edge Server Placement in Mobile Edge computing Using Trees Social Relations Optimization Algorithm and DVFS technique." (2022). https://doi.org/10.21203/rs.3.rs-1959758/v2
[36] Zhu, Si-feng, En-lin Sun, Qing-hua Zhang, and Jiang-hao Cai. "Computing offloading decision based on multi-objective immune algorithm in mobile edge computing scenario." *Wireless Personal Communications* 130, no. 2 (2023): 1025-1043. https://doi.org/10.1007/s11277-023-10318-2
[37] Li, Hongjian, Peng Zheng, Tiantian Wang, Jingjing Wang, and Tongming Liu. "A multi-objective task offloading based on BBO algorithm under deadline constrain in mobile edge computing." *Cluster Computing* 26, no. 6 (2023): 4051-4067. https://doi.org/10.1007/s10586-022-03809-7

[38] 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

[39] Jeong, Seongah, Osvaldo Simeone, and Joonhyuk Kang. "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning." *IEEE Transactions on Vehicular Technology* 67, no. 3 (2017): 2049-2063. https://doi.org/10.1109/TVT.2017.2706308

[40] Nakamura, Yugo, Hirohiko Suwa, Yutaka Arakawa, Hirozumi Yamaguchi, and Keiichi Yasumoto. "Middleware for proximity distributed real-time processing of IoT data flows." In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pp. 771-772. IEEE, 2016. https://doi.org/10.1109/ICDCS.2016.101

[41] Alali, Safaa, and Abdulkarim Assalem. "Metaheuristics Method for Computation Offloading In Mobile Edge Computing: Survey." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 36, no. 1 (2023): 43-73. https://doi.org/10.37934/araset.36.1.4373

[42] Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar. "A review on genetic algorithm: past, present, and future." *Multimedia tools and applications* 80 (2021): 8091-8126. https://doi.org/10.1007/s11042-020-10139-6

[43] Jain, Meetu, Vibha Saihjpal, Narinder Singh, and Satya Bir Singh. "An overview of variants and advancements of PSO algorithm." *Applied Sciences* 12, no. 17 (2022): 8392. https://doi.org/10.3390/app12178392

[44] Dasheng, Liu. "Multi objective particle swarm optimization: Algorithms and applications." (2009).

[45] Turkcan, Ayten, and M. Selim Akturk. "A problem space genetic algorithm in multiobjective optimization." *Journal of Intelligent Manufacturing* 14 (2003): 363-378. https://doi.org/10.1023/A:1024605927329

[46] De Jong, Kenneth Alan. *An analysis of the behavior of a class of genetic adaptive systems*. University of Michigan, 1975.

[47] Alvarez-Benitez, Julio E., Richard M. Everson, and Jonathan E. Fieldsend. "A MOPSO algorithm based exclusively on pareto dominance concepts." In *International conference on evolutionary multi-criterion optimization*, pp. 459-473. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. https://doi.org/10.1007/978-3-540-31880-4_32