Penerbit Akademia Baru

Journal of Advanced Research in Computing and Applications

Journal homepage:
https://akademiabaru.com/submit/index.php/arca/index
2462-1927

Using Gray Wolf Optimization for Joint Request Offloading and Resource Scheduling in 5G Network which Use Mobile Edge Computing

Safaa Alali[1,*], Abdulkarim Assalem [2,♣], Boushra Alali [3,♦]

1 Department of Electronic and 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
3 Software Engineering, Faculty of Information Engineering, Al-Baath University, Homs, Syria

ARTICLE INFO

Article history:
Received 5 November 2024
Received in revised form 1 December 2024
Accepted 10 December 2024
Available online 31 December 2024

Keywords:
Mobile Edge Computing; 5G networks; gray wolf optimization; computational offloading

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. Therefore, we model the joint request offloading and resource scheduling (JRORS) problem as a mixed-nonlinear program to maximize the system welfare of requests. Then we proposed gray wolf optimization algorithm referred as (GWO) to solve this problem. The simulation results showed that (GWO) outperforms existing methods in terms of system welfare and can maintain good performance in a dynamic network.

1.Introduction

Integrating Mobile Edge Computing(MEC) with 5G wireless networks have raised increasing attention in last years. Especially for computation-sensitive applications such as Internet of things (IoT) applications [1] and mobile augmented reality/virtual reality (AR/VR) applications which are getting more widely applied in various fields such as education, art, manufacturing field and entertainment [2,3].

* *Corresponding author.*
E-mail address: en.safaa.alali@gmail.com (Safaa Nouman Alali)
♣ *Corresponding author.*
E-mail address: assalem1@gmail.com (Abdulkarim Assalem)
♦ *Corresponding author.*
E-mail address: boushraali44@gmail.com (Boushra Alali)

https://doi.org/10.37934/arca.37.1.6586

MEC is established at the possible closest location to the mobile devices, with moderate servers' capabilities placed at the edge of the network to achieve necessary user-centric requirements and his applications. With the exponential growth of mobile data traffic, offloading computationally intensive tasks from mobile devices to edge servers has become crucial. However, determining the optimal offloading strategy is a complex task due to competing objectives such as minimizing energy consumption, reducing latency, and optimizing resource allocation.

Metaheuristic optimization is a powerful tool that can effectively address various challenges encountered in 5G networks that rely on edge computing. One of the primary problems that can be solved through Metaheuristics optimization is computation offloading. Metaheuristics optimization algorithms can intelligently balance these objectives and provide efficient solutions. Additionally, Metaheuristics optimization can also enhance the performance of mobile edge computing by mitigating issues like task scheduling, workload distribution, and resource management. By using Metaheuristics algorithm, this approach can optimize the network's overall performance and significantly improve the quality of service for users. Therefore, embracing Metaheuristics optimization techniques in 5G networks utilizing edge computing capabilities can lead to more efficient and reliable systems, ensuring seamless connectivity and satisfying the demands of the ever-evolving digital era.

In this paper, the joint request offloading and resource scheduling problem in computational offloading process was solved by using Gray Wolf Optimization (GWO), after modelled it as a mixed-integer non-linear program for maximizing the system welfare. After that, the performance of this algorithm was compared with another used algorithm.

The rest of this paper is organized as follows: some essential related works was reviewed in Section 2. The system model was provided in Section 3. Section 4 discusses GWO - based offloading algorithm. Performance evaluation was presented in Section 5. Finally, we present the conclusions and future work in Section 6.

2. Related works

In this section, recent papers which used gray wolf optimization for solving problems related to computation offloading in MEC were surveyed.

In the reference [4], authors developed an artificial intelligence (AI) driven meta-heuristic Binary Gray Wolf Optimization (BGWO) algorithm for Virtual Network Function (VNF) deployment which hosted in the cloud and edge servers of the 5G Internet's hybrid cloud infrastructure. The two key design goals of VNF deployment in a 5G hybrid cloud were to reduce deployment cost and minimize service latency experienced by users (ie, to maximize their Quality-of-Experiences).However, these two service parameters oppose each other as the reduction of user service latency requires the deployment of a higher number of VNF instances, incurring additional costs. So in this work, the aforementioned VNF deployment problem was formulated as a Multi-objective Linear Programming (MOLP) problem that brings a trade-off between the two conflicting objectives. The results of simulated experiments demonstrated a significant improvement in minimizing VNF deployment costs and maximizing users' QoE up to 30% and 10%, respectively. That achieved a near-optimal solution in polynomial time.

In the reference [5], They developed an evolutionary meta-heuristic solution for the offloading problem, namely WOLVERINE, based on a Binary Multi-objective Grey Wolf Optimization algorithm that achieves a feasible solution within polynomial time computational complexity. So, they addressed the problem by developing a multi-objective optimization framework that jointly optimizes the latency, energy consumption, and resource usage cost. The experimental results depicted that the developed WOLVERINE system achieves as high as 33.33%, 35%, and 40% performance improvements in terms of execution latency, energy, and resource cost, respectively compared to the state-of-the-art.

In the reference [6], the authors tried to optimize the methodology of resource allocation in Edge Computing, seeking to improve the quality of service (QoS) to the user. For this, they developed an algorithm for efficient resource allocation using grey wolves optimization technique, named as Resource Allocation Technique for Edge Computing (RATEC). The algorithm adopted the meta-heuristic technique to choose the best Edge when allocating the resources of user equipment (UE). Here, they considered that the UEs are composed of processing, storage, time and memory resources. So the algorithm used these resources to calculate the fitness of each Edge and decide which one to allocate, if available. The RATEC had been compared with two other policies and had managed to serve a number most significant of UEs, reducing the number of services refused and presenting a low number of blockages while searching for an Edge.

In the reference [7], the authors studied the task offloading problem on mobile edge in vehicular networks. Specifically, they took computational resource constraints into consideration, and aimed to simultaneously reduce latency and energy consumption. For this purpose, they established an offloading model that consists of local edge computing resources, edge server resources of both macro and subsidiary base stations, as well as cloud computing server resources. Each task could be offloaded through one of five strategies, and was evaluated via a loss function determined by its latency and energy consumption. Based on this model, their goal was to solve a mixed-integer non-linear optimization problem (MINLP) whose objective function was the weighted sum of the task-specific loss functions. To address this optimization problem, they split it into two sub-problems, referred to as resource allocation and offloading. So they developed a method based on Block Coordinate Descent technique combining convex optimization and Gray Wolf algorithm (BCD-CONGW) that alternatively solved the two sub-problems, until convergence. The former sub-problem was convex and can be solved in polynomial time, whereas the latter was non-convex and hence NP-hard. So for the latter, they used relax discrete variables and Gray Wolf algorithm with elite strategy to approximate its optimal point. By numerical evaluations, they showed that their method outperforms existent methods in terms of latency and energy consumption.

In the reference [8], the authors studied the problem of scheduling security-critical tasks of crowdsourcing applications in a multiserver MEC environment. Then they formulated this scheduling problem as an integer program and proposed a family of convergent grey wolf optimizer (CGWO) metaheuristic algorithms to seek for the best scheduling solutions. The proposed CGWO used a task permutation to represent a candidate solution to the formulated scheduling problem, and employed a probability-based mapping scheme to map each search agent in grey wolf optimizer (GWO) onto a valid task permutation. They introduced a new position update strategy for generating the next generation of grey wolf population after each round of search. With this strategy, they proved their proposed CGWO guarantees its convergence to the global best solution and introduced a new position update strategy for generating the next generation of grey wolf population after each round of search. More importantly, they provided a thorough analysis on the movement trajectories of grey wolves during the evolutionary procedure, in order to determine appropriate parameter values such that CGWO would not be trapped in local optima. Experimental results justified the superiority of CGWO metaheuristics over the standard GWO in solving the crowdsourcing task scheduling problem.

In the reference [9], the authors proposed an optimal selection of offloading tasks using well-known metaheuristics, ant colony optimization algorithm, whale optimization algorithm, and Grey wolf optimization algorithm using variant design of these algorithms according to proposed problem through mathematical modelling. Here executing multiple tasks at the server tends to provide high response time that leads to overloading and put additional latency at task computation. They also graphically represented the trade-off between energy and delay that, how both parameters were inversely proportional to each other, using values from simulation. Results showed that Grey wolf optimization outperforms the others in terms of optimizing energy consumption and execution latency while selected optimal set of offloading tasks.

In the reference [10], the authors proposed a D2D-assisted MEC system to address the scheduling challenges posed by numerous independent computing tasks generated by multiple users. They considered splitting the user's task into multiple independent subtasks and calculating offloading separately to reduce processing delay. After that they represented this scheduling problem in the form of a task permutation and proposed an improved Grey Wolf Optimizer (IGWO) metaheuristic algorithm to search for the optimal scheduling solution. This approach, through improvements to the nonlinear convergence factor and dynamic weighting, enhances the optimization speed and accuracy of the Grey Wolf algorithm, effectively reducing task processing latency. Simulation results indicated that the IGWO metaheuristic algorithm outperforms other benchmark methods in addressing this scheduling problem.

In the reference [11], the authors emphasized that offloading strategy should consider enough factors, and the strategy should be made as quickly as possible. While most of the existing model only considers one or two factors, so they investigated a model considering three targets and improved it by normalizing each target in the model to eliminate the influence of dimensions. Then, they introduced gray wolf optimization algorithm to solve the improved model. After that, they proposed an algorithm hybrid whale optimization algorithm (WOA) with GWO named GWO-WOA to obtain better performance. And they tested the improved algorithm on proposed model. The simulation results have shown the advantages of proposed GWO-WOA algorithm for obtaining optimal offloading.

In the reference [12], The authors presented a task offloading and power assignment optimization algorithm for minimizing task completion time under an energy constraint upon the mobile device. They developed a grey wolf optimizer (GWO)-based metaheuristic algorithm to determine the offloading order and power assignment for task offloading. The proposed algorithm employed a position-based mapping scheme for converting each grey wolf into a high-quality offloading solution represented by a task sequence. For the ordered tasks in the converted solution, they developed a heuristic strategy to assign their transmission powers such that the fitness value of the corresponding grey wolf could be evaluated. They further introduced a selection mechanism for updating the wolf population, in which grey wolves with high fitness values had high probabilities of being selected in the next generation of grey wolves. Evaluation results demonstrated that the new mapping scheme and selection mechanism were beneficial for improving task offloading performance in terms of significant make span reductions.

In the reference [13], the authors enhanced the computation design by apply the AlexNet of the DL model which based on the convolutional neural network (CNN) used to train a large number of attributes. To provide an optimal solution, the metaheuristics algorithm of grey wolf optimization (GWO) was combined with an AlexNet which was named a novel grey wolf optimization-based AlexNet (GWOAN) algorithm. In the proposed GWOAN algorithm, the AlexNet hyperparameters (weights, biases and other parameters) were fine-tuned by GWO and then performed a classification. As a result, the GWOAN had achieved a higher scheduling task and low latency than the standard AlexNet, ResNet-18 and VGGNet-16 respectively.

In the reference [14], the authors proposed a hybrid task offloading approach (HybridTO) integrating Grey Wolf Optimizer and Particle Swarm Optimization. This approach aimed to optimize energy consumption and fulfil latency constraints in EC environments by taking into account various factors such as capacity constraints, proximity constraints, and latency requirements. Leveraging the collaborative capabilities inherent in EC servers, HybridTO offered a comprehensive solution to the task offloading problem. Through extensive simulations, they evaluated the performance of HybridTO against baseline approaches, demonstrating its superiority regarding energy usage, offloading utility and response delay, especially under conditions of limited resources. These results underscored the effectiveness of HybridTO as a promising solution for energy-efficient task offloading in EC environments.

In the reference [15], the authors developed the optimal strategy for associating the mobile users to MEC hosts and access points by optimally allocating the computational and radio resources to every user. Here, a well-performing optimization technique called Grey Wolf Optimization (GWO) was used for solving the objective by minimizing the overall user transmit power in terms of latency constraints with both computation and communication times. Finally, they used comparative analysis over other intelligent methods to prove the efficiency of the proposed model.

While previous research on this integrated optimization had identified several near-optimal solutions, they often came with considerable system and computational overheads. So the authors in reference [16] investigated in power control and user grouping to optimize spectral efficiency in NOMA uplink systems, aiming to reduce computational difficulty. To address this, they employed an improved Grey Wolf Optimizer (GWO). Although GWO was effective, it could sometimes converge prematurely and might lack diversity. To enhance its performance, this study introduced a new version of GWO, integrating Competitive Learning, Q-learning, and Greedy Selection. Competitive learning adopted agent competition, balancing exploration and exploitation and preserving diversity. Q-learning guided the search based on past experiences, enhancing adaptability and preventing redundant exploration of sub-optimal regions. Greedy selection ensured the retention of the best solutions after each iteration. The synergistic integration of these three components substantially enhanced the performance of the standard GWO. This algorithm was used to manage power and user-grouping in NOMA systems, aiming to strengthen system performance while restricting computational demands. The effectiveness of the proposed algorithm was validated through numerical evaluations. Simulated outcomes revealed that when applied to the joint challenge in NOMA uplink systems, it surpassed the spectral efficiency of conventional orthogonal multiple access. Moreover, the proposed approach demonstrated superior performance compared to the standard GWO and other state-of-the-art algorithms, achieving reduced system complexity under identical constraints.

In the reference [17], paper, the authors proposed a task offloading scheme that minimizes the overall energy consumption along with satisfying capacity and delay requirements. So they proposed a MEC-assisted energy-efficient task offloading scheme that leverages the cooperative MEC framework. To achieve energy efficiency, they proposed a novel hybrid approach established based on Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO) to solve the optimization problem. The proposed approach considered efficient resource allocation such as sub-carriers, power, and bandwidth for offloading to guarantee minimum energy consumption. The simulation results demonstrated that the proposed strategy was computational-efficient compared to benchmark methods. Moreover, it improved energy utilization, energy gain, response delay, and offloading utility.

In the reference [18], the authors introduced an optimal approach to connect mobile users with MEC hosts and access points, effectively distributing radio and computing resources to individual mobile users. The objective was achieved by lowering the total user transmit power while abiding by latency constraints in both the communication and computation phases, using the Grey Wolf

Optimization technique (GWO). A comparison to alternative smart techniques demonstrated the effectiveness of this approach in solving proposed problem.

Based on previous reference studies, GWO algorithm will be adopted to solve the problem of offloading requests and scheduling resources jointly, aiming to increase the system welfare, which includes reducing cost, increasing utility, and reducing energy consumption. Table 1 shows a comparison between studies that used an algorithm and the basic objective of each study.

Table 1
Comparison between reference studies which used GWO algorithm for solving different problems in edge computing

Ref.	Scenario	Objective	Method
[4]	Virtual Network Function (VNF) deployment which hosted in the cloud and edge servers of the 5G Internet's hybrid cloud infrastructure.	minimizing VNF deployment costs and maximizing users' QoE	an artificial intelligence (AI) driven meta-heuristic Binary Gray Wolf Optimization (BGWO) algorithm
[5]	collaborative edge computing	jointly optimizes the latency, energy consumption, and resource usage cost	a Binary Multi-objective Grey Wolf Optimization algorithm
[6]	Edge Computing	optimize the methodology of resource allocation in Edge Computing, seeking to improve the quality of service (QoS) to the user.	using grey wolves optimization technique, named as Resource Allocation Technique for Edge Computing (RATEC(
[7]	mobile edge in vehicular networks	aim to simultaneously reduce latency and energy consumption.	they developed a method based on Block Coordinate Descent technique combining convex optimization and Gray Wolf algorithm (BCD-CONGW) that alternatively solved the two sub-problems, until convergence.
[8]	scheduling security-critical tasks of crowdsourcing applications in a multiserver MEC environment.	seek for the best scheduling solutions.	proposed a family of convergent grey wolf optimizer (CGWO) metaheuristic algorithms.
[9]	mobile edge computing	optimizing energy consumption and execution latency while selected optimal set of offloading tasks.	using well-known metaheuristics, ant colony optimization algorithm, whale optimization algorithm, and Grey wolf optimization algorithm using variant design of these algorithms according to proposed problem through mathematical modeling
[10]	D2D-assisted MEC system	address the scheduling challenges posed by numerous independent computing tasks generated by multiple users	proposed an improved Grey Wolf Optimizer (IGWO) metaheuristic algorithm to search for the optimal scheduling solution.
[11]	mobile edge computing	they investigated a model considering three targets and improved it by normalizing each target in the model to eliminate the influence of dimensions	they introduced gray wolf optimization algorithm to solve the improved model
[12]	energy-constrained mobile edge computing	They presented a task offloading and power assignment optimization algorithm for minimizing task completion time	They developed a grey wolf optimizer (GWO)-based metaheuristic algorithm to determine the offloading order and

		under an energy constraint upon the mobile device.	power assignment for task offloading.
[13]	computation design by apply the AlexNet of the DL model which based on the convolutional neural network (CNN) used to train a large number of attributes.	achieve a higher scheduling task and low latency than the standard AlexNet	metaheuristics algorithm of grey wolf optimization (GWO) was combined with an AlexNet which was named a novel grey wolf optimization-based AlexNet (GWOAN) algorithm
[14]	-Constrained Edge Computing Environments	They aim to optimize energy consumption and fulfil latency constraints in EC environments by taking into account various factors such as capacity constraints, proximity constraints, and latency requirements.	They proposed a hybrid task offloading approach (HybridTO) integrating Grey Wolf Optimizer and Particle Swarm Optimization.
[15]	multi-access edge computing resources	They develop the optimal strategy for associating the mobile users to MEC hosts and access points by optimally allocating the computational and radio resources to every user.	Grey Wolf Optimization (GWO) was used for solving the objective by minimizing the overall user transmit power in terms of latency constraints with both computation and communication times.
[16]	5G uplink communication	They investigate in power control and user grouping to optimize spectral efficiency in NOMA uplink systems, aiming to reduce computational difficulty.	an improved Grey Wolf Optimizer (GWO)
[17]	mobile edge computing and resource-intensive mobile applications	minimizes the overall energy consumption along with satisfying capacity and delay requirements	novel hybrid approach established based on Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO) to solve the optimization problem.
[18]	Multi-Access Edge Computing Using Intelligent Mobile User Resource Allocation In 6G	They introduce an optimal approach to connect mobile users with MEC hosts and access points, effectively distributing radio and computing resources to individual mobile users.	using the Grey Wolf Optimization technique (GWO)
This paper	5G Network with micro_BS and macro_BS equipped with edge servers	Maximizing welfare system	using the Grey Wolf Optimization technique (GWO)

3. 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 servers, *N*, and a macro-BS with a deep cloud, *C*. As shown in Figure 1.

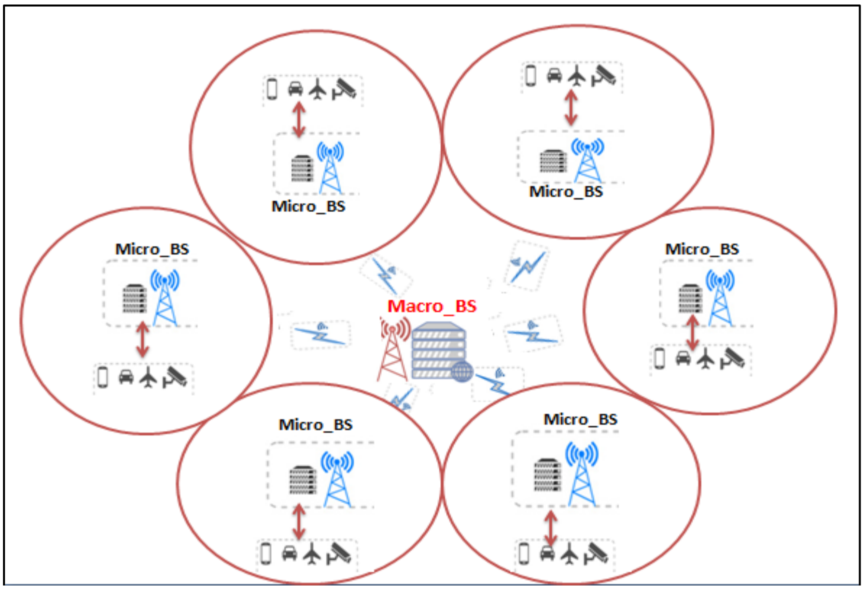

Fig. 1. System model

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 [19].

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*.

3.1. 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. The transmitting power allocation policy is defined as $P = \{p_{un} \mid u \in U, q \in Q\}$, in which p_{un} denotes the transmitting power from mobile user u to BS n. The location of each mobile user is assumed to be unchanged during the time interval, and the uplink rate $v_{un}(t)$ from mobile user u to BS n can be formulated as follows [19]:

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

where B and σ_0^2 represent the bandwidth of the uplink system and background white Gaussian noise power. The channel power gain. between mobile user u to BS n is defined as follows [20]:

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

where g_0 represents the channel power gain at the reference distance $d_0 = 1$ m and the transmitting power is 1W. The request offloading policy is defined as $X = \{x_{qn} \mid q \in Q, n \in N\}$, in which x_{qn} is a binary variable and $x_{qn} = 1$ 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 [19]:

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

The computing resource scheduling policy is defined as $Y = \{R_{qn} \mid q \in Q, n \in N\}$, in which 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 [19]:

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

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

$$t_q = t_{up}^q + t_{pro}^q \quad (5)$$

3.2. 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 [19]:

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

Given the average power consumption of BS and macro-BS, the energy consumed by executing request q is defined as [19]:

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

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

3.3. Problem Formulation

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 [21], we define the edge system utility for processing request q as:

$$k_p = \begin{cases} 1\,, & t_q \le Tg_q \\ 1 - \frac{1}{1+e^{\alpha(T_{avg}-t_q)/(T_{avg}-Tg_q)}}, & Tg_q < t_q \le T_{avg} \\ \frac{1}{1+e^{\alpha(t_q-T_{avg})/(Tb_q-T_{avg})}} & T_{avg} < t_q \le Tb_q \\ 0 & t_q > Tb_q \end{cases} \tag{8}$$

where

$$T_{avg} = \frac{Tg_q + Tb_q}{2} \tag{9}$$

and the edge system cost for processing request *q* is defined as [19]:

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

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 energy 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 [19]:

$$e_q = \varepsilon k_p + (1-\varepsilon) E_q^{pro} \tag{11}$$

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

$$W = \sum_n^N \sum_q^Q \left[x_{qn}\left(k_p - c_p\right) - \left(1 - x_{qn}\right) e_q \right] \tag{12}$$

The joint request offloading and computing resource scheduling problem is formulated as a system welfare maximization problem [19]:

$$P: \max_{X,Y} W \tag{13}$$

$$s.t \ \sum_{n\in N} x_{qn} \le 1, \forall \ q \in Q \tag{13a}$$

$$x_{qn} \in \{0,1\} \ \forall \ q \in Q, n \in N \tag{13b}$$

$$\sum_{q\in Q} R_{qn} \le R_n \ , \forall\, n \in N \tag{13c}$$

$$R_{qn} > 0, \forall \ q \in Q, n \in N \tag{13d}$$

Constraint 13a and constraint 13b 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 13c ensures that the total computing resources scheduled to requests should not exceed the BS's computing capacity. Constraint 13d ensures that BS must schedule a positive computing resource to each request that offloaded to it.

It is observed that the constraints (a) and (b) of the offloading policy *X*, and the constraints (c) and (d) of the offloading policy *Y* are separated from each other. Problem (13) can be divided into two problems, namely the request offloading (RO) problem and the computing resource scheduling (RS) problem. Hence, the RO problem of minimizing the extra cost of the edge system can be expressed as [19]:

$$\min_X M = \sum_n^N \sum_q^Q \left(1 - x_{qn}\right)\left(\varepsilon k_p + (1-\varepsilon) E_q^{pro}\right) \tag{14}$$

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

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

and the RS problem of maximizing the edge system welfare can be expressed as [19]:

$$\max_Y W = \sum_n^N \sum_q^Q \ \left[x_{qn}\left(k_p - c_p\right)\right] \tag{15}$$

$$\sum_{q\in Q} R_{qn} \le R_n \ , \forall\, n \in N \tag{15a}$$

$$R_{qn} > 0, \forall \ q \in Q, n \in N \tag{15b}$$

Therefore, this problem is a double decision-making problem which is very complex and involves a trade-off between two conflicting objectives. In this paper, we propose a Gray Wolf Optimization referred to as GWO, to solve this problem which dividing into problem (14) and problem (15).

4. Basic GWO algorithm

This section describes the GWO algorithm. GWO imitates the social hierarchy and the clever hunting displayed by a swarm of grey wolves. Naturally, grey wolves live in a group of between 5 to 12 individuals. Grey wolves sternly live in a social hierarchy. As depicted in Figure 2, the leaders of a group of grey wolves known as "alpha" are male and female wolves that are in charge of decision making on behalf of other wolves in the group. Such decisions include where to sleep, time to awake and hunting for preys. Generally, other wolves in the group must comply with the decision made by alpha. Nevertheless, in some cases, some classless actions in the social hierarchy of grey wolves are witnessed. In such cases, alpha can obey other wolves in the group. In meetings, other wolves sanction the decision made by alpha by lowering down their tails. It is worth noting that it is not compulsory that alpha should be the strongest wolf in the group. The chief responsibility of alpha is to oversee the group [22].

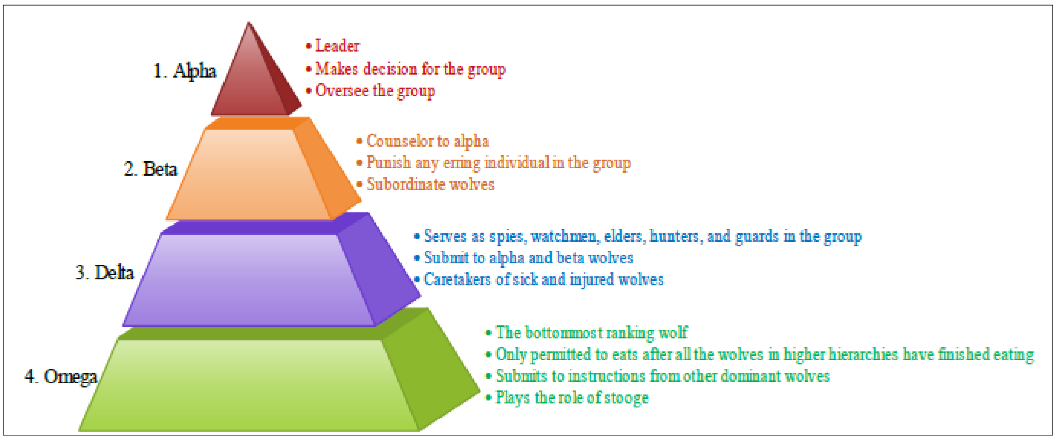

Fig. 2. Distribution of Grey Wolves Social Hierarchy and their responsibilities [22]

The most important characteristics of a group of grey wolves are their self-restraint and orderliness. After alpha, the next echelon in the social hierarchy of grey wolves is beta and the responsibility of beta is to assist alpha in decision-making processes. Any of the male or female wolves can be beta and beta can be the most suitable entrant to replace alpha at any time when old age catch up with any of them or any one of them is deceased. It is demanded that beta reverence and obey alpha, but he/she can control other wolves in his/her hierarchy. Beta serves as a counsellor for alpha and is in charge of punishing any erring individual in the group. The beta emphasizes the commands of alpha and provides responses of group members to alpha [22].

The lowest level in a group of grey wolves is an omega that acts like the victim. It is compulsory for the wolves at this level to submit themselves to the commands of other wolves of higher hierarchy and they are not permitted to eat food until other wolves in other groups have eaten. Though omega looks like the most trivial wolves in the group, without an omega it will be difficult to detect the existence of internal conflict and other problems. The reason for this is that omega is saddled with the responsibility of exposing the existence of cruelty in the group and the displeasure of other wolves. These make other wolves to be contented and also preserve the central organization of grey wolves. Occasionally, omega functions as child-minder in the group [22].

The rest of the wolves, apart from alpha, beta, and omega, are termed secondary (delta). The delta wolves submit to the alpha and beta wolves and rule over the omega wolves. They function as spies, watchmen, elders, hunters, and guards in the group. Spies are in charge of taking care of borderlines and area. They also raise warning alarm of any hazard faced by the group. Watchmen are responsible for safekeeping the group. The elders are veteran wolves that are qualified to be alpha and beta. Hunters assist alpha and beta in pursuing preys and getting food ready for the group, while guards take care of the feeble, sick, and injured wolves [23]. It is worthy to note that dominance decreases from top of the hierarchy downwards. Figure 2 summarises the social hierarchy of grey wolves and the roles each hierarchy plays in the group [22].

Besides the social hierarchy that exists in a group of grey wolves, collective hunting is another fascinating communal behaviours of grey wolves. The grey wolves' hunting includes the steps represented in Figure 3:

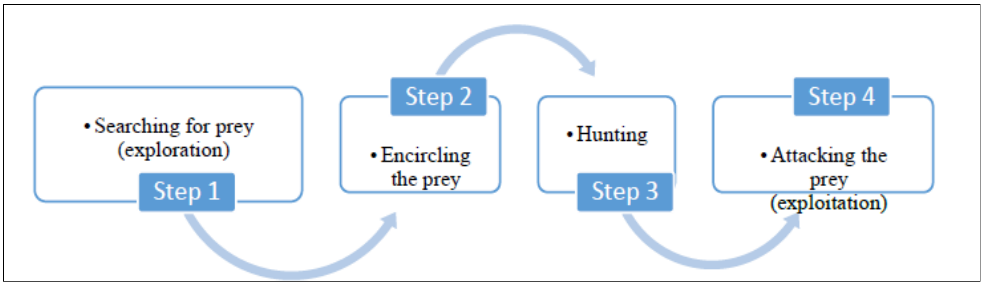

Fig. 3. Grey Wolf Hunting process [22]

The GWO algorithm modelled communal activities of grey wolves' group which are social hierarchy and hunting method.

4.1. Social Hierarchy

In the social hierarchy of grey wolves in the GWO algorithm, the best solution is represented as alpha α. Consequently, the second and third best solutions are represented as beta β and delta δ respectively, and another solution is taken as omega ω. In the GWO algorithm, hunting (optimization) is guided by α, β and δ wolves while ω go behind them.

4.2. Searching for the Prey (Exploration)

Grey wolves usually comb the environment for prey based on the positions of α, β, and δ. They wander away from each other to search for the location of prey and then congregate to attack the prey. Let us assume that $A^{\rightarrow}$ is a random vector that is between the range of -1 and 1 to compel the search agent to wander from the prey, which underlines the global search in GWO. When |A| is greater than 1 the grey wolf is compelled to deviate from the prey (local optimum) to look for superior results in the decision space [22].

This algorithm possesses an additional element ($C^{\rightarrow}$) that aids the algorithm to arrive at new solutions. According to Equation (19), the elements of a vector $C^{\rightarrow}$ are in the range of interval [0, 2]. The vector $C^{\rightarrow}$ supplies non-linear weights to the prey to arbitrarily accentuate ($c > 1$) or trivialise ($c < 1$) the influence of the prey in circumscribing the distance according to Equation (16). The GWO algorithm is able to explore more search space randomly through this singular factor. It also allows the search agent to escape been caught in the local optima during the process of optimization. Contrary to what we have in A, the decrement in C is nonlinear. The C vector is needed from the beginning to the end of the iteration process to enhance global search in the decision space as it prevents the search from been unable to move further in the local optima. C is employed as a barricade witnessed in the natural hunting process of grey wolves. This method of seeking out prey hinders grey wolves from swiftly advancing towards the prey. This is exactly the function of C in GWO algorithm all through the optimization operation [22].

4.3. Encircling the Prey

Based on the abovementioned, grey wolves during their hunting activity encompass their target. The grey wolves' action of surrounding their prey is stated as below [24]:

$$\vec{D} = \left|\vec{C}.\vec{X}_p - \vec{X}(t)\right| \tag{16}$$

$$\overrightarrow{X(t+1)} = \vec{X}_p(t) - \vec{A}.\vec{D} \tag{17}$$

Assuming t is iteration number; $A^{\rightarrow}$ and $C^{\rightarrow}$ are coefficient vectors; $X^{\rightarrow}p$ is a vector of the prey's positions; $X^{\rightarrow}$ is a vector of the grey wolf's positions; and $D^{\rightarrow}$ is computed vector employed to denote a new position of the grey wolf. $A^{\rightarrow}$ and $C^{\rightarrow}$ can be computed using the formulas below:

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \tag{18}$$

$$\vec{C} = 2.\vec{r}_2 \tag{19}$$

Assuming $A^{\rightarrow}$ is vector whose value is to reduce linearly from 2 to 0 over the iterations; and $r^{\rightarrow}{}_1$ and $r^{\rightarrow}{}_2$ are random vectors in [0, 1]. The position of a grey wolf at (x, y) can change depending on the position of prey at (x', y'). Various locations to the most ideal agent can be attained with regard to the present position by controlling $A^{\rightarrow}$ and$C^{\rightarrow}$, such as, by setting $A^{\rightarrow}$ to [1, 0] and C to [1, 1]. The new position of the grey wolf is now (x'- x, y').

The random vectors $r^{\rightarrow}{}_1$ and $r^{\rightarrow}{}_2$ allow the grey wolf to choose any position or node. Hence, a grey wolf can be positioned in any arbitrary position close to the prey. The position is computed using Equations (16) and (17). In a similar manner, grey wolves can change their position to any node of a hypercube in an n-dimensional decision space close to the optimal solution (position of the prey). They possess the capacity to differentiate the location of prey apart from others and encircle it. Generally, the hunting process is directed by α and β, while δ offer help for α. Hence, to mimic the stalking behaviour of grey wolves, it is presumed that α (most viable candidate for the solution), β and δ are more cognizant of the likely bearings of the prey [22]. Consequently, GWO retains three most ideal solutions attained to this point and obliges the omega wolves to bring up to date their positions to attain the ideal place in the decision space. According to [25] such a hunting behaviour can be modelled in an optimization algorithm by expressing it as:

$$\vec{D}_{\propto} = \left|\vec{C}_1.X_\alpha - \vec{X}\right|\vec{D}_\beta = \left|\vec{C}_2.X_\beta - \vec{X}\right|\vec{D}_\delta = \left|\vec{C}_3.X_\delta - \vec{X}\right| \tag{20}$$

$$\vec{X}_1 = \vec{X}_{\propto} - A_j.\left(\vec{D}_{\propto}\right), \overline{X}_2 = \vec{X}_\beta - A_2.\left(\vec{D}_\beta\right), \overline{X}_3 = \vec{X}_\delta - A_2.\left(\vec{D}_\delta\right) \tag{21}$$

$$\vec{X}(t+1) = \frac{x_1 + x_2 + x_3}{3} \tag{22}$$

Figure 4 depicts the way search agent makes the positions of α, β, and δ to be up to date in a two dimensional state space. Based on what we have in Figure 4, the ultimate position (solution) is in a sphere which is stipulated depending on the positions of α, β, and δ in the state space. Put differently, α, β, and δ evaluate the positions of prey and the remaining wolves and afterward make their new up to date positions arbitrarily close to the prey.

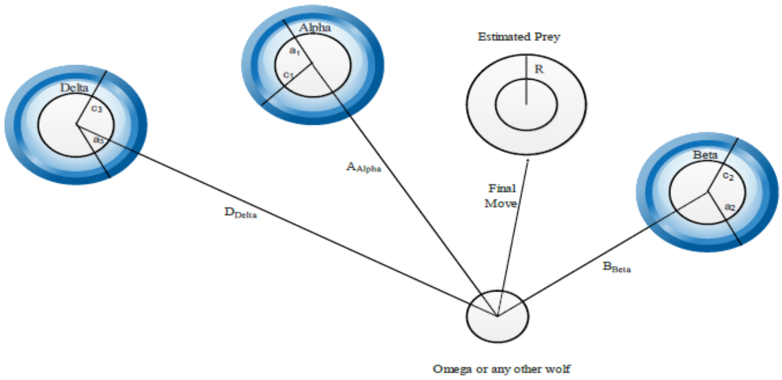

Fig. 4. Grey Wolf Hunting process [22]

4.4. Attacking the Prey

Based on the earlier discussion, grey wolves complete their stalking process by pouncing on the prey until it dies. To imitate the attacking process, the value of $\vec{a}$ is reduced in various iterations. Observe that as $\vec{a}$ reduces, the degree of variation of $\vec{a}$ also reduces. Alternatively stated, $\vec{a}$ is a variant in the interval [2a, 2a]. The value decreases from 2 to 0 as the iterations continues, and can be expressed as follows:

$$\vec{\alpha} = 2 - t.\frac{2}{max_i ter} \tag{23}$$

where $max_i ter$ is the sum of iterations done during the optimization and *t* is the iteration number. As soon as the stochastic value of $\vec{A}$ is in the interval [1, 1]. The subsequent position of a wolf can be between the present position of the wolf and the prey position. Figure 4 depicts that when |A|<1 grey wolves will show aggression toward the prey [22].

Through operators made available, the GWO algorithm allows the search agent to make its position up to date using the positions of α, β, and δ (approach the animal that is hunted). Nevertheless, GWO also possess some other operators that can be used to compute new solutions. Figures (5a) and (5b) show how grey wolves explore their environment and find the right prey before encircling and attacking the estimated prey [22].

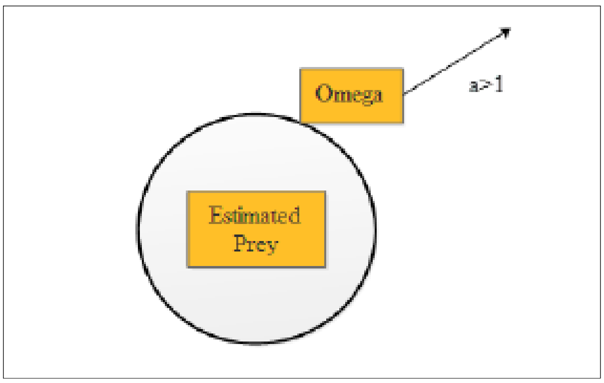

Fig. 5a. Diverging away from a prey to find the right prey [22]

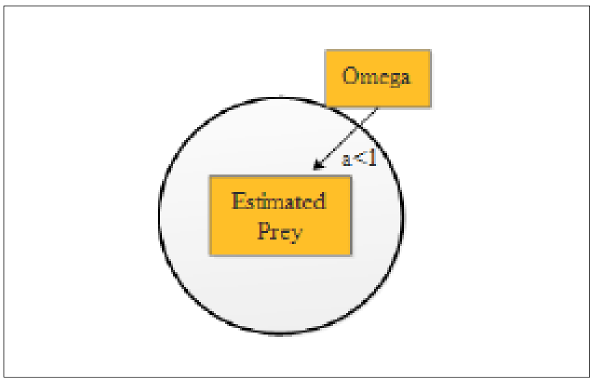

Fig. 5b. Finding a suitable location to encircle and attack the prey [22]

5. Performance Of GWO Algorithm

To evaluate the effectiveness of the proposed algorithm, we implemented GWO algorithm using Matlab R-2021a. The simulations were conducted on an Intel i5, 3.7Ghz PC with 4 GB RAM. We consider a multi-zone edge computing system consisting of mobile users, multiple BSs and a macro-BS. Each BS equipped with an edge server and covers a zone. 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 2.

Table 2
Parameters of simulation

Parameter	Value
Number of mobile users *U*	{20,30,40,50,60,70,80,90,100}
Number of BSs *N*	{2,4,6,8}
Workload of request w*q*	1500 (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)

5.1 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). Figure 6 shows the performance of GWO algorithm when the number of mobile users change.

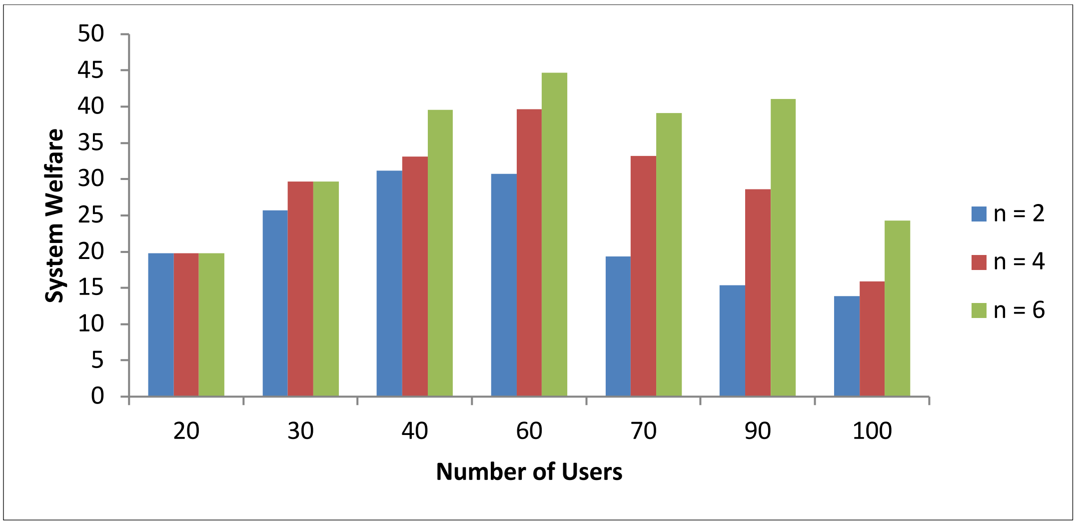

Fig. 6. System Welfare vs different number of mobile uses when Iq = 700 (KB)

It can be seen that with the increasing number of mobile users, the system welfare increases.

5.2 Effect of input data size of requests

Here, we evaluate the performance of GWO under different input data size of offloading requests, we assume wq = 1500 and the number of users is 60. Figure 7 show the performance of algorithm in this case.

Fig. 7. System Welfare under different data requests sizes when u=60

It is observed that: when the requests data size decreases, the system welfare increases because the computing resources of BS are sufficient to schedule the offloading requests which has less data size. Figure 8 show the system welfare when the number of users increases to u = 70.

Fig. 8. System Welfare under different data request when u=70

Comparing the Figure 7 with the Figure 8, it is observed that as the number of users increases, the system welfare decreases because the computational resources become insufficient to schedule more users. Figure 9 and Figure 10 show the system welfare when number of users increases to u = 90 and u = 100, respectively.

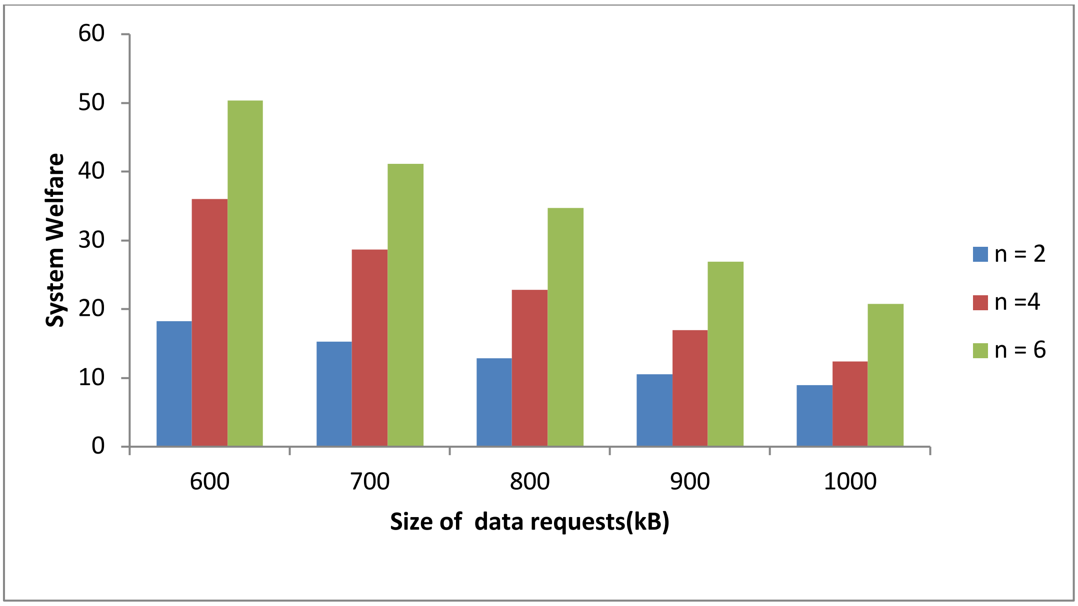

Fig. 9. System Welfare under different data request when u=90

Fig. 10. System Welfare under different data request when u=100

It is observed from Figure 9 and Figure 10 that as the number of users increases, the system well-being continues to decrease. In all cases, the system welfare is higher as the number of edge servers increases because sufficient computational resources are available to satisfy requests with larger data sizes.

5.3 Effect of the number of edge servers

To study the effect of used edge servers on system welfare, we only change the number of edge servers and fix the number of mobile users u=60, and the size of requests $I_q = 600kB$.

Table 3
System welfare against number of edge servers

The number of edge servers	N=4	N=6	N=8	N=10
system welfare when u=60	44.0689	49.4432	55.4252	59.4001
system welfare when u=80	42.1111	54.3907	54.4662	61.3696
system welfare when u=100	23.192	33.613	35.7098	44.1671

It is observed that the system welfare increases when the number of edge servers increases in all cases, due to the availability of sufficient computational resources.

5. Performance Evaluation

We compare the performance of GWO algorithm with MO-NSGA algorithm which proposed in [19]. 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), n = 4, Tgq = 0.5 (s) and Tbq = 0.65 (s). As shown in Figure 11, we evaluate the performance using system welfare rate of GWO, compared to the MO-NSGA algorithm against different number of mobile users. It can be seen that with the increasing number of mobile users, the system welfare increased and GWO achieved best result about 85% compared the MO-NSGA algorithm**.**

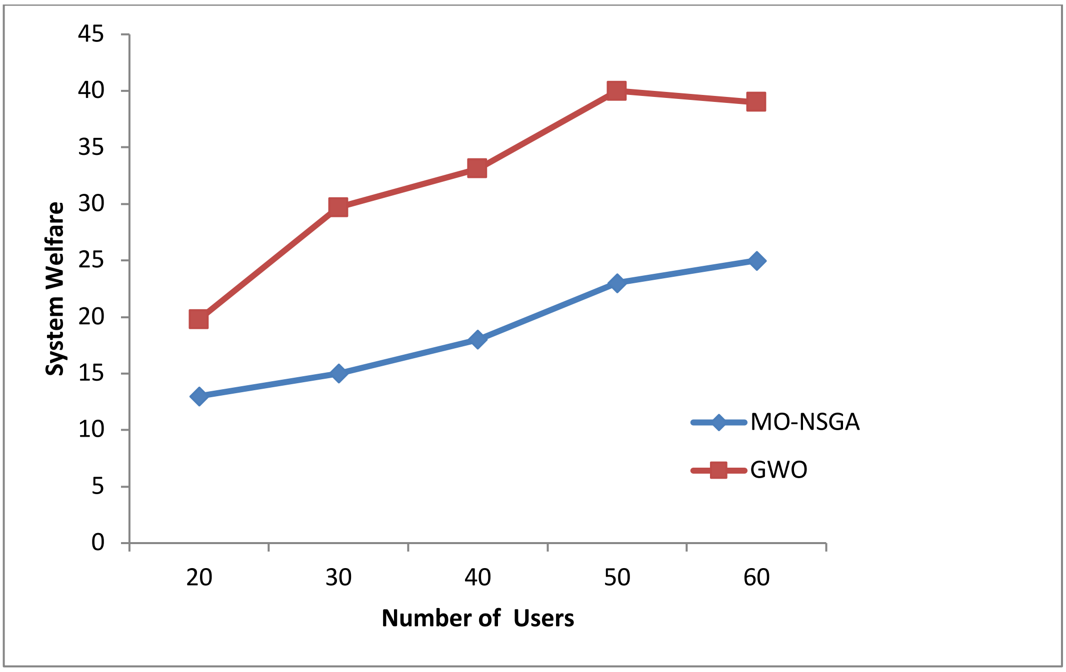

Fig. 11. Compression System Welfare of proposed algorithm and MO-NSGA algorithm when Iq = 700 (KB) and n=4

Similarly, as shown in Figure 12. It can be seen that with the increase of Iq, the system welfare also decreases, because a large amount of input data increases the transmitting delay. Even though, the GWO has the best performance in terms of system welfare rate, even under different request profiles.

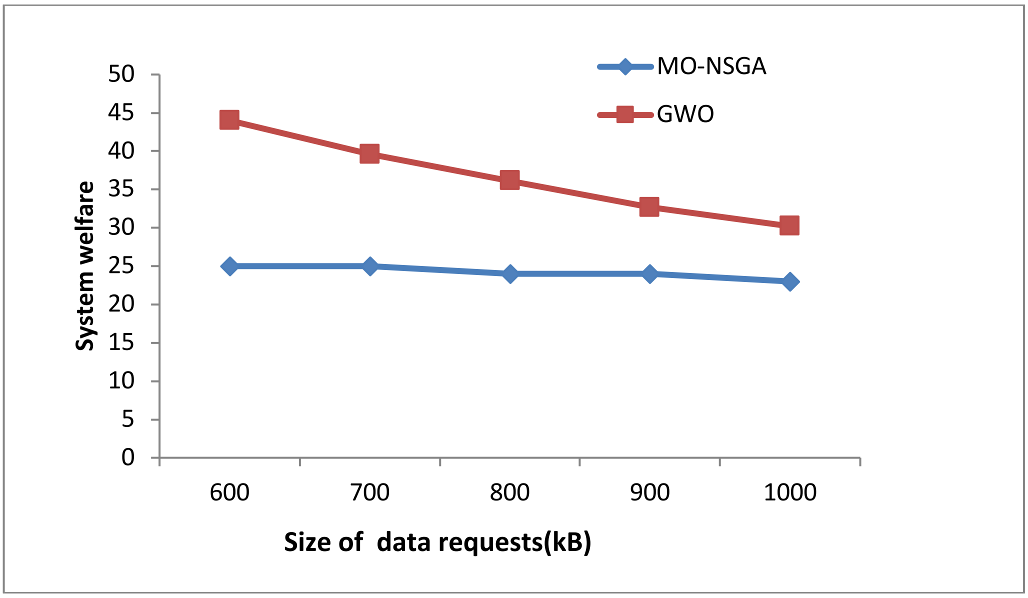

Fig. 12. Compression System Welfare of proposed algorithm and MO-NSGA algorithm when u=60 , n=4 and change the size of data requests size

6. Conclusions

In this paper, the request scheduling and offloading requests were studied problem in 5G networks with edge computing. The network consisting of a macro-BS, many micro-BSs and a large number of mobile users was considered. The NOMA protocol was used as the multiple access scheme between users and BSs. The problem involving jointly optimizing the request offloading for mobile users and the computing resource scheduling at the micro-BSs was formulated, by forming a mixed-integer non-linear program. Then the problem was analyzed as a double decision-making problem, and the Gray wolf optimization (GWO) algorithm was proposed to address it. Simulation results verified that our algorithm (GWO) outperforms the existing approaches in terms of system welfare and maintains a good performance in a dynamic MEC system. However, the proposed algorithm is not implemented in real-world applications. In the future, we will work on the design of edge computing resource scheduling algorithms for systems based on realistic applications, so as to solve the bottlenecks of practical problems.

Acknowledgement
“This research was not funded by any grant”

References

[1] Alhazmi, Omar. "A survivable internet of things scheme." *Journal of Advanced Research in Computing and Applications* 13, no. 1 (2018): 19-26.
[2] 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).
[3] 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, no. 1 (2016).
[4] Shahjalal, Mohammad, Nusrat Farhana, Palash Roy, Md Abdur Razzaque, Kuljeet Kaur, and Mohammad Mehedi Hassan. "A Binary Gray Wolf Optimization algorithm for deployment of Virtual Network Functions in 5G hybrid cloud." *Computer Communications* 193 (2022): 63-74. https://doi.org/10.1186/s13677-023-00570-z

[5] Nujhat, Nawmi, Fahmida Haque Shanta, Sujan Sarker, Palash Roy, Md Abdur Razzaque, Md Mamun-Or-Rashid, Mohammad Mehedi Hassan, and Giancarlo Fortino. "Task offloading exploiting grey wolf optimization in collaborative edge computing." *Journal of Cloud Computing* 13, no. 1 (2024): 23. https://doi.org/10.1186/s13677-023-00570-z

[6] Lieira, Douglas D., Matheus S. Quessada, André L. Cristiani, and Rodolfo I. Meneguette. "Resource allocation technique for edge computing using grey wolf optimization algorithm." In *2020 IEEE Latin-American Conference on Communications (LATINCOM)*, pp. 1-6. Ieee, 2020. https://doi.org/10.1109/LATINCOM50620.2020.9282316

[7] Cong, Yuliang, Ke Xue, Cong Wang, Wenxi Sun, Shuxian Sun, and Fengye Hu. "Latency-energy joint optimization for task offloading and resource allocation in mec-assisted vehicular networks." *IEEE Transactions on Vehicular Technology* 72, no. 12 (2023): 16369-16381. https://doi.org/10.1109/TVT.2023.3289236

[8] Lian, Zhichao, Jiangang Shu, Yi Zhang, and Jin Sun. "Convergent grey wolf optimizer metaheuristics for scheduling crowdsourcing applications in mobile edge computing." *IEEE Internet of Things Journal* (2023). https://doi.org/10.1109/JIOT.2023.3304909

[9] Abbas, Aamir, Ali Raza, Farhan Aadil, and Muazzam Maqsood. "Meta-heuristic-based offloading task optimization in mobile edge computing." *International Journal of Distributed Sensor Networks* 17, no. 6 (2021): 15501477211023021. https://doi.org/10.1177/15501477211023021

[10] Bao, Hongba, Hui Wang, Yiying Sun, and Sheng Pan. "An Improved Grey Wolf Optimizer for Task Scheduling in D2D-Assisted MEC Systems." In *2024 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 168-173. IEEE, 2024. https://doi.org/10.1109/IWCMC61514.2024.10592601

[11] Feng, Siling, Yinjie Chen, Qianhao Zhai, Mengxing Huang, and Feng Shu. "Optimizing computation offloading strategy in mobile edge computing based on swarm intelligence algorithms." *EURASIP Journal on Advances in Signal Processing* 2021 (2021): 1-15. https://doi.org/10.1186/s13634-021-00751-5

[12] Wang, Xiaoting, Zhimeng Zhou, Hanjing Chen, and Yi Zhang. "Task offloading and power assignment optimization for energy-constrained mobile edge computing." In *2021 Ninth International Conference on Advanced Cloud and Big Data (CBD)*, pp. 302-307. IEEE, 2022. https://doi.org/10.1109/CBD54617.2021.00058

[13] Sivakumar, P., Murali Murugan, S. K. Somasundaram, and S. Nagendra Prabhu. "A novel optimised GWOAN algorithm for scheduling task and power consumption in MEC network." *International Journal of Intelligent Systems Technologies and Applications* 21, no. 4 (2023): 386-401. https://doi.org/10.1504/IJISTA.2023.134989

[14] Alsadie, Deafallah. "Efficient Task Offloading Strategy for Energy-Constrained Edge Computing Environments: A Hybrid Optimization Approach." *IEEE Access* 12 (2024): 85089-85102. https://doi.org/10.1109/ACCESS.2024.3415756

[15] Unnisa, Nazeer, and Madhavi Tatineni. "Intelligent allocation strategy of mobile users for multi-access edge computing resources." In *2021 Emerging Trends in Industry 4.0 (ETI 4.0)*, pp. 1-7. IEEE, 2021. https://doi.org/10.1109/ETI4.051663.2021.9619420

[16] Sikkanan, Sobana, Chandrasekaran Kumar, Premkumar Manoharan, and Sowmya Ravichandran. "Exploring the potential of 5G uplink communication: Synergistic integration of joint power control, user grouping, and multi-learning Grey Wolf Optimizer." *Scientific Reports* 14, no. 1 (2024): 21406. https://doi.org/10.1038/s41598-024-71751-2

[17] Mahenge, Michael Pendo John, Chunlin Li, and Camilius A. Sanga. "Energy-efficient task offloading strategy in mobile edge computing for resource-intensive mobile applications." *Digital Communications and Networks* 8, no. 6 (2022): 1048-1058. https://doi.org/10.1016/j.dcan.2022.04.001

[18] Saeed, Mamoon M., Rashid A. Saeed, Rania A. Mokhtar, AbdulGuddoos SA Gaid, Zeinab E. Ahmed, and Mohammed S. Elbasheir. "Multi-Access Edge Computing Using Intelligent Mobile User Resource Allocation In 6G." In *2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, pp. 369-374. IEEE, 2024. https://doi.org/10.1109/MI-STA61267.2024.10599704

[19] 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

[20] 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

[21] Nakamura, Yugo, Teruhiro Mizumoto, Hirohiko Suwa, Yutaka Arakawa, Hirozumi Yamaguchi, and Keiichi Yasumoto. "In-situ resource provisioning with adaptive scale-out for regional iot services." In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 203-213. IEEE, 2018. https://doi.org/10.1109/SEC.2018.00022

[22] Dada, Emmanuel, Stephen Joseph, David Oyewola, Alaba Ayotunde Fadele, and Haruna Chiroma. "Application of grey wolf optimization algorithm: recent trends, issues, and possible horizons." *Gazi University Journal of Science* 35, no. 2 (2022): 485-504. https://doi.org/10.35378/gujs.820885

[23] Rezaei, Hossein, Omid Bozorg-Haddad, and Xuefeng Chu. "Grey wolf optimization (GWO) algorithm." *Advanced optimization by nature-inspired algorithms* (2018): 81-91. https://doi.org/10.1007/978-981-10-5221-7_9

[24] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." *Advances in engineering software* 69 (2014): 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[25] Jiang, He, Shuwei Zhang, Zhilei Ren, Xiaochen Lai, and Yong Piao. "Approximate Muscle Guided Beam Search for Three-Index Assignment Problem." *arXiv preprint arXiv:1703.01893* (2017).