MEC Network Resource Allocation Strategy Based on Improved PSO in 5G Communication Network

Yu Chen, Chongqing Industry Polytechnic College, China*

ABSTRACT

Relying on features such as high-speed, low latency, support for cutting-edge technology, internet of things, and multimodality, 5G networks will greatly contribute to the transformation of Web 3.0. In order to realize low-latency and high-speed information exchange in 5G communication networks, a method based on the allocation of network computing resource in view of edge computing model is proposed. The method first considers three computing modes: local device computing, local mobile edge computing (MEC) server computing, and adjacent MEC server computing. Then, a multi-scenario edge computing model is further constructed for optimizing energy consumption and delay. At the same time, the encoding-decoding mode is used to optimize PSO algorithm and combined with the improvement of fitness function, which can effectively support the communication network to achieve reasonable allocation of resources, ensuring efficiency of information exchange in the network. In the end, the results show that when the number of users is 500, the method can complete the task assignment within 44s.

KEYWORDS

5G Communication Network, Edge Computing, Fitness Function, Particle Swarm Optimization, Resource Allocation, Web 3.0

1. INTRODUCTION

Growth in mobile and Web traffic in new application requirements have attached requirements with higher levels in the service capability of 5th Generation (5G) mobile communication network (Sami, et al., 2021; Islambouli, et al., 2020, June; Mansour, et al., 2022). With the rise and development of the Metaverse, the emergence of computing- intensive and delay- sensitive applications with big amount makes users' requirements for service quality increase exponentially (Sami., & Mourad.., 2020; Inan., & Dikenelli., 2021). The final form of the Metaverse must be decentralized, and the current network ecology cannot fully meet the needs of Metaverse decentralization. Some people believe that the coming Web3.0 era is highly coincident with the network ecology required by the

DOI: 10.4018/IJSWIS.328526 *Corresponding Author

Metaverse. Web3.0 is expanding the data center to the edge (Chen, et al., 2022; Zhang, T. 2022). Compared with the current amount of Internet data, the amount of data generated and consumed in the Metaverse will be hundreds of times higher than the current amount. Relying on high-speed, low latency, and multimodal characteristics, 5G networks have greatly changed the possibilities of Web 3.0 applications. 5G networks can provide faster and more stable network services, support more new technologies, achieve the interconnection of everything, and greatly contribute to the transformation of Web 3.0. In the era of Web 3.0, the increase in the amount of task calculation results in local devices being unable to handle the corresponding computing tasks, while the cloud computing acts as solution of insufficiency in computing power (Tiwari, A., & Garg, R., 2022; Hussain., & Sayed., 2021). However, cloud computing also generates problems like costs of data transmission, cloud storage cost, Internet access management and security (Al-Qerem, et al., 2020; Stergiou, et al., 2021). Therefore, finding a reasonable network resource allocation method is crucial to support 5G networks to provide high-quality user services.

As Mobile Edge Computing (MEC) emerges, the device-cloud architecture is transformed with device-edge-cloud, thus reducing latency accordingly. Additionally, computing task throughput is improved by the strategy of allocating 3 computing reasonably, thus better meeting to the users' experience quality requirements can be better met and maximize economic benefits (Mychael, et al., 2022). In MEC environments, if the number of concurrent users is large, edge base stations may be overloaded. MEC reduces server load through swarm intelligence collaboration technology. Group intelligence collaboration technology uses a large number of base stations to complete tasks that cannot be completed by a single base station. Edge servers can also collaborate to perform tasks to balance network load. However, swarm intelligence collaboration technology requires the use of a large number of devices, which is suitable for the case of a large number of user devices and a small storage capacity of a single device. In PSO, each bird is considered a particle, and the bird swarm is considered a particle swarm, and each particle is encoded as a task resource scheduler. The main goal of PSO is to find the optimal particle from the population after multiple iterations of updates, that is, the optimal task resource scheduling program.

Relying on the model of edge computing, a method of communication network based on network resource allocation is proposed with major innovations as below:

1) Consider three network scenarios: local computing, offloading to MEC server in local area, and offloading within regional MEC server to build a multi-scenario task analysis mode. Aiming at optimizing delay in system computing and energy consumption, a mathematical model of strategic resource allocation for network computing is established to optimize the network operation state.
2) Using the encoding-decoding mode to optimize and design particle swarm algorithm, which improves the computing efficiency of PSO algorithm's computing efficiency, thus further optimizing the adaptation function of resource allocation model, realize the analysis on collaborative optimization of delay and energy consumption, which effectively support the efficient and stable operation of communication network.

2. RELATED WORK

According to 2020 Cisco White Paper, it is expected that from 2018 to 2023, global Internet users will show a rising trend, with 6% annual growth rate. By 2023, there will be nearly 300 million apps downloaded on mobile devices worldwide, producing trillions of gigabytes of data every day.

In the future, 5G communication networks will have to face new requirements, such as more applications that require complex computing, lower the latency of task execution, lower device energy consumption and higher service quality. The above service applications all need the support of a reasonable and reliable computing resource allocation strategy (Daniel, et al., 2022).

MEC utilizes servers deployed at network edge, thus providing computing and storage resources to users for enhancing the computing capabilities of user devices. Moreover, due to its "closeness" to the user or terminal devices, it can provide timely service response with minimal delay (Tang, & Hu., 2020; Jargis., et al., 2022). The advantages of MEC are: (1) Provide services to users through MEC server at the edge of network instead of remote cloud data centre, which greatly reduces the task transmission delay and transmission energy consumption (Wu, et al., 2021); (2) Minimize down and up network core traffic while saving a lot of bandwidth and better awareness in location and context, resulting in data security and privacy protection (Ge, et al., 2021).

After the network computing offloading scheme is determined, the terminal devices or terminal users offload tasks to MEC server based on the designed scheme. However, the server resources are limited (Wang, C., Feng, D., Zhang, S., et al., 2020), and how to allocate the limited MEC server resources is also a key issue.

Corresponding researches and analysis on the rational allocation of MEC resources. Ref. (Ren, & Xu., 2020) minimized the delay and energy consumption of file transmission under the constraints of given secure transmission conditions, and realizes the design of optimal codebook rate and the analysis of computing task allocation. Ref. (Yang, et al., 2019) took video and audio services as the background, enabling green MEC to achieve efficient resource allocation, reduce analysis energy consumption, and improve the average video bit rate of client. But single-user MEC system structure is the smallest and simplest system model. In a network environment, the coexistence of multiple users is the norm.

For MEC application scenario where multiple users coexist, scholars have carried out research on it. Ref. (Tao, et al., 2021) studied a multi-task MEC network with assistance of UAV considering the requirements of time-sensitive tasks. It effectively reduced energy consumption of IoT devices in total while meeting the needs of different types tasks. Ref. (Fan, et al., 2020) considered the scenarios covered by Wi-Fi and cellular networks, and combines linear programming with alternating techniques to effectively solve the non-convex problem of task offloading decision-making. This solution can greatly improve the system performance. In addition, Ref. (Zhang., & Du., 2020) considered reducing computing energy consumption and delay, and proposed a new learning algorithm using deep deterministic policy gradient and edge computing optimization offload algorithm based on candidate network optimization, this new learning algorithm is used for solving the uneven allocation of network resource. Ref. (Li., & Zhang., 2021) used genetic algorithm and divided-time-based resource allocation algorithm to search for optimal decisions to reduce cloud communication traffic.

However, it should be pointed out that most of current analysis methods are multi-user resource invocation research on a single MEC, and the introduction of adjacent MEC servers into communication network to be studied can more effectively improve the rationality of network resources. At the same time, optimizing and improving the solution algorithm based on mathematical model is an important guarantee for realizing the optimal allocation of network resources.

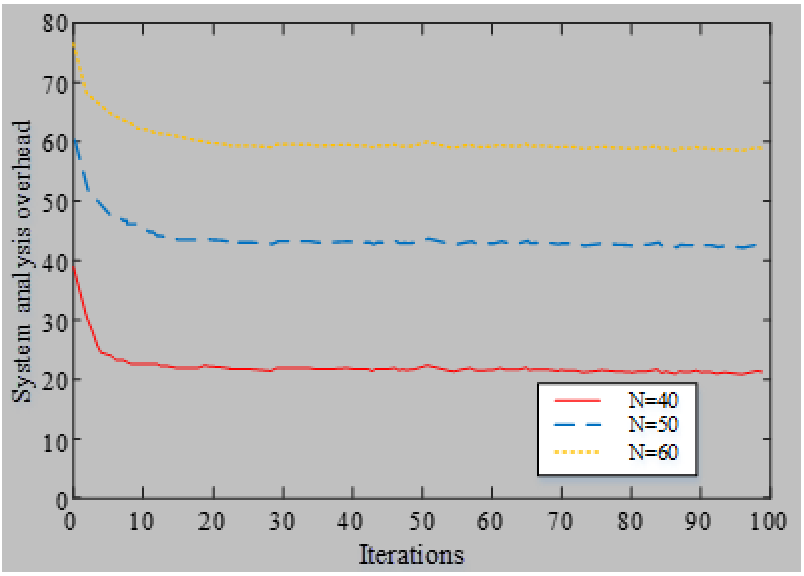
3. SYSTEM MODEL AND OPTIMIZATION PROBLEM MODELING

3.1 System Model

The communication network system model includes a Macro Base Station (MBS), M cell base stations and V users. Base stations and users are randomly distributed in the coverage area of MBS. The base stations and users are denoted as $M = \{1,2,\&,i,\& ,m\}$ and $V = \{1,2,\&,j,\&,v\}$. where i represents MBS. $i > 0$ and j represent cell base stations and users respectively.

MBS and cell base stations are deployed with MEC servers, providing users with computing offloading services. The computing task of user is denoted as $D_l = \{a_l, b_l, t_l^{\max}\}$, where a_l shows updated data amount to complete the computing task, including code, configuration files, etc.; $t_l^{\max}$ represents delay for task completion maximumly; b_l shows computing quantity required to process

Figure 1. Structure diagram of communication network system model



each bit of data. This parameter may vary for different tasks and is available through the task profiler. In other words, $h_l = a_l b_l$ computation is required to complete this computing task. In order to complete the computing task, users have freedom of local computing or offload the computing task to MBS or cell base station.

3.2 Communication Model

Based on the processing of MEC server, the amount of result data is very small and leads a huge gap with uploaded data amount, in the stage of task backhaul, delay and energy consumption are far less than the stage of task upload and execution, thus being ignored accordingly (Tri, et al., 2021). Model for communication just cares upload phrase of the task rather than temporary return phrase.

(1) Communication between users in the cell and base station

In the cell, the user uploads tasks to MEC server via the channel of wireless uplink transmission. For the convenience of calculation, it is assumed that the spectrum is allocated orthogonally in the same cell, and there is no interference. Define user j channel bandwidth as w_j, then the uplink transmission rate is:

$$s_j = w_j \log_2(1 + \frac{p_j r_j}{\sigma^2}) \tag{1}$$

where p_j represents the mobile user's power in transmission j, r_j symbolizes channel gain between the mobile user j and MBS, and σ^2 is Gaussian channel noise variance.

User device delay for transmission via uplink is:

$$T_j^1 = \frac{B_j}{s_j} \tag{2}$$

where B_j represents the task data size.

(2) Inter-cell base station communication

The transmission of data forwarding between local- area-based MEC server and nearby-area-based MEC server uses the two-way sided base stations with high speed of backhaul link, whose transmission is bidirectionally. While the research concern local-area-involved and nearby-area MEC server with data forwarding. Thus, here is an assumption: when e refers to rate of transmission within two base stations of places, guided by the backhaul link, delay for forwarding the two base stations with calculation task is:

$$T_j^2 = \frac{B_j}{e} \tag{3}$$

3.3 Computing Model

Define f_j as mobile device computing power (CPU), in view of local devices, delay for computing task execution is:

$$T_j^3 = \frac{G_j}{f_j} \tag{4}$$

G_j represents the amount of computing resources required to execute the task. Accordingly, required consumption of energy for computing tasks to execute on local devices is:

$$E_j^3 = \lambda_j G_j = 10^{-27}(f_j)^2 G_j \tag{5}$$

where λ_j refers coefficient of energy consumption, which represents the energy consumed by mobile devices per CPU cycle. Define the energy consumption model as $\lambda_j = 10^{-27}(f_j)$.

Considering the delay cost and cost for the execution of computing task in energy computing, μ is taken as delay in task execution parameter in preference, and η as energy consumption for task execution parameter in preference, which meets $\mu + \eta = 1$. Adjusted settings for preference parameter are done through types of tasks or users in specifically. Delay and energy consumption summarized in weight are computing tasks in execution costs, in total, local-device- based executing task calculation is:

$$Cost_j^3 = \mu T_j^3 + \eta E_j^3 \tag{6}$$

(1) Offload to the local area MEC server

In this network scenario, the user preferentially offloads tasks to MEC server, with local deploy based on cell macro station, thus processing the tasks, sending results to mobile devices. When $source_j$ is defined as the computing resources for local-based MEC server to assign tasks, its sum shall meet to constraint $\sum_{j=1} source_j \le source_{\max}$, where $source_{\max}$ indicates available computing resources for local MEC server with maximum. Delay for task processing is:

$$T_j^{4'} = \frac{G_j}{source_j} \tag{7}$$

The total delay of offloading tasks to MEC server in the local area refers to total delay for upload, processing, and download. Download delay could be ignored when it is too tiny. Thus, task offloading delay in total is:

$$T_j^4 = T_j^3 + T_j^{4'} \tag{8}$$

As with task offloading, in idle state, device energy consumption could be ignored for simple calculations. Energy consumption in processing task offloading involves task uploading energy consumption from devices to local MEC server. Thus, requiring task offloading energy consumption as:

$$E_j^4 = p_j T_j^3 = p_j \frac{G_j}{f_j} \tag{9}$$

where p_j refers mobile device power for transmission. In total, cost for offloading task to MEC server in local area is:

$$Cost_j^4 = \mu T_j^4 + \eta E_j^4 \tag{10}$$

According to equations (6) and (10), whether the offloading task via local MEC server improve tasks performances could be evaluated. When the cost from offloading tasks to local MEC server in system is lower than that of local-device-based computing task completion, it is good for the offloading task computing decision to the local MEC server, whose resources are limited. Thus, when too many users are using it in the same time, offloading tasks needs computing resources for local MEC servers will be exhausted for sure.

Hence, if offloading task amount is more than local-area-based MEC server in loading maximum, users can choose nearby MEC server to forward computing tasks and remain computing resources in server.

(2) Offload to the nearby regional MEC server

It is defined that MEC server' s computing resource allocation in the nearby area to tasks are $source'$, thus task-based calculation of computing resources need to meet the constraint

$\sum_{j=1} source'_j \leq source'_{\max}$, where $source'_{\max}$ represents the available computing resources remained for nearby MEC servers. Thus, required delay for nearby MEC server in task processing is:

$$T_j^5 = \frac{G_j}{source'_j} \tag{11}$$

If user offloads computing tasks to MEC server near his place, to forward task data, additional delay is required due to the distance between two servers. Besides, delay for nearby MEC server for task offloading equals to the delay sum of uploading, forwarding and processing. Thus, task offloading delay in total is:

$$T_j = T_j^1 + T_j^2 + T_j^3 + T_j^4 + T_j^5 \tag{12}$$

Likewise, required energy consumption to offload tasks involves task uploading or $E_j = E_j^4$. In total, offloading task costs to nearby regional MEC server is:

$$Cost_j = \mu T_j + \eta E_j \tag{13}$$

3.4 Problem Definition

Important indicators of measuring performances of systems are delay and energy consumption (Chang, et al., 2022; Gao, et al., 2022). when the cell-based tasks in execution delay and energy consumption amount is taken as total cost for system execution, which optimizes decisions about offloading, location of bandwidth and computing resources, problem in optimization model is:

$$\begin{aligned}
&\min_{X,Y} \sum_{j=1}^{J} Cost_j^3(1-x_j) + Cost_j^4(1-y_j) + Cost_j x_j y_j \\
&s.t. C1: \sum_{j=1}^{J} source_j \leq source_{\max} \\
&\quad C2: 0 \leq source_j \leq source_{\max} \\
&\quad C3: \sum_{j=1}^{J} source'_j \leq source'_{\max} \\
&\quad C4: 0 \leq source'_j \leq source'_{\max} \\
&\quad C5: \sum_{j=1}^{J} h_j \leq H_{\max} \\
&\quad C6: 0 \leq h'_j \leq H_{\max} \\
&\quad C7: x_j, y_j \in [0,1]
\end{aligned} \tag{14}$$

where X, Y are the vector sets of offloading decisions x_j and y_j respectively. H is the vector set of bandwidth resource allocation h_j, and $H_{\max}$ is the total system communication bandwidth. $source_j$, $source'_j$ are the computing resource allocation vector sets respectively, $source_{\max}$, $source'_j$ are the available maximum computing resource capacity provided by local-area and nearby- area MEC servers.

Constraints C1 and C2 represent local region MEC server computing resource constraints. Constraints C3 and C4 represent nearby regional MEC server computing resource constraints. Constraints C5 and C6 represent system communication bandwidth resource constraints. Constraint C7 ensures that the offloading decisions x_j, y_j are all $[0,1]$ variables.

Since there are integer variables $x_j, y_j \in [0,1]$ and continuous variables h_j, $source_j$, $source_j'$, and the variables are coupled, the problem is a MINLP problem. There are three ways to execute each task in the cell. When cell computing task number is N, the problem scale is 3^N, and the computing complexity rises along with task number increase accordingly.

4. PROBLEM SOLVING AND ALGORITHM DESIGN

4.1 Particle Encoding and Decoding

It is encoded by combining the length of subtasks and the number of base stations, and the length of a single particle is a one-dimensional matrix whose length is twice the number of subtasks. From the first element to the last element in the matrix, every two elements represent the priority of subtask and the scheduling sequence number of base station, for example: {2,4,5,2,1,1}. From right to left, tasks with priority 2 are executed on the fourth base station, and tasks with priority 5 are executed on the second base station. A task with a priority of 1 is executed at the first base station. The higher the priority, the more preferentially it is offloaded to the execution edge cloud. Meanwhile, try to make each edge base station call evenly.

During decoding, the order of subtasks executed at the edge base station is extracted from particles, and the execution time of each subtask before the execution of each subtask at edge base station and the execution time of subtasks adjacent to its parent task are comprehensively considered. The execution time is arranged for each subtask in order of priority, and the execution time is as early as possible.

The update formula of particles is adapted to optimize problems in continuous variables. Thus, some optimizations are made to solve non-integer- optimized discontinuous variables. The main method is to round each variable in the particle to the nearest integer. Because the variation range of each variable has a certain limit, a circular strategy is designed with the limit value as the radius. When the variable exceeds this radius, use gradient descent to bring the variable back within the limits.

4.2 Fitness Function

User attribute (user): the number of users V, user serial number $j = (1,2,3,\&,V)$, each user contains k tasks, $k = (1,2,3,\cdots,Q)$, and the user j can represent

$$user_j = \{Z_{j1}, Z_{j2}, \&, Z_{jk}\} \tag{15}$$

Each task contains w subtasks, $w = \{1,2,3,\cdots,W\}$, then the k task of j user can be represented by

$$Z_{jk} = \{z_{jk}, 1, z_{jk}, 2, \cdots, z_{jk}, W\} \tag{16}$$

The subtasks are as follows:

$$z_{jk,w} = \left\{r_{jkw}, s_{jkw}, t_{\max,jkw}\right\} \tag{17}$$

where r_{jkw} represents the data volume of subtask, s_{jkw} represents the unit data energy consumption of subtask, and $t_{\max,jkw}$ represents the maximum delay for the completion of subtask.

Local device (ge): geb_j represents the energy consumption of devices where the j user is located to transmit unit data, and gec_j represents the energy consumption of devices where the j user is located to process the unit data. ν_{jkw} indicates whether the w subtask in the k task of j user needs to be offloaded to processing edge cloud. $\nu_{jkw} \in [0,1]$, 0 means that the subtask is processed locally, and 1 means that it is offloaded to edge base station for processing.

Edge Base Station (BS): M refers to base station amount, that is $i = (1,2,3,\&,M)$, and $e_{BS,i}$ represents the per unit of energy consumption for data processing by the i algorithm based on edge base station.

Channel (CH): The number N, the spectral bandwidth and bandwidth of each channel are SB, the signal-to-noise power is σ^2, and the channel transmission unit data energy consumption e_{ij}. Channel transmission unit data energy consumption e, user j and base station i channel gain λ_{ij}, there is channel multiplexing. u_{jkw} represents the influence factor of surrounding environment when the w subtask uses the s channel to transmit data, $d \in N$.

Referring to the formula proposed by Shannon, rate of transmission of data uplink in any subtask is as follows

$$d_{ikw}^{j} = SB * \log_2(1 + geb_j * \lambda_{ij} / (u_{ikw}^{j} + \sigma^2)) \tag{18}$$

Then the task data uplink transmission rate of k task of j user is as follows:

$$d_{ij} = \sum_{w=1}^{W} \nu_{jkw} * d_{ikw}^{j} \tag{19}$$

ν_{jkw} can filter out subtasks that are not offloaded to edge base stations. In total, subtask offloading with consumption to the edge cloud equals to energy consumption for data transmission and the task processing, as shown in equation (20):

$$e_{jkw} = e_{ij} * geb_j * r_{jkw} / d_{ikw} + s_{jkw} * e_{BS,i} \tag{20}$$

At this time $\nu_{jkw} = 1$, the power consumption of subtasks that are not offloaded to edge base station is as equation (21):

$$e_{jkw} = s_{jkw} * gec_j \tag{21}$$

At this time $\nu_{jkw} = 0$, then in total, energy consumption of k task of user j is shown in equations (22) and (23):

$$e_{ij} = \sum_{w=1}^{w_1} e_{jkw} + \sum_{w=1}^{w_2} e_{jkw} \tag{22}$$

$$w_1 + w_2 = w \tag{23}$$

Considering the limited resources and task-related requirements of edge base stations, equations (19), (22) and (23) should satisfy equations (24) and (25):

$$r_{jkw} / d_{ikw} \leq t_{\max,jkw} \tag{24}$$

$$geb_j \ll gec_j \tag{25}$$

To sum up, the fitness functions are $f_1 = \max(d_{ij})$ and $f_2 = \min(e_{ij})$. for improving algorithm performance, the author modifies to $f = -\max(d_{ij}) + \min(e_{ij})$.

4.3 Improved PSO Solution Steps

The determining factors of particle swarm speed update mainly include three parts: the first part is mainly its own original speed, which can also be called "inertial effect". The second part is mainly about the optimal location of one's own memory, which can also be called "cognitive influence". The third part is mainly the optimal location of all particle memories, which can also be called "social influence". Finally, the particle with swarm is for offloading computing.

Algorithm 1 is an offloading process based on PSO algorithm.

Input:

1) Local task set $N = \{n_1, n_2, \cdots, n_k\}$, MEC server set $M = \{m_1, m_2, \cdots, m_i\}$, channel gain matrix H.

2) Algorithm control parameters: particle swarm size $V = 30$, iteration number $\max Gen = 120$. The speed boundary $s_{\max}$ is twice more than MEC servers, and the location boundary $i_{\max}$ refers to MEC servers. Learning factor $c_1 = 1.3$, $c_2 = 1.3$, inertia factor $\xi = 0.5$, penalty factor $\kappa = 1.0 \times 10^{-2.5}$.

Output:

The optimal fitness function $fitness(\ )$.

Initialization:

1) Initialize the random position r_k and velocity v_k of each particle, where k represents the k particle.

2) fitness value initialization: using local task set K, MEC service period set M and the channel gain matrix H to calculate the delay and energy consumption

3) Initialize the optimal distribution of particles and the optimal global distribution: set particle position as allocation of optimal task scheme p_{best} individually, thus taking group optimal allocation scheme g_{best} with the smallest fitness value

Iterative calculation:

4) Let the number of iterations be $t = 0$

5) while $t \leq \max Gen$

a) Speed update. Using particle dimension individually to update the velocity $X[k]$, when it is more than $v_{\max}$, let $X[k] = v_{\max}$, the updated equation is:

5)while $t \leq \max Gen$

$$X[k] = \xi * X[k] + c_1 * rand()(p_{best} - V[k]) + c_2 * rand()(g_{best} - V[k]) \quad (26)$$

where, $rand()$ is a random number from 0 to 1.

b) Update location with independent dimension in particle with $V[k]$, when $V[k]$ position is greater than $p_{\max}$, let $V[k] = p_{\max}$. The update equation for particle position is:

$$V[k] = V[k] + X[k] \quad (27)$$

c) Update the optimal distribution of particles and global optimal distribution. If the value of updated fitness is lower than that of the current fitness, optimal allocation scheme p_{best} of the particle g_{best} and the optimal fitness function $fitness(\ )$.

d) $t = t + 1$

6) End

Output result:

7) Obtain the optimal distribution vector $V[k] = g_{best}$ and the optimal fitness function $fitness(\ )$.

8) When optimal solution is dissatisfied, step 1 follows. Finally, by means of centralized control, $v_j(j = 1, 2, \cdots, M) = 0$ puts the task into local devices for execution, and $v_k = i(i = 1, 2, \cdots, M)$ puts the task into MEC server corresponding to the number i for execution.

5. EXAMPLE VERIFICATION AND RESULT DISCUSSION

The algorithm mentioned in chapter also uses Python language for simulation experiments to prove the superiority of the resource allocation method in the proposed algorithm. The experiment evaluates and analyzes the performance of the proposed algorithm by changing relevant parameters.

In the scenario of MEC system simulation with multiple users and cells, located in central cell 1, the MBS is deployed with an MEC server, and the coverage radius of MBS is 750m. two- regional MBSs connected via backhaul link have rate of forwarding at 15MB/s. Setting distribution of mobile devices to cell at amount of $25 \sim 100$ Bandwidth of the system communication is 50MHz, available computing resources for local MEC server at maximum is 15GHz to 110GHz.

Task data size and computing resource amount both obey a normal distribution. The user's preference parameters for both delay and energy consumption are set to 0.5, as summarized in the research, the simulated parameters are shown in Tab. 1.

To discusse the proposed resource allocation method, the simulation verification under different user access numbers is firstly implemented. Figure 2 shows the system cost as a function of number of iterations.

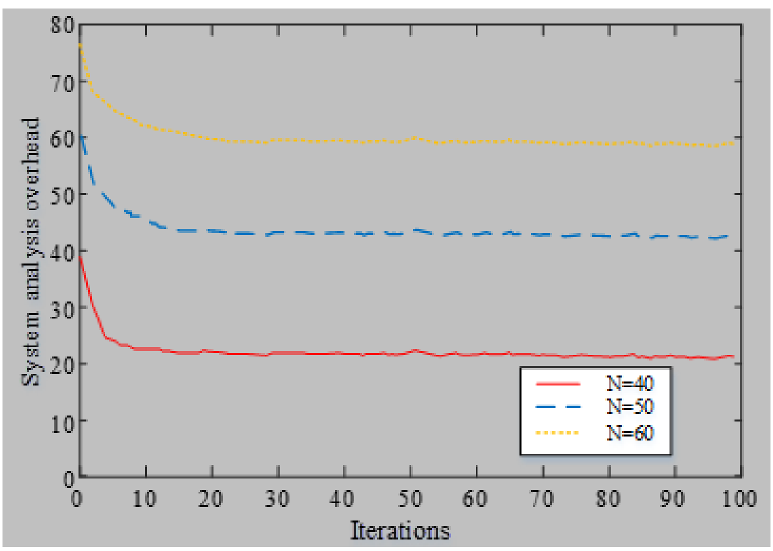
In Figure 2, in system cost, there is a gradual decrease in the iterative process of the method. Through limited iteration amount, it transfers to be stable. Thus, the algorithm constantly generating new solutions with feasibility and jump-out-of local state at optimal, reducing system cost accordingly. Besides, the convergence of proposed algorithm convergence and device number are in line with each other.

Figure 3 shows average task completion time compared with the number of users under different numbers of edge servers.

Table 1. Resource allocation simulation parameter table

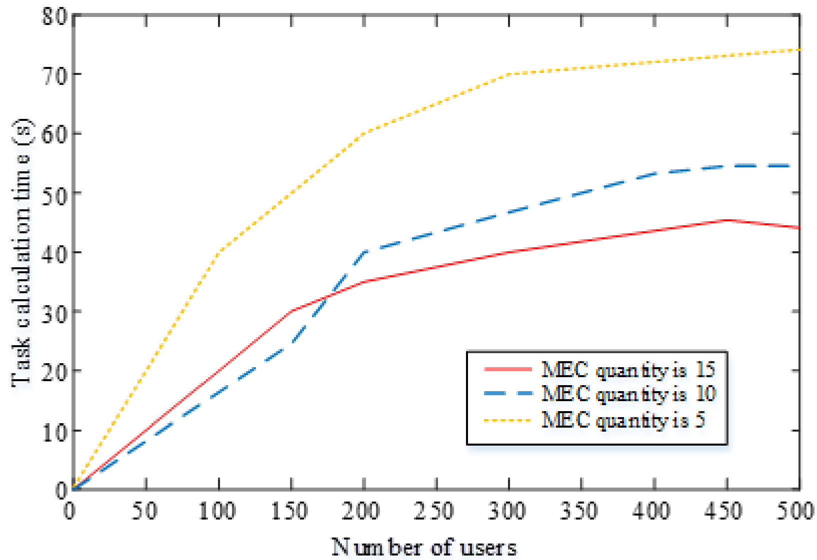
Item	Details
Number of mobile devices	25, 50, 75, 100
System communication bandwidth	
Mobile device transmit power	
Wireless channel gain	
Gaussian channel noise	2.5×10^{-12} W
Computing resources of local nearby MEC server at maximum	
Maximum computing resources of MEC server in the nearby area	
Mobile device computing power	
task data size	
Task computing resources	
Data transfer rate between base stations	
Number of particles in the algorithm	
Inertia weighting factor	
Acceleration constant	
Maximum number of iterations	

Figure 2. Algorithm operation overhead



From Figure 3, when users' number is small, the computing resources when MEC is 5 is sufficient. At this time, improving the computing power of MEC has little effect, and MEC suits to requirements on computing tasks of mobile devices. As the number of users gains substantial increase, the effect of increasing the computing power of MEC becomes more and more obvious. When users' number rises to 300, completion time for task with 5 edge servers and MEC of 15 are already 30 seconds

Figure 3. The average task completion time under different number of edge servers

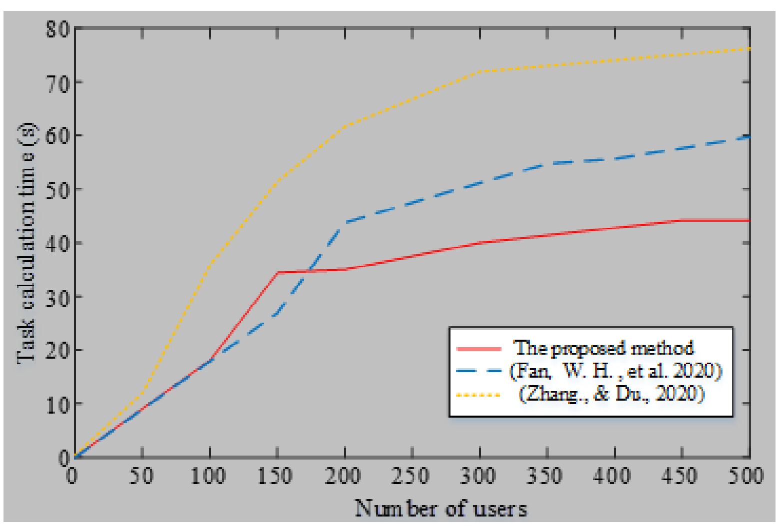


apart. When there are increases in users' number, the difference in the average completion time of tasks between different numbers of edge servers becomes larger and larger. When the number of users increases, edge servers increase and effectively reduce the average task completion time.

Furthermore, Reference (Fan, et al., 2020) and Reference (Zhang., & Du., 2020) are for comparative analysis, and the proposed allocation algorithm runs in the same environment to the verification of proposed algorithm in optimizing its performance. Figure 4 compares users' task completion time in different methods averagely.

In Figure 4, it shows the method proposed by the paper can effectively reduce the task completion time. When users reach to 500, algorithm proposed in the paper can realize data analysis within 44s,

Figure 4. The average completion time of tasks under different methods



which is about 16s and 32s shorter than the (Fan, et al., 2020) and (Zhang., & Du., 2020). Besides, the edge server cannot complete all the computing tasks within the specified time, the computing tasks are queued up. Therefore, in average, completion time for users to finish the task of three algorithms is on the rise with the growth of the number of users. The reason is that the proposed method integrates multi-scenario task offloading methods such as device execution and offloading locally to MEC server, and offloading to the nearby area MEC server, which greatly improves the rationality of network task computing allocation. The comparison method does not involve the integrated analysis of multiple computing scenarios, and the computing model is relatively simple and single.

In addition, this paper sets mobile device number of the system to 50 for offloading task number based on different methods in multi-computing scenarios. Figure 5 shows the amount of offloading tasks based on different algorithms with individual execution mode.

Figure 5 shows offload task amount of algorithms for local device, local-area and nearby regional MEC servers, whose number of computing tasks of proposed method are 5, 17 and 28 respectively. More computing task offloading are to the nearby MEC servers, saving energy and reducing latency. However, the comparison method only implements task allocation on the local device and local MEC server, which will result delay and energy consumption in large amount. The Figure 4 also shows the disadvantage of comparison method in computing efficiency. In this paper, adjacent MEC servers are introduced into the model building to participate in network resource allocation and to optimize and improve PSO algorithm, thus improving network resource allocation quality and achieve the maximum and optimal satisfaction of network user service requirements.

In addition, this paper also conducts corresponding research on the energy consumption of resource allocation methods. Figure 6 shows the energy consumption of algorithm under different algorithms.

Figure 6 indicates interaction of energy consumption of different methods and the users' number. The increase in users' number increases energy consumption. Owing to coding system, which optimizes PSO algorithm, and the improved adaptation function of algorithm, the optimization and convergence of proposed algorithm can be effectively enhanced, and the optimization of network resource allocation can be achieved. When users reach to 150, energy consumption calculation is 132.4J, which is smaller than the comparison method. When users is 450, the method used in the paper can still maintain a low energy consumption, and the computing cost is 198.5J, which is 114.1J and 129.7J lower than those in (Fan, et al., 2020) and (Zhang., & Du., 2020).

Figure 5. Resource allocation under different methods

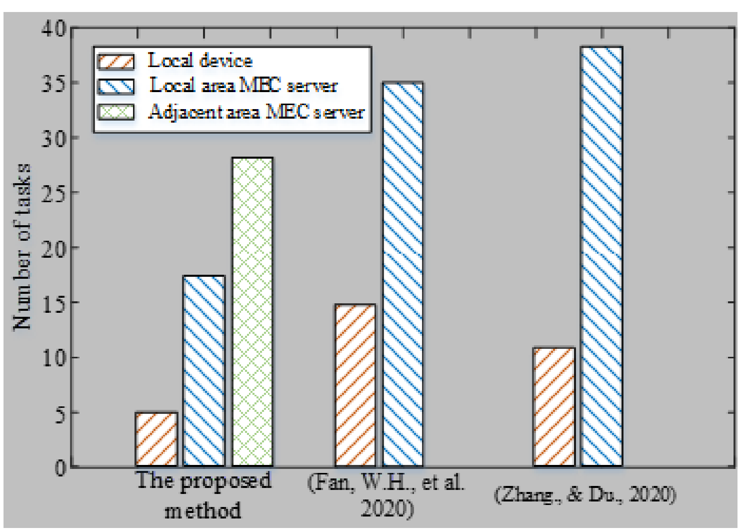
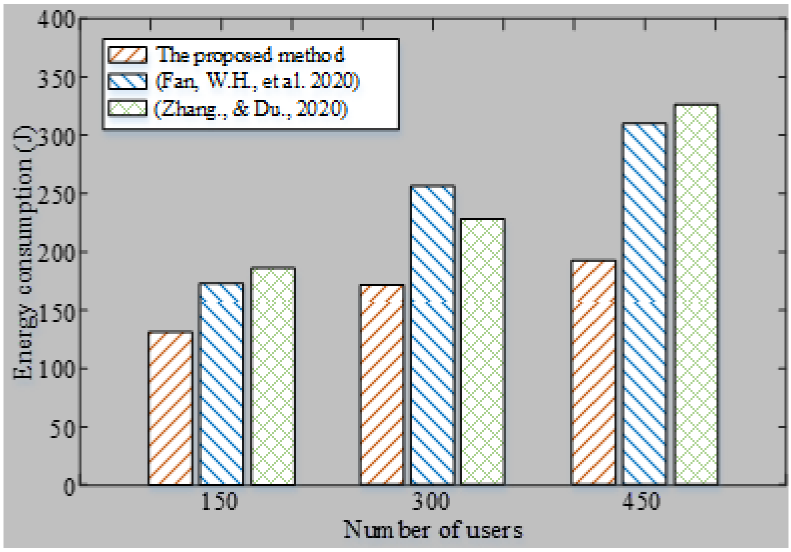


Figure 6. Computing energy consumption under different methods



6. CONCLUSION

For 5G communication network, a method with resource allocation and basis of edge computing is proposed. The method integrates three computing modes: local device computing, local MEC server computing and adjacent MEC server computing to construct a multi-scenario edge computing model, which can satisfy the rationality of resource allocation methods. Besides, the encoding-decoding mode is used to optimize PSO algorithm, it further supports the communication network to achieve reasonable resource allocation. Results simulation indicate the feasibility of this method in ensuring the full use of network resources and support the efficient operation of communication network, and it has good infrastructure support significance for accelerating the transformation and landing of Metaverse and Web 3.0.

However, the proposed strategy also has some limitations:

1) At present, the research on MEC system is only in the theoretical stage. Due to the limitation of experimental conditions, this paper fails to comprehensively and practically concern various factors of environment, and it is only verified by simulation experiments. In future study, algorithm verification combined with the real MEC system platform to further improve our proposed strategy should be implemented.
2) The proposed model focuses more on resource utilization and real-time network response, while neglecting the privacy of user data. In the era of Web 3.0, data privacy will be highly valued. Therefore, in the next work, blockchain and Federated learning technologies will be introduced into the designed architecture, effectively solving the privacy problem of user data on the premise of ensuring reliable network service performance, which will better meet the needs of the Metaverse and Web 3.0 era.

Conflicts of interest: Not applicable.
Availability of data and material: Not applicable.
Code availability: Not applicable.
Funding statement: This work was not supported by any found projects.

REFERENCES

Al-Qerem, A., Alauthman, M., Almomani, A., & Gupta, B. B. (2020). IoT transaction processing through cooperative concurrency control on fog–cloud computing environment. *Soft Computing*, *24*(8), 5695–5711. doi:10.1007/s00500-019-04220-y

Chang, D., Wang, Y., & Fan, R. (2022). Forecast of Large Earthquake Emergency Supplies Demand Based on PSO-BP Neural Network. *Tehnicki Vjesnik (Strojarski Fakultet)*, *29*(2), 561–571. doi:10.17559/TV-20211120092137

Chen, C., Zhang, L., Li, Y. H., Liao, T., Zhao, S., Zheng, Z., Huang, H., & Wu, J. (2022). When Digital Economy Meets Web3.0: Applications and Challenges. *IEEE Open Journal of the Computer Society*, *3*, 233–245. doi:10.1109/OJCS.2022.3217565

Daniel, , Lew & Ooi. (2022). Mobile Interactive System in Virtual Classroom based on TPACK: A Study from Students' Perspectives. Journal of Logistics. *Informatics and Service Science*, *9*(3), 159–171.

Fan, W. H., Han, J. T., & Yao, L. (2020). Latency-energy optimization for joint Wi-Fi and cellular offloading in mobile edge computing networks. *Computer Networks*, *181*(1), 1–11.

Gao, M., Fang, S., Wang, J., Zhang, X., & Cao, Y. (2022). A Dual Frequency Predistortion Adaptive Sparse Signal Reconstruction Algorithm. *Tehnicki Vjesnik (Strojarski Fakultet)*, *29*(2), 580–589. doi:10.17559/TV-20210728035851

Ge, L. P., Zhou, J. H., & Zheng, Z. K. (2021). Dynamic Hierarchical Caching Resource Allocation for 5G-ICN Slice. *IEEE Access : Practical Innovations, Open Solutions*, *9*(1), 134972–134983. doi:10.1109/ACCESS.2021.3116602

Hussain, A. I., & Sayed, A. Z. (2021). Optimal user association of lte/wi-fi/wi-gig bands in 5g cellular networks. *International Journal on Semantic Web and Information Systems*, *17*(2), 22–40. doi:10.4018/IJSWIS.2021040102

Inan, E., & Dikenelli, O. (2021). A Semantic-Embedding Model-Driven Seq2Seq Method for Domain-Oriented Entity Linking on Resource-Restricted Devices. *International Journal on Semantic Web and Information Systems*, *17*(3), 73–87. doi:10.4018/IJSWIS.2021070105

Islambouli, R., Sweidan, Z., Mourad, A., & Abou-Rjeily, C. (2020, June). Towards trust-aware IoT hashing offloading in mobile edge computing. *2020 International Wireless Communications and Mobile Computing (IWCMC),* 2216-2221.

Jargis, , Razzaque, & Rahman. (2022). A Stackelberg Game-Based Dynamic Resource Allocation in Edge Federated 5G Network. *IEEE Access : Practical Innovations, Open Solutions*, *10*(1), 10460–10471.

Li, Z. J., & Zhang, X. L. (2021). Resource allocation and offloading decision of edge computing for reducing core network congestion. *Computer Science*, *48*(3), 281–288.

Mansour, E., Shahzad, F., Tekli, J., & Chbeir, R. (2022). Data redundancy management for leaf-edges in connected environments. *Computing*, *104*(7), 1565–1588. doi:10.1007/s00607-021-01051-4

Mychael, , Arief, & Edi. (2022). Mobile Device Security: A Systematic Literature Review on Research Trends. *Methods and Datasets. Journal of System and Management Sciences*, *12*(2), 66–78.

Ren, P. Y., & Xu, Q. (2020). Delay and energy minimization for MEC - based secure communication. *Journal of Communication*, *41*(11), 52–63.

Sami, H., & Mourad, A. (2020). Dynamic on-demand fog formation offering on-the-fly IoT service deployment. *IEEE eTransactions on Network and Service Management*, *17*(2), 1026–1039. doi:10.1109/TNSM.2019.2963643

Sami, H., Otrok, H., Bentahar, J., & Mourad, A. (2021). AI-based resource provisioning of IoE services in 6G: A deep reinforcement learning approach. *IEEE eTransactions on Network and Service Management*, *18*(3), 3527–3540. doi:10.1109/TNSM.2021.3066625

Stergiou, C. L., Psannis, K. E., & Gupta, B. B. (2021). InFeMo: Flexible big data management through a federated cloud system. *ACM Transactions on Internet Technology*, *22*(2), 1–22. doi:10.1145/3426972

Tang, L., & Hu, H. (2020). Computation Offloading and Resource Allocation for the Internet of Things in Energy-Constrained MEC-Enabled HetNets. *IEEE Access : Practical Innovations, Open Solutions*, *8*(1), 47509–47521. doi:10.1109/ACCESS.2020.2979774

Tao, L. W., & Zhao, M. X. (2021). Collaborative offloading for UAV-enabled time-sensitive MEC network. *EURASIP Journal on Wireless Communications and Networking*, *2021*(1), 1–17. doi:10.1186/s13638-020-01861-8

Tiwari, A., & Garg, R. (2022). Adaptive Ontology-Based IoT Resource Provisioning in Computing Systems. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–18. doi:10.4018/IJSWIS.306260

Tri, D. T. (2021). Modeling Data Redundancy and Cost-Aware Task Allocation in MEC-Enabled Internet-of-Vehicles Applications. *IEEE Internet of Things Journal*, *8*(3), 1687–1701. doi:10.1109/JIOT.2020.3015534

Wang, C., Feng, D., Zhang, S., & Chen, Q. (2020). Video Caching and Transcoding in Wireless Cellular Networks With Mobile Edge Computing: A Robust Approach. *IEEE Transactions on Vehicular Technology*, *69*(8), 9234–9238. doi:10.1109/TVT.2020.2997344

Wu, Y. C., Dinh, T. Q., Fu, Y., Lin, C., & Quek, T. Q. S. (2021). A Hybrid DQN and Optimization Approach for Strategy and Resource Allocation in MEC Networks. *IEEE Transactions on Wireless Communications*, *20*(7), 4282–4295. doi:10.1109/TWC.2021.3057882

Yang, S. R., Tseng, Y. J., Huang, C. C., & Lin, W.-C. (2019). Multi-access edge computing enhanced video streaming: Proof-of-concept implementation and prediction/QoE models. *IEEE Transactions on Vehicular Technology*, *68*(2), 1888–1902. doi:10.1109/TVT.2018.2889196

Zhang, T. (2022). Teaching and Training System Based on WEB3.0 Technology. *2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, 1-4. doi:10.1109/ICATIECE56365.2022.10046862

Zhang, W. X., & Du, Y. W. (2020). Deep reinforcement learning-based optimization of lightweight task offloading for multi-user mobile computing. *Journal of Measurement Science and Instrumentation*, *2020*(1), 1–14.

Yu Chen, Master of Language Study, graduated from Chongqing University in 2013. Worked in Office of Academic Affairs, Chongqing Industry Polytechnic College. His research interests include communication technology.