




Privacy-preserving edge computing offloading scheme based on whale optimization algorithm

Zhenpeng Liu^{1,2} · Jingyi Wang¹ · Zilin Gao¹ · Jianhang Wei^{2,3} 

Accepted: 8 August 2022 / Published online: 29 August 2022
© The Author(s) 2022

Abstract

Aiming at the problem of user's task offloading in mobile edge computing and the potential leakage of location privacy during the offloading process, a privacy-preserving computing offloading scheme based on whale optimization algorithm is proposed. Using differential privacy technology to obfuscate the user's location information, the user can make task offloading decisions according to the obfuscated distance. Considering the delay, energy consumption, and their weighted sum, the offloading problem is modeled as a convex optimization problem. Then, the whale optimization algorithm is adopted to solve this optimization problem to achieve a balance between privacy protection and resource consumption. Experiments are conducted to verify the relationship between the degree of privacy leakage, the computation-offloading cost and real distance, privacy-preserving impact factor, the respective weights of time delay and energy consumption. The experimental results show that the offloading scheme proposed in this paper has good performance in terms of cost and privacy protection.

Keywords Edge computing · Computing offloading · Differential privacy · Whale optimization algorithm

1 Introduction

With the popularization of smart terminals and the Internet of Things, various terminal applications continue to emerge, which not only bring rich entertainment experience to users, but also consume more computing resources and energy consumption

✉ Jianhang Wei
wei@hbu.edu.cn

¹ School of Electronic Information Engineering, Hebei University, Baoding 071002, China

² Information Technology Center, Hebei University, Baoding 071002, China

³ Network and Experiment Management Center, Xinjiang University of Science and Technology, Korla 841000, China

of terminals [1]. The computing capability of the user terminal device is limited, and cannot complete the processing of a large number of computing tasks in a short period of time. Mobile edge computing (MEC) is a new computing model [2]. By deploying computing resources at the edge of the network and spatially adjacent to end users, MEC can reduce service delay and terminal energy consumption [3]. Computing offloading technology [4, 5], as one of the key technologies of MEC, transmits compute-intensive terminal applications to adjacent MEC nodes for processing through wireless links, and utilizes sufficient computing resources and energy of MEC servers to reduce the latency and energy consumption of terminal processing tasks, effectively improving service quality and user experience [6]. However, while edge computing provides users with high-quality and convenient computing services, users' personal privacy is also under great threat [7]. During task offloading, mobile devices tend to offload more tasks to the edge server when the wireless channel conditions between the user and the edge server are good, while they tend to perform more computing tasks locally when the wireless channel conditions are poor. The wireless channel condition between the user and the edge server is closely related to the distance between them. The smaller the distance between the user and the edge server, the better the wireless channel condition, and vice versa. Therefore, an untrusted edge server or an attacker can infer the wireless channel information by monitoring the user's task offloading ratio, thereby inferring the user's location information. Most previous studies have only considered the cost of task offloading [8], and there has not been much research on the location privacy issues that may occur in the process of computing offloading. This paper studies the location privacy leakage problem that may occur during computation offloading in edge computing environment, and propose a privacy-preserving edge computing offloading scheme based on whale optimization algorithm (WOPP), which uses the whale optimization algorithm to select the offloading strategy that minimizes the cost while using differential privacy technique to protect the user's real location information to achieve the balance between privacy protection and cost control.

The main contributions of this article are as follows:

1. Aiming at the problem of location privacy leakage that may occur in the process of computing offloading, a privacy protection mechanism is proposed, which uses differential privacy technology to obfuscate the user's real location;
2. The latency and energy consumption cost generated by the computing offloading process are modeled as a convex optimization problem, applying the whale optimization algorithm to solve this optimization problem, and an offloading strategy that minimizes the cost is obtained.
3. The effects of different parameters on the WOPP algorithm are obtained through experiments, and the effectiveness of the WOPP algorithm is verified by comparing it with other classical computing offloading schemes.

2 Related work

At this stage, research schemes on computing offloading are mainly carried out from three different optimization objectives: minimizing delay time, minimizing energy consumption, and balancing delay time and energy consumption [9–11]. Jiang et al. [12] studied the multi-user task offloading problem in the multi-server environment in edge computing, and transformed the task offloading problem of minimizing energy consumption into a constrained multi-dimensional multi-knapsack problem, and proposed a multi-pointer network (Multi-pointer network) to solve the problem. Wang et al. [13] developed an intelligent pricing mechanism to coordinate the computational offloading method of multi-layer devices to solve the problems of network congestion and node overload, each MEC server utilizes multi-agent reinforcement learning to determine its offloading strategy and resource allocation, thereby reducing the total energy consumption. Li et al. [14] proposed a task offloading algorithm based on double deep Q-learning net (DDQN) and a federated learning (FL) adaptive task offloading algorithm in MEC. The algorithm combines the QoS model and the deep reinforcement learning algorithm to obtain an optimal offloading policy according to the local link and node state information in the channel coherence time to address the problem of time-varying transmission channels and reduce the computing energy consumption and task processing delay. Lv et al. [15] explored the joint optimization problem of computing offloading and resource allocation for various IoT services in SD-MEC (software defined mobile edge computing) networks. In order to minimize the delay and power consumption of system utilities, a new distributed DL-based computation offloading and power resource allocation algorithm is proposed. However, the above literature mainly aims at reducing the cost of computational offloading and does not consider the privacy leakage that may occur during the computational offloading process. In recent years, people have realized the importance of privacy, and privacy-preserving mechanisms have been applied to various domains to secure users' privacy [16, 17], and some privacy-preserving mechanisms oriented to task offloading have been proposed. Zhao et al. [18] proposed a privacy-preserving computing offloading method based on k-anonymity, aiming at the problem that users' offloading tasks and offloading frequency in mobile edge computing (MEC) may cause users to be locked out by attackers. Considering that most existing blockchain mining service computing offloading frameworks ignore users' privacy, Nguyen et al. [19] proposed a MEC-based user privacy model for mobile network, in which mobile devices select an efficient offloading decision through a constrained Markov decision process. The literature [20] proposed a privacy and energy co-aware data aggregation computation offloading scheme and used a fog-assisted three-layer secure computing architecture to ensure data offloading security. The WOPP scheme proposed in this paper considers the relationship between user location information and user task offloading decisions, as well as the trade-off between privacy protection and cost control.

3 System model and problem definition

3.1 System scenario

This paper assumes that the system consists of a terminal device and a mobile edge server. The terminal device has m tasks waiting to be processed. The task set of the user terminal is modeled as $TASK = \{task_1, task_2, \dots, task_m\}$, and the i th task is modeled as $task_i = \{U_i, D_i, R_i, r_i^s\}$, where U_i represents the data volume of the i th task, D_i represents the data volume of the processing result of the i th task, and R_i represents the number of CPU cycles required by the CPU of the user terminal device or MECS to process each bit task, r_i^s represents the number of MECS resources required to process the i th task. Model the MECS as $mecs = (f_e, P^D, rs)$, where f_e is the processing capability of the mecs, which can be expressed by the number of CPU cycles per second, P^D denotes the transmitting power of the MECS, and rs denotes the number of computational resources owned by the MECS. The user terminal device is modeled as $User = (f_l^c, P_l, P^U)$, where f_l^c denotes the computing capability of the user terminal, which can be expressed by the number of CPU cycles per second, P_l denotes the computing power of the user terminal device, and P^U denotes the transmitting power of the user terminal device to send data to the MECS. The offloading decision of the i th task is represented by the offloading ratio λ_i .

The mobile edge computing model is shown in Fig. 1. The user terminal device can connect wirelessly with the MECS and offload all or part of the tasks to the MECS for processing, and the MECS helps the user terminal device to process and return the results to the user terminal device, which is shown in Fig. 2.

3.2 Latency model

3.2.1 Local computing latency

When part of the i th task is allocated to the user device for computation, the local execution time T_l^i of the task can be calculated as in Eq. (1)

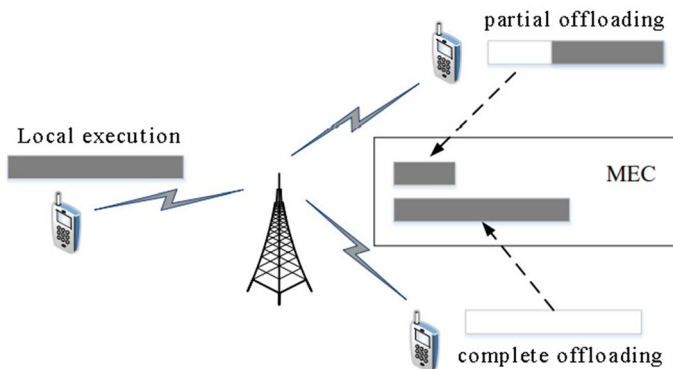


Fig.1 Schematic diagram of computational offloading

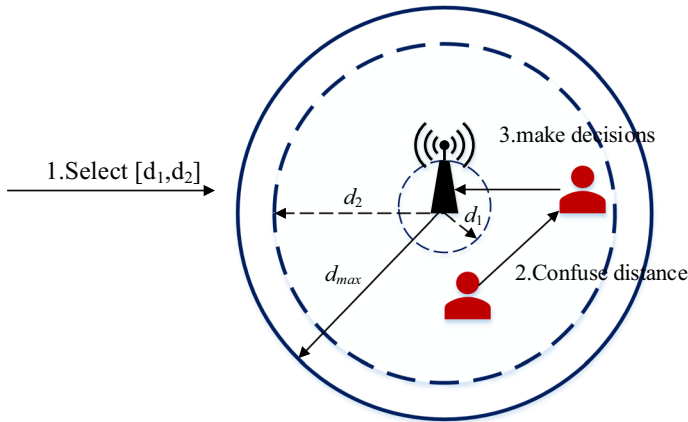


Fig. 2 Privacy protection mechanism process

$$T_i^i = \frac{(1 - \lambda_i)U_i * R_i}{f_i^c} \tag{1}$$

3.2.2 External computing latency

When part of the task is offloaded to MECS for execution, the delay includes transmission delay T_u^i , MECS processing delay T_c^i and result return delay T_b^i . Since the result return delay is usually small, it is not considered. According to Shannon’s formula, the transmission rate V_u at which the user terminal equipment sends the task to the MECS and the data transmission rate V_d at which the MECS sends the result back to the user terminal equipment are shown in Eqs. (2) and (3)

$$V_u = B * \log_2(1 + \frac{P^U * d^{-r}}{\sigma^2}) \tag{2}$$

$$V_d = B * \log_2(1 + \frac{P^D * d^{-r}}{\sigma^2}) \tag{3}$$

where B denotes the bandwidth between the user terminal device and the MECS, d^{-r} denotes the channel coefficient between the user terminal device and the MECS, d denotes the distance between the user terminal device and the MECS, r denotes the fading factor of the channel, and σ^2 denotes the noise power of the channel.

$$T_u^i = \frac{\lambda_i U_i}{V_u} \tag{4}$$

$$T_c^i = \frac{D_i * R_i}{f_e} \quad (5)$$

Thus, the total delay T_e^i generated by the user terminal device offloading the i th task to the MECS for processing is Eq. (6).

$$T_e^i = T_u^i + T_c^i + T_b^i \quad (6)$$

3.2.3 Total delay

Since local execution and external execution are parallel, the total latency should be the greater of the two.

$$T^i = \max\{T_l^i, T_e^i\} \quad (7)$$

3.3 Energy consumption model

3.3.1 Local computing energy consumption

When a task is processed at the user terminal device, the user terminal device CPU generates energy consumption. Assume that the energy consumption of the i -th task of the user terminal equipment processed at the user terminal equipment is E_l^i , as shown in Eq. (8)

$$E_l^i = T_l^i * P_l \quad (8)$$

3.3.2 External computing energy consumption

The energy consumption generated when user terminal device offloads the task to the MECS for processing includes the transmission energy consumption generated by the local user terminal device when the user terminal sends tasks to the MECS, the energy consumption generated by the MECS when it processes the task from the user terminal device, and the energy consumption generated by the MECS when it returns computation results to the user terminal device. Since the MECS is always powered, this paper mainly considers the energy consumption generated by the local user terminal device uploading data to the MECS. Assume that the energy consumption of the user terminal device uploading the i th task is E_u^i , as in Eq. (9).

$$E_u^i = P^U * T_u^i \quad (9)$$

3.3.3 Total energy consumption

$$E^i = E_l^i + E_u^i \tag{10}$$

3.4 Problem model

Assume that for the i th task, the total cost of the system to process the user terminal device task is C^i as in Eq. (11).

$$C^i = \alpha T^i + \beta E^i \tag{11}$$

In edge computing scenarios, latency and energy consumption are the two most commonly used metrics to measure the performance of offloading schemes [21]. Considering the cost required by the task during execution comprehensively, α is defined as the weight coefficient of task execution delay, which indicates the user’s concern about delay; β is the weight coefficient of task execution energy consumption, which indicates the user’s concern about energy consumption, satisfying $\alpha + \beta = 1$. In this paper, the delay and energy consumption are considered comprehensively, and the goal is to minimize the weighted sum of system delay and energy consumption, which is defined as the problem Q.

The problem Q: $\min_{k=1}^N C(\lambda_i)$.

4 Privacy protection mechanism

4.1 Mechanism description

In order to protect the user’s location privacy, a new distance confusion probability density function is used, which can be used by the user to confuse the real distance between it and the edge server to avoid the leakage of the user’s location information. Suppose the real distance between the user and the edge server is d and the obfuscated distance is d^* , let d_{\max} represent the maximum coverage radius of the edge server, d_1 and d_2 represent the upper and lower bounds of the range of the user’s obfuscated distance, respectively, $\Delta d = d_2 - d_1$, with $d_1 < d_2$ and $d_1, d_2 \in [0, d_{\max}]$. With the confusion range, the differential privacy mechanism is applied to confuse the distance between the user and the edge server. Existing differential privacy schemes based on Laplace mechanism are difficult to be directly applied to the scenario where task offloading protects user’s location privacy. Therefore, considering the limitation of the confusion range $[d_1, d_2]$ under the condition that the total probability of confusing the distance in the range $[d_1, d_2]$ is guaranteed to be 1, the probability density function of confusing the distance d into the distance d^* is set as follows:

$$f(d^*) = \begin{cases} \frac{e^{-\frac{e|d^*-d|}{\Delta d}}}{2\Delta d} + \frac{e^{-\frac{e(d_1-d)}{\Delta d}} + e^{-\frac{e(d_2-d)}{\Delta d}}}{2\Delta d}, & \text{if } d^* \in [d_1, d_2] \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

Accordingly, the user can obfuscate the real distance between himself and the edge server, and the untrustworthy edge server or the attacker can only infer the user's obfuscated location by the user's task offloading ratio at the same time, and at this time, due to the randomness provided by differential privacy for adjacent data sets, it is difficult for an attacker to deduce the user's real location in reverse from the obfuscated location, thereby protecting the user's location information.

The *KL* divergence (Kullback–Leibler divergence) [22] is used to measure the degree of fit between the mechanism with privacy protection and the mechanism without privacy protection when the real distance between the user and the edge server is d . Suppose $Q(d^*|d)$ represents the probability distribution of task offloading by the user according to the real distance without privacy protection, $P(d^*|d)$ represents the probability distribution of user obfuscation according to the true distance when a differential privacy mechanism is added to protect privacy, then the degree of fitting $Q(d^*|d)$ with $P(d^*|d)$ is:

$$D_{\text{KL}}(P||Q) = \int_{d_1}^{d_2} Q(d^*|d) \log \frac{Q(d^*|d)}{P(d^*|d)} dd^* \quad (13)$$

According to the definition of *KL* divergence, if the value of $D_{\text{KL}}(P||Q)$ is smaller, it indicates that the higher the degree of fit between $P(d^*|d)$ and $Q(d^*|d)$, the higher the probability of leakage of information about the true distance between the user and the edge server, and the worse the degree of privacy protection. Therefore, when the real distance between the user and the edge server is d , the degree of privacy leakage of the user is the inverse of the above formula:

$$\text{PL}_{d_1, d_2} = - \int_{d_1}^{d_2} Q(d^*|d) \log \frac{Q(d^*|d)}{P(d^*|d)} dd^* \quad (14)$$

In general, when the wireless network between the user and the edge server is in good condition (closer), the confused distance should be set to be relatively close to the edge server in a high probability, so that after the task offloading decision is made based on the confusion distance, it is guaranteed that the user will offload more tasks to the edge server to save the user's resource consumption at the real location. In order to ensure the effectiveness of the task offloading decision made by the user based on the confusion distance d^* on the real distance d , it must be ensured that the confused distance d^* takes values on the left and right sides close to the real distance d , then there is a confusion range $d_1 \leq d \leq d_2$, and the closer d_1 , d_2 and d are, the better the decision-making utility of task offloading at the confusion distance. However, when the range Δd between d_1 and d_2 shrinks, although the task offloading decision based on the confusion distance can optimize the user's energy consumption and computational delay at the real distance, the degree of privacy leakage of the user's location information becomes larger. Therefore, before task offloading according to the confused location, the lower and upper bound values of the confusion range should be adjusted to achieve a balance between privacy protection and task offloading utility based on the user's privacy protection needs and the wireless channel between the user and the edge server. The formula for adjusting the

lower bound value d_1 and the upper bound value d_2 of the confusion range to balance the utility of privacy protection and task offloading is:

$$\min_{d_1, d_2} E + T + \omega \cdot PL \tag{15}$$

where E is the user’s energy consumption, T is the user’s computational delay cost, and ω as an impact factor reflects the user’s attention to the degree of privacy leakage. In the most extreme case, users do not pay any attention to the leakage of their location privacy, at this time, $\omega = 0$.

When the user’s confusion range is determined, a confused distance d^* can be randomly selected from $[d_1, d_2]$ according to the distance confusion probability density function, and a task offloading decision can be made based on the confusion distance to complete the task offloading.

4.2 Theoretical analysis

For the real distance d between the user and the edge server and its proximity distance d' , it can be proved that after applying the privacy protection mechanism proposed in this paper, the probability $Pr(d^*|d)$ of confusion from the real distance d to d^* and the probability $Pr(d^*|d')$ of confusion from the proximity distance d' to d^* satisfies the definition of ϵ -differential privacy.

$$\begin{aligned} \frac{Pr(d^*|d)}{Pr(d^*|d')} &= \frac{\frac{\epsilon}{2\Delta d} e^{-\frac{\epsilon|d^*-d|}{\Delta d}} + \frac{e^{\frac{\epsilon(d_1-d)}{\Delta d}} + e^{-\frac{\epsilon(d_2-d)}{\Delta d}}}{2\Delta d}}{\frac{\epsilon}{2\Delta d} e^{-\frac{\epsilon|d^*-d'|}{\Delta d}} + \frac{e^{\frac{\epsilon(d_1-d')}{\Delta d}} + e^{-\frac{\epsilon(d_2-d')}{\Delta d}}}{2\Delta d}} \\ &= \frac{e^{-\frac{\epsilon|d^*-d|}{\Delta d}} + \frac{e^{\frac{\epsilon(d_1-d)}{\Delta d}} + e^{-\frac{\epsilon(d_2-d)}{\Delta d}}}{\epsilon}}{e^{-\frac{\epsilon|d^*-d'|}{\Delta d}} + \frac{e^{\frac{\epsilon(d_1-d')}{\Delta d}} + e^{-\frac{\epsilon(d_2-d')}{\Delta d}}}{\epsilon}} \\ &\leq \max \left(\frac{e^{-\frac{\epsilon|d^*-d|}{\Delta d}}}{e^{-\frac{\epsilon|d^*-d'|}{\Delta d}}}, \frac{e^{\frac{\epsilon(d_1-d)}{\Delta d}} + e^{-\frac{\epsilon(d_2-d)}{\Delta d}}}{e^{\frac{\epsilon(d_1-d')}{\Delta d}} + e^{-\frac{\epsilon(d_2-d')}{\Delta d}}} \right) \\ &\leq \max \left(e^{\frac{\epsilon|d^*-d'|}{\Delta d}}, \max \left(\frac{e^{\frac{\epsilon(d_1-d)}{\Delta d}}}{e^{\frac{\epsilon(d_1-d')}{\Delta d}}}, \frac{e^{-\frac{\epsilon(d_2-d)}{\Delta d}}}{e^{-\frac{\epsilon(d_2-d')}{\Delta d}}} \right) \right) \\ &\leq \max \left(e^\epsilon, \max(e^\epsilon, e^\epsilon) \right) \end{aligned}$$

$$= e^{\epsilon}$$

The above proofs show that the privacy protection mechanism proposed in this paper satisfies the definition of ϵ -differential privacy.

5 Task offloading mechanism

For the selected confusion interval $[d_1, d_2]$, the confusion position d^* is generated based on the real distance d , and the whale optimization algorithm is used to make a task offloading decision with the objective of minimizing the resource consumption under the confusion location: the unloading ratio λ_r .

The whale algorithm simulates a special hunting mechanism of humpback whales—a bubble net foraging method which can be simply expressed as follows: the whale dives around the prey at a water depth of 10~15 m, swims in a spiral posture around the prey in a gradually contracting range toward the surface, and protrudes bubbles of different sizes while swimming, and the exhaled bubbles form a circular or square bubble net. The bubbles form a ring or a square bubble net, and then the prey will be attacked by the whale. In this paper, the optimal prey is equivalent to the optimal unloading strategy, and the process of searching for the prey around the whale is equivalent to the process of optimizing finding the optimal unloading decision vector.

The three stages of whale predation are mathematically modeled below: (1) random search for prey; (2) surround prey; (3) bubble attack on prey.

5.1 Prey search (global search)

The random search for prey in the initial position corresponds to the global search stage of the whale algorithm. At this time, the coefficient vector $|A| > 1$, the mathematical model of this stage is as follows:

$$D = |C \cdot X_{\text{rand}} - X(t)| \quad (16)$$

$$X(t + 1) = X_{\text{rand}} - A \cdot D \quad (17)$$

where $X(t)$ represents the position vector of the current whale, X_{rand} represents the position of a random whale in the current whale population, A and C represent the coefficient vector, and the calculation formula of the coefficient vector is as follows:

$$A = 2a \cdot r - a \quad (18)$$

$$C = 2r \quad (19)$$

$$a = 2 - 2 \cdot \frac{t}{T_{\text{max}}} \quad (20)$$

where a linearly decreases from 2 to 0 during the iteration; r is a random vector of $[0, 1]$; t denotes the current number of iterations, and T_{\max} represents the maximum number of iterations.

5.2 Encircling prey

In the prey-encircling stage, each whale represents an independent individual in the algorithm, the position of each individual in the search space represents a solution of the optimization process, the optimal prey position is unknown in the search space, and the optimal candidate solution of the WOA (Whale Optimization Algorithm) is location of the optimal whale (prey). After the optimal solution is established, other search agents gradually approach the optimal whale search agent (prey). the way the whale circles its prey is represented by the following mathematical model:

$$D = |C \cdot X^*(t) - X(t)| \tag{21}$$

$$X(t + 1) = X^*(t) - A \cdot D \tag{22}$$

where t represents the current number of iterations, $X(t)$ represents the current position vector of the whale, $X^*(t)$ represents the position of the best whale obtained so far, and $X(t + 1)$ represents the position vector of the target prey.

5.3 Bubble-net attacking method (local search)

The bubble predation method simulates the local search process of whales, when the coefficient vector $|A| < 1$ in the mathematical model and mathematically models the whale bubble predation. Two strategies for updating the position are designed as follows:

- (1) Shrinking encircling mechanism

The update of the whale’s position is represented by Eq. (19). When $|A| < 1$, the searcher’s next position may exist at any position between the current position and the prey.

- (2) Spiral predation mechanism

The whale searches through a spiral motion, and the formula for position update is as follows:

$$D' = |X^*(t) - X(t)| \tag{23}$$

$$X(t + 1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \tag{24}$$

where indicates the distance of the current whale to the prey (best solution obtained so far), b is a constant for defining the shape of the logarithmic spiral, l is a random number in $[-1, 1]$.

Both mechanisms are carried out simultaneously during bubble-net attacking period with a probability of 0.5, calculated as follows:

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D, & p < 0.5 \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t), & p \geq 0.5 \end{cases} \quad (25)$$

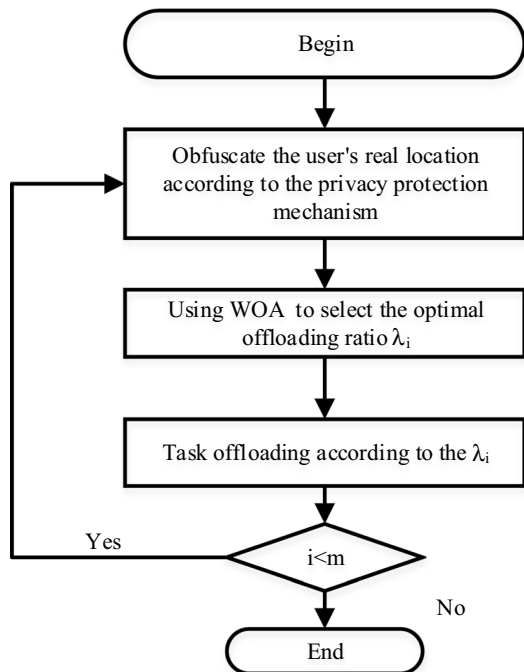
where p is a random number between $[0, 1]$.

5.4 Fitness function

In the whale algorithm, the objective function of the problem is usually used as a fitness function to evaluate the pros and cons of each solution. For a certain unloading decision λ_i (candidate solution of the problem), assuming its fitness evaluation function $f(\lambda_i)$. The formula is as follows:

$$f(\lambda_i) = C(\lambda_i) \quad (26)$$

Fig. 3 Process flow diagram of WOOP



5.5 Algorithm process

Figure 3 depicts the basic flow of the privacy-preserving computing offloading scheme based on whale optimization algorithm (WOPP). The whole scheme is divided into two stages. The first stage is to obfuscate the user's real location to protect the privacy of the location; the second stage is to use WOA to select the offload ratio, and the user can offload tasks according to the optimal offload ratio λ_i .

In order to depict the entire implementation of the algorithm in more detail, the pseudo-code of the algorithm is given here.

Algorithm *Privacy protection offloading scheme based on WOA*

1. **Procedure1** Confuse distance
 2. *input: d_{max} , d , ω , $[d_1, d_2]$*
 3. *output: \mathbf{d}^**
 4. Update d by Eq.(12)
 5. return \mathbf{d}^*
 6. **Procedure2** WOA
 7. Initialize the whales population X_i
 8. Calculate the fitness of each search agent
 9. X^* =the best search agent
 10. **while**(t <maximum number of iterations)
 11. **for** each search agent
 12. Update a , A , C , l , and p
 13. **if1**($p < 0.5$)
 14. **if2**($|A| < 1$)
 15. Update the position of the current search agent by the Eq.(22)
 16. **else if2**($|A| > 1$)
 17. Select a random search agent (X_{rand})
 18. Update the position of the current search agent by the Eq.(17)
 19. **end if2**
 20. **else if1**($p \geq 0.5$)
 21. Update the position of the current search by the Eq.(24)
 22. **end if1**
 23. **end for**
 24. Check if any search agent goes beyond the search space and amend it
 25. Calculate the fitness of each search agent by Eq.(11)
 26. Update X^* if there is a better solution
 27. $t=t+1$
 28. **end while**
 29. return X^*
-

6 Experimental simulation and result analysis

The experimental environment is Windows10 operating system, Intel(R) Core(TM) i5-1135G7, CPU (2.40 GHz), 16 GB RAM. In order to verify the effectiveness of the WOPP algorithm for computational offloading, experiments were conducted using MATLAB. We utilized the dataset for Shanghai Telecom's base stations [23–25] to simulate the scenario and perform data offloading. The dataset covers 3233 SBSs and 9481 subscribers. Each entry in the dataset consists of a user identifier, traffic flow, session time, location zone number, and sector ID. In addition, each SBS has its own location zone number and sector ID, so we can obtain its geographic location using the Google Map API. The experiments verify the impact of three important indicators, the weight α , the real distance d between the user and the edge server, and the user privacy leakage impact factor ω on the effect of the WOPP algorithm. The effectiveness of the WOPP algorithm is verified by comparing it with PSO (Particle Swarm Optimization) [26], GA (Genetic Algorithm) [27], WOA [28]. The main parameter settings for the simulation experiments are shown in Table 1.

6.1 The relationship between d and the cost and privacy leakage

The performance of the WOPP algorithm regarding resource consumption and privacy protection for different real distances d between users and edge servers is shown in Fig. 3, where the real distance d between the user and the edge server is randomly taken within [50, 250]m, the privacy leakage impact factor ω of the user is 0.002, and the privacy budget ϵ of the user regarding differential privacy is 0.1.

Figure 4a shows that in order to better balance the relationship between user resource overhead and privacy leakage, when the distance between the user and the edge server increases, the lower bound d_1 and the upper bound d_2 of the confusion interval become larger centered on the true distance d . The Δd of the confusion interval $[d_1, d_2]$ does not change significantly. When the distance increases, the wireless channel condition between the user and the edge server will gradually become worse, and then the user will perform the task more locally, and the resource cost of the user will gradually increase, and when the distance increases, the degree of privacy leakage increases with the distance after the user selects a suitable confusion interval. It is namely that the degree of privacy leakage does not change significantly with the increase in distance when the size of the confusion interval does not change significantly, as shown in Fig. 4b.

6.2 The relationship between ω and the cost and the degree of privacy leakage

The performance of the WOPP algorithm regarding resource consumption and privacy leakage with different privacy leakage impact factors ω of users is shown in Fig. 5, where the real distance $d = 150m$ between the user and the edge server,

Table 1 Parameter setting

Parameter	Value
B	Random integer in 10~50 MHz
f_e	Random integer in 50~60 GHz
f_l^c	7 GHz
σ^2	10^{-9} dBm
P^D	Random integer 30~80dBm
N	50
$Maxgen$	Maxgen=500

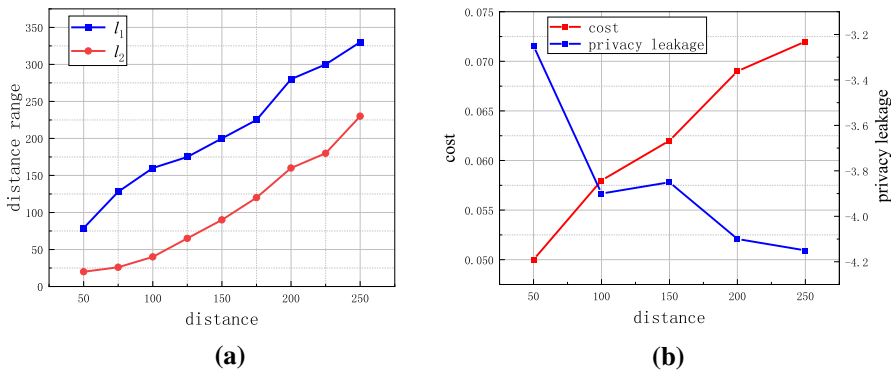


Fig. 4 The relationship between d and the cost and privacy leakage

the privacy leakage impact factor ω of the user is one of $\{0.001,0.002,0.003,0.004,0.005\}$, and the privacy budget ϵ of the user is 0.1.

Figure 5a shows that the Δd of the confusion intervals $[d_1, d_2]$ is getting larger when the user’s privacy leakage impact factor ω increases, this is because, when the user’s privacy leakage impact factor ω increases, the user is more concerned about the leakage of location privacy. Therefore, the privacy leakage is smaller only when the confusion interval is larger. Figure 5b shows that the increasing Δd leads to a progressively larger resource overhead for the user, which is because the utility of the user’s task offloading decision based on the confusion distance becomes lower with respect to the true distance when the confusion interval becomes larger. In turn, the larger the confusion interval, the progressively smaller the user’s privacy leakage.

6.3 Influence of weight α on WOPP algorithm

It can be seen from Fig. 6 that in the same experimental environment, when the weight α increases, WOPP is more biased to optimize the latency of user terminal task offloading, and the system produces smaller latency at this time, however, this will weaken the optimization of system energy consumption, so the system energy

consumption will increase while reducing the system latency. In addition, in Fig. 6a, we can see that the impact of the weight change on the system latency is not significant when the task volume of the user terminal device is small, because when the task volume is small, the resources of MECS are sufficient, and the tasks are offloaded to MECS for execution at this time, so the resulting latency is similar, but from Fig. 6b, we can see that the impact of the weight change on the system energy consumption at small task sizes is larger than that on the delay, because the channel conditions are different when users send data to each MECS, so the impact of weight on energy consumption is relatively obvious.

6.4 The relationship between task volume and delay and energy consumption

Comparing WOPP with PSO, GA, and WOA, under these four schemes, the delay and energy consumption of the system processing user terminal tasks vary with the number of tasks as shown in Fig. 7a, b.

From Fig. 7a, b, it can be seen that the latency and energy consumption generated by the system processing user terminal device tasks increase continuously with the increase in task volume, which is because the workload requires more time for transmission and computation, and the latency and energy consumption of WOA increase less than PSO and GA when the task volume increases, which is because WOA searches the solution space more adequately and can search for the global optimum in the continuous iterative process in order to make the system generate less latency and energy consumption when processing the tasks of all user devices. Considering the cost of privacy protection, the latency and energy consumption of WOPP are higher than those of WOA, but lower than those of PSO and GA.

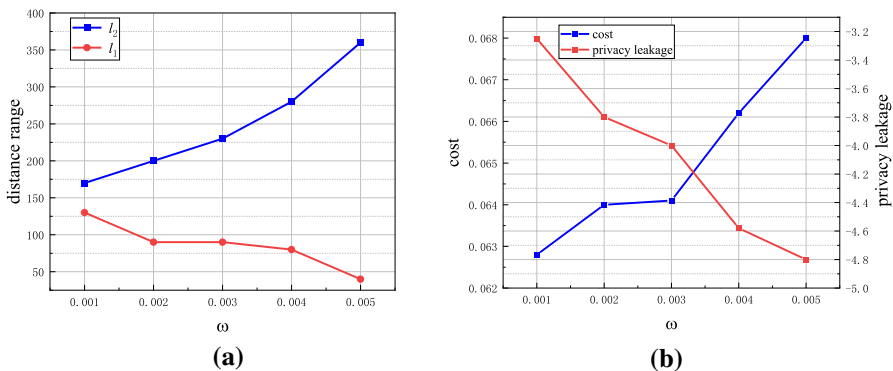


Fig. 5 The relationship between ω and the cost and privacy leakage

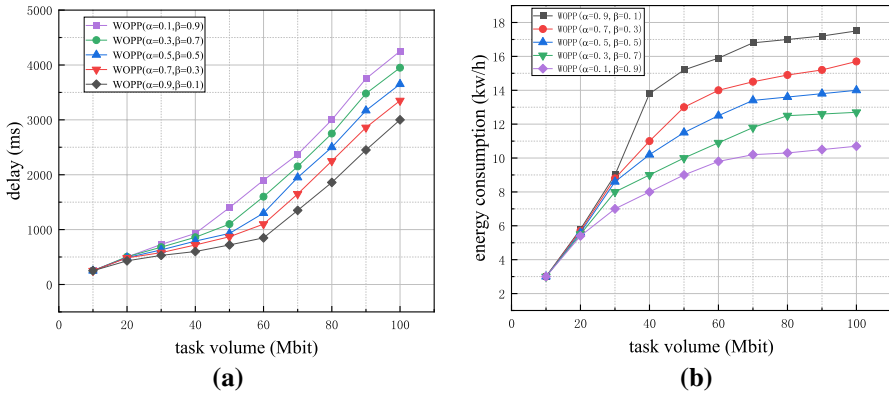


Fig. 6 Effect of weight on system delay and energy consumption

7 Summary

During the offloading process of mobile edge computing, since there is a certain relationship between the user’s offloading decision and the real distance between the user and the edge server, an untrusted edge server or an attacker can infer the wireless channel information by monitoring the user’s task offloading ratio, thereby inferring the user’s location information, resulting in location privacy leakage. Aiming at this problem, a privacy-preserving edge computing offloading scheme WOPP based on whale optimization algorithm is proposed. This scheme combines the privacy-preserving mechanism and the offloading ratio optimization selection mechanism, first using differential privacy to obfuscate the user’s distance, and then using an optimization algorithm to select the best offloading ratio based on the obfuscated location, which protects the user’s location privacy while controlling the offloading cost. It is experimentally verified that the degree of privacy leakage and the

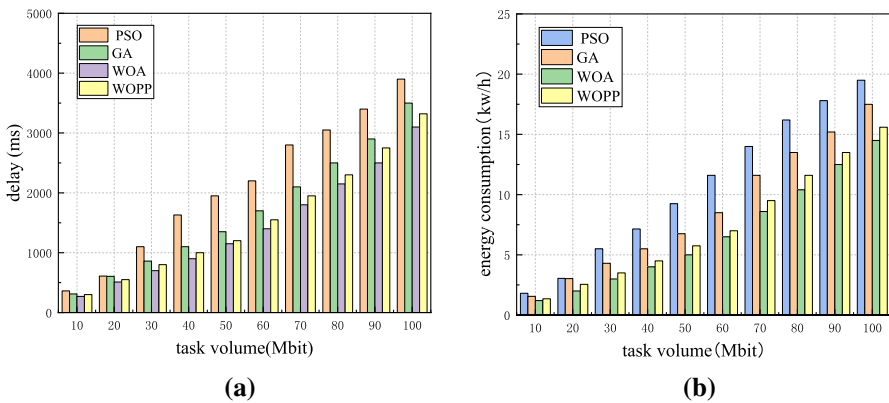


Fig. 7 Effect of increasing task volume on total delay and energy consumption

computation-offloading cost are closely related to real distance, privacy-preserving impact factor, the respective weights of time delay and energy consumption, which are compared by PSO, GA, WOA, etc. It is shown that the WOPP algorithm is highly usable in controlling the offloading cost.

Funding Natural Science Foundation of Hebei Province, No. F2019201427, Zhenpeng Liu, Integration of Cloud Computing and Big Data, Innovation of Science and Education of China, 2017A20004, Zhenpeng Liu.

Data availability The experimental data used to support the findings of this study are available from the corresponding author upon request.

Declarations

Conflict of interest This research was supported by the Natural Science Foundation of Hebei Province, China under Grant No. F2019201427 and Fund for Integration of Cloud Computing and Big Data, Innovation of Science and Education of China under Grant No. 2017A20004.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Dixit S, Kathavate S, Gautham SK (2022) An overview on mobile edge cloud system. *IOT Smart Syst* 251:719–727
2. Shi W, Zhang X, Wang Y et al (2019) Edge computing: state-of-the-art and future directions. *J Comp Res Develop* 56(1):69
3. Huda SMA, Moh S (2022) Survey on computation offloading in UAV-Enabled mobile edge computing. *J Netw Comput Appl* 201(5):103341
4. Islam A, Debnath A, Ghose M et al (2021) A survey on task offloading in multi-access edge computing. *J Syst Architect* 118:102225
5. Zheng H, Yu S, Cui X et al (2021) Survey on computing offloading in edge computing. *Comput Syst Appl* 30(12):9
6. Hosseinzadeh M, Wachal A, Khamfroush H, et al. (2022) QoS-aware priority-based task offloading for deep learning services at the edge. In *IEEE 19th Annual Consumer Communications and Networking Conference (CCNC2022)*, pp 319–325
7. Elgendy IA, Yadav R (2022) Survey on mobile edge-cloud computing: a taxonomy on computation offloading approaches. In *Security and Privacy Preserving for IoT and 5G Networks*, pp 117–158
8. Ni J, Lin X, Shen XS (2019) Toward edge-assisted Internet of Things: from security and efficiency perspectives. *IEEE Network* 33(2):50–57
9. Xie R, Lian X, Jia Q et al (2018) Survey on computation offloading in mobile edge computing. *J Commun* 39(11):138
10. Mach P, Becvar Z (2017) Mobile edge computing: A survey on architecture and computation offloading. *IEEE Commun Surv Tutor* 19(3):1628–1656

11. Wang F, Xu J, Wang X et al (2017) Joint offloading and computing optimization in wireless powered mobile-edge computing systems. *IEEE Trans Wirel Commun* 17(3):1784–1797
12. Jiang Q, Zhang Y, Yan J (2020) Neural combinatorial optimization for energy-efficient offloading in mobile edge computing. *IEEE Access* 8(2):35077–35089
13. Wang P, Di B, Song L et al (2022) Multi-layer computation offloading in distributed heterogeneous mobile edge computing networks. *IEEE Trans Cognitive Commun Netw* 8(2):35077–35089
14. Li J, Yang Z, Wang X et al (2022) Task offloading mechanism based on federated reinforcement learning in mobile edge computing. *Digital Commun Netw*. <https://doi.org/10.1016/j.dcan.2022.04.006>
15. Wang Z, Lv T, Chang Z (2022) Computation offloading and resource allocation based on distributed deep learning and software defined mobile edge computing[J]. *Comput Netw* 205(3):108732
16. Bingur R, Jothilakshmi S, Srinivasu N (2022) A comprehensive review on security and privacy preservation in cloud environment. *Sustain Commun Netw Appl* 93(1):719–738
17. Mu R, Gong B, Ning Z et al (2022) An identity privacy scheme for blockchain-based on edge computing. *Concurr Comput Pract Exp* 34(1):e6545
18. Zhao X, Peng J, You W et al (2021) A privacy-preserving computation offloading method based on k-anonymity. *J Electron Inf Technol* 43(4):892–899
19. Zhao P, Tao J, Kangjie L et al (2022) Deep reinforcement learning-based joint optimization of delay and privacy in multiple-user MEC systems. *IEEE Trans Cloud Comput* 1:1
20. Nguyen DC, Pathirana PN, Ding M et al (2020) Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning. *IEEE Trans Netw Serv Manage* 17(4):2536–2549
21. Chen S, You Z, Ruan X (2020) Privacy and energy co-aware data aggregation computation offloading for fog-assisted IoT networks. *IEEE Access* 8(4):72424–72434
22. Erven TV, Harremoës P (2014) Divergence and Kullback-Leibler divergence. *IEEE Trans Inf Theory* 60(7):3797–3820
23. Li Y, Zhou A, Ma X, Wang S (2022) Profit-aware edge server placement. *IEEE Internet Things J* 9(1):55–67
24. Guo Y, Wang S, Zhou A, Xu J, Yuan J, Hsu C (2020) User allocation-aware edge cloud placement in mobile edge computing. *Softw Pract Exp* 50(5):489–502
25. Wang S, Guo Y, Zhang N, Yang P, Zhou A, Shen X (2021) Delay-aware microservice coordination in mobile edge computing: a reinforcement learning approach. *IEEE Trans Mob Comput* 20(3):939–953
26. Marini F, Walczak B (2015) Particle swarm optimization (PSO). A tutorial. *Chemometr Intell Lab Syst* 149(partB):153–165.
27. Mirjalili S (2019) Genetic algorithm. *Evol Algorithms Neural Netw* 780:43–55
28. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95(5):51–67

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.