

Secure and Multi-Step Computation Offloading and Resource Allocation in Ultra-Dense Multi-Task NOMA-Enabled IoT Networks

Tianqing Zhou, Yanyan Fu, Dong Qin, Xuefang Nie, Nan Jiang, and Chunguo Li

arXiv:2303.06353v1 [cs.IT] 11 Mar 2023

Abstract—Ultra-dense networks are widely regarded as a promising solution to explosively growing applications of Internet-of-Things (IoT) mobile devices (IMDs). However, complicated and severe interferences need to be tackled properly in such networks. To this end, both orthogonal multiple access (OMA) and non-orthogonal multiple access (NOMA) are utilized at first. Then, in order to attain a goal of green and secure computation offloading, under the proportional allocation of computational resources and the constraints of latency and security cost, joint device association, channel selection, security service assignment, power control and computation offloading are done for minimizing the overall energy consumed by all IMDs. It is noteworthy that multi-step computation offloading is concentrated to balance the network loads and utilize computing resources fully. Since the finally formulated problem is in a nonlinear mixed-integer form, it may be very difficult to find its closed-form solution. To solve it, an improved whale optimization algorithm (IWOA) is designed. As for this algorithm, the convergence, computational complexity and parallel implementation are analyzed in detail. Simulation results show that the designed algorithm may achieve lower energy consumption than other existing algorithms under the constraints of latency and security cost.

Index Terms—ultra-dense networks, secure computation offloading, user association, channel selection, power control, PSO, WOA, IoT.

I. INTRODUCTION

With the staggering development of the mobile Internet of Things (IoT), a great many of new delay-sensitive and computing-intensive applications emerge, such as smart homes, virtual reality, augmented reality, autonomous driving, etc. [1]–[3]. Although the computing power of IoT mobile devices (IMDs) has achieved a qualitative leap, due to the limited computing resources and battery capacity, they cannot support these applications well [4]–[7]. To address such an

issue, mobile edge computing (MEC) is widely regarded as a promising option, which provides a large number of computing resources for IMDs (users) at the edge of networks. In MEC networks, any task of IMDs (users) can be partially or completely offloaded to some neighboring edge servers for computing. Evidently, through such an operation, the workloads and energy consumption of users may be reduced greatly.

In order to further shorten the distance between users and computing centers, ultra-dense networks are widely advocated and have attracted increasing attention, where base stations (BSs) are equipped with MEC servers [8]. Through the deployment of ultra-dense BSs, the service coverage can be enhanced greatly, and the uplink transmission power of users may be reduced significantly. However, such a deployment often results in complicated and severe network interferences. In addition, during the computation offloading, offloaded tasks are vulnerable to malicious attacks. To attain the goal of secure communications, some additional computation overheads yield for secure preventive services, resulting in extra computation latency and energy consumption.

It is evident that the design of secure and green offloading mechanisms is an important topic in ultra-dense networks. Specifically, under the limited network resources, the central issue remains how to protect users' data, mitigate network interferences and reduce users' energy consumption in such networks.

A. Related Work

In wireless networks, although spectrum sharing is beneficial to improving spectrum utilization, it will inevitably incur severe interferences within and between regions. It means that a reasonable resource management strategy needs to be introduced, especially for ultra-dense networks. To this end, some relevant efforts have been made as follows. In [9], joint spectrum, power, computation offloading decisions and resource allocation were optimized to minimize the energy consumed by users in densely deployed small cell networks. Such work considered distinct channels for macro BSs (MBS) and small BSs (SBSs), but let users utilize the same channels at some BS. In [10], offloading decisions, transmission duration and computing rate were jointly optimized to minimize the overall delay of tasks under both non-orthogonal multiple access (NOMA) and orthogonal frequency division multiple access (OFDMA). In [11], Li *et al.* jointly optimized

This work was supported by National Natural Science Foundation of China under Grant Nos. 62261020, 61861017, 62062034, 62001201, 62171119, 61861018, 61961020, 61862025 and 61963017, National Key Research and Development Program of China under Grant No. 2020YFB1807201, Natural Science Foundation of Jiangxi Province of China under Grant Nos. 20212BAB202004, 20212BAB202004 and 20212BAB212001, Key Research and Development Plan of Jiangsu Province Grant No. BE2021013-3, Special 03 Project and 5G Project of Jiangxi Province under Grant No. 20203ABC03W07. The corresponding author is Chunguo Li.

T. Zhou, Y. Fu, X. Nie and N. Jiang are with the School of Information Engineering, East China Jiaotong University, Nanchang 330013, China (email: zhoutian930@163.com; fuyanyan3640@163.com; Xuefang-nie@163.com; jiangnan1018@acm.org).

D. Qin is with School of Information Engineering, Nanchang University, Nanchang 330031, China (e-mail: qindong@seu.edu.cn).

C. Li is with School of Information Science and Engineering, Southeast University, Nanjing 210096, China (email: chunguoli@seu.edu.cn).

uplink transmission power, offloading decisions and weight coefficients of delay and energy consumption to minimize energy consumption for ultra-dense networks with NOMA and time division multiple access (TDMA). In [12], Lu *et al.* jointly optimized task offloading, BS selection, channel and computing resource allocation to minimize the total system cost consisting of delay and energy consumption caused by users and BSs for ultra-dense networks with OFDMA.

At the same time, multi-task offloading has attracted more and more attention in recent years. Some related work has been done as follows. In [13], joint task offloading decisions and bandwidth allocation were considered to minimize total system cost defined as the weighted sum of energy consumption and task delay. In such work, each user has multiple independent tasks. In [14], joint multi-task offloading decisions, computing and spectral resource allocation were optimized to minimize task latency while guaranteeing the energy available to the users. Such an investigation was made under the user-assisted MEC system. In [15], offloaded workloads and local computing rates were jointly optimized to minimize the weighted sum of energy consumed by NOMA transmission and local execution of smart terminals for a multi-task NOMA system. In [16], the amount of offloaded data was optimized to minimize task delay for a multi-server and multi-task scenario. In [17], the computation offloading was performed to minimize the average energy-time cost of all users for a MEC system with multiple dependent tasks. In [18], joint resource allocation and partial computation offloading were performed to minimize system energy consumption for heterogeneous edge networks with multiple separable tasks. In [19], joint multi-task offloading and resource allocation were executed to minimize the weighted sum of delay and energy consumption under task-overflowed situations.

It is easy to find that aforementioned one-step computation offloading cannot utilize computing resources well, and computation delay may increase with the number of tasks significantly. To fully utilize these resources in networks, especially in ultra-dense networks, multi-step computation offloading has been regarded as a good option. So far, multi-step computation offloading was rarely studied and is still an open topic. Some existing efforts made towards it can be listed as follows. In [20], joint user association, multi-step offloading decision, power and computation resources were optimized to minimize network-wide energy consumption for ultra-dense multi-task networks under users' latency constraints. In [21], joint device association, multi-step computation offloading and resource allocation were performed to minimize the network-wide energy consumption for ultra-dense multi-task networks under OFDMA and proportional computing resource allocation. In [22], joint device association, channel allocation and multi-part collaborative offloading were optimized to minimize the average delay under the affordable cost of network operators.

Although computation offloading can reduce energy consumed by mobile terminals and task delay greatly, offloaded data is vulnerable to malicious attacks. In view of this, secure computation offloading has attracted increasing attention.

Although computation offloading can reduce energy consumed by mobile terminals and task delay greatly, offloaded

data is vulnerable to malicious attacks. In view of this, secure computation offloading has attracted increasing attention. There exists some related work listed as follows. In [23], Han *et al.* jointly optimized computing and communicational resources to maximize the secrecy energy efficiency of computation offloading in a NOMA system. In [24], He *et al.* jointly optimized offloading ratio and uplink transmission power to energy-plus-payment cost, where some jamming signals broadcasted by edge servers were used for impeding eavesdropping. In [25], Wang *et al.* jointly optimized uplink transmission power, offloading timeslots, task allocation and local processing frequency to minimize system energy consumption under physical layer security techniques. In [26], Wu *et al.* jointly optimized task partition, power allocation, codeword transmission rate and confidential data rate to minimize the weighted sum of energy consumption under physical layer security and NOMA techniques. In [27], Bai *et al.* jointly optimized offloading and attacking decisions to maximize the expected reward of the edge system, which involves both the service delay and security risks. In [28], Liu *et al.* jointly optimized task partition, uplink transmission power and offloading rate to maximize the requirement satisfaction of all users, which is quantized as a combination of delay, energy consumption and security decisions.

It is evident that the above-mentioned work concentrated on physical-layer assisted secure offloading mechanisms. When many attackers cooperate with each other, such mechanisms cannot guarantee the task security well. In view of this, some secure offloading schemes based on cryptographic algorithms have attracted increasing attention. In [29], Elgendy *et al.* jointly optimized security decisions, resource allocation and computation offloading to minimize the energy consumption and delay of the entire system. In addition, they also jointly optimized security decisions, offloading policy, task compression and resource allocation to minimize the weighted sum of energy consumption for a multi-task MEC system [30]. After that, according to the execution time, energy consumption, CPU and memory usage, the computation offloading was dynamically performed in [31], where a new security layer was added to protect the transferred data in the cloud. In [32], Zahed *et al.* jointly optimized security service assignment, cooperative task offloading and caching to minimize the total system cost quantized as a combination of security breach cost and energy consumption.

Among the above-mentioned efforts, few concentrate on the design of secure multi-task multi-step computing offloading mechanisms, especially for ultra-dense networks. In addition, most of them utilize the frequency spectrum of ultra-dense networks in a pure OFDMA or full-frequency reusing manner. However, since such manners may result in low spectrum efficiency or severe network interferences, they may be unreasonable and impractical for such networks.

B. Contributions and Organization

Unlike most efforts, in ultra-dense multi-task IoT networks, we first consider the BS clustering, OFDMA and NOMA to mitigate network interferences and improve frequency

spectrum utilization. After that, we try to develop a secure and green computation offloading scheme to minimize the energy consumed by IMDs under constraints of latency and secure costs, which jointly optimizes the device association, channel selection, task partition, security service assignment, uplink transmission power and computing resource allocation. Specifically, the main work and contributions of this paper can be summarized as follows:

- 1) *Joint BS Clustering, OFDMA and NOMA Used for Ultra-Dense IoT Networks:* To mitigate the complicated and severe interferences, and improve frequency spectrum utilization, we consider the following operations for ultra-dense IoT networks. At first, SBSs are first divided into several clusters using K-means according to their physical positions. Secondly, the whole system frequency band is cut into two parts used by MBS and SBS separately. Thirdly, we let each cluster own some orthogonal sub-channels (frequency bands), but different clusters have distinct subchannels. At last, we consider that IMDs served by SBSs in the same cluster perform uplink transmission in a NOMA manner, but ones served by MBSs utilize frequency bands equally. As far as we know, such joint BS Clustering, OFDMA and NOMA should be a new investigation for ultra-dense IoT networks.
- 2) *Secure Multi-Step Multi-Task Computation Offloading in Ultra-Dense IoT Networks:* In ultra-dense multi-task IoT networks, we consider secure one-step and two-step computation offloading. In a one-step manner, a part of any task of an IMD is offloaded to the associated MBS. In a two-step manner, a part of any task of an IMD is first offloaded to the associated SBS, and then a part of the partial task received by the SBS is further offloaded to a nearby MBS. To attain the goal of secure green communications, any offloaded part needs to be encrypted, and receivers decrypt received parts. To the best of our knowledge, such secure multi-step multi-task computation offloading should be a new topic in ultra-dense IoT networks.
- 3) *Problem Formulation of Secure Multi-Step Multi-Task Computation Offloading in Ultra-Dense IoT Networks:* To achieve the goal of green and secure computation offloading in ultra-dense multi-task IoT Networks, under joint BS clustering, OFDMA and NOMA, proportional allocation of computational resources, and the constraints of latency and security costs, we jointly optimize device association, channel selection, security service assignment, power control and computation computing resources to minimize the total energy consumed by all IMDs. Evidently, it should be a new formulation.
- 4) *Design Algorithm to Solve the Formulated Problem:* Considering that the formulated problem is in a nonlinear mixed-integer form, we design an improved whale optimization algorithm (IWOA). Specifically, we first improve conventional WOA by changing parameters, settings and rules in three phases consisting of searching for prey, shrinking encirclement and bubble-net attacking. In addition, we replace the current best agent with the

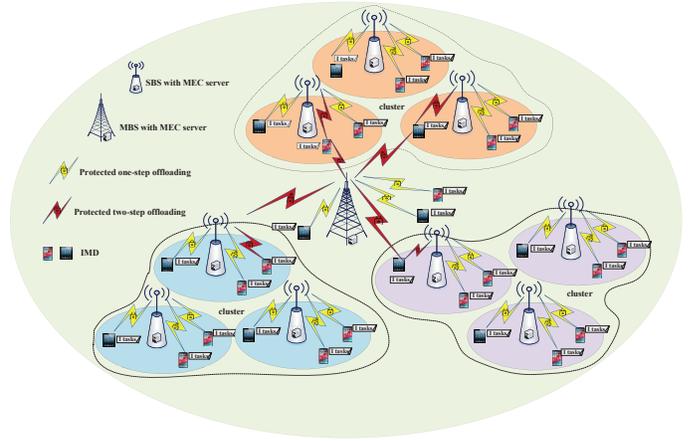


Fig. 1. Ultra-dense multi-task IoT networks with secure multi-step offloading.

historically best agent, and search for prey in the nearby area of the historically best agent.

- 5) *Analyses of Convergence, Computation Complexity and Simulation:* As for the designed algorithm in this paper, we provide some detailed analyses of the convergence and computation complexity. At last, we investigate its effectiveness by introducing other existing algorithms for comparison in the simulation.

The rest of this article is organized as follows. The second section introduces the system model, including the network model, communication model, computing model, security model, and multi-task model; the third section gives the optimization problem formulation of minimizing the energy consumption of the whole network under the constraints of IMDs delay and the total cost of security vulnerabilities; Section IV develops the IWOA-IPSO algorithm to solve the stated problem; Section V provides a detailed algorithm analysis, including three parts: convergence, computational complexity, and parallel implementation; Section VI presents the simulation results and analysis; Section VII presents conclusions and directions for further research in the future.

II. SYSTEM MODEL

In this section, network, communication, security and computation models are given in detail.

A. Network Model

In this paper, we concentrate on ultra-dense multi-task IoT networks with secure multi-step offloading, which is illustrated in Fig.1. In such networks, the number of SBSs is greater than or equal to the one of IMDs; each BS is equipped with a MEC server; all SBSs are connected to nearby MBS via wired links; each IMD has K independent delay-sensitive and computing-intensive tasks to execute within a security breach cost and a specific deadline. Without loss of generality, we consider that there exists an MBS and \bar{S} SBSs in Fig.1, where \bar{S} SBSs are indexed from 1 to \bar{S} in the set $\bar{S} = \{1, 2, \dots, \bar{S}\}$; the index of MBS is 0; $\mathcal{S} = \bar{S} \cup \{0\}$ represents the set of all BSs; U IMDs are indexed from 1 to U in the set $\mathcal{U} = \{1, 2, \dots, U\}$;

the tasks of each IMDs are indexed from 1 to K in the set $\mathcal{K} = \{1, 2, \dots, K\}$.

In Fig.1, when an IMD is associated with some SBS, a part of any task of this IMD is offloaded to such BS after encrypting. This BS first decrypts it and then transmits its part to nearby MBS after encrypting. Significantly, the associated SBS executes the remaining part, and MBS calculates the received part after decrypting. When an IMD is associated with some MBS, a part of any task of this IMD is offloaded to such BS after encrypting. This BS executes it after decrypting. Evidently, SBSs concentrate on secure two-step offloading, but MBSs adopt secure one-step offloading.

To mitigate cross-tier interferences, the system frequency band is cut into two parts used by MBS and SBS separately, where the widths of them are $\eta\varpi$ and $(1 - \eta)\varpi$ respectively; ϖ is the width of the system frequency band; $0 \leq \eta \leq 1$ is the band division factor. To further improve the spectrum efficiency, SBSs are divided into W clusters using K-means according to their physical positions, where each cluster has N orthogonal subchannels (frequency bands) used by SBSs in this cluster, and IMDs associated with these SBSs can utilize the same subchannel through a NOMA manner; N subchannels are indexed from 1 to N in the set \mathcal{N} ; $N = \text{round}((1 - \eta)\varpi/(\omega M))$, $\text{round}()$ is a rounding function, and ω is the bandwidth of a subchannel. Significantly, IMDs associated with some MBS utilize frequency band $\eta\varpi$ equally.

B. Communication Model

Under the aforementioned resource utilization manner, there just exist intra-cluster interferences exist for any task of IMDs. In view of this, the uplink data rate of IMD i associated with SBS $s \in \bar{\mathcal{S}}$ on subchannel n can be given by

$$\begin{cases} R_{i,s,n} = \omega \log_2 \left(1 + \frac{p_i \bar{h}_{i,s}}{\sum_{u \in \mathcal{Q}_{i,s,n}} p_u \bar{h}_{u,s} + \sigma^2} \right), \\ \mathcal{Q}_{i,s,n} = \{i \in \mathcal{U}\} \setminus \{i = u\} : \\ \bar{h}_{u,s} \leq \bar{h}_{i,s}; a_u = a_i = n; b_u, b_i \in \mathcal{W}_s, \end{cases} \quad (1)$$

where a_i and b_i are the channel and SBS indices selected by IMD i respectively; $\bar{h}_{i,s}$ is channel gain between IMD i and BS s ; p_i is the transmission power of IMD i ; σ^2 is the noise power; \mathcal{W}_s denotes the cluster that SBS s belongs to.

Since IMDs associated with an MBS utilize frequency bands equally, and these bands are different from the ones used by SBSs, there are no intra-tier and cross-tier interferences. Considering that IMDs often transmit tasks one by one on some channel, we can assume that each MBS has N virtual subchannels, which correspond to only one channel in reality. That is to say, any IMD can use one of them to transmit a task at some time slot, which means that such an IMD utilizes a real channel to do it. Based on this, the uplink data rate of IMD i associated with MBS 0 on subchannel n can be given by

$$R_{i,0,n} = \eta\varpi \left(\sum_{u \in \mathcal{U}} x_{u,0} \right)^{-1} \log_2 \left(1 + p_i \bar{h}_{i,0} / \sigma^2 \right), \quad (2)$$

where $\sum_{u \in \mathcal{U}} x_{u,0}$ is the number of IMDs associated with MBS 0; $x_{i,s}$ denotes the association index of IMD i at BS s ; $x_{i,s} = 1$ if IMD i is associated with the BS s ; otherwise, $x_{i,s} = 0$.

C. Security Model

In the reality, offloaded tasks often have different security requirements. However, they may be vulnerable to malicious attacks, eavesdropping, and spoofing. To tackle such an issue, data encryption and decryption are widely regarded as promising solutions, which utilize different cryptographic algorithms. As revealed in [32], as the strength and robustness of security protection algorithms increase, the energy and latency overhead increase significantly. In addition, these preventive measures prevent security breaches completely. Therefore, quantifying security risks is an important topic in the design of secure offloading strategies.

To guarantee secure offloading, offloaded tasks are encrypted and decrypted using different cryptographic algorithms in this paper. When an IMD is associated with some SBS, a part of any task of this IMD is offloaded to such BS after encrypting. This BS first decrypts it and then transmits its part to nearby MBS after encrypting. Significantly, the associated SBS executes the remaining part, and MBS calculates the received part after decrypting. When an IMD is associated with some MBS, a part of any task of this IMD is offloaded to such BS after encrypting. This BS executes it after decrypting.

As we know, distinct cryptographic algorithms correspond to distinguishable security levels. We assume that security protection levels are indexed from v_1 to v_L in the set $\mathcal{V} = \{v_1, v_2, \dots, v_l, \dots, v_L\}$, and $v_l = l$ represents protection level (robustness) of the cryptographic algorithm l . In addition, the computation capacities of cryptographic algorithm l are $\bar{\theta}_l$ (in CPU cycles/bit) and $\hat{\theta}_l$ (in CPU cycles/bit) for encrypting and decrypting one bit, and the corresponding energy consumptions are assumed to be the same, i.e., θ_l (in mJ/bit). Significantly, $\bar{\theta} = \{\bar{\theta}_l, \forall l \in \mathcal{L}\}$, $\hat{\theta} = \{\hat{\theta}_l, \forall l \in \mathcal{L}\}$ and $\bar{\theta} = \{\bar{\theta}_l, \forall l \in \mathcal{L}\}$.

When task k of IMD i adopts cryptographic algorithm l to offload its parts securely, its failure probability [33] can be given by

$$\bar{p}_{i,k,l} = \begin{cases} 1 - e^{-\nu_{i,k}(\rho_{i,k} - v_l)}, & \text{if } v_l < \rho_{i,k}, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where $\nu_{i,k}$ is the security risk coefficient of task k of IMD i ; $\rho_{i,k}$ is the expected security level of task k of IMD i . As revealed in (3), cryptographic algorithm l successfully protects task m of IMD i if its security level is greater or equal to the expected one. Otherwise, algorithm l fails in protecting such a task.

The security breach cost [33] of task k of IMD i can be given by

$$\varphi_{i,k} = \sum_{s \in \mathcal{S}} \sum_{l \in \mathcal{L}} \lambda_k x_{i,s} y_{i,k,l} \bar{p}_{i,k,l}, \quad (4)$$

where λ_k is the finance loss (in \$) of task k if it fails; $y_{i,k,l}$ is the security decision index of the task k of IMD i , $y_{i,k,l} = 1$ if cryptographic algorithm l is selected for tackling task k of IMD i , 0 otherwise. Then, overall security breach cost of IMD i can be given by

$$\psi_i = \sum_{k \in \mathcal{K}} \varphi_{i,k} = \sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}} \lambda_k x_{i,s} y_{i,k,l} \bar{p}_{i,k,l}, \quad (5)$$

where $\mu_{i,k}$ is the cost incurred by failure security protection of task k of IMD i .

D. Computational Model

Task k of IMD i is denoted as $\mathcal{D}_{i,k} \triangleq (d_{i,k}, c_{i,k}, \tau_i^{\max}, \rho_{i,k})$, where $d_{i,k}$ represents the data size of task k of IMD i ; $c_{i,k}$ is the number of CPU cycles used to calculate one bit of task k of IMD i ; τ_i^{\max} is the deadline of IMD i .

1) Local computation: When IMD i is associated with BS s , the size of locally processed data of task k of IMD i is $d_{i,k} - \bar{d}_{i,s,k}$, where $\bar{d}_{i,s,k}$ is the data size of task k offloaded from IMD i to BS s . In addition, the local executing time $\tau_{i,s,k}^{LOC}$ used for processing the task k of IMD i associated with BS s can be given by

$$\tau_{i,s,k}^{LOC} = \frac{(d_{i,k} - \bar{d}_{i,s,k})c_{i,k}}{f_i^{UE}} + \sum_{l \in \mathcal{L}} \frac{y_{i,k,l} \bar{\theta}_l \bar{d}_{i,s,k}}{f_i^{UE}}, \quad (6)$$

where f_i^{UE} is the computing capability (capacity) of IMD i ; the two items on the right side of (6) are the computing time and encrypting time respectively.

2) Offloading to SBS: When IMD i is associated with SBS s , the following steps need to be executed for any task k . At first, the part $\bar{d}_{i,s,k}$ of $d_{i,k}$ is offloaded from IMD i to SBS s after encrypting. Secondly, SBS s decrypts $\bar{d}_{i,s,k}$, and then executes $\bar{d}_{i,s,k} - \hat{d}_{i,s,k}$. Thirdly, the part $\hat{d}_{i,s,k}$ of $\bar{d}_{i,s,k}$ is offloaded from SBS s to nearby MBS after encrypting. Fourthly, MBS executes $\hat{d}_{i,s,k}$ after decrypting. Consequently, the remote time $\tau_{i,s,k}^{BS}$ used for processing the task k of IMD i associated with SBS s can be given by

$$\begin{aligned} \tau_{i,s,k}^{BS} &= \sum_{n \in \mathcal{N}} \frac{z_{i,n} \bar{d}_{i,s,k}}{R_{i,s,n}} + \frac{(\bar{d}_{i,s,k} - \hat{d}_{i,s,k})c_{i,k}}{f_{i,s,k}} + \frac{\hat{d}_{i,s,k}}{R_0} \\ &+ \frac{\hat{d}_{i,s,k}c_{i,k}}{f_{i,0,k}} + \sum_{l \in \mathcal{L}} \frac{y_{i,k,l} \hat{\theta}_l \bar{d}_{i,s,k}}{f_{i,s,k}} \\ &+ \sum_{l \in \mathcal{L}} \frac{y_{i,k,l} \bar{\theta}_l \hat{d}_{i,s,k}}{f_{i,s,k}} + \sum_{l \in \mathcal{L}} \frac{y_{i,k,l} \hat{\theta}_l \hat{d}_{i,s,k}}{f_{i,0,k}}, \end{aligned} \quad (7)$$

where $z_{i,n}$ denotes the association decision of IMD i on subchannel n ; $z_{i,n} = 1$ if IMD i selects subchannel n , $z_{i,n} = 0$ otherwise. R_0 is the wired backhaul rate between SBS and MBS; $f_{i,s,k}$ is the computing capability allocated to task k of IMD i by SBS s ; on the right side of (7), the first four items are the time used for uploading $\bar{d}_{i,s,k}$ from IMD i to SBS s , the one used for computing $\bar{d}_{i,s,k} - \hat{d}_{i,s,k}$ at SBS s , the one used for uploading $\hat{d}_{i,s,k}$ from SBS s to nearby MBS, and the one used for computing $\hat{d}_{i,s,k}$ at MBS, respectively. The last three items are the time used for decrypting $\bar{d}_{i,s,k}$ at SBS s , the one used for encrypting $\hat{d}_{i,s,k}$, and the one used for decrypting $\hat{d}_{i,s,k}$, respectively.

According to the ratio of CPU cycles used for tackling task k of IMD i to total utilized cycles at associated BS s , the computing capability of BS s is allocated to the computing and secure operations of such a task. Specifically, when IMD i is associated with SBS s , the computing capability $\bar{f}_{i,s,k}$ assigned to k of IMD i by SBS s can be given by

$$\bar{f}_{i,s,k} = \frac{f_s^{BS} (\Gamma_{i,s,k} + \sum_{l \in \mathcal{L}} y_{i,k,l} \bar{\Gamma}_{i,s,k,l})}{\sum_{u \in \mathcal{U}} \sum_{j \in \mathcal{K}} x_{u,s} (\Gamma_{u,s,j} + \sum_{l \in \mathcal{L}} y_{u,j,l} \bar{\Gamma}_{u,s,j,l})}, \quad (8)$$

$$\begin{cases} \Gamma_{i,s,k} = (\bar{d}_{i,s,k} - \hat{d}_{i,s,k})c_{i,k}, \\ \bar{\Gamma}_{i,s,k,l} = \hat{\theta}_l \bar{d}_{i,s,k} + \bar{\theta}_l \hat{d}_{i,s,k}, \end{cases} \quad (9)$$

where f_s^{BS} represents total computing capability of SBS s ; $\Gamma_{i,s,k}$ is the CPU cycles used for processing $\bar{d}_{i,s,k} - \hat{d}_{i,s,k}$; $\bar{\Gamma}_{i,s,k,l}$ is the CPU cycles used for decrypting $\bar{d}_{i,s,k}$ and encrypting $\hat{d}_{i,s,k}$.

Since IMDs associated with SBSs can further offload partial tasks to nearby MBSs for processing, and ones associated with MBSs can directly upload tasks to these BSs for execution, the data processed at any MBS should include the following two parts. Consequently, the CPU cycles used for computing and decrypting the data offloaded from SBSs to this MBS selected by IMDs, which is given by $\sum_{u \in \mathcal{U}} \sum_{s \in \bar{\mathcal{S}}} x_{u,s} \sum_{j \in \mathcal{K}} \Upsilon_{u,s,j}$, where $\Upsilon_{u,s,j} = \hat{d}_{u,s,j} c_{u,j} + \sum_{l \in \mathcal{L}} y_{u,j,l} \hat{\theta}_l \hat{d}_{u,s,j}$. In addition, the CPU cycles used for computing and decrypting the data offloaded from IMDs to MBS selected by them, which is given by $\sum_{u \in \mathcal{U}} x_{u,0} \sum_{j \in \mathcal{K}} \bar{\Upsilon}_{u,0,j}$, where $\bar{\Upsilon}_{u,0,j} = \bar{d}_{u,0,j} c_{u,j} + \sum_{l \in \mathcal{L}} y_{u,j,l} \bar{\theta}_l \bar{d}_{u,0,j}$. Under the proportional computing allocation mentioned previously, when IMD i is associated with MBS 0, the computing capability $\bar{f}_{i,0,k}$ assigned to k of IMD i by MBS 0 can be given by

$$\bar{f}_{i,0,k} = \frac{f_0^{BS} (\sum_{s \in \bar{\mathcal{S}}} x_{i,s} \Upsilon_{u,s,j} + x_{i,0} \bar{\Upsilon}_{u,0,j})}{\sum_{u \in \mathcal{U}} \sum_{j \in \mathcal{K}} (\sum_{s \in \bar{\mathcal{S}}} x_{u,s} \Upsilon_{u,s,j} + x_{u,0} \bar{\Upsilon}_{u,0,j})}. \quad (10)$$

3) Offloading to MBS: When IMD i is associated with MBS 0, the following steps need to be executed for any task k . At first, the part $\bar{d}_{i,s,k}$ of $d_{i,k}$ is offloaded from IMD i to MBS 0 after encrypting. Secondly, MBS 0 decrypts $\bar{d}_{i,s,k}$, and then executes it. Consequently, the remote time $\tau_{i,0,k}^{BS}$ used for processing the task k of IMD i associated with MBS 0 can be given by

$$\tau_{i,0,k}^{BS} = \sum_{n \in \mathcal{N}} \frac{z_{i,n} \bar{d}_{i,0,k}}{R_{i,0,n}} + \frac{\bar{d}_{i,0,k}c_{i,k}}{f_{i,0,k}} + \sum_{l \in \mathcal{L}} \frac{y_{i,k,l} \bar{d}_{i,0,k} \hat{\theta}_l}{f_{i,0,k}}, \quad (11)$$

where the items on the right side of (11) represent the time used for uploading $\bar{d}_{i,s,k}$ from IMD i to MBS 0, the one used for computing $\bar{d}_{i,s,k}$ at MBS 0, and the one used for decrypting $\bar{d}_{i,s,k}$ at MBS 0, respectively.

We assume that all computation tasks of each IMD are executed sequentially to satisfy practical implementations. However, local execution and computation offloading can be performed for any task in a parallel manner. Therefore, the total time τ_i used for completing all task of IMD i can be given by

$$\tau_i = \sum_{k \in \mathcal{K}} \max \left(\sum_{s \in \mathcal{S}} x_{i,s} \tau_{i,s,k}^{LOC}, \sum_{s \in \mathcal{S}} x_{i,s} \tau_{i,s,k}^{BS} \right), \quad (12)$$

Then, the total energy consumed by all IMDs can be given by

$$\begin{aligned} \epsilon &= \sum_{i \in \mathcal{U}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} \varsigma x_{i,s} (d_{i,k} - \bar{d}_{i,s,k}) c_{i,k} f_i^2 \\ &+ \sum_{i \in \mathcal{U}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} \sum_{l \in \mathcal{L}} x_{i,s} y_{i,k,l} \hat{\theta}_l \bar{d}_{i,s,k} \\ &+ \sum_{i \in \mathcal{U}} \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} \sum_{n \in \mathcal{N}} x_{i,s} z_{i,n} p_i \bar{d}_{i,s,k} / R_{i,s,n}, \end{aligned} \quad (13)$$

where ς is the energy coefficient of chip architecture; the three items on the right side of (13) are total computing, encrypting and uploading energy consumptions of IMDs, respectively.

III. PROBLEM FORMULATION AND SOLUTION

A. Problem Formulation

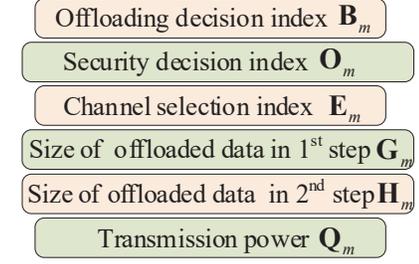
To achieve the goal of green and secure computation of offloading in ultra-dense multi-task IoT Networks, we try to minimize the total energy consumption of all IMDs under joint BS clustering, OFDMA and NOMA, proportional allocation of computational resources, and the constraints of latency and security costs, jointly optimizing the device association, channel selection, security service assignment, power control and computation computing resources. Mathematically, it can be formulated as

$$\begin{aligned}
& \min_{\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{p}, \bar{\mathbf{D}}, \hat{\mathbf{D}}} \epsilon(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{p}, \bar{\mathbf{D}}, \hat{\mathbf{D}}) \\
& \text{s.t. } C_1 : \tau_i \leq \tau_i^{\max}, \forall i \in \mathcal{U}, \\
& C_2 : \psi_i \leq \psi_i^{\max}, \forall i \in \mathcal{U}, \\
& C_3 : \sum_{s \in \mathcal{S}} x_{i,s} = 1, \forall i \in \mathcal{U}, \\
& C_4 : \sum_{l \in \mathcal{L}} y_{i,k,l} = 1, \forall i \in \mathcal{U}, \forall k \in \mathcal{K}, \\
& C_5 : \sum_{i \in \mathcal{U}} z_{i,n} = 1, \forall n \in \mathcal{N}, \\
& C_6 : \vartheta \leq p_i \leq p_i^{\max}, \forall i \in \mathcal{U}, \\
& C_7 : x_{i,s} \in \{0, 1\}, \forall i \in \mathcal{U}, s \in \mathcal{S}, \\
& C_8 : y_{i,k,l} \in \{0, 1\}, \forall i \in \mathcal{U}, \forall k \in \mathcal{K}, l \in \mathcal{L}, \\
& C_9 : z_{i,n} \in \{0, 1\}, \forall i \in \mathcal{U}, \forall n \in \mathcal{N}, \\
& C_{10} : \theta \leq \hat{d}_{i,s,k} \leq \bar{d}_{i,s,k} \leq d_{i,k}, \forall i \in \mathcal{U}, s \in \mathcal{S}, k \in \mathcal{K},
\end{aligned} \tag{14}$$

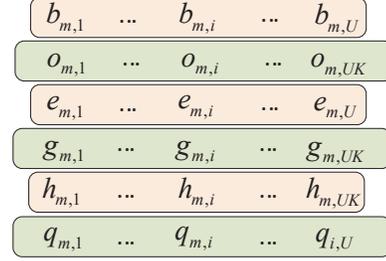
where $\mathbf{X} = \{x_{i,s}, \forall i \in \mathcal{U}, \forall s \in \mathcal{S}\}$, $\mathbf{Y} = \{y_{i,k,l}, \forall i \in \mathcal{U}, \forall k \in \mathcal{K}, \forall l \in \mathcal{L}\}$, $\mathbf{Z} = \{z_{i,n}, \forall i \in \mathcal{U}, \forall n \in \mathcal{N}\}$, $\mathbf{p} = \{p_i, \forall i \in \mathcal{U}\}$, $\bar{\mathbf{D}} = \{\bar{d}_{i,s,k}, \forall i \in \mathcal{U}, \forall s \in \mathcal{S}, \forall k \in \mathcal{K}\}$, $\hat{\mathbf{D}} = \{\hat{d}_{i,s,k}, \forall i \in \mathcal{U}, \forall s \in \mathcal{S}, \forall k \in \mathcal{K}\}$; ϑ takes a small enough value to avoid zero division, e.g., 10^{-20} ; C_1 indicates that the task execution time of IMD i cannot exceed its deadline τ_i^{\max} ; C_2 means that total security breach cost of IMD i cannot exceed its maximum acceptable cost ψ_i^{\max} ; C_3 and C_7 indicate that an IMD can just be associated with only one BS; C_4 and C_8 indicate that the task k of IMD i can just select only one cryptographic algorithm; C_5 and C_9 indicate that an IMD can just select only one subchannel; C_6 gives the lower bound (ϑ) and upper bound (p_i^{\max}) of the transmission power of IMD i ; C_{10} means that the offloaded parts $\bar{d}_{i,s,k}$ and $\hat{d}_{i,s,k}$ are greater than or equal to ϑ , but less than or equal to the data size $d_{i,k}$ of task k of IMD i . Meanwhile, $\hat{d}_{i,s,k}$ must be less than or equal to $\bar{d}_{i,s,k}$.

B. Algorithm Design

As revealed in [34], WOA is a gradient-free method and can relax the computations of gradients. In addition, it is insensitive to the initial feasible solutions, which may affect the convergence and performance of other traditional methods greatly. Moreover, WOA has been equipped with adaptive mechanisms that balance its explorative and exploitative behaviors appropriately, which can increase the probability of avoiding locally optimal solutions. At last, since WOA is flexible and easy to be implemented, it is applicable to common optimization problems rather than particular ones. So far, WOA has been regarded as a promising solution



(a) The location of whale m



(b) Specific location structures of whales

Fig. 2. Encoding structures of whales.

to the optimization problem in wireless and communication networks. In view of this, we develop IWOA to solve the formulated problem (14) by improving WOA in [34], which consists of encircling prey, bubble-net attacking (exploitation phase) and prey search (exploration phase). To utilize IWOA to solve (14), we need to encode whales, define their fitness function, and initialize their values. Then, the procedures of encircling prey, bubble-net attacking and search for prey are given in detail.

1) Encode whale

The optimization parameters $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{p}, \bar{\mathbf{D}}, \hat{\mathbf{D}}$ of problem (14) are encoded as $\mathbf{B}_m, \mathbf{O}_m, \mathbf{E}_m, \mathbf{Q}_m, \mathbf{G}_m, \mathbf{H}_m$ respectively, where $\mathbf{B}_m = \{b_{m,i}, i \in \mathcal{U}\}$, and $b_{m,i}$ is the BS index selected by IMD i in the individual (whale) m ; $\mathbf{O}_m = \{o_{m,i}, i \in \bar{\mathcal{U}}\}$, $\bar{\mathcal{U}} = \{1, 2, \dots, K, K+1, \dots, 2K, \dots, UK\}$, and $o_{m,i}$ is the index of cryptographic algorithm selected by virtual IMD i in the individual m ; $\mathbf{E}_m = \{e_{m,i}, i \in \mathcal{U}\}$, and $e_{m,i}$ is the channel index selected by IMD i in the individual m ; $\mathbf{Q}_m = \{q_{m,i}, i \in \mathcal{U}\}$, and $q_{m,i}$ is the transmission power of IMD i in the individual m ; $\mathbf{G}_m = \{g_{m,i}, i \in \bar{\mathcal{U}}\}$, and $g_{m,i}$ is the amount of data offloaded from IMD i to its associated SBS in the individual m ; $\mathbf{H}_m = \{h_{m,i}, i \in \bar{\mathcal{U}}\}$, and $h_{m,i}$ is the amount of data offloaded from IMD i or its associated SBS to nearby MBS in the individual m . Significantly, $\mathcal{M} = \{1, 2, \dots, M\}$ represents the population consisting of M individuals (whales).

The coding and structure of individuals are shown in Fig.2.

2) Fitness function

To assess the fitness of individuals (whales), fitness func-

tions need to be designed properly. Seen from (14), we can easily observe that constraints C_1 and C_2 are in nonlinear, mixed-integer and coupling forms, and hard to be met in whales' actions. In view of this, they are introduced into the fitness function as penalty terms, which can be explicitly used to prevent individuals from falling into the infeasible region. In this way, the established population can always find a feasible optimal solution.

To minimize the energy consumed by all IMDs under the constraints C_1 and C_2 , the fitness function of individual m can be defined as

$$\begin{aligned} F(\mathbf{B}_m, \mathbf{O}_m, \mathbf{E}_m, \mathbf{Q}_m, \mathbf{G}_m, \mathbf{H}_m) \\ = -\epsilon(\mathbf{B}_m, \mathbf{O}_m, \mathbf{E}_m, \mathbf{Q}_m, \mathbf{G}_m, \mathbf{H}_m) \\ - \sum_{i \in \mathcal{U}} \alpha_i \max(0, \tau_i - \tau_i^{\max}) \\ - \sum_{i \in \mathcal{U}} \beta_i \max(0, \psi_i - \psi_i^{\max}), \end{aligned} \quad (15)$$

where α_i and β_i are the penalty factors of IMD i .

3) Population initialization

In order to meet the constraints C_3 - C_{10} , initial population can be generated use the following rules. Specifically, any individual m can be initialized into

$$\begin{cases} b_{m,i}^0 = \text{randi}(\mathcal{S}), \forall i \in \mathcal{U}, \\ o_{m,i}^0 = \text{randi}(\mathcal{L}), \forall i \in \bar{\mathcal{U}}, \\ e_{m,i}^0 = \text{randi}(\mathcal{N}), \forall i \in \bar{\mathcal{U}}, \\ q_{m,i}^0 = \text{rand}(p_i^{\max}), \forall i \in \mathcal{U}, \\ g_{m,i}^0 = \text{rand}(d_{u,k}), \forall i \in \bar{\mathcal{U}}, \\ h_{m,i}^0 = \text{rand}(g_{m,i}^0), \forall i \in \bar{\mathcal{U}}, \\ [u, k] = \text{ind2sub}([UK], i), \forall i \in \bar{\mathcal{U}}, \end{cases} \quad (16)$$

where $[u, k] = \text{ind2sub}([UK], i)$ returns the row subscript m and column subscript k of $U \times K$ matrix corresponding to the linear index i ; $\text{randi}(\mathcal{Z})$ outputs an element from the set \mathcal{Z} randomly, and $\text{rand}(\gamma)$ generates a random number between 0 and γ .

4) Encircle prey

Humpback whales can recognize the locations of prey and then encircle them completely. Therefore, all whales are agents that search for prey. In conventional WOA [34], the current best agent is assumed to be the target prey, and all whales update their positions towards it during iteration. To ensure global convergence of WOA, we replace the current best agent with the historically best agent. The former refers to the individual (whale) that owns the highest fitness function value among all individuals in the current iteration, but the latter refers to the one that has the highest fitness function value among all individuals in the previous and current iterations. Mathematically, the behavior of encircling prey \bar{m} of individual (whale) m can be formulated as

$$b_{m,i} = \text{round}(\kappa_1 b_{\bar{m},i} - \kappa_2 |\kappa_3 b_{\bar{m},i} - b_{m,i}|), \forall i \in \mathcal{U}, \quad (17)$$

$$o_{m,i} = \text{round}(\kappa_1 o_{\bar{m},i} - \kappa_2 |\kappa_3 o_{\bar{m},i} - o_{m,i}|), \forall i \in \bar{\mathcal{U}}, \quad (18)$$

$$e_{m,i} = \text{round}(\kappa_1 e_{\bar{m},i} - \kappa_2 |\kappa_3 e_{\bar{m},i} - e_{m,i}|), \forall i \in \mathcal{U}, \quad (19)$$

$$q_{m,i} = \kappa_1 q_{\bar{m},i} - \kappa_2 |\kappa_3 q_{\bar{m},i} - q_{m,i}|, \forall i \in \mathcal{U}, \quad (20)$$

$$g_{m,i} = \kappa_1 g_{\bar{m},i} - \kappa_2 |\kappa_3 g_{\bar{m},i} - g_{m,i}|, \forall i \in \bar{\mathcal{U}}, \quad (21)$$

$$h_{m,i} = \kappa_1 h_{\bar{m},i} - \kappa_2 |\kappa_3 h_{\bar{m},i} - h_{m,i}|, \forall i \in \bar{\mathcal{U}}, \quad (22)$$

where $\text{round}(\gamma)$ represents a rounding operation on γ ; $|\gamma|$ is the absolute value of γ ; \bar{m} is the index of historically best agent (individual);

$$\begin{cases} \kappa_1 = \sin(t\pi/2T + \pi) + 1, \\ \kappa_2 = 2(2r_1 - 1)(1 - \sin(t\pi/2T)), \\ \kappa_3 = 2r_2, \end{cases} \quad (23)$$

r_1 and r_2 are random numbers between 0 and 1; t is iteration index; T is the number of iterations.

Inspired by the efforts in [35], [36], adaptive nonlinear weights κ_1 and κ_2 are introduced for updating the positions of individuals (whales) in (17)-(22). In addition, these weights are also used for bubble-net attacks and searching for prey. It is easy to find that such weights can balance exploitation and exploration well.

5) Bubble-net attacking

Bubble-net attacking of humpback whales involves shrinking encircling and spiral movement simultaneously, which are performed in equal probability. By performing these actions, the new position of any agent will be located between its current position and the position of the historically best agent. It means that a local optimum of problem (14) can be found using bubble-net attacking. To mimic the helix-shaped movement of whales, the spiral equation between the positions of prey \bar{m} of any individual (whale) m can be given by

$$b_{m,i} = \text{round}(\kappa_1 b_{\bar{m},i} + \kappa_4 |\kappa_3 b_{\bar{m},i} - b_{m,i}|), \forall i \in \mathcal{U}, \quad (24)$$

$$o_{m,i} = \text{round}(\kappa_1 o_{\bar{m},i} + \kappa_4 |\kappa_3 o_{\bar{m},i} - o_{m,i}|), \forall i \in \bar{\mathcal{U}}, \quad (25)$$

$$e_{m,i} = \text{round}(\kappa_1 e_{\bar{m},i} + \kappa_4 |\kappa_3 e_{\bar{m},i} - e_{m,i}|), \forall i \in \mathcal{U}, \quad (26)$$

$$q_{m,i} = \kappa_1 q_{\bar{m},i} + \kappa_4 |\kappa_3 q_{\bar{m},i} - q_{m,i}|, \forall i \in \mathcal{U}, \quad (27)$$

$$g_{m,i} = \kappa_1 g_{\bar{m},i} + \kappa_4 |\kappa_3 g_{\bar{m},i} - g_{m,i}|, \forall i \in \bar{\mathcal{U}}, \quad (28)$$

$$h_{m,i} = \kappa_1 h_{\bar{m},i} + \kappa_4 |\kappa_3 h_{\bar{m},i} - h_{m,i}|, \forall i \in \bar{\mathcal{U}}, \quad (29)$$

where κ_4 is used for adaptively adjusting the spiral amplitude and avoiding falling into local optimum [37], and it can be given by

$$\begin{cases} \kappa_4 = \exp(a_3 + 5 \cos(\pi(1 - t/T))) \cos(2a_3\pi), \\ a_3 = (-2 - t/T) * r_3 + 1, \end{cases} \quad (30)$$

where r_3 is a random number between 0 and 1.

Besides spiral movement, any whale also needs to perform shrinking encircling action during the bubble-net attacking phase, which can be formulated as (17)-(22).

6) Search for prey

In the prey search of conventional WOA, whales are forced to move toward a random whale. Through such an operation, the search space of this algorithm can be extended. However, its global search capability may greatly rely on the selection of the random whale, and it may be easy to fail into local optimum. To tackle this issue, Cauchy's inverse cumulative distribution function may be used for the mutation operations of whales since its long tail [37]. Inspired by this point, Cauchy's inverse cumulative distribution is used for formulating the prey search of whales. Mathematically, the behavior

of prey search of any individual (whale) m can be formulated as

$$b_{m,i} = \text{round}(b_{m,i} + \kappa_2 \tan(\pi(r_1 - 0.5))), \forall i \in \mathcal{U}, \quad (31)$$

$$o_{m,i} = \text{round}(o_{m,i} + \kappa_2 \tan(\pi(r_1 - 0.5))), \forall i \in \bar{\mathcal{U}}, \quad (32)$$

$$e_{m,i} = \text{round}(e_{m,i} + \kappa_2 \tan(\pi(r_1 - 0.5))), \forall i \in \mathcal{U}, \quad (33)$$

$$q_{m,i} = q_{m,i} + \kappa_2 \tan(\pi(r_1 - 0.5)), \forall i \in \mathcal{U}, \quad (34)$$

$$g_{m,i} = g_{m,i} + \kappa_2 \tan(\pi(r_1 - 0.5)), \forall i \in \bar{\mathcal{U}}, \quad (35)$$

$$h_{m,i} = h_{m,i} + \kappa_2 \tan(\pi(r_1 - 0.5)), \forall i \in \bar{\mathcal{U}}, \quad (36)$$

It is noteworthy that weight κ_2 is used for adaptively adjusting the magnitude of mutation.

6) Search for prey in the nearby area of historically best agent

In order to improve the convergence rate, avoid premature convergence and thus achieve a better solution, we further force whales (individuals) to search for prey in the nearby area of the historically best agent once more. Mathematically, in the nearby area of the historically best agent \bar{m} , new positions of any individual (whale) m can be generated by

$$\bar{b}_{m,i} = \text{round}(b_{\bar{m},i}(1 + 0.5r_4)), \forall i \in \mathcal{U}, \quad (37)$$

$$\bar{o}_{m,i} = \text{round}(o_{\bar{m},i}(1 + 0.5r_4)), \forall i \in \bar{\mathcal{U}}, \quad (38)$$

$$\bar{e}_{m,i} = \text{round}(e_{\bar{m},i}(1 + 0.5r_4)), \forall i \in \mathcal{U}, \quad (39)$$

$$\bar{q}_{m,i} = q_{\bar{m},i}(1 + 0.5r_4), \forall i \in \mathcal{U}, \quad (40)$$

$$\bar{g}_{m,i} = g_{\bar{m},i}(1 + 0.5r_4), \forall i \in \bar{\mathcal{U}}, \quad (41)$$

$$\bar{h}_{m,i} = h_{\bar{m},i}(1 + 0.5r_4), \forall i \in \bar{\mathcal{U}}, \quad (42)$$

where r_4 is a random number between 0 and 1.

As for the newly generated positions using (37)-(42), we decide whether or not to save them in a greedy approach [37]. Specifically, the original positions of whales should be replaced with them when $F(\mathbf{B}_m, \mathbf{O}_m, \mathbf{E}_m, \mathbf{Q}_m, \mathbf{G}_m, \mathbf{H}_m) \leq F(\bar{\mathbf{B}}_m, \bar{\mathbf{O}}_m, \bar{\mathbf{E}}_m, \bar{\mathbf{Q}}_m, \bar{\mathbf{G}}_m, \bar{\mathbf{H}}_m)$, where $\bar{\mathbf{B}}_m = \{\bar{b}_{m,i}, i \in \mathcal{U}\}$, $\bar{\mathbf{O}}_m = \{\bar{o}_{m,i}, i \in \bar{\mathcal{U}}\}$, $\bar{\mathbf{E}}_m = \{\bar{e}_{m,i}, i \in \mathcal{U}\}$, $\bar{\mathbf{Q}}_m = \{\bar{q}_{m,i}, i \in \mathcal{U}\}$, $\bar{\mathbf{G}}_m = \{\bar{g}_{m,i}, i \in \bar{\mathcal{U}}\}$ and $\bar{\mathbf{H}}_m = \{\bar{h}_{m,i}, i \in \bar{\mathcal{U}}\}$. Otherwise, the original positions of whales should not be changed.

Until now, the whole procedure used for IWOA can be summarized as Algorithm 1.

IV. ALGORITHM ANALYSIS

In this section, the convergence, computational complexity, and parallel implementation of IWOA will be analyzed in detail.

Algorithm 1: Improved WOA (IWOA)

```

1: Input: Number  $T$  of iterations.
2: Output:  $\mathbf{B}$ ,  $\mathbf{O}$ ,  $\mathbf{E}$ ,  $\mathbf{Q}$ ,  $\mathbf{G}$  and  $\mathbf{H}$  at  $t$ -th iteration.
3: Initialization:
4: Initialize iteration index:  $t = 1$ .
5: Initialize the population consisting of  $M$  agents using (16).
6: Calculate the fitness values of all agents using (15).
7: Find the historically best agent among all agents.
8: While  $t \leq T$  do
9: Update  $\kappa_1$ ,  $\kappa_2$ ,  $\kappa_3$  and  $\kappa_4$  using (23) and (30).
10: Generate the probability  $r_5$  randomly.
11: If  $r_5 < 0.5$  holds, then
12:   If  $|\kappa_2| \geq 1$  holds, then
13:     All agents search prey using (31)-(36).
14:   Else
15:     All agents encircle prey using (17)-(22).
16:   EndIf
17: Else
18:   All agents perform bubble-net attacks using (24)-(29).
19: EndIf
20: Calculate fitness value  $\chi_m = F(\mathbf{B}_m, \mathbf{O}_m, \mathbf{E}_m, \mathbf{Q}_m, \mathbf{G}_m, \mathbf{H}_m)$ 
21:   of any agent  $m$  using (15).
22: Find the current best agent, and replace historically best agent
23:   with it if its fitness value is higher than historically best agent.
24: Any agent searches for prey in the nearby area of historically best
25:   agent, and generates new position  $\{\bar{\mathbf{B}}_m, \bar{\mathbf{O}}_m, \bar{\mathbf{E}}_m, \bar{\mathbf{Q}}_m, \bar{\mathbf{G}}_m, \bar{\mathbf{H}}_m\}$ 
26:   using (37)-(42).
27: Calculate fitness value  $\bar{\chi}_m = F(\bar{\mathbf{B}}_m, \bar{\mathbf{O}}_m, \bar{\mathbf{E}}_m, \bar{\mathbf{Q}}_m, \bar{\mathbf{G}}_m, \bar{\mathbf{H}}_m)$ 
28:   of any agent  $m$  using (15).
29: If  $\bar{\chi}_m > \chi_m$  holds, then
30:    $\{\mathbf{B}_m, \mathbf{O}_m, \mathbf{E}_m, \mathbf{Q}_m, \mathbf{G}_m, \mathbf{H}_m\}$  is replaced with  $\{\bar{\mathbf{B}}_m, \bar{\mathbf{O}}_m, \bar{\mathbf{E}}_m,$ 
31:    $\bar{\mathbf{Q}}_m, \bar{\mathbf{G}}_m, \bar{\mathbf{H}}_m\}$ 
32: EndIf
33: Update the iteration index:  $t = t + 1$ .
34: EndWhile

```

A. Convergence Analysis

The convergence of IWOA can be established as follows.

Theorem 1: IWOA converges to global optimum solution after a large number of iterations.

Proof: In Algorithm 1 (IWOA), all whales perform the encircling prey, bubble-net attacking (exploitation phase) and prey search (exploration phase) in Steps 9-19. When the iteration index t gradually approaches T , κ_2 is closer and closer to zero. Evidently, when $t = T$, all whales don't search for prey using (31)-(36), they encircle prey and perform bubble-net attacks in equal probability. In other words, IWOA only contains two operations consisting of shrinking encirclement and spiral update at this time. Such operations are performed by whales in equal probability. Even if a common whale falls into a local optimum solution during the spiral update, it may jump out of such a solution when the shrinking operation is done.

It is noteworthy that the historically best agent (whale/individual) always remains in the population of IWOA. In addition, all agents led by this agent perform the operations of shrinking encirclement and spiral update. It means that these two operations force all agents to move toward the historically best agent. Evidently, when the number of iterations of IWOA tends to infinity, it can finally converge to the global optimum solution.

In Steps 20-32 of IWOA, all agents are forced to search for prey in the nearby area of the historically best agent. Such an operation can refine the solutions found by agents, and

improve the historically best agent. It is evident that Steps 20-32 of IWOA can speed up the global convergence of this algorithm.

In general, IWOA converges to a global optimum solution after a large number of iterations. \square

B. Complexity Analysis

The computational complexity of IWOA is analyzed as follows.

Proposition 1: The computational complexity of IWOA is $\mathcal{O}(\max\{TMUK, TMNU^2\})$ after T iterations in the worst scenario that all IMDs share each channel.

In Steps 6-7, the computational complexity is mainly dependent on the calculation of the fitness values of all agents. These fitness values are tightly related to energy consumption, delay and security breach cost of all IMDs. In fact, it is easy to find that the computational complexities of energy consumption and delay mainly come from the calculations of data rates and computing capabilities. To calculate the data rates and computing capabilities, we first convert $\bar{\mathbf{B}}_m$ and $\bar{\mathbf{E}}_m$ into $\mathbf{a} = \{a_i, \forall i \in \mathcal{U}\}$ and $\mathbf{b} = \{b_i, \forall i \in \mathcal{U}\}$ for any individual m , respectively. In addition, \mathbf{O}_m is converted into the indices of cryptographic algorithms for any individual m . Through these conversions, the calculations of data rates and computing capabilities can be greatly reduced since we just need to consider the utilized BSs, channels and cryptographic algorithms for any IMD.

In (1), $\sum_{u \in \mathcal{Q}_{i,s,n}} p_u \bar{h}_{u,s}$ can be calculated before calculating $R_{i,s,n}$. Similarly, $\sum_{u \in \mathcal{U}} x_{u,0}$ can be calculated before calculating $R_{i,0,n}$. Consequently, under the given a_i and b_i , the calculation of $R_{i,s,n}$ may have a complexity of $\mathcal{O}(NU^2)$ in the worst scenario that all IMDs share each channel, and the one of $R_{i,0,n}$ may have a complexity of $\mathcal{O}(NU)$. In (8), $\sum_{l \in \mathcal{L}} y_{i,k,l} \bar{\Gamma}_{i,s,k,l}$ and $\sum_{u \in \mathcal{U}} \sum_{j \in \mathcal{K}} x_{u,s} (\Gamma_{u,s,j} + \sum_{l \in \mathcal{L}} y_{u,j,l} \bar{\Gamma}_{u,s,j,l})$ can be calculated before calculating $\bar{f}_{i,s,k}$ for SBS s . In (10), $\sum_{s \in \bar{\mathcal{S}}} x_{i,s} \Upsilon_{u,s,j}$ and $\sum_{u \in \mathcal{U}} \sum_{j \in \mathcal{K}} (\sum_{s \in \bar{\mathcal{S}}} x_{u,s} \Upsilon_{u,s,j} + x_{u,0} \Upsilon_{u,0,j})$ can be calculated before calculating $\bar{f}_{i,0,k}$ for MBS 0. Consequently, under the given a_i and b_i , the computational complexity of $\bar{f}_{i,s,k}$ is $\mathcal{O}(USK)$ for any BS s .

Based on the above-mentioned analyses, under the given \mathbf{a} , \mathbf{b} and indices of cryptographic algorithms, the computational complexity of delay is $\mathcal{O}(\max\{UK, NU^2\})$ for all IMDs in the worst scenario, and the one of total energy consumption ϵ is still $\mathcal{O}(\max\{UK, NU^2\})$ in the worst scenario. In addition, it is easy to find that the computational complexity of security breach cost is $\mathcal{O}(UK)$ for all IMDs under the given indices of cryptographic algorithms. In general, Steps 6-7 have a computational complexity of $\mathcal{O}(\max\{MUK, MNU^2\})$ in the worst scenario.

Evidently, the computational complexity of Steps 9-19 is $\mathcal{O}(MUK)$. The one of Steps 20-33 mainly comes from the calculation of fitness values of all agents, which is $\mathcal{O}(\max\{MUK, MNU^2\})$ in the worst scenario. After T iterations, the one of IWOA is $\mathcal{O}(\max\{TMUK, TMNU^2\})$ in the worst scenario. \square

TABLE I
SIMULATION PARAMETERS

Parameter	Value
System bandwidth ϖ	20 MHz
Noise power σ^2	10^{-11} mW
IMD power p_i^{\max}	23 dBm
Deadline τ_i^{\max}	5~10 s
Data size $d_{i,k}$	200~500 KB
Size M of population	32
Number W of clusters	5
Number K of tasks	3
Finance loss λ_k	1~5 K\$
Number of SBSs at each macrocell	30
Number L of cryptographic algorithms	6
Maximal security breach cost ψ_i^{\max}	5~10 K\$
Security risk coefficient $\nu_{i,k}$	1~3
Expected security level $\rho_{i,k}$	{5, 6}
Wired backhauling rate r_0	1 Gbps
Computation capacity f_s^{BS}	20 GHz
Computation capacity f_i^{UE}	1 GHz
$c_{i,k}$ used for computing one bit of $d_{i,k}$	50~100 cycles/bit
Pathloss between MBS 0 and IMD i	$128.1 + 37.6 \log_{10}(\ell_{i,0})$
Pathloss between SBS s and IMD i	$140.7 + 36.7 \log_{10}(\ell_{i,s})$
Log-normal shadowing fading	Standard deviation of 8 dB

C. Parallel implementation

As revealed in the previous section, the computational complexity of IWOA mainly comes from the calculations of the fitness values of all agents. Such calculations will lead to relatively high computational complexity if the number of agents is too large. In order to reduce computational complexity and improve the efficiency of designed algorithm, all agents should calculate their fitness values in a parallel manner, which has been widely advocated in reality. Certainly, any one of three operations consists of encircling prey, bubble-net attacking and prey search can be also performed by all agents in parallel.

V. NUMERICAL RESULTS

Without loss of generality, IMDs and ultra-dense SBSs are randomly deployed into a macrocell, where the number of SBSs is greater than or equal to the number of IMDs. At the same time, we consider $\hat{\theta} = [100, 200, 250, 300, 350, 1050]$ cycles/bit, $\hat{\theta} = [90, 280, 350, 300, 400, 1700]$ cycles/bit and $\hat{\theta} = [2.5296, 5.0425, 6.837, 7.8528, 8.7073, 26.3643] \times 10^{-7}$ J/bit [38]. Moreover, other important parameters are summarized in TABLE I, where $\ell_{i,s}$ is the distance (in km) between BS s and IMD i .

To highlight the effectiveness of IWOA, the following algorithms are introduced for comparison.

Computation at Mobile Terminals (CMT): All IMDs complete their computation tasks by themselves in allowable maximum computing capacity.

Computation at MEC Servers (CMS): All computation tasks are offloaded from IMDs to BSs with the best channel gains. In addition, cryptographic algorithms with minimum security breach costs are always selected for these tasks. According to the ratio of CPU cycles used for tackling them, the

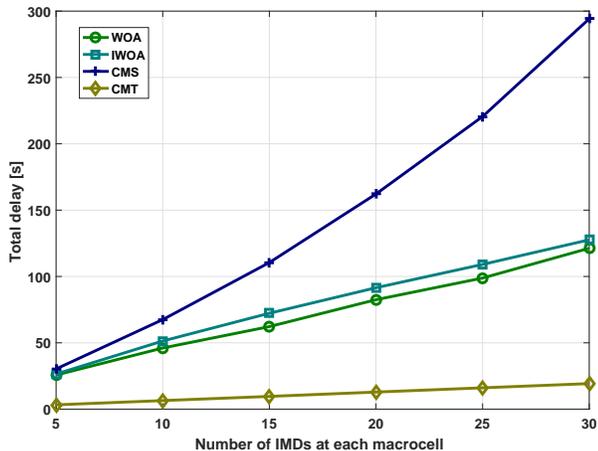


Fig. 3. Impacts of the number of IMDs at each macrocell on total task delay.

computation capacities of any BS are allocated to its served tasks proportionally.

Whale Optimization Algorithm (WOA): To solve the problem (14), WOA in [34] is introduced.

In the simulation, we mainly investigate the impacts of the number of IMDs at each macrocell, and the frequency spectrum partitioning factor on the offloading performance. Due to the consideration of the historically best agent, Cauchy's inverse cumulative distribution function used for the search of prey, and the search of prey in the nearby area of the historically best agent in IWOA, IWOA may achieve lower total local energy consumption than WOA in general, and the former may also achieve higher fitness (function) value than the latter, which will be illustrated in the subsequent simulation. In addition, CMS may achieve the lowest total local energy consumption among all algorithms since it has not locally executed tasks, but CMT may achieve the highest one among all algorithms since it lets all tasks of IMDs be executed locally in allowable maximum computing capacity. As we know, in order to achieve lower local energy consumption, lower local and/or remote computation capacities may be used, resulting in higher task delay. Consequently, among all algorithms, CMS may achieve the highest total delay, CMT may have the lowest one, and IWOA may have a higher one than WOA. Under some large enough penalty factors, the latency and cost constraints of IMDs may be guaranteed strictly in WOA and IWOA. Since the support ratios of cost constraints of all algorithms are always 1, such a performance metric will not be illustrated in the following simulation, where the cost support ratio refers to the ratio of IMDs whose costs are less than or equal to the total security breach costs of them to all IMDs.

Fig.3 shows the impacts of the number of IMDs at each macrocell on total task delay. As illustrated in Fig.3, the total task delay of all algorithms may increase with the number of IMDs at each macrocell. Such a performance trend can be easily inferred according to the definition of total task delay in (12).

Fig.4 shows the impacts of the number of IMDs at each

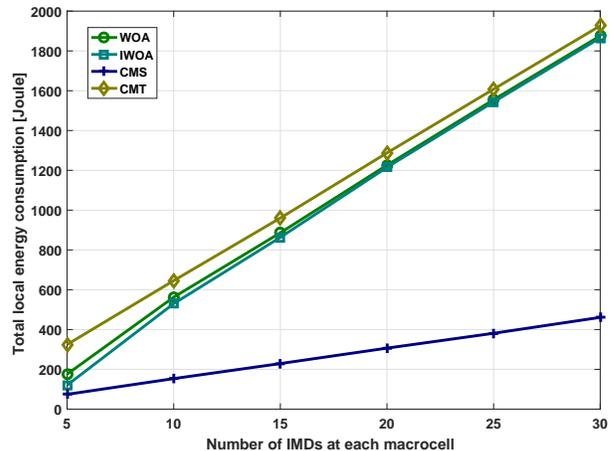


Fig. 4. Impacts of the number of IMDs at each macrocell on total local energy consumption.

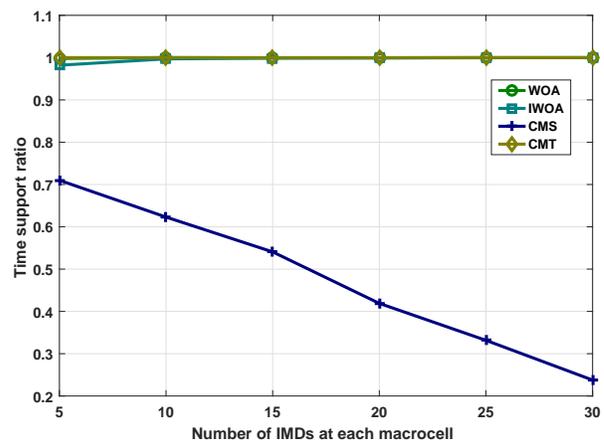


Fig. 5. Impacts of the number of IMDs at each macrocell on time support ratio.

macrocell on total local energy consumption. As illustrated in Fig.3, the total local energy consumption of all algorithms may increase with the number of IMDs at each macrocell. Such a performance trend can be easily inferred according to the definition of total local energy consumption in (13).

Fig.5 shows the impacts of the number of IMDs at each macrocell on the time support ratio, where the mentioned ratio refers to the ratio of IMDs whose task delay is less than or equal to the deadlines of them to all IMDs. As illustrated in Fig.5, WOA, IWOA and CMT almost certainly meet the latency constraints of all IMDs. Under some large enough penalty factors, the latency constraints of all IMDs in WOA and IWOA are forced to be met. Since CMT has no uplink transmission delay and encrypting delay, and it always completes the computation tasks of all IMDs in the allowable maximum computing capacity, the latency constraints of all IMDs in it can be guaranteed strictly. Unlike other algorithms, the time support ratio of CMS may decrease with the number of IMDs at each macrocell. According to the rules of CMS, we can easily know that tasks of IMDs may be always offloaded to

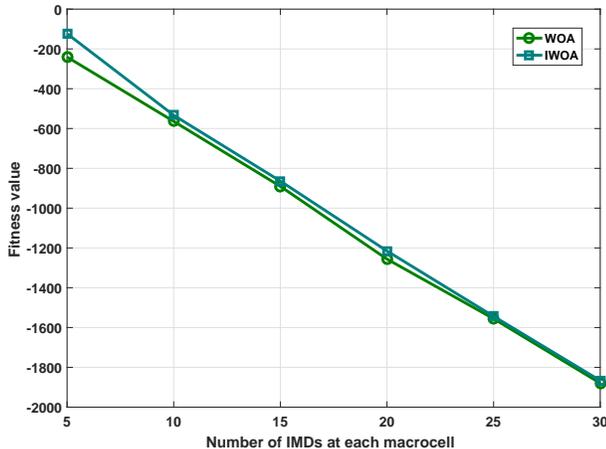


Fig. 6. Impacts of the number of IMDs at each macrocell on fitness value.

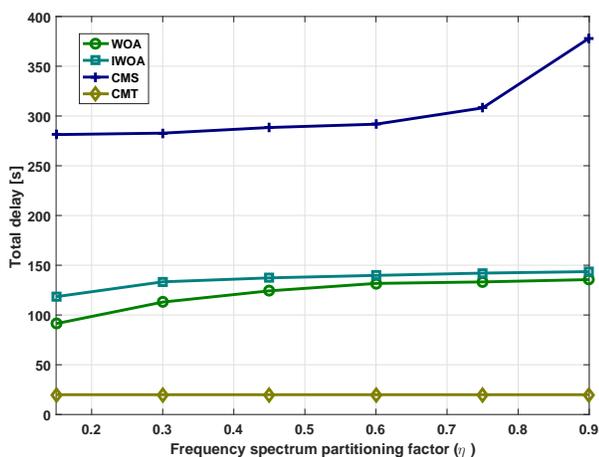


Fig. 7. Impacts of frequency spectrum partitioning factor on total task delay.

BSs with the best channel gains. When the number of IMDs at each macrocell increases, loads of these BSs are getting heavier, resulting in the latency constraints of more and more IMDs can not be guaranteed.

Fig.6 shows the impacts of the number of IMDs at each macrocell on the fitness (function) value. As illustrated in Fig.6, the fitness values of WOA and IWOA may decrease with the number of IMDs at each macrocell. Such a performance trend can be easily inferred according to the definition of fitness value in (15).

Fig.7 shows the impacts of frequency spectrum partitioning factor η on total task delay. As illustrated in Fig.7, besides CMT, the total task delay of other algorithms may increase with η . Since CMT doesn't utilize the uplink frequency spectrum, the total task delay of CMT should not change with η . It is easy to find that the number of NOMA channels decreases with η . Consequently, co-channel interferences may become severer and severer, resulting in increasing task delay in WOA, IWOA and CMS. Significantly, in the simulation, we find that the number of IMDs associated with SBSs in

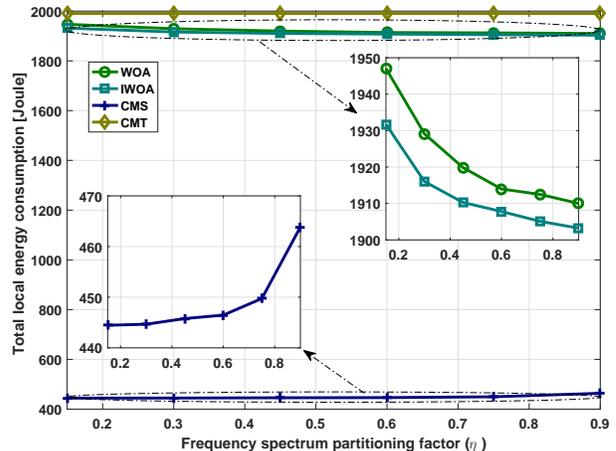


Fig. 8. Impacts of frequency spectrum partitioning factor on total local energy consumption.

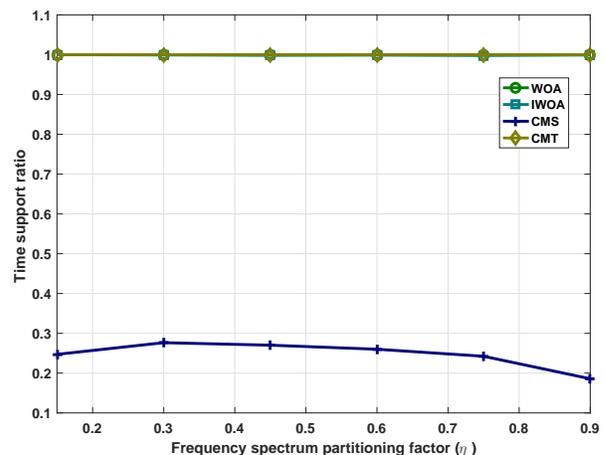


Fig. 9. Impacts of frequency spectrum partitioning factor on time support ratio.

CMS is distinctly greater than the one in WOA and IWOA. It means that the total task delay of CMS may increase with η constantly. However, the total task delay of WOA and IWOA may initially increase with η but then doesn't change with it.

Fig.8 shows the impacts of the frequency spectrum partitioning factor η on total local energy consumption. As illustrated in Fig.8, the total local energy consumption of CMT doesn't change with η since it has no relation to such a factor. The total local energy consumption of WOA and IWOA may decrease with η since the spectrum resources of MBSs selected by most IMDs increase. However, the total local energy consumption of CMS may increase with η since the spectrum resources of SBSs selected by a lot of IMDs decrease and IMDs served by these SBSs receive severer and severer co-channel interferences. Significantly, the opposite performance trend between CMS and whale optimization algorithms may be tightly dependent on the association results of IMDs.

Fig.9 shows the impacts of the frequency spectrum partitioning factor on the time support ratio. As illustrated in

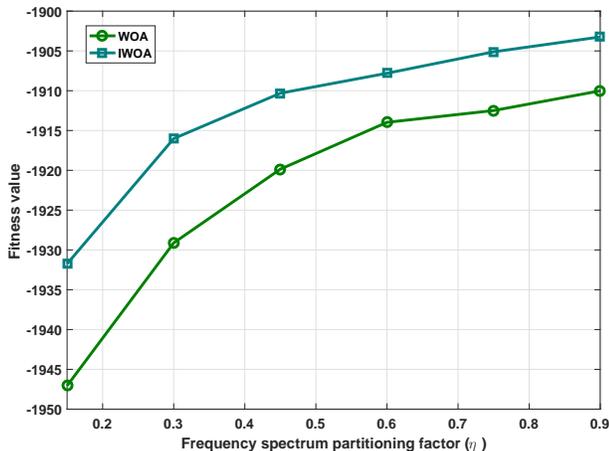


Fig. 10. Impacts of frequency spectrum partitioning factor on fitness value.

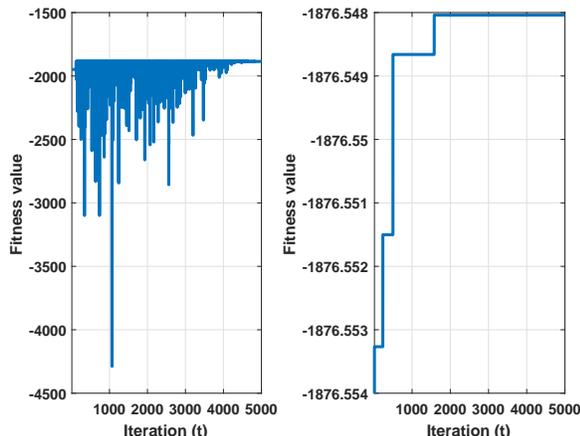


Fig. 11. Convergence comparison of WOA and IWOA.

Fig.9, WOA, IWOA and CMT almost certainly meet the latency constraints of all IMDs. As revealed in Fig.5, the latency constraints of all IMDs in WOA, IWOA and CMT can be guaranteed strictly. Unlike other algorithms, the time support ratio of CMS may initially increase with η but then decrease with it. In CMS, most IMDs are associated with MBSs according to the best gain association. An increased η results in increased uplink data rates of IMDs associated with MBSs, resulting in an increased time support ratio. However, an increased η also results in decreased uplink data rates of IMDs associated with SBSs because of fewer spectrum resources and severer co-channel interferences. It may result in a decreased time support ratio.

Fig.10 shows the impacts of the frequency spectrum partitioning factor on the fitness (function) value. As illustrated in Fig.10, the fitness values of WOA and IWOA may increase with η . Seen from Fig.8, the total local energy consumption of WOA and IWOA decreases with η . According to the definition of fitness value in (15), we can easily know that decreased total local energy consumption may result in increased fitness value.

Fig.11 shows the convergence of WOA and IWOA. As illustrated in Fig.11, IWOA has a higher convergence rate than WOA. In addition, the former can achieve higher fitness value than the latter. Evidently, by considering the historically best agent, Cauchy's inverse cumulative distribution function used for the search of prey, and the search of prey in the nearby area of the historically best agent, IWOA may achieve better performance than WOA.

VI. CONCLUSION

As for ultra-dense multi-task IoT networks, both OMA and NOMA are first used to mitigate network interferences and improve spectrum utilization. Then, under the proportional allocation of computational resources and the constraints of latency and security cost, we jointly optimize device association, channel selection, security service assignment, power control and multi-step computation offloading to minimize the total energy consumption of all IMDs. Considering that the finally formulated problem is in a nonlinear mixed-integer form and hard to tackle, we design IWOA to solve it. After that, the convergence, computational complexity and parallel implementation are analyzed in detail. Simulation results show that IWOA may achieve lower energy consumption than other existing algorithms under the constraints of latency and security cost. Future work can include further improvement of IWOA, and the application of data compression and other intelligent algorithms.

REFERENCES

- [1] J. Zhao, X. Sun, Q. Li, *et al.*, "Edge caching and computation management for real-time internet of vehicles: an online and distributed approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2183-2197, Apr. 2021.
- [2] L. Qian, W. Wu, W. Lu, *et al.*, "Secrecy-based energy-efficient mobile edge computing via cooperative non-orthogonal multiple access transmission," *IEEE Trans. Commun.*, vol.69, no.7, pp. 4659-4677, Jul. 2021.
- [3] J. Zhao, Q. Li, Y. Gong, *et al.*, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol.68, no.8, pp. 7944-7956, Aug. 2019.
- [4] F. Li, H. Yao, J. Du, *et al.*, "Auction design for edge computation offloading in SDN-based ultra dense networks," *IEEE Trans. Mob. Comput.*, vol.21, no.5, pp. 1580-1595, May 2022.
- [5] R. Zhang, P. Cheng, Z. Chen, *et al.*, "Calibrated bandit learning for decentralized task offloading in ultra-dense networks," *IEEE Trans. Commun.*, vol.70, no.4, pp. 2547-2560, Apr. 2022.
- [6] Q. Zhu, X. Wang, and Z. Qian, "Energy-efficient small cell cooperation in ultra-dense heterogeneous networks," *IEEE Commun. Lett.*, vol.23, no.9, pp. 1648-1651, Sep. 2019.
- [7] J. Zhao, S. Ni, L. Yang, *et al.*, "Multiband cooperation for 5G HetNets: a promising network paradigm," *IEEE Veh. Technol. Mag.*, vol.14, no.4, pp. 85-93, Dec. 2019.
- [8] T. Zhou, D. Qin, X. Nie, *et al.*, "Energy-efficient computation offloading and resource management in ultradense heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol.70, no.12, pp. 13101-13114, Dec. 2021.
- [9] F. Guo, H. Zhang, H. Ji, *et al.*, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2651-2664, Dec. 2018.
- [10] Y. Li, Y. Wu, M. Dai, *et al.*, "Hybrid NOMA-FDMA assisted dual computation offloading: a latency minimization approach," *IEEE Trans. Netw. Sci. Eng.*, vol.9, no.5, pp. 3345-3360, Sep. 2022.
- [11] L. Li, Q. Cheng, X. Tang, *et al.*, "Resource allocation for NOMA-MEC systems in ultra-dense networks: a learning aided mean-field game approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1487-1500, Mar. 2021.

- [12] Y. Lu, X. Chen, Y. Zhang, *et al.*, "Cost-efficient resources scheduling for mobile edge computing in ultra-dense networks," *IEEE Trans. Netw. Serv. Manage.*, vol.19, no.3, pp. 3163-3173, Sep. 2022.
- [13] M. H. Chen, B. Liang, and M. Dong, "Multi-user multi-task offloading and resource allocation in mobile cloud systems," *IEEE Trans. Wireless Commun.*, vol.17, no.10, pp. 6790-6805, Oct. 2018.
- [14] M. Sun, X. Xu, X. Tao, *et al.*, "Large-scale user-assisted multi-task online offloading for latency reduction in D2D-enabled heterogeneous networks," *IEEE Trans. Netw. Sci. Eng.*, vol.7, no.4, pp. 2456-2467, Oct. 2020.
- [15] Y. Wu, B. Shi, L. P. Qian, *et al.*, "Energy-efficient multi-task multi-access computation offloading via NOMA transmission for IoTs," *IEEE Trans. Ind. Inf.*, vol.16, no.7, pp. 4811-4822, Jul. 2020.
- [16] H. Zhang, Y. Yang, X. Huang, *et al.*, "Ultra-low latency multi-task offloading in mobile edge computing," *IEEE Access*, vol.9, pp. 32569-32581, Feb. 2021.
- [17] J. Chen, Y. Yang, C. Wang, *et al.*, "Multitask offloading strategy optimization based on directed acyclic graphs for edge computing," *IEEE Internet Things J.*, vol.9, no.12, pp. 9367-9378, Jun. 2022.
- [18] J. Bi, H. Yuan, K. Zhang, *et al.*, "Energy-minimized partial computation offloading for delay-sensitive applications in heterogeneous edge networks," *IEEE Trans. Emerg. Topics Comput.*, vol.10, no.4, pp. 1941-1954, Oct. 2022.
- [19] H. Tang, H. Wu, Y. Zhao, *et al.*, "Joint computation offloading and resource allocation under task-overflowed situations in mobile-edge computing," *IEEE Trans. Netw. Serv. Manage.*, vol.19, no.2, pp. 1539-1553, Jun. 2022.
- [20] Y. Dai, D. Xu, S. Maharjan, *et al.*, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Trans. Veh. Technol.*, vol.67, no.12, pp. 12313-12325, Dec. 2018.
- [21] T. Zhou, Y. Yue, D. Qin, *et al.*, "Joint device association, resource allocation, and computation offloading in ultra-dense multi-device and multi-task IoT networks," *IEEE Internet Things J.*, vol.9, no.19, pp. 18695-18709, Oct. 2022.
- [22] H. Zhang, Y. Yang, B. Shang, *et al.*, "Joint resource allocation and multi-part collaborative task offloading in MEC systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8877-8890, Aug. 2022.
- [23] S. Han, X. Xu, S. Fan, *et al.*, "Energy efficient secure computation offloading in NOMA-based mMTC networks for IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5674-5690, Jun. 2019.
- [24] X. He, R. Jin, H. Dai, *et al.*, "Physical-layer assisted secure offloading in mobile-edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4054-4066, Jun. 2020.
- [25] J. Wang, H. Yang, M. Cheng, *et al.*, "Joint optimization of offloading and resources allocation in secure mobile edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8843-8854, Aug. 2020.
- [26] W. Wu, F. Zhou, R. Q. Hu, *et al.*, "Energy-efficient resource allocation for secure NOMA-enabled mobile edge computing networks," *IEEE Trans Commun.*, vol. 68, no. 1, pp. 493-505, Jan. 2020.
- [27] Y. Bai, L. Chen, L. Song, *et al.*, "Risk-aware edge computation offloading using bayesian stackelberg Game," *IEEE Trans. Netw. Serv. Manage.*, vol. 17, no. 2, pp. 1000-1012, Jun. 2020.
- [28] S. Liu, Y. Yao, L. Guo, *et al.*, "Satisfaction-maximized secure computation offloading in multi-eavesdropper MEC networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4227-4241, Jun. 2022.
- [29] I. A. Elgendy, W. Zhang, Y.-C. Tian, *et al.*, "Resource allocation and computation offloading with data security for mobile edge computing," *Future Gener. Comput. Syst.*, vol. 100, pp. 531-541, Nov. 2019.
- [30] I. A. Elgendy, W. -Z. Zhang, Y. Zeng, *et al.*, "Efficient and secure multi-user multi-task computation offloading for mobile-edge computing in mobile IoT networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 17, no. 4, pp. 2410-2422, Dec. 2020.
- [31] I. A. Elgendy, W. -Z. Zhang, C. -Y. Liu, *et al.*, "An efficient and secured framework for mobile cloud computing," *IEEE Trans. Cloud Comput.*, vol. 9, no. 1, pp. 79-87, 1 Jan.-Mar. 2021.
- [32] M. I. A. Zahed, I. Ahmad, D. Habibi, *et al.*, "Green and secure computation offloading for cache-enabled IoT networks," *IEEE Access.*, vol. 8, pp. 63840-63855, 2020.
- [33] W. Jiang, K. Jiang, X. Zhang, *et al.*, "Energy optimization of security-critical real-time applications with guaranteed security protection," *Journal Syst. Arch.*, vol. 61, no. 7, pp. 282-292, Aug. 2015.
- [34] Q. V. Pham, S. Mirjalili, N. Kumar, *et al.*, "Whale optimization algorithm with applications to resource allocation in wireless networks," *IEEE Trans. Veh. Technol.*, vol.69, no.4, pp. 4285-4297, Apr. 2020.
- [35] Z. Guo, P. Wang, Y. Ma, *et al.*, "Whale optimization algorithm based on adaptive weight and Cauchy mutation," *Microelect. Comput.*, vol. 34, no. 9, pp. 20-25, Sep. 2017.
- [36] Z. Wu, Y. Mu, "Improved whale optimization algorithm," *Appl. research Comput.*, vol.37, no. 12, pp. 3618-3621, Dec. 2020.
- [37] L. Liu, Ke Bai, Z. Dan, *et al.*, "Whale optimization algorithm with global search strategy," *J. Chinese Comput. Syst.*, vol. 41, no. 9, pp. 1820-1825, Sep. 2020.
- [38] Y. Zhang, Y. Liu, J. L. Zhou, *et al.*, "Slow-movement particle swarm optimization algorithms for scheduling security-critical tasks in resource-limited mobile edge computing", *Future Gener. Comput. Syst.*, vol. 112, pp. 148-161, Nov. 2020.