Wireless Communications and Mobile Computing

Artificial Intelligence-Empowered Resource Orchestration for Quality of Service Provisioning in 6G

Lead Guest Editor: Wen Wu
Guest Editors: Qiang Ye, Ying Chen, Ning Zhang, and Muhammad Ismail

Artificial Intelligence-Empowered Resource Orchestration for Quality of Service Provisioning in 6G

Wireless Communications and Mobile Computing

Artificial Intelligence-Empowered Resource Orchestration for Quality of Service Provisioning in 6G

Lead Guest Editor: Wen Wu
Guest Editors: Qiang Ye, Ying Chen, Ning Zhang, and Muhammad Ismail

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3646272, 13 pages
https://doi.org/10.1155/2022/3646272

Research Article

Pillar-Based Cooperative Perception from Point Clouds for 6G-Enabled Cooperative Autonomous Vehicles

Jian Wang,[1] **Xinyu Guo**,[1] **Hongduo Wang**,[1] **Pin Jiang**,[2] **Tengyun Chen**,[2] **and Zemin Sun**[1]

[1]*College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[2]*Wireless X Labs, Huawei Technologies, Shenzhen 518000, China*

Correspondence should be addressed to Jian Wang; wangjian591@jlu.edu.cn

Received 8 April 2022; Accepted 7 July 2022; Published 25 July 2022

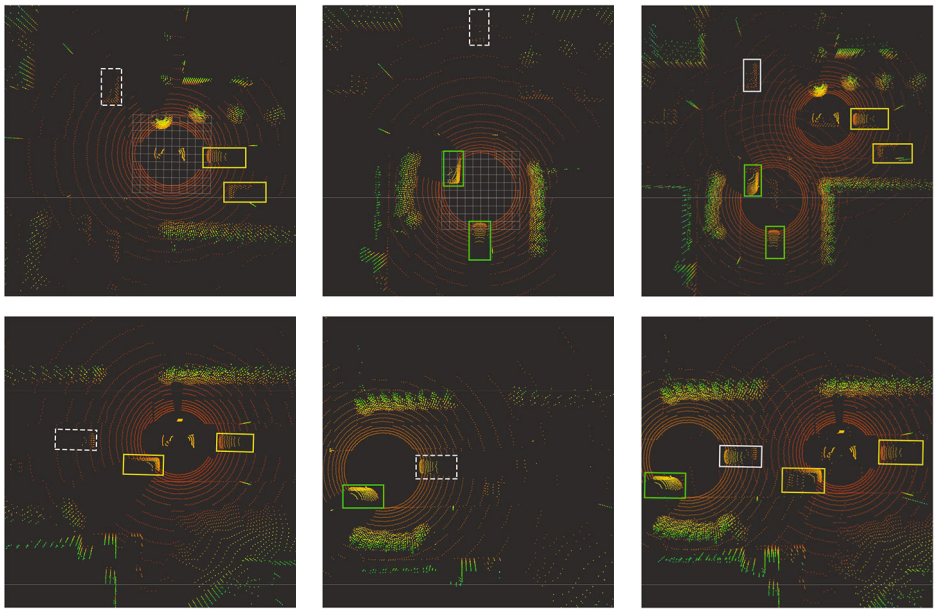
Academic Editor: Ying Chen

3D object detection is a significant aspect of the perception module in autonomous driving; however, with current technology, data sharing between vehicles and cloud servers for cooperative 3D object detection under the strict latency requirement is limited by the communication bandwidth. The sixth-generation (6G) networks have accelerated the transmission rate of the sensor data significantly with extreme low-latency and high-speed data transmission. However, which sensor data format and when to transmit it are still challenging. To address these issues, this study proposes a cooperative perception framework combined with a pillar-based encoder and Octomap-based compression at edges for connected autonomous vehicles to reduce the amount of missing detection in blind spots and further distances. This approach satisfies the constraints on the accuracy of the task perception and provides drivers or autonomous vehicles with sufficient reaction time by applying fixed encoders to learn a representation of point clouds (LiDAR sensor data). Extensive experiment results show that the proposed approach outperforms the previous cooperative perception schemes running at 30 Hz, and the accuracy of the object bounding box results in further distances (greater than 12 m). Furthermore, this approach achieves a lower total delay for the procession of the fusion data and the transmission of the cooperative perception message. To the best of our knowledge, this study is the first to introduce a pillar-based encoder and Octomap-based compression framework for cooperative perception between vehicles and edges in connected autonomous driving.

1. Introduction

Object perception is indispensable in sensing the surrounding environment and improving driving safety in autonomous vehicles (AVs). However, the perception capability of onboard sensors on AVs could be influenced by external factors such as obstacles and weather conditions [1]. To address this issue, European Telecommunications Standards Institute (ETSI) proposes the generation rules of Cooperative Perception Messages (CPMs) [2], which specify that CAVs can generate CPMs for cooperative perception when vehicles receive the object results from the object detection algorithm using their sensor data. As a result, CPMs improve the perception capability of connected autonomous vehicles (CAVs) by expanding the effective sensing range and reducing the missed detections [3].

Nevertheless, CAVs still cannot perceive the missed vehicles, which are not detected by local sensor data, because the content of CPMs only includes the object results. Figure 1 is the comparison between CPMs and fusion sensor data detection. There are two cases to explain the undetected object in which each vehicle can only perceive a sparse point cloud. The above pictures are the situation at an intersection, and the below pictures are for bidirected roads. Figure 1(a) denotes the object detection result (yellow bounding boxes) from vehicle *A*. Also, Figure 1(b) represents the result (green bounding boxes) from vehicle *B*. Finally, the fusion result shows that the undetected vehicle (white boxes) can be detected in Figure 1(c).



(a) (b) (c)

Figure 1: Illustration of the detection results of CPMs and fusion sensor data detection.

Cooper [4] is the basic idea of cooperative perception to transmit CPMs with LiDAR data from multiple CAVs and point cloud fusion for detecting 3D objects. However, transmitting raw sensor data (such as point cloud) is challenging even though the sixth-generation (6G) communication technology. Significantly, the data size of 1 frame of LiDAR sensor with 64 beams can be about 3-4 MB, and the sample rate of typical LiDAR is 30 Hz. Therefore, the network capacity is at least 960 Mb/s. Besides, the raw point cloud is a sparse representation due to the measuring method by the laser scanner. As a result, the transmission of raw sensor data is inefficient and wastes limited communication resources.

To overcome the above challenges, this work proposes a cooperative perception scheme integrating pillar-based encoder sensor data and Octomap-based compression. To the best of our knowledge, this pillar-based fusion for cooperative perception at edges is the first study that tackles the problem of missing object detection. Furthermore, the proposed solution reduces the communication delay and improves the perception capabilities of CAVs. The main contributions of this study can be summarized as follows. First, this study proposes a new data transmission type to exchange cooperative perception messages between CAVs and edges for exploring the potential of "intermediate outputs" in 3D object detection algorithms. Second, this detection scheme performance is achieved while running at 30 Hz, and the precision of distant vehicles (more than 20 m) is improved compared with the feature map fusion solution [5]. Therefore, autonomous vehicles have sufficient reaction time to handle traffic emergencies. Finally, the scheme running in the edges does not require a specific onboard object detection algorithm. In other words, CAVs do not rely on object results from the edges. The proposed solution can be seen as enhancing the perception capability of CAVs by 6G communication.

The remainder of this work is organized as follows. Section 2 presents the preliminaries and background. Section 3 introduces the proposed object perception scheme on CAVs, including the phase to transmit the specific cooperative perception messages and receive the semantic and precise object results. Then, this study designs a pillar-based fusion algorithm on edges to detect cooperative perception results. Finally, we propose an integrating CAV-based and edge-based scheme for cooperative perception. Simulation results and conclusions are given in Section 4 and Section 5.

2. Preliminaries and Background

Given state of the art in the field of 3D object detection in autonomous driving, this work starts by briefly reviewing the data choice, including data type, fusion level [6], compression, and reconstruction in general. And then this study focuses on algorithms specific to object detection from LiDAR point clouds. After that, this study describes the outlook and work done in the scheme for cooperative perception.

2.1. Data Choice

2.1.1. Type. Rapid development of 3D object detection has motivated increasing studies to design efficient representations to detect vehicles in point clouds and images. Images can provide rich color properties and detailed texture information. Numerous research deduced the 3D object detection in image [7–11] benefits from nature 2D image-based object detection technology. At first, the 2D bounding boxes are

obtained by processing images in these algorithms. After that, the bounding boxes are converted from 2D to 3D based on different methods, such as template matching [7, 8, 12, 13] and geometric properties [14–16]. However, images lack the depth information and are susceptible to lighting conditions. The accuracy of detection algorithms is bounded by the capability of the depth estimation and light compensation. Point clouds can provide accurate 3D position, speed, and depth information of objects and have advantages of spatial dimension over 2D images [4]. LiDAR is less affected by the environment; it can work even in dark or bad weather and generate real-time, high-resolution point clouds of the surrounding environment. Besides, the point cloud representation (x, y, z, r), indicating each spatial position information and reflectivity, is easy to process. Therefore, this work is dedicated to exchanging data from point clouds.

2.1.2. Fusion Level. This study uses the classification of sensor data fusion defined in [6]: low-level, feature-level, and high-level. Briefly, low-level fusion is raw data from the onboard sensor without processing. Although the low-level fusion method keeps the original spatial information of objects, the transmission of low-level fusion requires ultra-high bandwidth and is hard to apply in the 6G network. On the other hand, feature-level fusion seeks a preprocessing output from raw data before getting the object results. Hence, it can reduce the data size and keep the main features. Finally, high-level fusion means onboard sensors independently process raw data to generate an object list and fuse all the lists. Accordingly, it is less complex to implement in practice, but it cannot detect the missed objects which never detected before fusion. Considering the advantages and disadvantages of different levels fusion, this study pays attention to the feature-level fusion.

2.1.3. Compression. The point cloud provides a highly realistic view of the surrounding environment; it also costs large bandwidth to transmit uncompressed data. Therefore, compression of point clouds is necessary. The function proposed in current research can be summarized into three modes: 1D traversal, 2D projection, and 3D correlations [17]. As for 1D prediction methods, it is to construct a prediction tree to convert the geometry data into a 1D representation. Although this model is relatively simple to achieve, it does not consider the 3D spatiality of point clouds. 2D projection focuses on converting the 3D point cloud into 2D representation by projection or mapping [18]. Because of the 3D correlations of point clouds, the most common way to compress is to convert the point clouds to 3D representation [17]. Therefore, the 3D representation is used in this study. There are various approaches to achieve this purpose, such as octree-based coder [19], hierarchical clustering coder [20], and context-based intracoder [21]. The octree-basedcoder is adopted by this work because the lossless encoding is suitable for fusion.

2.1.4. Reconstruction. The received data cannot be processed straightforwardly because it is generated on different positions and angles. Thus, edges need to reconstruct the exchanging point cloud data into its UTM (Universal Transverse Mercator) coordinate system. In order to merge point clouds, cooperative perception messages need GPS and IMU (Inertial Measurement Unit) as additional information to calculate the offset information (Euler angles, unit quaternions, and rotation vectors [22]). Euler angles represent a rotation with yaw, pitch, and roll angles for $z-$, $y-$ and $x-$ axis that are θ, φ, and ψ, respectively. It can be calculated by using the IMU value difference between the transmitter and receiver. Besides, the quaternions can be represented as follows:

$$q = q_w + q_x i + q_y j + q_z k. \tag{1}$$

Given Euler angles θ, φ, and ψ, the quaternions can be calculated as follows:

$$\begin{aligned}
q_w &= \cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)\cos\left(\frac{\varphi}{2}\right) + \sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)\sin\left(\frac{\varphi}{2}\right),\\
q_x &= \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)\cos\left(\frac{\varphi}{2}\right) - \sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)\sin\left(\frac{\varphi}{2}\right),\\
q_y &= \cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)\sin\left(\frac{\varphi}{2}\right) + \sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)\cos\left(\frac{\varphi}{2}\right),\\
q_z &= \sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)\cos\left(\frac{\varphi}{2}\right) - \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)\sin\left(\frac{\varphi}{2}\right).
\end{aligned} \tag{2}$$

Subsequently, the rotation matrix $\mathbf{R}$ can be written as follows:

$$\mathbf{R} = \begin{bmatrix} 1 - 2\left(q_y^2 + q_z^2\right) & 2\left(q_x q_y - q_w q_z\right) & 2\left(q_w q_y + q_x q_z\right) \\ 2\left(q_x q_y + q_w q_z\right) & 1 - 2\left(q_x^2 + q_z^2\right) & 2\left(q_y q_z - q_w q_x\right) \\ 2\left(q_x q_z - q_w q_y\right) & 2\left(q_w q_x + q_y q_z\right) & 1 - 2\left(q_x^2 + q_y^2\right) \end{bmatrix}. \tag{3}$$

Finally, the coordinate of point clouds can be calculated from different sensors as follows:

$$\mathbf{P}_{\mathbf{transform}} = \mathbf{R} \cdot \mathbf{P}_{\mathbf{original}}, \tag{4}$$

where P_{original} is the original coordinates from its sensor, and $P_{\text{transform}}$ is the coordinate transformed to the UTM system.

2.2. 3D Object Detection in Point Clouds. A 3D object detection algorithm is divided into three parts: data representation, feature extraction (i.e., backbone), and detection network (detection head). Data representation organizes the point cloud generated by LiDAR into a proper structure to which the convolution operations can be applied. The main approaches are voxel-based, point-based, frustum-based, pillar-based, and 2D projection-based [23]. First, voxel is short for volume pixel, dividing point clouds into 3D grids at a specific resolution in a 3D Cartesian coordinate

system. Many voxel-based methods are proposed in the earlier studies such as VoxelNet [24], SECOND [25], and Voxel-FPN [26]. The advantage is that it can easily migrate neural network operations such as convolution based on 3D grids. The disadvantage is that the efficiency is low due to a large amount of discrete computation and encoder. Besides, point-based methods process the raw data and generate a sparse representation and then aggregate the features of adjacent points, and the feature of each point is extracted [23]. However, this method poses stringent requirements on the hardware compare with the voxel-based method. Pillar-based method organizes point clouds in the x-y plane; since the point cloud is three-dimensional, point clouds are formed into vertical columns, called pillars, such as PointPillars [27]. Due to the exclusion of the z-axis and fixed encoder, the pillar-based method has a significant processing rate improvement compared to other algorithms, and it also achieves top efficiency and performance on recent 3D object detection benchmarks. Therefore, this study chooses the pillar-based method as data representation due to its speed and accuracy. The convolutional neural network (CNN) architectures [28] as feature extraction can be seen as the state of art in images and point clouds. Besides, the detection head used in this work, as the layer to compensate for CNN positioning capabilities, is Single Shot Detector proposed in [29].

2.3. Current Cooperative Perception Schemes Proposed. The benefits of cooperative perception have been studied in recent years [30–33]. Motivated by its potential, many researches focused on several areas, including communication protocol [34], generation rules of CPMs [1], and algorithm design. In CarSpeak [34], a content-centric communication protocol around the needs of cooperative perception has been proposed. It focuses on sensor information sharing at the medium access control (MAC) layer and is fully integrated with autonomous driving. On the other hand, Thandavarayan et al. [1, 3] has analyzed the performance of cooperative perception messages and offered generation rules to define which information should be included. In algorithm design, Chen et al. [4, 5] and Guo et al. [35] have proposed a series of feature map fusion methods on CAVs by vehicle-to-vehicle (V2V) communication. Aoki et al. [32] proposed the scheme with deep reinforcement learning (DRL) to select the data to transmit. Besides, the infrastructure sensor-supported cooperative perception has been proposed in [36, 37]. However, most of these studies only rely on the sensor data from assist CAVs, which cannot operate independently with local data. 3D object detection as a kind of computation-intensive tasks generated by CAVs is limited by battery and computing capacity. Therefore, some works focus on dynamic task offloading for mobile edge computing [38, 39]. The collaborative task offloading for vehicles and cloud faces challenges by the increase of the scales of task offloading problem and solution space size. These studies are meaningful and significant but beyond the scope of this paper. Distinguished from the previous works, this study focuses on the algorithm on edges to enhance the object detection of missed objects.

3. Pillar-Based Cooperative Perception from Point Clouds

Inspired by the advantages of the fixed encoder such as PointPillars [27] and Octomap compression [40], this study proposes the pillar-based cooperative perception framework from point clouds on CAVs and edges. This study presents two schemes for cooperative perception: Cooperative Perception on Vehicles (CPV) and Cooperative Perception on Edges (CPE). This section presents the detail and architecture framework of two schemes.

3.1. Cooperative Perception on Vehicles. The proposed architecture of CPV, depicted in Figure 2, consists of three components: Octomapcompression, pillarupload, and listfusion. These three components are executed sequentially. Besides, pillarupload and listfusion start to execute when the vehicles receive corresponding messages from edges. It should be noted that the 3D object detection algorithm in this work is PointPillars [27]. However, it does not mean that CPVs only rely on PointPillars. The detection algorithm can be replaced by others. At first, the vehicle process downsampled 3D LiDAR point cloud represented as (x, y, z, r) with 3D position and reflection value r into Octomap [40]. Octomap based on tree structure uses probabilistic occupancy estimates to support lossless compression. Specifically, CAVs convert the range of point clouds into a cube space with side length L. This space will be divided recursively into eight subvoxels until the minimum voxel size is r_o. The leaf node in the Octomap only stores the occupancy probability and one pointer of an inner node with eight pointers to its children or a null pointer (no child node), as shown in Figure 3. The occupancy probability of the node is the maximum occupancy of all the eight subvoxels as follows:

$$O(n) = \max_i o(n_i), i = 1, \cdots, 8. \tag{5}$$

After that, CAVs transmit the Octomap data to edges. Then, the CPVs execute the next component when it has received cuboid regions (how to generate cuboid regions on edges will detail in Section 3.2) from edges. Then, CAVs discretize the points into pillars P, which is no control in the z-dimension based on the cuboid regions. The point p in the pillar represents with nine dimensions as follows:

$$p = \left(x, y, z, r, x_c, y_c, z_c, x_p, y_p\right), \tag{6}$$

where c denotes the center point with distance to the mean of all points, and the p denotes the offset from the center point. Besides, if there are too many points in a pillar, this scheme randomly discards the points. Conversely, the zero points pad into the pillars with less than N points. Once the generation of pillars is finished, CAVs transmit the feature pillars to edges with their locations.

Finally, CAVs receive the object list L_{cloud} from edges. Vehicles execute the listfusion component to obtain final 3D bounding box of objects around itself. The detail of fusion algorithm based on weightbox is shown in Algorithm 1.

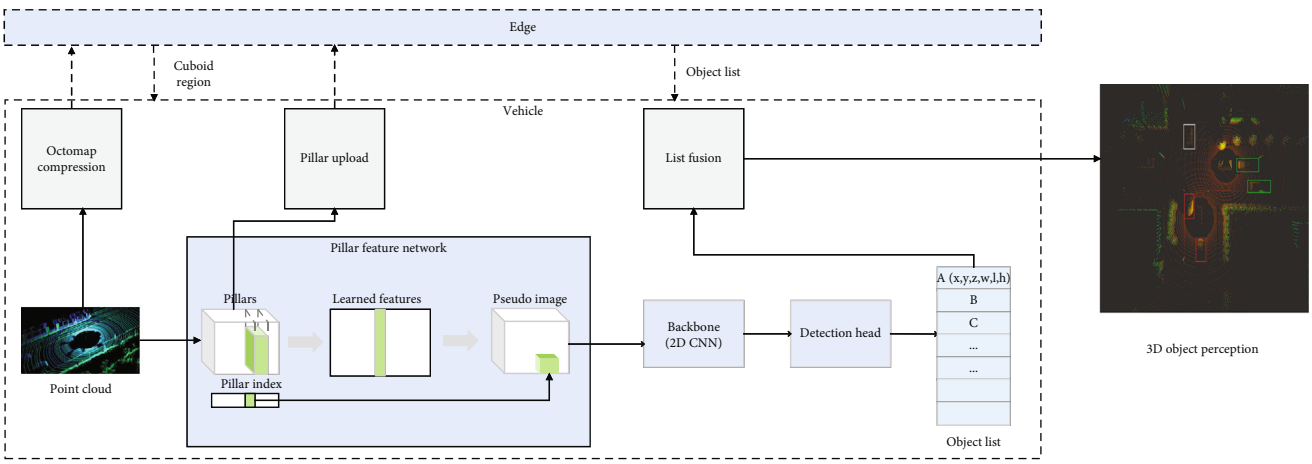


Figure 2: Architecture of the Cooperative Perception on Vehicles (CPV).

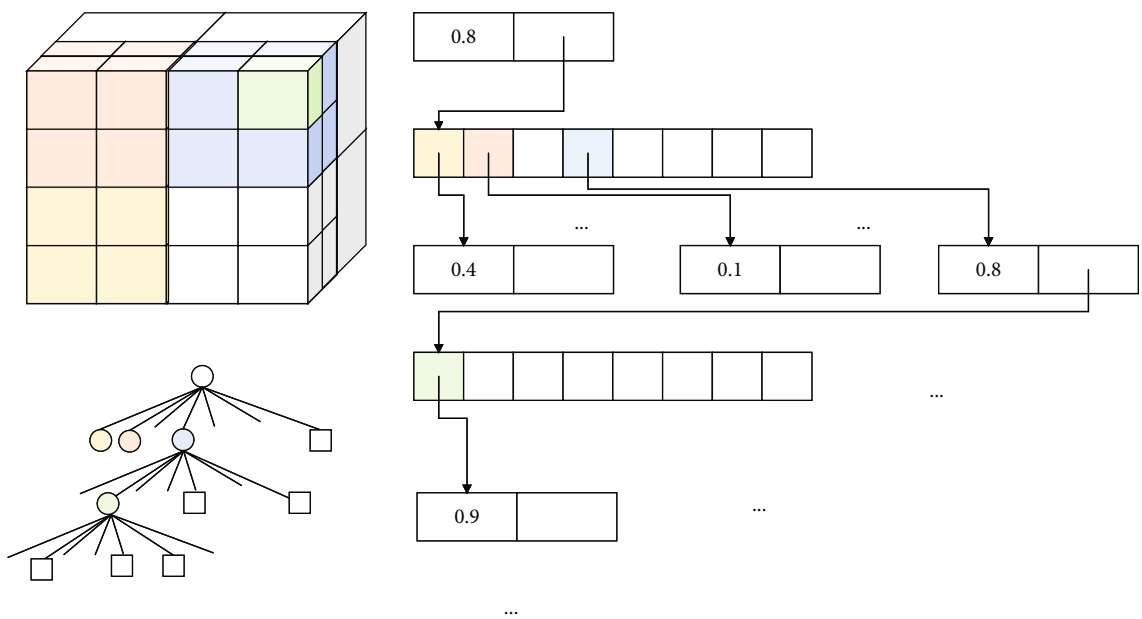


Figure 3: Illustration of an Octomap storing tree, including the model and corresponding tree representation and the structure of a leaf node.

In the algorithm, vehicles first initialize the parameters such as groups that are the same object candidates M and the threshold of IOU (Intersection over Union) Thr. Then, the proposed algorithm removes the points out of range R_{vehicle}. Secondly, this algorithm filters out the candidates of a 3D bounding box for the same object and stores the candidates in a group of the list M. Therefore, M includes multiple groups. Finally, the algorithm outputs the object list L_{final} after processing the list M by weighted average in the same index. This algorithm can effectively reduce the maintenance detection list because these out-of-range bounding boxes have useless value for the vehicle.

3.2. Cooperative Perception on Edges. There are four main components: Octomapfusion, cuboidregionextraction, pillar fusion, and objectdetection. These components are executed sequentially. It should be noted that the 3D object detection algorithm is based on PointPillars [27] in this work. Besides, Octomapfusion and pillarfusion can be executed when the edges have received corresponding basic messages from vehicles.

3.2.1. Octomap Fusion. The CPE executes Octomap fusion component as shown in Figure 4 when the edge receives messages from CAVs, including Octomap and the locations of vehicles. At first, all the Octomaps should transfer into the UTM coordinate system by using data reconstruction mentioned in Section 2.1.4. After that, there are four conditions: (1) unknown, (2) samedepth, (3) coarse, and (4) finer, shown in Figure 5, to merge the leaf node n into final map T_{fusion} from one Octomap T_i due to the hierarchical tree-structure. If the leaf node in T_i is not constructed on the T_{fusion} (the state is unknown), the Octomap adds the node into T_{fusion}, as follows:

Data: Object list from cloud L_{cloud} and object list from vehicle $L_{vehicle}$
Result: Object list L_{final}
List M, $R_{vehicle}$, Thr, α, β initialization
/*Remove the objects which are out of range of the vehicle */
while index in List L_{cloud} **do**
  if $\|L_{cloud}[index] - Location(Vehicle)\| > 2 * R_{vehicle}$ **then**
    remove $L_{cloud}[index]$
/*Storage the objects in same location to M */
while j in List L_{cloud} **do**
  //Thr is the threshold of IOU
  if IOU $s(L_{cloud}[j], L_{vehicle}) \geq Thr$ **then**
    $M[j(L_{vehicle})][:] = L_{cloud}[j]$
  else
    Insert $(L_{cloud}[j])$ to $L_{vehicle}$
    $M[j(L_{vehicle})][:] = L_{cloud}[j]$
/*Calculate the final bounding box and loction of object in List M */
while k in List M **do**
//Weighted Average
  $L_{vehicle}[k] = \alpha \cdot L_{vehicle}[k] + \beta \cdot \sum_{i=1}^{sum(M[k])} M[k][i]/sum(M[k])$
  $L_{final} = L_{vehicle}$
RETURN L_{final}

Algorithm 1: Object list fusion based on weightbox.

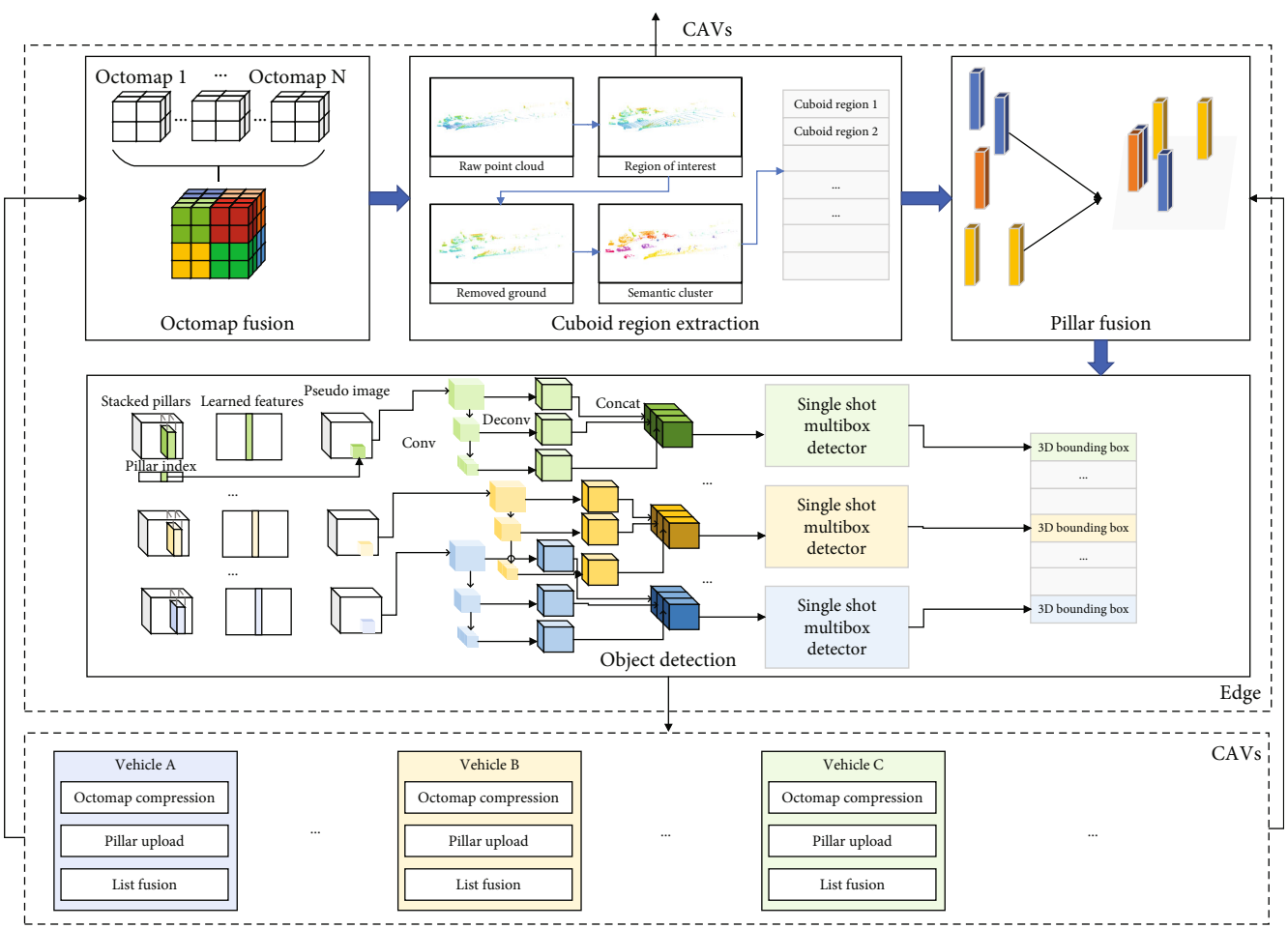


Figure 4: Architecture of the Cooperative Perception on Edges (CPE).

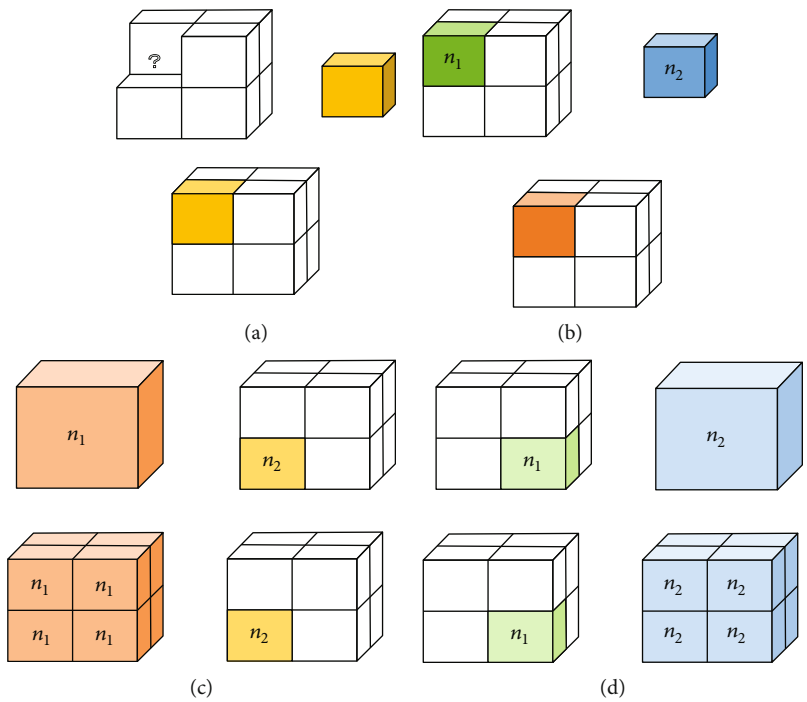


Figure 5: Illustration of an Octomap fusion by different situations, including (a) unknown, (b) same depth, (c) coarse, and (d) finer.

$$L(n_1) = L(n_2), \tag{7}$$

where the logarithmic occupancy probability of n is $L(n)$,and n_1, n_2 denotes the leaf node in T_{fusion} and in T_i, respectively. If $L(n_1)$ and $L(n_2)$ are in samedepth, the probability will be updated, as follows:

$$L(n_1) = L(n_1) + L(n_2). \tag{8}$$

In condition (3), if $L(n_2)$ is represented by coarse resolution node $L(n_1)$, $L(n_1)$ is divided into eight child nodes $L(n_1^i)$ with the same occupied probability and update as follows:

$$\begin{aligned} L(n_1^i) &= L(n_1), i = 1, \cdots, 8, \\ L(n_1^i) &= L(n_1^i) + L(n_2). \end{aligned} \tag{9}$$

Similarly, the update occupied probability is represented in condition (4), as follows:

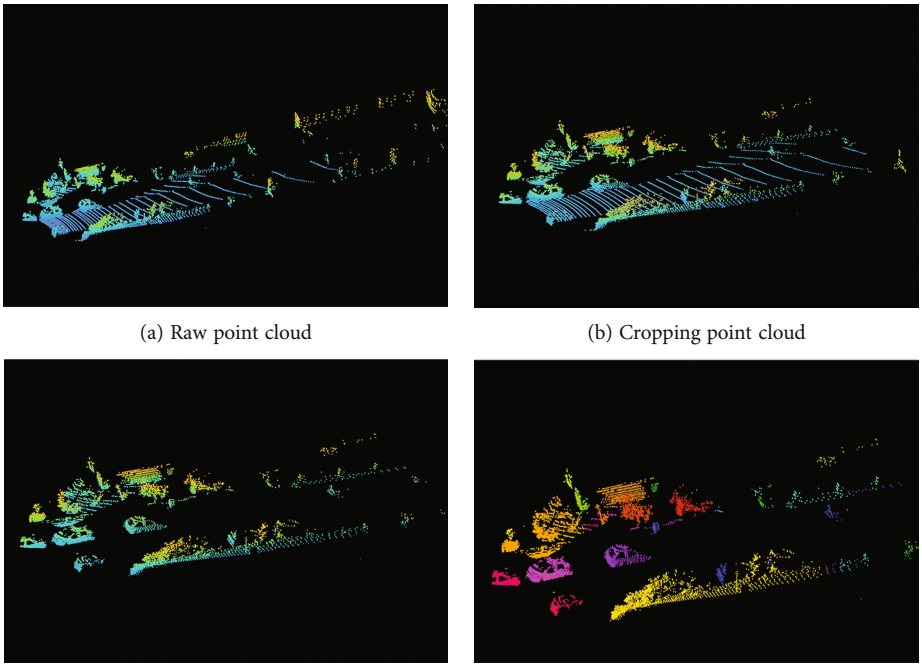
$$\begin{aligned} L(n_2^i) &= L(n_2), \qquad i = 1, \cdots, 8, \\ L(n_1) &= L(n_1) + L(n_2^i). \end{aligned} \tag{10}$$

The final Octomap can be transferred into point clouds C_{fusion} after all the T_i merge into T_{fusion}. The detail of the Octomap fusion algorithm on the edge is shown in Algorithm 2.

3.2.2. Cuboid Region Extraction. The cuboid region extraction will be executed after the edge gets point clouds C_{fusion}. Point clouds C_{fusion} are further classified into clusters by Euclidean, and each cluster is converted into a coarse-grained 3D bounding box. At first, point clouds outside the X_{limit}, Y_{limit}, and

Data: Octomap from Vehicles $T_i, i = 1, \cdots, N$
Result: Octomap T_{fusion}
$T_{fusion}, i = 0$ initialization;
while i $\leq N$ **do**
  i=i+1
  T_i reconstruction based on Eq. (4)
  // Octomap Fusion(T_{fusion}, T_i)
  for Leaf node $L(n_2)$ in T_i **do**
    if $L(n_2)$ not in T_{fusion} **then**
      Add node: $L(n_1) = L(n_2)$
    else
      The depth of $L(n_2)$ is d_2
      for Leaf node $L(n_1)$ in T_{fusion} **do**
        The depth of $L(n_1)$ is d_1
        if $d_2 = d_1$ **then**
          $L(n_1) = L(n_1) + L(n_2)$
          else $d_2 - d_1 = 1$ && $L(n_1).child$ is $NULL$ **then**
            Add child node to T_{fusion}: $L(n_1^i) = L(n_1)$;
            $L(n_1^i) = L(n_1^i) + L(n_2)$
        else if $d_1 - d_2 = 1$ && $L(n_2).child$ is $NULL$ **then**
          Add child node to T_i: $L(n_2^i) = L(n_2)$
          $L(n_1) = L(n_1) + L(n_2^i)$
RETURN T_{fusion}

Algorithm 2: Octomap fusion on the edge.



(a) Raw point cloud (b) Cropping point cloud

(c) Removed ground plane (d) Segment out obstacle

Figure 6: The point clouds of cuboidregionextraction at four different stages: (a) the raw point clouds, (b) the region of interest (RoI) by removing outside points, (c) the points removed ground plane, and (d) the semantic results by using different colors.

Z_{limit} are removed. After that, the ground plane is fitted from the survived points. And the unrecognized obstacle can be clustered by Euclidean because of removing the ground plane. In the end, all the bounding boxes are proposed by clusters. The cuboid region extraction is an adapted version of the traditional and efficient point cloud segmentation algorithm implemented by PCL [41], and its algorithm flowchart is shown in Figure 6.

3.2.3. Pillar Fusion. After vehicles transmit the pillars data, CPE executes *pillarfusion* component on edge as shown in Figure 4. For fixed backbone architecture, this study first feeds the point clouds to pseudoimages. The received pillars based on their locations fuse into different cuboid regions (can also be called as subtasks). In a subtask, there are some pillars in the overlapping area as shown in Figure 4. The approach in this study to fuse these pillars into one subtask can be represented in the following equation:

$$P = \{P_{\text{nol}} \cup P_{\text{ol}} = \{\text{sample}(p_i)\}\}, p_i \in P_A \cap \cdots \cap P_B, \tag{11}$$

where P_{nol} denotes the nonoverlapping areas of subtask in ground plane, P_{ol} randomly samples from all the pillars p_i in the overlapping areas, and P_A, P_B denote the pillars from vehicle A and B.

3.2.4. Object Detection. The fused pillars feed with a linear layer followed by BatchNorm [42], ReLU [43], and max pooling [44] to generate and reshape tensor size (C, W, H). After that, the backbone and detection head are similar to PointPillars [27].

The loss function is based on ground truth boxes and anchors defined by $(x, y, z, w, l, h, \theta)$. The total loss is as follows:

$$\mathscr{L} = \frac{\beta_{\text{cls}}}{N_{\text{cls}}}\mathscr{L}_{\text{cls}} + \frac{\beta_r}{N_r}\left(\mathscr{L}_\theta + \mathscr{L}_{\text{reg}}\right) + \frac{\beta_a}{N_a}\mathscr{L}_a, \tag{12}$$

with $\beta_{\text{cls}} = 1.0$, $\beta_r = 2.0$, and $\beta_a = 0.1$. $\mathscr{L}_{\text{reg}}$ is regression loss for (x, y, z, w, l, h) by Smooth$L1$-loss [45] in the following equation:

$$\mathscr{L}_{\text{reg}} = \sum_{\alpha \in (x,y,z,w,l,h)} L_{1,\text{smooth}}(\Delta\alpha), \tag{13}$$

which

$$\begin{aligned} L_{1,\text{smooth}}(x) &= \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < 1 \\ |x| - \frac{1}{2} & \text{otherwise,} \end{cases} \\ \Delta x &= \frac{x_{gt} - x_a}{\sqrt{\left(w_a{}^2 + l_a{}^2\right)}}, \\ \Delta y &= \frac{y_{gt} - y_a}{\sqrt{\left(w_a{}^2 + l_a{}^2\right)}}, \\ \Delta z &= \frac{z_{gt} - z_a}{h_a}, \\ \Delta w &= \log w_{gt} - \log w_a, \\ \Delta l &= \log l_{gt} - \log l_a, \\ \Delta h &= \log h_{gt} - \log h_a. \end{aligned} \tag{14}$$

The Smooth L1-loss is less sensitive to outliers than other regression loss. Smooth L1-loss can be interpreted as a combination of L1-loss and L2-loss. It behaves as L1-loss when the absolute value of the argument is high, and it behaves like L2-loss when the absolute value of the argument is close to zero. $\mathscr{L}_\theta$ is angle regression loss as follows:

$$\mathscr{L}_\theta = L_c\left(\theta_{gt}^c, \theta_a^c\right) + D\left(\theta_{gt}^r, \theta_a^r\right), \tag{15}$$

where superscript c and r represent angle class and residual, respectively, L_c is orientation classification loss, and D is residual prediction loss.

For the detection classification loss $\mathscr{L}_{cls}$, this study uses the Focal Loss:

$$\mathscr{L}_{cls} = -\alpha(1 - p^a)^\gamma \log p^a, \tag{16}$$

with $\alpha = 0.25$ and $\gamma = 2$. Besides, this study uses a softmax loss on the $\mathscr{L}_a$.

4. Simulations

This section presents simulation results to compare the performance with baseline and F-Cooper [5] under the same datasets and scenarios. In order to have the same settings, this study carries out experiments with our dataset for cooperative 3D object detection. The dataset is simulated scenarios by the gazebo [46]. All data is collected from vehicles equipped with a 64-beam LiDAR, one GPS, and one IMU. There are the detailed parameters in Table 1. The parameter setting of LiDAR is suitable for most products on the market. Besides, the distance of "Near" and "Far" vehicle is defined by [5], and the other parameters (β_{cls}, β_r, β_a, α, and γ) refer to [27]. In the experiments, the framework proposed, and F-Cooper runs on a computing device with an NVIDIA RTX 3090 GPU.

Table 1: Parameters [5, 27].

Name	Value
Vertical sampling rate	64
Horizontal sampling rate	1024
Angular resolution	0.08
X-axle detection range R_x	[0, 25.6]
Y-axle detection range R_y	[-40, 40]
Z-axle detection range R_z	[-3, 1]
"Near" vehicle	0-12.5 m
"Far" vehicle	≥12.5 m
X-axle pillar size P_x	0.2 m
Y-axle pillar size P_y	0.2 m
Transmission rate	1 Gbps
β_{cls}	1.0
β_r	2.0
β_a	0.1
α	0.25
γ	2
Confidence score threshold	0.5

4.1. Detection Precision Analysis. Table 2 shows the performance of the proposed algorithm against the baseline and F-Cooper [5] by comparing the detected vehicles with the confidence score threshold at 0.5. There are three scenarios in the experiments, including multilane roads test, road intersection 1 test, and road intersection 2 test. The difference between these two road intersection tests is the number of vehicles. The former with more vehicles can be seen as traffic jams. Moreover, the latter can be thought of as an uncongested intersection. This study chooses vehicle one as ego vehicle for convenience compared with "Near" and "Far" results.

In a multilane road scenario, vehicle one can have a good "Near" detection accuracy but a weak vision of "Far" vehicles with 22.22% precision. Besides, vehicle 2 can only detect 16.67% of nearby objects and cannot sense any vehicles at a "Far" distance due to the severe occlusion. Next, F-Cooper and our proposed algorithm achieve a similar detection precision at "Near" distance compared with baseline. These two approaches can significantly improve detection precision at "Far" vehicles with 42.78% and 37.5%, respectively, compared with 22.22% in baseline test. It means that the drivers can have more reaction slot windows due to the more accurate perception of the further vehicle.

In the road intersections 1 case, the CAVs are not affected by occlusion, which means that detection precision without fusion can perform well at "Near" and "Far" distances, such as 75% and 42.86% detection on vehicle 1, respectively. On the other hand, the fusion methods can perform better with 85.71% by our proposed approach and 80.21% by F-Cooper at near distance and 50% and 46.42% at long distance, respectively.

Road intersection 2 is a particular case because the vehicles in this scenario are far away from each other. Therefore, it mainly focuses on the result of the far distance to show the performance. It can be noted that the detection precision on vehicle 1 is 25%, and on vehicle 2 is 40%. Besides, the F-Cooper and method proposed in this study can achieve 36.41% and 48.81%, respectively. This result signifies that our method can improve approximately 10% on fusion precision.

4.2. Data Volume Evaluation. The data size generated from different scenarios is approximately equal. Therefore, this study only shows the data from road intersection 1, which means that vehicles are detected near and far. In Figure 7, the bars depict the amount of data that needs to transmit and processed compared to the raw point cloud and original Pillars.

It can be seen that the raw data gathered directly from the 64-beam LiDAR region of interest (RoI) is about 4 MB. Similarly, the original data volume of pillars ranges from 3.4 MB to 3.7 MB because of lots of null points. After

Table 2: Precision comparison among baseline, *F*-Cooper, and proposed algorithm on a receiver.

Scenario	Baseline (vehicle 1)		Baseline (vehicle 2)		*F*-Cooper [5]		Ours	
	Near	Far	Near	Far	Near	Far	Near	Far
Multilane roads	66.66%	22.22%	16.67%	0.00%	65.51%	42.78%	71.43%	37.50%
Road intersection 1	75.00%	42.86%	66.67%	36.36%	80.21%	46.42%	85.71%	50.00%
Road intersection 2	N/A	25.00%	N/A	40.00%	N/A	36.41%	N/A	48.81%

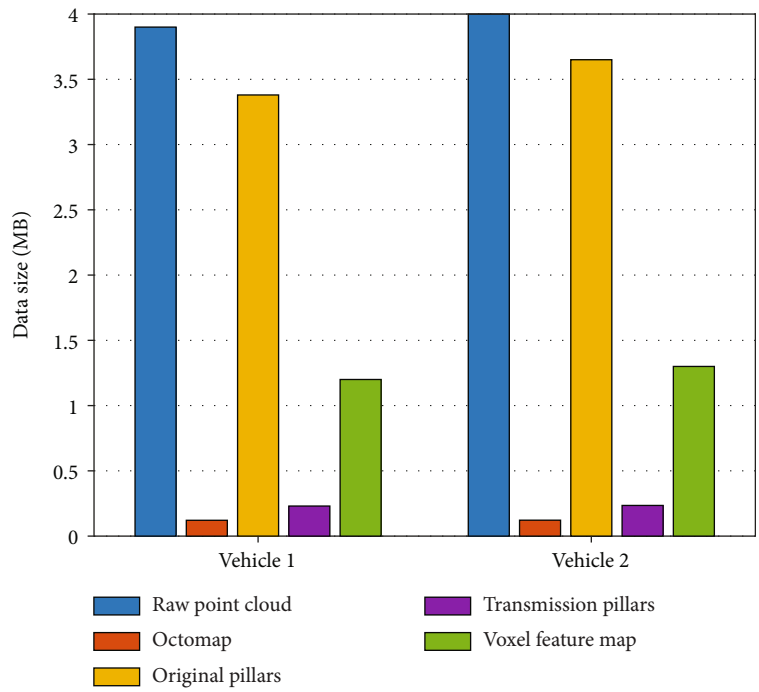


Figure 7: Comparison of data volume using different strategies. The data is collected from Road intersection 1.

processing and compression, the data size of different strategies falls off sharply to less than a quarter of raw data volume. The average voxel feature map mentioned by *F*-Cooper is about 1.25 MB. The data volume satisfies the transmission constraint of 6G communication (about 125 KB per millisecond in this experiment setting). The processing time of the feature map is too long to meet the real-time running before transmission. This study will discuss this in the following subsection. Moreover, point clouds compressed into Octomap significantly reduce data size, approximately 124.6 KB. Besides, the data volume of pillars transmitted by CAVs is about 238.29 KB. Both Octomap and transmission pillars can process within time slots that meet real-time requirements (details will be shown later).

It should be noted that the Octomap compression is necessary for transmission by the difference in data volume between the original pillars and transmission pillars. The dominant factors that impact the detection precision in PointPillars [27] are the width and length of a pillar. The smaller pillar size is able to get a more accurate object detection result. While ensuring the granularity of pillar, Octomap compression can significantly reduce the data size before transmitting pillars.

4.3. Time Consumption Evaluation. As shown in Figure 8, the total time used for different components proposed in the architecture of this study is all less than 15 ms. Besides, the time consumption of this whole framework is approximately 39 ms, which is satisfied with the 30 Hz requirement. It can be noted that Octomapcompression, and pillarupload consume the most transmission time because the data size of Octomap and pillars is much greater than the object list used in the other components. However, the time spent is still less than 2 ms. As a result, the transmission time is almost negligible compared with the processing time. The total time of cuboidregionextraction and objectdetection includes Octomapfusion and pillarfusion, respectively, before running the algorithm in this work. It can be seen that the primary time spent on the process is in Octomapcompression and objectdetection. Therefore, the raw data compression technology and 3D object perception algorithm will be mainly focused on improving cooperative detection in the future.

Besides, this study shows the time-consuming comparison with different strategies mentioned before in Figure 9. Although communicating with edges/vehicles by 6G, the raw data exchanging in Cooper [4] and feature map sharing in *F*-Cooper spend 32.76 ms and 10.24 ms in transmission,

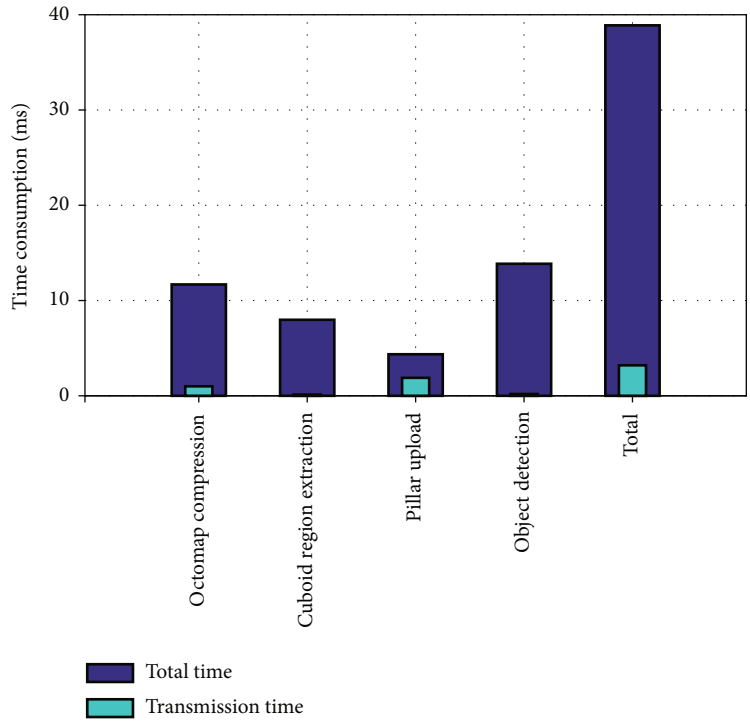


Figure 8: Comparison of time-consuming used in different components.

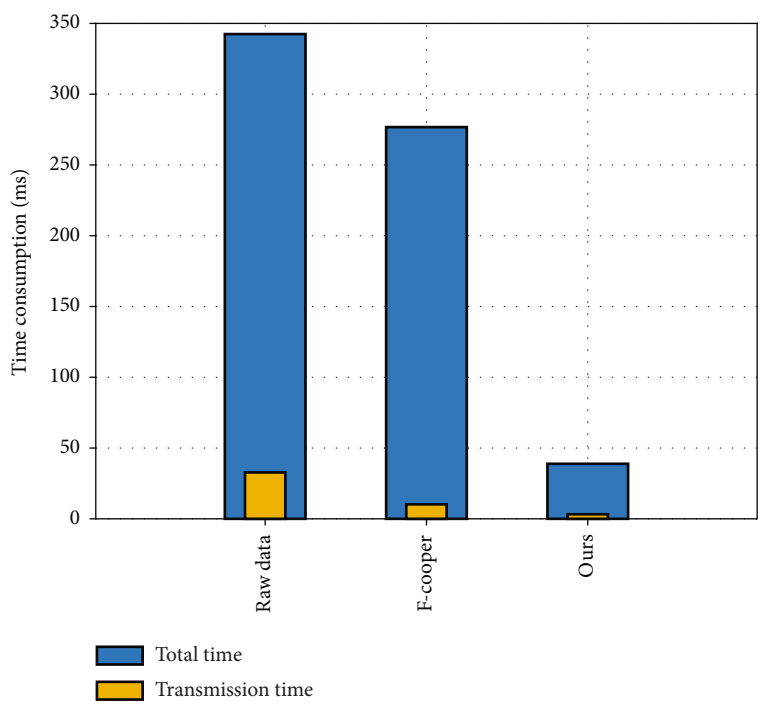


Figure 9: Comparison of time-consuming used by different strategies, including raw data, *F*-Cooper, and the proposal.

respectively. This time consumption does not meet the low latency requirement of object detection. Besides, both Cooper and *F*-Cooper are based on VoxelNet, a 3D CNN backbone. It can be noted that the processing time based on VoxelNet is approximately 237 ms, which is almost 15 times longer than that of the algorithm based on PointPillars.

5. Conclusions

In this study, a novel pillar-based perception framework has been proposed for cooperative autonomous vehicles and edges. The simulation results and analysis show that the proposed algorithm is able to achieve a 30 Hz running and

higher detection accuracy than previous schemes in terms of "Far" vehicles. It can be deployed for both CAVs and roadside edges as the improvement of perception in real-world scenarios.

In the future, the authors will focus on more data exchange types for CAVs to achieve higher detection accuracy under low delay constraints. Besides, we will also pay attention to the object detection algorithm with a higher frame rate, such as 60 Hz, without detection precision loss.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Jilin Province Science and Technology Project (No. 20200501012GX).

References

[1] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Generation of cooperative perception messages for connected and automated vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16336–16341, 2020.

[2] *Intelligent transport system (its); vehicular communications; basic set of applications; analysis of the collective-perception service (cps)*, ETSI 103 562 V2.1.1, Tech. Rep, 2019.

[3] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Analysis of message generation rules for collective perception in connected and automated driving," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 134–139, 2019.

[4] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: cooperative perception for connected autonomous vehicles based on 3d point clouds," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 514–524, 2019.

[5] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pp. 88–100, 2019.

[6] J. Shi, W. Wang, X. Wang et al., "Leveraging spatio-temporal evidence and independent vision channel to improve multi-sensor fusion for vehicle environmental perception," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 591–596, 2018.

[7] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2147–2156, 2016.

[8] X. Chen, K. Kundu, Y. Zhu et al., "3d object proposals for accurate object class detection," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[9] S. Song and M. Chandraker, "Joint sfm and detection cues for monocular 3d localization in road scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3734–3742, 2015.

[10] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3d voxel patterns for object category recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1903–1911, 2015.

[11] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler, "Detailed 3d representations for object recognition and modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2608–2623, 2013.

[12] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau, "Deep manta: a coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2040–2049, 2017.

[13] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, 2018.

[14] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7074–7082, 2017.

[15] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "Gs3d: an efficient 3d object detection framework for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1019–1028, 2019.

[16] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7644–7652, 2019.

[17] C. Cao, M. Preda, and T. Zaharia, "3d point cloud compression: a survey," in *The 24th International Conference on 3D Web Technology*, pp. 1–9, 2019.

[18] H. Houshiar and A. Nüchter, "3d point cloud compression using conventional image compression for efficient data transmission," in *2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT)*, pp. 1–8, 2015.

[19] Y. Huang, J. Peng, C.-C. J. Kuo, and M. Gopi, "Octree-based progressive geometry coding of point clouds," in *PBG@ SIGGRAPH*, pp. 103–110, 2006.

[20] Y. Fan, Y. Huang, and J. Peng, "Point cloud compression based on hierarchical point clustering," in *in 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1–7, 2013.

[21] D. C. Garcia and R. L. de Queiroz, "Intra-frame context-based octree coding for point-cloud geometry," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 1807–1811, 2018.

[22] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[23] D. Fernandes, A. Silva, R. Névoa et al., "Point-cloud based 3d object detection and classification methods for self-driving applications: A survey and taxonomy," *Information Fusion*, vol. 68, pp. 161–191, 2021.

[24] Y. Zhou and O. Tuzel, "Voxelnet: end-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4490–4499, 2018.

[25] Y. Yan, Y. Mao, and B. Li, "Second: sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[26] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-fpn: multiscale voxel feature aggregation for 3d object detection from lidar point clouds," *Sensors*, vol. 20, no. 3, p. 704, 2020.

[27] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[29] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[30] S.-W. Kim, B. Qin, Z. J. Chong et al., "Multivehicle cooperative driving using cooperative perception: design and experimental validation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 663–680, 2015.

[31] Y. Wang, G. De Veciana, T. Shimizu, and H. Lu, "Performance and scaling of collaborative sensing and networking for automated driving applications," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2018.

[32] S. Aoki, T. Higuchi, and O. Altintas, "Cooperative perception with deep reinforcement learning for connected vehicles," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 328–334, 2020.

[33] R. Hussain and S. Zeadally, "Autonomous cars: research results, issues, and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2018.

[34] S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, and D. Rus, "CarSpeak," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 259–270, 2012.

[35] J. Guo, D. Carrillo, S. Tang et al., "Coff: cooperative spatial feature fusion for 3-d object detection on autonomous vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11078–11087, 2021.

[36] M. Gabb, H. Digel, T. Müller, and R. W. Henn, "Infrastructure-supported perception and track-level fusion using edge computing," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1739–1745, 2019.

[37] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, "Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors," in *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[38] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Science and Technology*, vol. 10, 2021.

[39] Y. Chen, W. Gu, and K. Li, "Dynamic task offloading for internet of things in mobile edge computing via deep reinforcement learning," *International Journal of Communication Systems*, vol. e5154, 2022.

[40] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: a probabilistic, flexible, and compact 3d map representation for robotic systems," in *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, vol. 2, 2010.

[41] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[42] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[43] A. F. Agarap, "Deep learning using rectified linear units (relu)," 2018, http://arxiv.org/abs/1803.08375.

[44] J. Nagi, F. Ducatelle, G. A. Di Caro et al., "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 342–347, 2011.

[45] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[46] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3, pp. 2149–2154, 2004.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7444916, 13 pages
https://doi.org/10.1155/2022/7444916

Research Article

Private Computing Offloading in Edge Cloud via Collaborative Online Learning

Lin Wang,[1] **Lei Yang**,[2] **Mingchuan Zhang**,[1] **Jianxin Zhang**,[3] **Zhibin Cao**,[2] **and Qingtao Wu**[1]

[1]*School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China*
[2]*CITIC Heavy Industries Corporation Limited, Information Technology Management Center, Luoyang 471003, China*
[3]*Academy of Industrial Internet Security, Beijing Qihu Technology Company Ltd., Beijing 100088, China*

Correspondence should be addressed to Mingchuan Zhang; zhang_mch@haust.edu.cn

Received 14 April 2022; Revised 16 June 2022; Accepted 27 June 2022; Published 14 July 2022

Academic Editor: Qiang Ye

Computing offloading based on mobile edge computing (MEC) for mobile devices (MDs) has received great attentions in recent years. Strategy selection is an extremely important part of computing offloading, so how to make an optimal decision quickly and accurately during the computing offloading is a difficult point. Furthermore, MDs are likely to leak personal privacy when interacting with edge cloud, and there is also an issue about commercial privacy leakage between different cloud service suppliers. In this paper, we propose the privacy-protected edge cloud computing offloading (EPCO) algorithm based on online learning to improve the efficiency of computing offloading while ensuring the privacy of system users. Simultaneously, EPCO also supports different MDs customize their privacy level. We prove that adding privacy protection mechanism is almost no effect on the convergence of the algorithm. The simulation results validate our conclusion using a real-world dataset.

1. Introduction

Mobile devices (MDs) have become extremely popular in recent years due to their mobility and convenience [1, 3]. Meanwhile, the functionality of the application for MDs becomes increasingly powerful [3], which leads to lack of local resources of MDs, such as computing resources, storage, and energy [4, 5]. To this end, computing offloading for MDs has emerged. Researchers proposed mobile cloud computing (MCC) that source starvation can be resolved by sending computing tasks of MDs to remote cloud for execution [6, 7]. However, since cloud servers are often far away from MDs, the data needs to be transmitted for a long distance, which results in a long response time. To this end, much research in recent years has focused on mobile edge computing (MEC) [8], which sends computing tasks to edge cloud servers (ECSs) [9]. ECSs are typically deployed around MDs, which enables a short physical distance between MDs and servers, resulting a shorter latency [10]. The work of this paper is based on the edge cloud network.

Strategy selection is an important part of computing offloading between ECSs and MDs [1]. When a MD decides to offload its computing tasks to an ECS, it must first make a decision to select an optimal server for computing offloading. Researchers have used game theory in past research to solve the problem of selecting servers for computing offloading [11], which was also significantly effective at the time. However, with the increasing demands of users on the quality of network services and the challenges of big data [12, 13], most of the previous studies are outdated. In recent years, online learning algorithms have been greatly developed and used in various fields to help improve system's performance [14–18]. Therefore, we consider using online learning algorithms to solve strategy selection problem of computing offloading. Furthermore, not only the efficiency of computing offloading should be considered, but the privacy of system

users should also be concerned. However, few researches involve the above two aspects.

Privacy protection is an important part of the computing offloading [19]. The privacy issues we consider include the following two parts: the privacy of the MDs and the service suppliers. On the one hand, a MD's privacy may be exposed during data transmission or forwarding if there is a malicious third party involved. The malicious third party can infer the characteristics of the MD by accessing the computing offloading records. For example, a large amount of computation indicates the importance of the MD and the distance item exposes the location of the MD. With multiple of these side information, it is possible to identify a user in real world. For instance, He et al. [19] proposed a privacy-aware task offloading algorithm, which enabled low delay and energy consumption while maintaining an appropriate level of privacy. Min et al. [20] proposed a privacy-aware offloading algorithm that can improve the offloading performance, save energy, and enable privacy of healthcare IoT devices. Although these studies focus on the privacy issue in offloading, they only avoid the possibility of privacy leakage through some transmission method, so the privacy protection effect of these method is limited. On the other hand, since there is commercial competition between service suppliers, the privacy between them should also be considered. Therefore, privacy protection is another important part of the computing offloading. However, the research that considers both strategy selection and privacy protection is barely known.

To overcome above challenge, we consider introducing differential privacy into our computing offloading scenario. Differential privacy which proposed by Dwork et al. [21] has received great attention in the field of privacy protection in recent years. Differential privacy uses random noise to ensure that the private information of the individual will not be disclosed when the result of a query requests to disclose visible information. Zhang et al. and Hassan et al. [22, 23] use differential privacy techniques to address the risk of privacy leakage in their systems. This paper is motivated by the unresolved privacy risks in computing offloading scenarios. The privacy of mobile device users, such as location and device usage, may be leaked through data exchange during the offloading process, which is a potential privacy breach risk for users. According to the research of He et al. [19] and Min et al. [20], privacy risk has indeed become an important issue in computing offloading. Although they paid attention to the privacy issues in computing offloading, these algorithms only avoided the possibility of privacy leakage through a certain transmission method and did not fundamentally solve the problem of privacy risks. Therefore, we introduce differential privacy technology and propose an algorithm EPCO that protects the privacy of multiparty users in computing offloading. In this paper, the system structure is shown in Figure 1. For instance, a healthcare device needs to compute a large amount of monitoring data, so part of the computing tasks should be offloaded to the edge cloud for computing. The device sends the encrypted offloading data to the edge cloud, and then, each service supplier in the edge cloud gives an optimal offloading plan through online learning algorithms, and finally, the device makes a decision. Our main contributions are as follows:

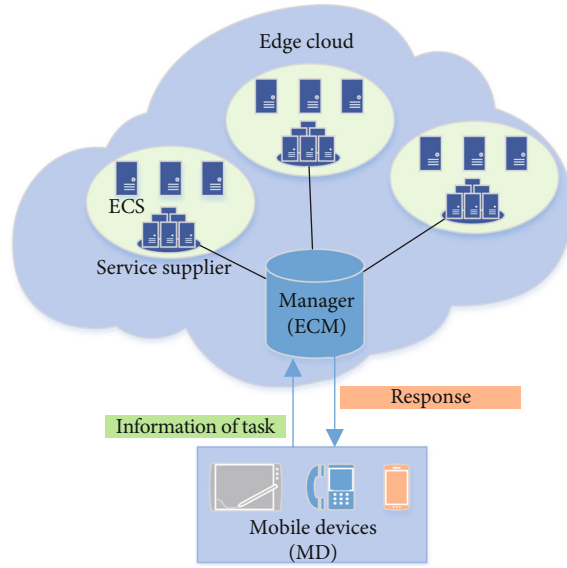


Figure 1: The structure of edge cloud in this paper.

(i) In this paper, we propose EPCO algorithm for MD and edge cloud to perform computing offloading based on online learning and differential privacy technology

(ii) EPCO preserves the privacy of both MDs and service suppliers. Moreover, we support different MDs to customize their own privacy protection levels. We proved it through theoretical derivation

(iii) We verified the theory through simulation experiments. The results show that EPCO guarantees the efficiency of computing offloading while protecting the privacy of system users

2. Related Work

In this section, we introduce related work from two aspects: optimal offloading and privacy management.

2.1. Optimal Offloading. Computing offloading alleviates the limitation of MDs' resources by sending computing tasks to the remote cloud for execution [24]. There has been a lot of research work on computing offloading in the past two decades [25–27]. Selecting ECSs for MDs is an indispensable part of computing offloading [28]. Previous decisions about server selection usually used the game theory method, which mainly concerned about energy conservation, network environment perception, and so on. Jošilo and Dán [29] made the decision to choose the wireless access points for mobile users during the computing offloading process. They proved that there is a Nash equilibrium in the model they propose, which can maximize the benefits of all users. However, this solution requires multiple interactions between users and computing resources to obtain better results, which is very

unfriendly to time-sensitive applications. Barrameda and Samaan [30] considered to use tree execution dependency trees to enhance the accuracy, which is an important technical indicator in computing offloading. However, this solution is designed for a central cloud with a large number of computing resources. First, it needs to use a large amount of additional computing to run the algorithm, and secondly, the service response time cannot be guaranteed. Although these studies have solved some specific problems, they are gradually unable to adapt to the more complicated situation such as the challenges of big data and the personalized needs of users.

In recent years, online learning has been used in various fields to help improve system performance. Shahrampour et al. [31] used online learning in object recognition to help identify the current object through historical information from other modes. Sakulkar and Krishnamachari [32] proposed two online learning algorithms to help them solve the power allocation problem modelled as a Markov decision process. We consider applying online learning to the scenario of computing offloading, which can help us further improve the efficiency of computing offloading. Cao and Cai [33] used machine learning technology to solve the decision problems of MD for achieving Nash equilibrium points in the noncooperative game that they proposed, which shows the prospect of machine learning in computing offloading research. Therefore, these studies bring us new ideas for strategy selection of computing offloading. We consider that if we try to apply online learning theory and technology to the scenario of computing offloading, this can help us further improve the efficiency of computing offloading.

2.2. Privacy Management. As more and more people pay attention to personal privacy, the issue of user privacy protection should also be considered in computing offloading. Differential privacy is a popular research in recent years in terms of privacy protection. Differential privacy was first proposed by Dwork [34] and gave provable differential privacy protection. In recent years, people have realized the importance of privacy, and privacy has been used in various research fields to protect users' privacy [35, 36]. Shin et al. [37] proposed a novel matrix factorization algorithm that guarantees per-user privacy under local differential privacy. In addition, they reduced communication overhead between the server and users by dimensionality reduction. Piao et al. [38] proposed an algorithm that can reduce query sensitivity and improved the effectiveness of published data. The above researches on differential privacy only provide same level of privacy protection, which is not practical in many applications. Dobbe et al. [39] proposed a customized local differential privacy mechanism to solve the privacy protection problem in multiagent distributed optimization problems. They proposed an approach for determining sensitivity, and they derived analytical bounds for some quadratic problems. The customizable ideas mentioned here have been adopted by us. In this paper, we allow different MDs to customize their privacy protection levels.

Since few researchers have paid attention to the privacy protection of computing offloading before, there are not many related research contributions, but we will continue to pay attention to the research progress in this area.

3. System Model

3.1. Computing Offloading. Consider an edge cloud network with a set $\mathscr{M} = \{1, \cdots, M\}$ of mobile device and an edge cloud manager (ECM) that manages a set $\mathscr{V} = \{1, \cdots, V\}$ of service suppliers. Each supplier has a set $\mathscr{S} = \{1, \cdots, S\}$ of servers that can provide computing service. Each mobile device $i \in \mathscr{M}$ has a task that has been determined to perform computing offloading. As shown in Figure 1, at each slot t, a MD sends a d-dimensional context in $\mathscr{X} := [0, 1]^d$ denoted by x_t to all suppliers, where x_t is added to Laplace noise based on the privacy protection requirements of different MDs. By receiving x_t, the ECM first broadcasts it to all suppliers. Each supplier then selects an optimal ECS and sends the information of this ECS to the ECM. The ECM provides the MD with the optimal ECS in this network that denoted by s_t. Subsequently, the MD decides whether to perform computing offloading.

Each $s_t \in \mathscr{S}$ has a two-dimensional vector denoted by $\mathbf{w}_t = (w_t^p, w_t^q)$, where w_t^p and w_t^q denote the reward in the price and reliability for performing computing offloading, respectively. w_t^p and w_t^q are given by $w_t^p = y_{s_t}^p(x_t) + \pi_t^p$ and $w_t^q = y_{s_t}^q(x_t) + \pi_t^q$, respectively, where $y_{s_t}^k(x_t)$, $k \in \{p, q\}$ denotes the expected reward of selecting ECS s in k given context x. π_t^k, $k \in \{p, q\}$ is a random noise, which satisfies $\mathbb{E}[\pi_t^k | s_{1:t}, x_{1:t}, \pi_{1:t-1}^p, \pi_{1:t-1}^q] = 0$. We assume that π_t^k, $k \in \{p, q\}$ is conditionally 1-sub-Gaussian. Formally, this means that

$$\forall \lambda \in \mathbb{R}\mathbb{E}\left[\exp^{\lambda \pi_t^k} | s_{1:t}, x_{1:t}, \pi_{1:t-1}^p, \pi_{1:t-1}^q\right] \leq \exp\left(\frac{\lambda^2}{2}\right). \quad (1)$$

Let $y_*^p(x) := \max_{s \in \mathscr{S}} y_s^p(x)$ and $y_*^q(x) := \max_{s \in \mathscr{S}} y_s^q(x)$ denote the expected reward of a ECS s in the price and the reliability for context x, respectively. Let $s^*(x)$ denote the optimal ECS for the context x.

Assumption 1. We assume that for all $k \in \{p, q\}$, $s \in \mathscr{S}$ and x, $x' \in \mathscr{X}$, $y_s^k(x)$ satisfy the following condition:

$$\left| y_s^k(x) - y_s^k\left(x'\right) \right| \leq L \left\| x - x' \right\|^{\alpha}, \quad (2)$$

where $L > 0$, $0 < \alpha \leq 1$. Assumption 1 means that if the offloading price and reliability of two ESCs are similar, it is expected that the cost of their offloading are similar.

Initially, the MD does not know any reward of ECSs. The MD learns the reward of ECSs over time. In order to evaluate the performance of our method, we define the 2D regret of the ECS as the tuple $(R^p(T), R^q(T))$, where

$$R^k(T) := \sum_{t=1}^{T} \quad y_*^k(x_t) - \sum_{t=1}^{T} \quad y_{s_t}^k(x_t), \quad k \in \{p, q\}. \quad (3)$$

When $R^p(T) = O(T^{\beta 1})$ and $R^q(T) = O(T^{\beta 2})$, we consider that the 2D regret is $O(T^{\max(\beta 1, \beta 2)})$.

3.2. Differential Privacy

Definition 2 (differential privacy). An algorithm $\mathscr{G}$ has ε differential privacy if there is only one entry different in all pairs $\mathscr{D}, \mathscr{D}' \in \mathbb{R}^d$, and all set of outcomes $\mathscr{R} \in \text{Range}(\mathscr{G})$.

$$\frac{\mathbf{P}[\mathscr{G}(\mathscr{D}) \in \mathscr{R}]}{\mathbf{P}\left[\mathscr{G}\left(\mathscr{D}'\right) \in \mathscr{R}\right]} \leq \exp(\varepsilon). \tag{4}$$

This definition mentioned above applies only to the identical level of privacy protection used by all suppliers. We now consider that each supplier i in our system specifies its own privacy ε_i.

Definition 3 (local differential privacy). An algorithm $\tilde{\mathscr{G}}$ has M nodes in the system, and we say that the algorithm $\tilde{\mathscr{G}}$ is ε_i locally private for node $i, i = 1, \cdots, M$ if for any $\mathscr{R}_i \in \text{Range}(\tilde{\mathscr{G}}_i)$ it satisfies that

$$\frac{\mathbf{P}\left\{\tilde{\mathscr{G}}_i(\mathscr{D}_1, \cdots, \mathscr{D}_i, \cdots, \mathscr{D}_M) \in \mathscr{R}_i\right\}}{\mathbf{P}\left\{\tilde{\mathscr{G}}_i\left(\mathscr{D}_1, \cdots, \mathscr{D}'_i, \cdots, \mathscr{D}_M\right) \in \mathscr{R}_i\right\}} \leq \exp(\varepsilon_i). \tag{5}$$

And we say that the algorithm $\tilde{\mathscr{G}}$ is $(\varepsilon_1, \cdots, \varepsilon_M)$-differentially private, if $\tilde{\mathscr{G}}$ is ε_i-differentially locally private for all suppliers, where $s = 1, \cdots, M$.

Definition 4 (sensitivity). The sensitivity of the function $y : X \mapsto R^d$ is as follows:

$$\Delta f = \max_{X, X' \|X - X'\|_1 = 1} \left\| f(X) - f\left(X'\right) \right\|_1. \tag{6}$$

Definition 5 (sensitivity). The sensitivity of the function $h : G \mapsto R^d$ is as follows:

$$\Phi_i = \max_{G_i, G'_i \|G_i - G'_i\|_1 = 1} \left\| g(G_i) - g\left(G'_i\right) \right\|_1. \tag{7}$$

3.3. The Learning Algorithm. In this section, we detail our proposed EPCO as shown in Algorithm 1 (EPCO(1)), Algorithm 2 (EPCO(2)), and Algorithm 3 (EPCO(3)). Since the computing offloading decision of each ECS for different MDs has stochastic distributions, we decide to let our proposed system learn an ECS's performance by online learning method. According to the sample mean reward of each ECS for the same context vector update, service suppliers learn the performance of each ECS. In order to understand the EPCO, we divide it into three algorithms which are named as EPCO(1), EPCO(2), and EPCO(3), respectively. As shown in Figure 2, the MDs run EPCO(1) to customize their privacy protection level. ECM runs EPCO(2) to interact with the MD and send the best option among all agents to the MD for decisions. Service suppliers run EPCO(3) to select the optimal ECS and interact with the ECM.

Input: $\dot{x}$.
Output: x.
1: MD is ready for computing offloading;
2: $f(\dot{x}) = \dot{x} + Lap(\Delta f/\varepsilon)$;
3: Set $x = f(\dot{x})$;
4: Send x to the edge cloud;

Algorithm 1: EPCO(1).

First, we analyse the privacy problem of MDs. Since different MDs have different requirements for privacy protection, MDs are allowed to customize the privacy level of each user. In order to protect personal privacy when MDs send computing information, the information is added with a noise which is drawn from the Laplace distribution in EPCO(1). Then, we discuss the privacy issues of the suppliers. When suppliers have selected optimal ECSs for MDs, they send the information to ECM. Since any supplier can access to this information in ECM, the Laplace mechanism is used in EPCO(2) to protect the privacy of service suppliers.

In this paper, the context space $\mathscr{X}$ is divided into m^d identical hypercubes with side length n^{-l}. Let C denote the subspace context space of $\mathscr{X}$. For ECS s and each $c \in \mathscr{C}$, EPCO maintains a counter $N_{s,c}$ recording the number of times that s was selected for the context that belongs to c. When a MD needs to perform a computing offloading, it first sends a message to the ECSs containing information about the computing task. In order to protect the privacy of the MD, it adds Laplace noise to this information in EPCO(1). Upon each context data of a MD arrival, the suppliers first identify to which subspace c the context belongs. Then, each service supplier first calculates the indices for the rewards (line 5 in EPCO(3)), which is given as follows:

$$h_{s,c}^k := \hat{y}_{s,c}^k + \mu_{s,c} + \eta_s^k, \quad k \in \{p, q\}, \tag{8}$$

where $\hat{y}_{s,c}^k = y_{s,c}^k / N_{s,c}$ estimates the sample mean of the reward for the selection of s in subspace c. $\hat{y}_{s,c}^p$ and $\hat{y}_{s,c}^q$ denote the price objective and reliability objective, respectively. $\mu_{s,c} = \sqrt{2S_{n,T}/N_{s,c}}$, where $S_{n,T} = (1 + 2\log(4|\mathscr{S}|n^d T^{3/2}))$ denotes the uncertainty of the reward estimate, which is commonly used to tradeoff exploration and exploitation in online learning [40]. η_s^k is a random noise obeying the Gaussian distribution. Then, the Upper Confidence Bound (UCB) for $y_s^k(x)$ is $h_{s,c}^k + b$ for context x in subspace c, where $b := Ld^{\alpha/2} n^{-\alpha}$ denotes the uncertainty due to context partition. Its main purpose is to inflate the reward of ECSs that are seldomly selected, which is more conducive to exploring more suitable servers than just servers with high estimated reward.

We add Laplace noise η_s^k to the index function to protect the privacy of ECSs. When $\mu_{s_p^*,c} + \eta_s^k \leq \gamma b$, it means that the

Input: x_i.
Output: s^*.
1: Receive x_i from the MD;
2: Broadcast x_i to each service suppliers;
3: Receive optimal ECSs from all service suppliers;
4: Set $s_i^* = \arg\ \max_{\{v \in \mathscr{V}\}} \{J_{v,s}\}$;
5: Send s_i^* to the MD;
6: Observe the decision of the MD and send it to the service suppliers

Algorithm 2: EPCO(2).

Input: x_i, γ.
Output: s^*.
1: Initialize: $\mathscr{C} \subseteq \mathscr{X}, N_s = 0, \forall s \in \mathscr{S}_i$;
2: Receive x_i from the ECM;
3: **for** $t = 1, 2, \cdots, T, x_t \in c$ **do**
4: Compute $h_{s,c}^k, k \in \{p, q\}$ via (4);
5: $I_{s,c}^k := h_{s,c}^k + \eta_s^k, k \in \{p, q\}$
6: **if** $\mu_{s_p^*,c} + \eta_s^k > \gamma b$ **then**
7: Set $s_p^* = s^{p^*}$;
8: **else**
9: Find the candidate optimal set of ECSs $\mathscr{S}^*$ via (5);
10: Set $s_p^* = \arg\ \max_{s \in \mathscr{S}^*} g_{s,c}^q$;
11: **end if**
12: Send s^* and $h_{s^*,c}$ to the ECM;
13: Receive $\mathbf{w}_t = (w_t^p, w^q, t)$;
14: $\widehat{y}_{s_t,c}^k \longleftarrow (\widehat{y}_{s_t,c}^k N_{s_t,c}^k + w_t^k)/(N_{s_t,c}^k + 1), k \in \{p, q\}$
15: $N_{s_t,c}^k \longleftarrow N_{s_t,c}^k + 1$
16: $t \longleftarrow t + 1$;
17: **end for**

Algorithm 3: EPCO(3).

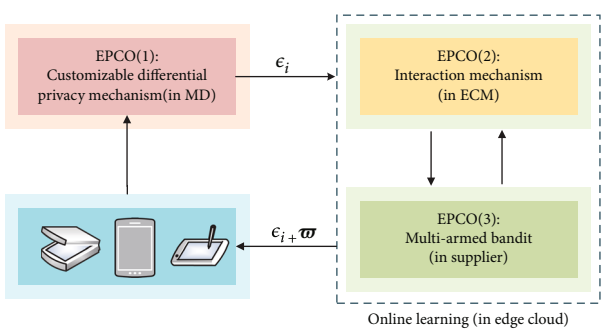


Figure 2: Algorithm mechanism.

confidence of s_p^* is high, and EPCO(3) calculates the candidate set of the optimal ECSs, which is given as follows:

$$\widehat{\mathscr{S}}^* := \left\{ s \in \mathscr{S} : h_{s,c}^p \geq \widehat{y}_{s_p^*,c}^p - \mu_{s_p^*,c} - \eta_s^k - 2b \right\} = \left\{ s \in \mathscr{S} : \widehat{y}_{s,c}^p \geq \widehat{y}_{s_p^*,c}^p - \mu_{s_p^*,c} - \eta_s^k - \mu_{s,c^*} - 2b \right\}. \tag{9}$$

When $\mu_{s_p^*,c} + \eta_s^k \geq \gamma b$, EPCO(3) just set $s_p^* = \arg\ \max_{s \in \mathscr{S}} h_{s,c}^p$ to improve the confidence of s_p^* (lines 6-7 in EPCO(3)). Simultaneously, an optimal ECS $\widehat{s}$ is selected by the exponential mechanism (lines 12-15 in EPCO(2)). We use $h_{s,c}$ to denote the total reward of ECS s that can be compared, which is given as follows:

$$J_{s,c} = \psi \widehat{y}_{s,c}^p + (1 - \psi) \widehat{y}_{s,c}^q, \quad \psi \in [0, 1], \tag{10}$$

where ψ represents a MD's preference and is adjusted according to the actual needs of a MD. For example, if a MD requires strict service payment, the value of ψ is larger. However, if a MD requires strict reliability, the value of ψ is relatively small. We select the ECS with the highest total reward and sent it to the MD (line 12 in EPCO(3)). Finally, according to the computing offloading decision of the MD (line 7 in EPCO(2)), the service suppliers update the sample mean reward and the counter (lines 14-16 in EPCO(3)).

4. Regret Analysis

In this section, we prove that the 2D regret of EPCO is sublinear functions of T. The regret $R^k(T)$ is due to selecting suboptimal ECSs from $\mathscr{S}_i$ by time T.

Let

$$R_c^k(T) := \sum_{t=1}^{T} y_*^k(x_c(t)) - \sum_{t=1}^{T} y_{\varepsilon_s^\delta}^k(x_c(t)), \quad k \in \{p, q\} \tag{11}$$

denotes the regret for objective k in round $\mathscr{T}_c$. The best fixed ECS is denoted by $y_*^k(x)$, $y_*^k(x) = \max_{s \in \mathscr{S}} y_*^k(x)$, and $k \in \{p, q\}$. Then, we have the total regret for selecting suboptimal ECSs

$$R^k(T) = \sum_{c \in \mathscr{C}} R_c^k(T). \tag{12}$$

Then, the corresponding expected regret is given as follows:

$$\mathbb{E}\left[R^k(T)\right] = \sum_{c \in \mathscr{C}} \mathbb{E}\left[R_c^k(T)\right]. \tag{13}$$

Let $\underline{h}_{s,c}^k(t) := \widehat{y}_{s,c}^k(t) - \mu_{s,c}(t) - \eta_s^k(t)$ and $\bar{h}_{s,c}^k(t) := \widehat{y}_{s,c}^k(t) + \mu_{s,c}(t) + \eta_s^k(t)$ denote lower and upper bounds for $k \in \{p, q\}$, respectively. Then, $\underline{h}_{s,c}^k(t) - b$ and $\bar{h}_{s,c}^k(t) + b$ are the lower

and upper confidence bounds, respectively. Let

$$Q_{s,c}^k := \bigcup_{t=1}^{N_c(T)} \left\{ y_s^k(x_s(t)) \notin \left[\underline{h}_{s,c}^k(t) - b, \bar{h}_{s,c}^k(t) + b \right] \right\} \tag{14}$$

denote that the service supplier is not confident about the reward estimate by time T with the context x in subspace c. Then, we partition the regret into following and bound them, respectively.

$$\begin{aligned} \mathbb{E}\left[R_c^{k_1}(T)\right] &= \mathbb{E}\left[R_c^{k_1}(T)|Q\right]P(Q) + \mathbb{E}\left[R_c^{k_1}(T)\middle|\tilde{Q}\right]P\left(\tilde{Q}\right) \\ &\le A_{\max}^{k_1} N_c(T)P(Q) + \mathbb{E}\left[R_c^{k_1}(T)\middle|\tilde{Q}\right], \end{aligned} \tag{15}$$

where $\tilde{Q}$ denotes the complement of event Q and A^k is the maximum difference between the expected reward in optimal server and other server for objective k. We use s to denote the server selected in EPCO(3) algorithm, $s*$ to denote the optimal server, and $\hat{s}$ to denote the server whose index is highest. Next, we will bound the items in Equation (15). We first bound $P(Q)$.

Lemma 6. *For any $c \in \mathscr{C}$, we have the following:*

$$\mathbf{P}(Q_c) \le \frac{1}{n^d T}. \tag{16}$$

Proof. Let $W_{s,c}^k(t)$ denote the random reward of server s in objective k in round t. We know

$$\begin{aligned} W_s^k, c(t) &= y_s^k(x_c(t)) + \pi_c^k(t) + \eta_c^k, \\ y_s^k, c(t) &= \frac{\sum_{l=1}^{t-1} W_s^k, c(l)\mathbf{I}(s_c(l) = s)}{N_{s,c}(t)}. \end{aligned} \tag{17}$$

□

We define upper and lower bounds of the random reward as follows:

$$\begin{aligned} \widehat{W}_s^k, c(t) &= \widehat{y}_s^k(x_c(t)) + \pi_c^k(t) + \eta_c^k, \\ \check{W}_s^k, c(t) &= \check{y}_s^k(x_c(t)) + \pi_c^k(t) + \eta_c^k. \end{aligned} \tag{18}$$

Let

$$\begin{aligned} \widehat{y}_s^k(x_c(t)) &= \frac{\sum_{l=1}^{t-1} \widehat{W}_s^k, c(l)1(s_c(l) = s)}{N_{s,c}(t)}, \\ \check{y}_s^k(x_c(t)) &= \frac{\sum_{l=1}^{t-1} \check{W}_s^k, c(l)1(s_c(l) = s)}{N_{s,c}(t)}. \end{aligned} \tag{19}$$

Then, we have the following:

$$\begin{aligned} \widehat{L}_{s,c}^k(t) &:= \widehat{y}_s^k(x_c(t)) - \mu_{s,c}(t) - \eta_c^k, \\ \widehat{U}_{s,c}^k(t) &:= \widehat{y}_s^k(x_c(t)) + \mu_{s,c}(t) + \eta_c^k, \\ \check{L}_{s,c}^k(t) &:= \check{y}_s^k(x_c(t)) - \mu_{s,c}(t) - \eta_c^k, \\ \check{U}_{s,c}^k(t) &:= \check{y}_s^k(x_c(t)) + \mu_{s,c}(t) + \eta_c^k. \end{aligned} \tag{20}$$

Since when $N_{s.c}(t) = 0$, $\mathbf{P}(y_s^k(x_c(t)) \in [\widehat{L}_{s,c}^k(t) - b, \widehat{U}_{s,c}^k(t) + b]) = 0$, so below, we only focus on the case of $N_{s.c}(t) > 0$, which can be expressed as follows:

$$\begin{aligned} &\left\{ y_s^k(y_c(t)) \in \left[L_{s,c}^k(t) - b, U_{s,c}^k(t) + b \right] \right\} \subset \\ &\left\{ y_s^k(y_c(t)) \in \left[\widehat{L}_{s,c}^k(t) - b, \widehat{U}_{s,c}^k(t) + b \right] \right\} \cup \\ &\left\{ y_s^k(y_c(t)) \in \left[\check{L}_{s,c}^k(t) - b, \check{U}_{s,c}^k(t) + b \right] \right\}. \end{aligned} \tag{21}$$

From the Hölder continuity, we have the following derivation:

$$\begin{aligned} &\widehat{y}_s^k(x_c(t) = \sup_{x \in c} y_s^k(y) = y_s^k\left(x'\right), \\ &\widehat{y}_{s,c}^k - y_s^k(x_c(t)) \le L\left\|x' - x_c(t)\right\|^\alpha = L\left(\sqrt{d \cdot \frac{1}{n^2}}\right)^\alpha = L\left(\frac{\sqrt{d}}{n}\right)^\alpha. \end{aligned} \tag{22}$$

Then, integrating the above derivation, we have the following:

$$y_s^k(x_c(t)) \le \widehat{y}_{s,c}^k \le y_s^k(x_c(t)) + L\left(\frac{\sqrt{d}}{n}\right)^\alpha, \tag{23}$$

$$y_s^k(x_c(t)) - L\left(\frac{\sqrt{d}}{n}\right)^\alpha \le \check{y}_s^k(x_c(t)) \le y_s^k(x_c(t)). \tag{24}$$

Using Equation (23) and Equation (24), the question can be expressed as follows:

$$\begin{aligned} &\left\{ y_s^k(x_c(t)) \in \left[\widehat{L}_{s,c}^k(t) - b, \widehat{U}_{s,c}^k(t) + b \right] \right\} \subset \left\{ \widehat{y}_s^k(x_c(t)) \in \left[\widehat{L}_{s,c}^k(t), \widehat{U}_{s,c}^k(t) \right] \right\}, \\ &\left\{ y_s^k(x_c(t)) \in \left[\check{L}_{s,c}^k(t) - b, \check{U}_{s,c}^k(t) + b \right] \right\} \subset \left\{ \check{y}_s^k(x_c(t)) \in \left[\check{L}_{s,c}^k(t), \check{U}_{s,c}^k(t) \right] \right\}. \end{aligned} \tag{25}$$

Thus, plugging Equation (23) and Equation (24) into Equation (21), we obtain the following:

$$\begin{aligned} &\left\{ y_s^k(x_c(t)) \in \left[L_{s,c}^k(t) - b, U_{s,c}^k(t) + b \right] \right\} \subset \\ &\left\{ \widehat{y}_s^k \in \left[\widehat{L}_{s,c}^k(t), \widehat{U}_{s,c}^k(t) \right] \right\} \cup \left\{ \check{y}_s^k \in \left[\check{L}_{s,c}^k(t), \check{U}_{s,c}^k(t) \right] \right\} p. \end{aligned} \tag{26}$$

Using Equation (26), we have the following:

$$\mathbf{P}(\tilde{U}) \le \mathbf{P}\left(\bigcup_{t=1}^{N_c(T)} \left\{\hat{y}_s^k \in \left[\hat{L}_{s,c}^k(t), \hat{U}_{s,c}^k(t)\right]\right\}\right) + \mathbf{P}\left(\bigcup_{t=1}^{N_c(T)} \left\{\check{y}_s^k \in \left[\check{L}_{s,c}^k(t), \check{U}_{s,c}^k(t)\right]\right\}\right). \tag{27}$$

Using the concentration inequality, the right side of the above inequality is bounded as follows:

$$\mathbf{P}(\tilde{U}) \le \frac{1}{2|\mathcal{S}|n^d T}. \tag{28}$$

Using the union bound, we have the following:

$$\begin{aligned} \mathbf{P}(\tilde{U}) &\le \frac{1}{2n^d T}, \\ \mathbf{P}(\tilde{U}) &\le \frac{1}{pn^d T}. \end{aligned} \tag{29}$$

Using the result of Lemma 10, $\mathbf{P}(U)$ and $\mathbf{P}(\tilde{U})$ can be bounded as follows:

$$\begin{aligned} \mathbf{P}(U) = \mathbf{P}\left(\cup_{k\in\{p,q\}} U_c^k\right) = m^d \cdot \mathbf{P}(U_c) &\le \frac{1}{T}, \\ \mathbf{P}(\tilde{U}) &\ge 1 - \frac{1}{T}. \end{aligned} \tag{30}$$

Lemma 7. *Under Assumption 1, $y_{s^*(t)}^p(x_c(t))$ and $y_{s(t)}^p(x_c(t))$ are generated by EPCO(3) algorithm. On event $\tilde{U}$, we have the following:*

$$y_{s^*(t)}^p(x_c(t)) - y_{s(t)}^p(x_c(t)) \le U_{s,c}^p(t) - L_{s,c}^p(t) + 2(\gamma+2)b. \tag{31}$$

Proof. There are two cases here. When $\mu_{\hat{s}_p^*,c} + \eta_s^k \le \gamma b$, we have the following:

$$U_{s,c}^p(t) \ge L_{\hat{s},c}^p - 2b \ge U_{\hat{s},c}^p(t) - \mu_{\hat{s}_p^*,c} - 2\eta_s^k - 2b \ge U_{\hat{s},c}^p(t) - 2(\gamma+1)b. \tag{32}$$

When $\mu_{\hat{s}_p^*,c} + \eta_s^k \ge \gamma b$, we have the following:

$$U_{s,c}^p(t) = U_{\hat{s}^*,c}^p(t) \ge U_{\hat{s}^*,c}^p(t) - 2(\gamma+1)b. \tag{33}$$

According to the above two cases, we obtain the following:

$$U_{s,c}^p(t) \ge U_{\hat{s}^*,c}^p(t) - 2(\gamma+1)b \ge U_{s^*,c}^p(t) - 2(\gamma+1)b, \tag{34}$$

$$U_{\hat{s}^*,c}^p(t) \ge U_{s^*,c}^p(t). \tag{35}$$

On event $\tilde{U}$, we have the following:

$$y_{s*,c}^p(x_c(t)) \le U_{s^*,c}^p(t) + b \le U_{\hat{s}^*,c}^p(t) + 2(\gamma+1)b + b, \tag{36}$$

$$y_{s,c}^p(x_c(t)) \le U_{s,c}^p(t) - b. \tag{37}$$

Combined with Equations (34)–(37), we obtain the following:

$$y_{s^*,c}^p(x_c(t)) - y_{s,c}^p(x_c(t)) \le U_{s,c}^p(t) - L_{s,c}^p(t) + 2(\gamma+2)b. \tag{38}$$

□

Lemma 8. *Under Assumption 1, $y_{s^*(t)}^q(x_c(t))$ and $y_{s(t)}^q(x_c(t))$ are generated by EPCO(3) algorithm. On event $\tilde{U}$, we have the following:*

$$y_{s^*(t)}^q(x_c(t)) - y_{s(t)}^q(x_c(t)) \le U_{s,c}^q(t) - L_{s,c}^q(t) + 2b. \tag{39}$$

Proof. We know that when $\mu_{\hat{s}_p^*,c} + \eta_s^k \le \gamma b$ holds, all servers $\hat{\mathcal{S}}^*$ are in interval $[L_{\hat{s}^*,c}^p(t) - 2b, U_{\hat{s}^*,c}^p(t)]$. Then, we show that $U_{s^*,c}^p(t)$ also satisfies this condition. On event $\tilde{U}$, we have the following:

$$y_{s^*,c}^p(x_c(t)) \in \left[L_{s^*,c}^p(t) - b, U_{s^*,c}^p(t) + b\right], \tag{40}$$

$$y_{\hat{s}^*,c}^p(x_c(t)) \in \left[L_{\hat{s}^*,c}^p(t) - b, U_{\hat{s}^*,c}^p(t) + b\right], \tag{41}$$

$$y_{s^*(t)}^q(x_c(t)) \ge y_{s^*,c}^p(x_c(t)). \tag{42}$$

By Equations (40)–(42), we obtain the following:

$$U_{s^*,c}^p \ge y_{s^*,c}^p(x_c(t)) - b \ge y_{\hat{s}^*,c}^p(x_c(t)) - b \ge L_{\hat{s}^*,c}^p(t) - 2b. \tag{43}$$

□

Since the selected server s is in $[L_{\hat{s}^*,c}^p(t) - 2b, U_{\hat{s}^*,c}^p]$, we have $U_{s,c}^q \ge U_{s^*,c}^q$. Using this result, we obtain the following:

$$y_{s,c}^q \ge U_{s,c}^q - b, \tag{44}$$

$$y_{s^*,c}^q \le U_{s^*,c}^q + b \le U_{s,c}^q + b. \tag{45}$$

From Equation (44) and Equation (45), we have the following:

$$y_{s^*,c}^q - y_{s,c}^q \le U_{s,c}^q - L_{s,c}^q + 2b. \tag{46}$$

Lemma 9. *Under Assumption 1, $\mu_{\hat{s}_p^*,c}$ are generated by EPCO(3) algorithm, and the upper limit of the number of rounds for $\mu_{\hat{s}_p^*,c} + \eta_s^k > \gamma b$ is as follows:*

$$|\mathcal{S}|\left(\frac{2S_{n,T}}{\left(\gamma b - \eta_s^k\right)^2} + 1\right). \tag{47}$$

Proof. Since

$$\mu_{\tilde{s}_p^*,c} = \sqrt{\frac{2S_{n,T}}{N_{\tilde{s}^*,c}}},$$
$$N_{\tilde{s}^*,c} = \frac{2S_{n,T}}{\mu_{\tilde{s}_p^*,c}} < \frac{2S_{n,T}}{\left(\gamma b - \eta_s^k\right)^2}. \tag{48}$$

□

And each such event increases the value of $N_{\tilde{s}^*,c}$ by one. The number of rounds for $\mu_{\tilde{s}_p^*,c} + \eta_s^k > \gamma b$ is bounded by 2 $S_{n,T}/(\gamma b - \eta_s^k)^2 + 1$. Summing all the servers together obtains the final result.

Lemma 10. *Under Assumption 1, $R_c^p(t)$ is generated by EPCO(3) algorithm. On event $\tilde{U}$, we have the following:*

$$R_c^p(t) \le |\mathcal{S}|A_{\max}^p + 2B_{n,T}\sqrt{|\mathcal{S}|N_c(t)} + 2(\gamma+2)zN_c(t), \tag{49}$$

where $B_{n,T} := 2\sqrt{2S_{n,T}}$.

Lemma 11. *Under Assumption 1, $R_c^q(t)$ is generated by EPCO(3) algorithm. On event $\tilde{U}$, we have for all $c \in \mathcal{C}$*

$$R_c^q(t) \le |\mathcal{S}|A_{\max}^q\left(\frac{2S_{n,T}}{\gamma^2 z^2} + 1\right) + 2zN_c(t) + 2B_{n,T}\sqrt{|\mathcal{S}|N_c}. \tag{50}$$

More detailed proof of Lemma 10 and Lemma 11 is presented in [41] (see Lemma 10 and Lemma 11 in [41]).

Theorem 12. *Under Assumption 1, $R^k(t)$ is generated by EPCO(3) algorithm, and we have for any $k \in p, q$ as follows:*

$$\mathbf{P}\left(R^k(t) < \xi_k(t)\right) \ge 1-, \quad \forall t \in \{1, \cdots, T\}, \tag{51}$$

where

$$\xi_p(t) = n^d|\mathcal{S}|A_{\max}^p + 2B_{n,T}\sqrt{|\mathcal{S}|n^d t} + 2(\gamma+2)zt,$$
$$\xi_q(t) = n^d|\mathcal{S}|A_{\max}^q\left(\frac{2S_{n,T}}{\gamma^2 z^2} + 1\right) + 2zt + 2B_{n,T}\sqrt{|\mathcal{S}|n^d t}. \tag{52}$$

Proof. Combining Equation (12), Lemma 10, and Lemma 11, we have the following:

$$R^p(t) \le n^d|\mathcal{S}|A_{\max}^p + 2B_{n,T}\sum_{c\in\mathcal{C}}\sqrt{|\mathcal{S}|N_c(t)} + 2(\gamma+2)zN_c(t),$$
$$\le n^d|\mathcal{S}|A_{\max}^p + 2B_{n,T}\sqrt{|\mathcal{S}|n^d t} + 2(\gamma+2)zt, R^q(t)$$
$$\le n^d|\mathcal{S}|A_{\max}^q\left(\frac{2S_{n,T}}{\gamma^2 z^2} + 1\right) + 2zN_c(t) + 2B_{n,T}\sum_{c\in\mathcal{C}}\sqrt{|\mathcal{S}|N_c}$$
$$\le n^d|\mathcal{S}|A_{\max}^q\left(\frac{2S_{n,T}}{\gamma^2 z^2} + 1\right) + 2zt + 2B_{n,T}\sqrt{|\mathcal{S}|n^d t}. \tag{53}$$

□

Theorem 13. *Under Assumption 1, $R^k(t)$ is generated by EPCO(3) algorithm, and m and γ satisfy $n = \lceil T^{1/(3\rho+d)}\rceil$ and $\gamma > 0$, and we have the following:*

$$\mathbb{E}[R^p(T)] \le A_{\max}^p + 2^d|\mathcal{S}|A_{\max}^q T^{d/3\rho+d} + 2(\gamma+2)Ld^{\rho/2}T^{2\rho+d/3\rho+d}$$
$$+ 2^{d/2+1}2B_{n,T}\sqrt{|\mathcal{S}|}T^{1.5\rho+d/3\rho+d}, \mathbb{E}[R^q(T)]$$
$$\le 2^{d/2+1}2B_{n,T}\sqrt{|\mathcal{S}|}T^{1.5\rho+d/3\rho+d}$$
$$+ A_{\max}^q + \left(2Ld^{\rho/2} + \frac{A_{\max}^q|\mathcal{S}|2^{1+2\rho+d}B_{n,T}}{\gamma^2L^2d^\rho}\right)T^{2\rho+d/3\rho+d}$$
$$+ 2^d A_{\max}^q|\mathcal{S}|T^{d/3\rho+d}. \tag{54}$$

Proof. According to the Theorem 12 and Equation (15), $\mathbb{E}[R^k(T)]$ is bounded as follows:

$$\mathbb{E}\left[R^k(T)\right] \le \mathbb{E}\left[R^k(T)\middle|\tilde{U}\right] + \sum_{c\in\mathcal{C}} A_{\max}^k N_c(t)\mathbf{P}(U) \le \mathbb{E}\left[R^k(T)\middle|\tilde{U}\right]$$
$$+ \frac{\sum_{c\in\mathcal{C}} A_{\max}^k N_c(t)}{T = \mathbb{E}\left[R^k(T)\middle|\tilde{U}\right] + A_{\max}^k} \tag{55}$$

Then, we obtain the following:

$$\mathbb{E}[R^p(T)] \le \xi_p(t) + A_{\max}^p,$$
$$\mathbb{E}[R^q(T)] \le \xi_q(t) + A_{\max}^q. \tag{56}$$

When $n = \lceil T^{1/(3\rho+d)}\rceil$, we have the following:

$$\mathbb{E}[R^p(T)] \le A_{\max}^p + 2^d|\mathcal{S}|A_{\max}^q T^{d/3\rho+d} + 2(\gamma+2)Ld^{\rho/2}T^{2\rho+d/3\rho+d}$$
$$+ 2^{d/2+1}2B_{n,T}\sqrt{|\mathcal{S}|}T^{1.5\rho+d/3\rho+d}, \mathbb{E}[R^q(T)]$$
$$\le 2^{(d/2)+1}2B_{n,T}\sqrt{|\mathcal{S}|}T^{1.5\rho+d/3\rho+d} + A_{\max}^q$$
$$+ \left(2Ld^{\rho/2} + \frac{A_{\max}^q|\mathcal{S}|2^{1+2\rho+d}B_{n,T}}{\gamma^2L^2d^\rho}\right)T^{2\rho+d/3\rho+d}$$
$$+ 2^d A_{\max}^q|\mathcal{S}|T^{d/3\rho+d}. \tag{57}$$

The result shows that not only the regret of EPCO is sublinear, which is $O(T^{(2\rho+d)/(3\rho+d)})$, but also added privacy differential mechanism does not affect its convergence. □

5. Differentially Private

The privacy protection mechanism applied in this paper uses *differential privacy* mechanism, which is originally introduced by Dwork et al. [21].

Theorem 14. *EPCO(1) preserves $(\varepsilon_i, 0)$-differential privacy for MD i where*

$$\varepsilon_i = \Phi_i \sum_{t=1}^{T} \frac{1}{\rho_i^t}, \tag{58}$$

and each MD i of the noise η_i^t is independently selected according to the Laplace distribution, where the density function is $p(\eta_i^t) = (1/2\rho_i^t)\exp(-\|\eta_i^t\|/\rho_i^t)$ for $i = 1, \cdots, M$ and $t = 1, \cdots, T$.

Proof. We first show that $\tilde{\mathscr{G}}_i$ is locally $(\varepsilon_i, 0)$-differential privacy for MD i. Now we start by studying the quantity of interest

$$\frac{\mathbf{P}\left\{\tilde{\mathscr{G}}_i^T(\mathscr{D}_1, \cdots, \mathscr{D}_i, \cdots, \mathscr{D}_M) \in \mathscr{W}\right\}}{\mathbf{P}\left\{\tilde{\mathscr{G}}_i^T\left(\mathscr{D}_1, \cdots, \mathscr{D}'_i, \cdots, \mathscr{D}_M\right) \in \mathscr{W}\right\}} \tag{59}$$

of MD i. We use a random variable W_i^t, for $t = 1, \cdots, T$, to denote the output of $\tilde{\mathscr{G}}_i^T$ with input $(\mathscr{D}_1, \cdots, \mathscr{D}_i, \cdots, \mathscr{D}_M)$ and W'^t_i to denote the output of $\tilde{\mathscr{G}}_i^T$ with input $(\mathscr{D}_1, \cdots, \mathscr{D}'_i, \cdots, \mathscr{D}_M)$. □

According to the definition above, we can rewrite the issue as follows:

$$\frac{\mathbf{P}\left\{W_i^1 = w_i^1, \cdots, W_i^T = w_i^T\right\}}{\mathbf{P}\left\{W'^1_i = w_i^1, \cdots, W'^T_i = w_i^T\right\}}. \tag{60}$$

We use W^t to denote the tuple $(W_1^t, \cdots, W_M^t)$. Then, we have the following:

$$\begin{aligned}\frac{\mathbf{P}\left\{W_i^1 = w_i^1, \cdots, W_i^T = w_i^T\right\}}{\mathbf{P}\left\{W'^1_i = w_i^1, \cdots, W'^T_i = w_i^T\right\}} &= \frac{\mathbf{P}\left\{W^1 = w^1, \cdots, W^T = w^T\right\}}{\mathbf{P}\left\{W'^1 = w^1, \cdots, W'^T = w^T\right\}} \\ &= \prod_{t=1}^{T} \frac{\mathbf{P}\left\{W^t = w^t \middle| W^\tau = w^\tau, \tau < t\right\}}{\mathbf{P}\left\{W'^t = w^t \middle| W'^\tau = w^\tau, \tau < t\right\}}.\end{aligned} \tag{61}$$

We form this process into a Markov chain where the random vector a^t denotes the Lagrangian $(\nu_1^t, \cdots, \nu_M^t)$, which is presented in [39] (see Theorem 3.1 in [39]). Then, we have the following relationship:

$$\begin{aligned}\frac{\mathbf{P}\left\{W^t = w^t \middle| W^\tau = w^\tau, \tau < t\right\}}{\mathbf{P}\left\{W'^t = w^t \middle| W'^\tau = w^\tau, \tau < t\right\}} &= \frac{\mathbf{P}\left\{W^t = w^t \middle| a^{t-1} = \nu^{t-1}\right\}}{\mathbf{P}\left\{W'^t = w^t \middle| a'^{t-1} = \nu'^{t-1}\right\}} \\ &= \frac{\mathbf{P}\left\{W^t = w^t \middle| a^{t-1} = \nu^{t-1}\right\}}{\mathbf{P}\left\{W'^t = w^t \middle| a'^{t-1} = \nu^{t-1}\right\}} \le \exp\left(\frac{\left\|y\left(\nu_i^{t-1}\right) - y\left(\nu'^{t-1}_i\right)\right\|}{\rho_i^t}\right) \\ &\le \exp\left(\frac{\Phi_i}{\rho_i^t}\right).\end{aligned} \tag{62}$$

With Φ_i in Definition 5, we obtain the following:

$$\begin{aligned}&\frac{\mathbf{P}\left\{\mathscr{G}_i^T(\mathscr{D}_1, \cdots, \mathscr{D}_i, \cdots, \mathscr{D}_M) \in \mathscr{W}\right\}}{\mathbf{P}\left\{\mathscr{G}_i^T\left(\mathscr{D}_1, \cdots, \mathscr{D}'_i, \cdots, \mathscr{D}_M\right) \in \mathscr{W}\right\}} \\ &= \prod_{t=1}^{T} \frac{\mathbf{P}\left\{W^t = w^t \middle| W^\tau = w^\tau, \tau < t\right\}}{\mathbf{P}\left\{W'^t = w^t \middle| W'^\tau = w^\tau, \tau < t\right\}} \le \prod_{t=1}^{T} \exp\left(\frac{\Phi_i}{\rho_i^t}\right) \\ &= \exp\left(\sum_{t=1}^{T} \left(\frac{\Phi_i}{\rho_i^t}\right)\right).\end{aligned} \tag{63}$$

This result proves the privacy guarantee in Equation (58).

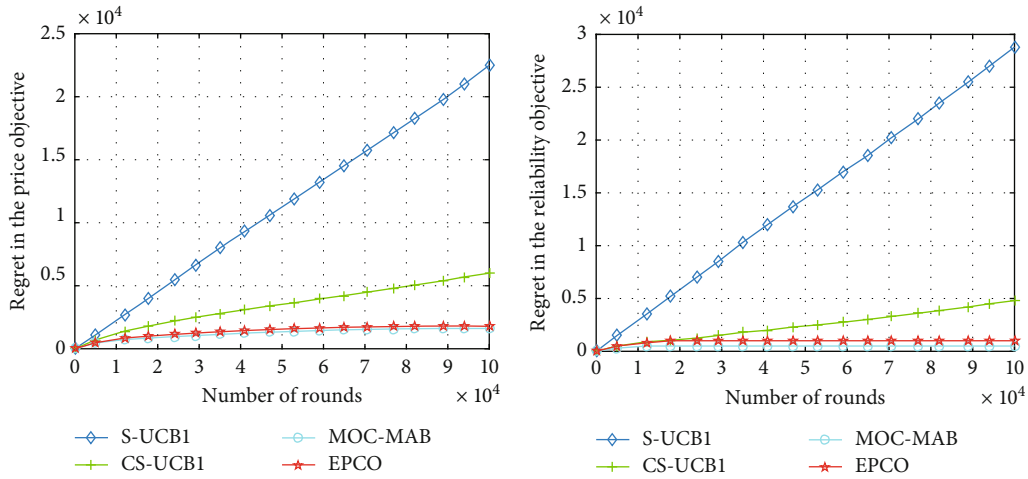
Thus, Theorem 14 proves that our proposed EPCO(1) can guarantee the MD's privacy and different MDs have different privacy protection levels.

Theorem 15. *EPCO(1) preserves $(\varepsilon, 0)$-differential privacy for the contextual information of ECSs.*

Proof. Let $D = (h_1, \cdots, h_T)$ denote true information of a supplier and $\mathscr{D}'$ denote a dataset which differs from $\mathscr{D}$ in only one data. We define the reward that adds noise as I_t. Then, for different suppliers s_1 and s_2, we have the following:

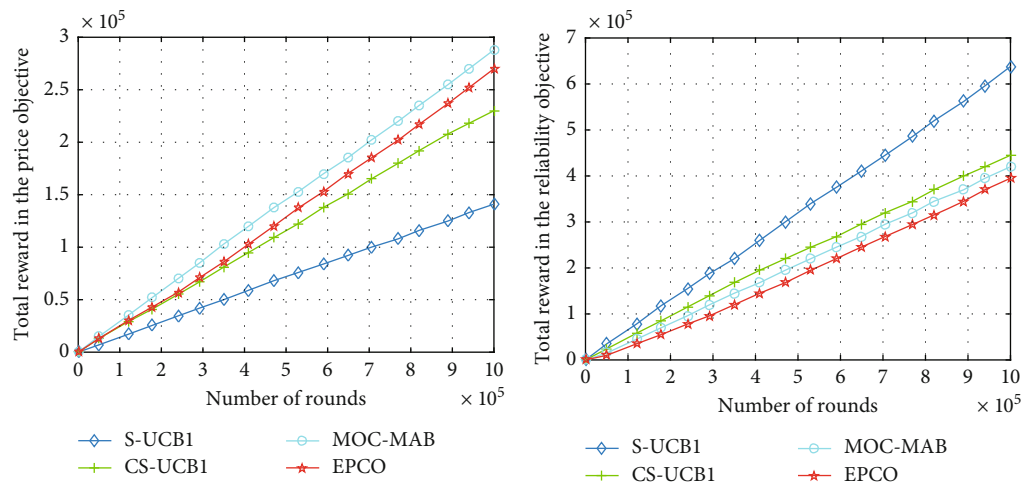
$$\begin{aligned}\frac{\mathbf{P}[\mathscr{M}(s_1, t) = I_t]}{\mathbf{P}[\mathscr{M}(s_2, t) = I_t]} &= \frac{\exp\left(-\left(\varepsilon' \left|h_{s_1,t} - I_t\right| / \Delta f\right)\right)}{\exp\left(-\left(\varepsilon' \left|h_{s_2,t} - I_t\right| / \Delta f\right)\right)} \\ &= \exp\left(\frac{\varepsilon'}{\Delta f}\left(\left|h_{s_2,t} - I_t\right| - \left|h_{s_1,t} - I_t\right|\right)\right) \\ &\le \exp\left(\frac{\varepsilon'}{\Delta f}\left|h_{s_1,t} - h_{s_2,t}\right|\right) \\ &= \exp\left(\frac{\varepsilon'}{\Delta f}\left\|h_{s_1,t} - h_{s_2,t}\right\|_1\right) \le \exp\left(\varepsilon'\right).\end{aligned} \tag{64}$$

Employing Theorem 3.6 in [42] (see page 32 in [42]), Theorem 15 is proved. □



(a) Regret of price objective (b) Regret of reliability objective

Figure 3: Regret of multiobjective.



(a) Reward of price objective (b) Reward of reliability objective

Figure 4: Reward of multiobjective.

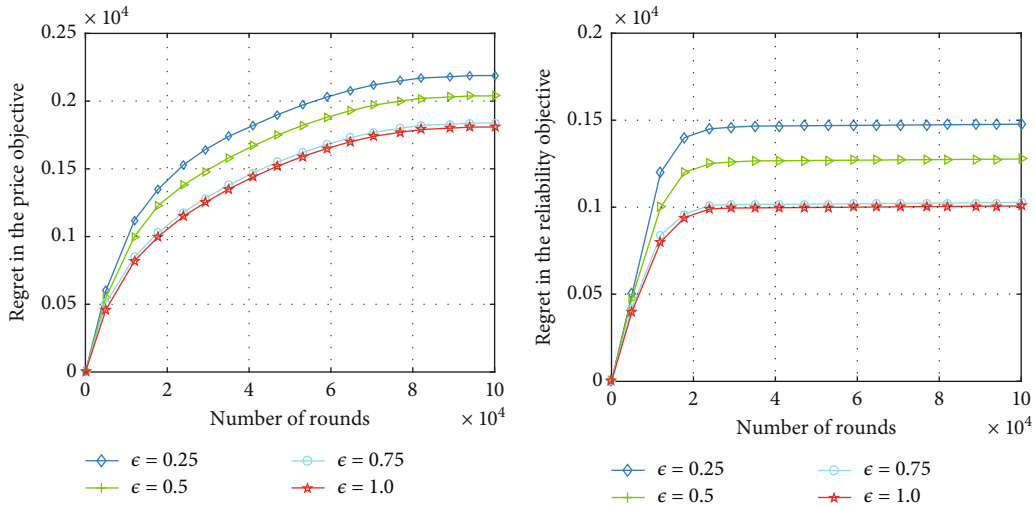
Therefore, Theorem 15 proves that service supplier fails to extract information about ECSs in ECM by the rewards. In summary, Theorem 14 and Theorem 15 prove that EPCO can preserve both privacy of MDs and service suppliers synchronously. In addition, EPCO also supports different MDs to customize their privacy protection levels.

6. Simulation Results

In order to verify the efficiency and privacy of our proposed algorithm, we conducted the following simulation experiments with real-world datasets [43]. We compare EPCO with P-UCB1, S-UCB1 [44], and MOC-MAB [41] algorithms. We use Python 3.6 version to implement these algorithms. We run these algorithms on an Acer computer with Intel(R) Core(TM) i5-4460 @ 3.2 GHz and 8 GB RAM. The operating system is Windows 10 Professional. We set L, α, and γ to 1. We give the sets $\mathscr{P} = \{0.2,0.4,0.6\}$ and $\mathscr{R} = \{0.1,0.25,0.5,1\}$. We let the time horizon $T = 10^5$ and $T = 10^6$ to satisfy different experiments. We take $\mathscr{X} = [0, 1]^2$, and the context is chosen randomly from $\mathscr{X}$ at each round. We use 6 arms in each algorithm and every algorithm runs 20 times. We take the average result of the simulation experiment.

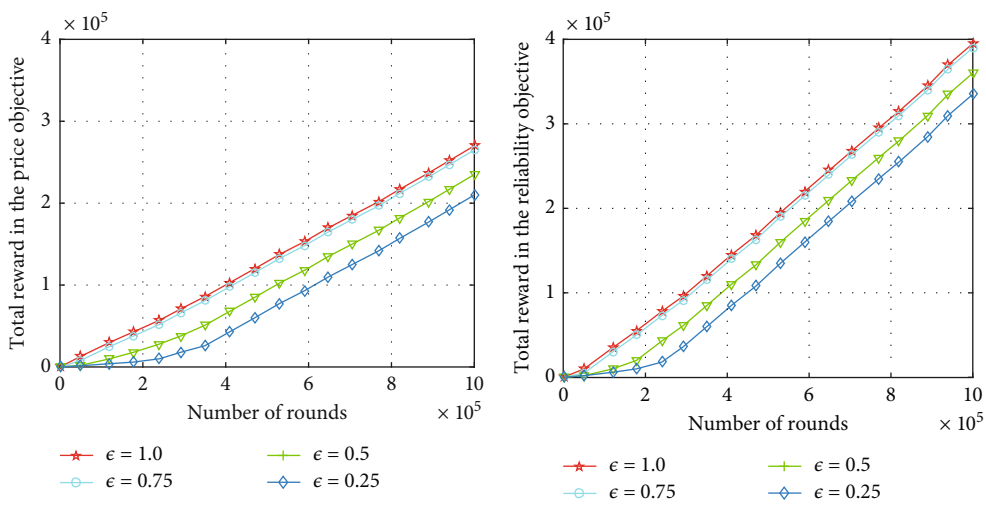
6.1. Analysis of Regret. Figure 3 shows the change of the regret of the algorithm in price and reliability objectives over time. The simulation results show that the regret of EPCO is at a lower position to both price and reliability objectives. But its regret is not the lowest, which is slightly higher than the MOC-MAB algorithm. Because we added a privacy protection mechanism to EPCO, it affected the convergence of the algorithm, but this effect was almost negligible.

6.2. Analysis of Rewards. We compare the rewards of these algorithms in two objectives, and the results are shown in



(a) Impact on price objective (b) Impact on reliability objective

Figure 5: The relationship between regret and privacy.



(a) Impact on price objective (b) Impact on reliability objective

Figure 6: The relationship between reward and privacy.

Figure 4. Compared with other multiobjective optimization algorithms, EPCO performs relatively well. Figure 4(a) shows that the EPCO algorithm performs well in price objective, outperforming other algorithms, but slightly inferior to the MOC-MAB algorithm. Figure 4(b) shows that the EPCO algorithm also performs well in reliability objective. This is because although our algorithm has high efficiency, due to the addition of a privacy protection mechanism, a small part of the error is caused, which affects the reward of different objectives.

6.3. Analysis of Privacy Protection. The effect of different privacy protection levels on the performance of the EPCO algorithm is shown in Figures 5 and 6. We set the privacy parameter ε to 0.25, 0.5, 0.75, and 1, respectively, representing different levels of privacy protection. Figure 5 shows the relationship between different privacy levels and regret. Obviously, when the value of ε is larger, the regret of the EPCO algorithm is smaller. This is because as the value of ε increases, the availability of data increases, resulting in a smaller regret. Similarly, in Figure 6, as the value of ε increases, the availability of data also increases, resulting in a larger reward for the EPCO algorithm.

6.4. Discussion. The simulation results show that EPCO's regret is at a low level in P-UCB1, S-UCB1, and MOC-MAB, because P-UCB1 and S-UCB1 are not suitable for multiobjective optimization. The regret of MOC-MAB is lower than that of EPCO. This is due to the extra work done by EPCO to protect the privacy of users. With ε taking 0.25, 0.5, 0.75, and 1.0, respectively, the regret and reward EPCO also produced different consequences. We noticed that when the value of the privacy parameter ε is larger, the regret of EPCO is smaller, the reward and privacy leakage is larger.

This is because a larger ε has little effect on the calculation performance of the algorithm. Although there is a difference in privacy leakage values from the data point of view, this difference is almost impossible to detect in real scenarios; that is to say, regardless of whether the privacy parameter ε is set to 0.1 or 0.5, this means that user privacy is almost impossible to be leaked.

6.5. Lessons Learned. The purpose of our simulation experiment is to verify the performance of EPCO from three perspectives, namely, regret, reward, and privacy. We used the Python programming language to implement the EPCO algorithm and used real-world datasets for operations. By comparing with three multiobjective optimization algorithms from the perspectives of regret and reward, namely, P-UCB1, S-UCB1, and MOC-MAB, it is concluded that EPCO does a good job in these two aspects. Meanwhile, it is concluded that EPCO can also effectively protect the privacy of users in the system through the method of privacy leakage. This is because differential privacy mechanism integrates well with online learning algorithms. No matter whether ε is set to 0.1 or 1.0, there is no trending change in the performance of EPCO. That is to say, theoretically, the value of b can be adjusted as large as possible within a certain range, which can not only ensure the privacy of users but also minimize the impact of privacy protection mechanisms on the calculation performance of the algorithm.

7. Conclusion

We have proposed a privacy-protected algorithm and EPCO algorithm, for two types of users of computing offloading. One is for MDs whose privacy is protected by a customizable differential privacy mechanism. The other is for ECS suppliers whose privacy is protected by using a common differential privacy mechanism. In addition, we proved no significant impact on the performance of the EPCO algorithm with our privacy protection mechanisms. Simultaneously, an online learning algorithm is introduced to improve the computing offloading efficiency and a detailed theoretical proof of the method is given. The simulation results verify the effectiveness of the EPCO algorithm.

Data Availability

The data that support the findings of this study are found in "More Google cluster data" [43].

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. (62002102) and No. (62176113), in part by the Leading Talents of Science and Technology in the Central Plain of China under Grant No. (224200510004), and in part by the Luoyang Major Scientific and Technological Innovation Projects under Grant No. (2101017A).

References

[1] M. Li, Q. Wu, J. Zhu, R. Zheng, and M. Zhang, "A computing offloading game for mobile devices and edge cloud servers," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 2179316, 10 pages, 2018.

[2] W. Zhang, D. Yang, W. Wu et al., "Optimizing federated learning in distributed industrial IoT: a multiagent approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3688–3703, 2021.

[3] W. Wu, N. Chen, C. Zhou et al., "Dynamic ran slicing for service-oriented vehicular networks via constrained learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2076–2089, 2021.

[4] W. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: a survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.

[5] R. Zheng, J. Zhu, M. Zhang, R. Liu, Q. Wu, and Y. Li, "A novel resource deployment approach to mobile microlearning: from energy-saving perspective," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 7430860, 15 pages, 2019.

[6] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: can offloading computation save energy?," *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[7] Q. Wu, M. Zhang, R. Zheng, Y. Lou, and W. Wei, "A QoS-satisfied prediction model for cloud-service composition based on a hidden Markov model," *Mathematical Problems in Engineering*, vol. 2013, Article ID 387083, 7 pages, 2013.

[8] F. Song, M. Zhu, Y. Zhou, I. You, and H. Zhang, "Smart collaborative tracking for ubiquitous power IoT in edge-cloud interplay domain," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6046–6055, 2020.

[9] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Network*, vol. 32, no. 5, pp. 112–117, 2018.

[10] W. Shangguang, Z. Yali, X. Jinliang, Y. Jie, and H. Ching-Hsien, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019.

[11] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: a computation offloading game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.

[12] F. Song, Z. Ai, Y. Zhou, I. You, K.-K. R. Choo, and H. Zhang, "Smart collaborative automation for receive buffer control in multipath industrial networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1385–1394, 2020.

[13] F. Song, Z. Ai, H. Zhang, I. You, and S. Li, "Smart collaborative balancing for dependable network components in cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 6916–6924, 2021.

[14] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 361–373, 2017.

[15] X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, "Online deep reinforcement learning for computation offloading in

blockchain-empowered mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8050–8062, 2019.

[16] M. Zhang, Y. Zhou, Q. Ge, R. Zheng, and Q. Wu, "Decentralized randomized block-coordinate Frank–Wolfe algorithms for submodular maximization over networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2021.

[17] M. Zhang, B. Hao, Q. Ge, J. Zhu, R. Zheng, and Q. Wu, "Distributed adaptive subgradient algorithms for online learning over time-varying networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4518–4529, 2022.

[18] J. Zhu, Q. Wu, M. Zhang, R. Zheng, and K. Li, "Projection-free decentralized online learning for submodular maximization over time-varying networks," *Journal of Machine Learning Research*, vol. 22, pp. 1–42, 2021.

[19] X. He, J. Liu, R. Jin, and H. Dai, "Privacy-aware offloading in mobile-edge computing," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, 2017.

[20] M. Min, X. Wan, L. Xiao et al., "Learning-based privacy-aware offloading for healthcare IoT with energy harvesting," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4307–4316, 2019.

[21] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds., pp. 265–284, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[22] Y. Zhang, Y. Mao, and S. Zhong, "Joint differentially private Gale-Shapley mechanisms for location privacy protection in mobile traffic offloading systems," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 10, pp. 2738–2749, 2016.

[23] M. Ul Hassan, M. H. Rehmani, R. Kotagiri, J. Zhang, and J. Chen, "Differential privacy for renewable energy resources based smart metering," *Journal of Parallel and Distributed Computing*, vol. 131, pp. 69–80, 2019.

[24] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[25] Y. Kim, H. Lee, and S. Chong, "Mobile computation offloading for application throughput fairness and energy efficiency," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 3–19, 2019.

[26] M. Zhang, M. Yang, Q. Wu, R. Zheng, and J. Zhu, "Smart perception and autonomic optimization: a novel bio-inspired hybrid routing protocol for MANETs," *Future Generation Computer Systems*, vol. 81, pp. 505–513, 2018.

[27] R. Zheng, J. Chen, M. Zhang, Q. Wu, J. Zhu, and H. Wang, "A collaborative analysis method of user abnormal behavior based on reputation voting in cloud environment," *Future Generation Computer Systems*, vol. 83, pp. 60–74, 2018.

[28] Y. Chen, N. Zhang, Y. Zhang, and X. Chen, "Dynamic computation offloading in edge computing for Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4242–4251, 2019.

[29] S. Jošilo and G. Dán, "A game theoretic analysis of selfish mobile computation offloading," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, 2017.

[30] J. Barrameda and N. Samaan, "A novel statistical cost model and an algorithm for efficient application offloading to clouds," *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 598–611, 2018.

[31] S. Shahrampour, M. Noshad, J. Ding, and V. Tarokh, "Online learning for multimodal data fusion with application to object recognition," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, pp. 1259–1263, 2018.

[32] P. Sakulkar and B. Krishnamachari, "Online learning schemes for power allocation in energy harvesting communications," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 4610–4628, 2018.

[33] H. Cao and J. Cai, "Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: a game-theoretic machine learning approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 752–764, 2018.

[34] C. Dwork, "Differential privacy," in *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052 of Lecture Notes in Computer Science, , pp. 1–12, Springer, 2006.

[35] F. Song, Y.-T. Zhou, Y. Wang, T. Zhao, I. You, and H.-K. Zhang, "Smart collaborative distribution for privacy enhancement in moving target defense," *Information Sciences*, vol. 479, pp. 593–606, 2019.

[36] J. Zhu, C. Xu, J. Guan, and D. O. Wu, "Differentially private distributed online algorithms over time-varying directed networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 4–17, 2018.

[37] H. Shin, S. Kim, J. Shin, and X. Xiao, "Privacy enhanced matrix factorization for recommendation with local differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1770–1782, 2018.

[38] C. Piao, Y. Shi, J. Yan, C. Zhang, and L. Liu, "Privacy-preserving governmental data publishing: a fog-computing-based differential privacy approach," *Future Generation Computer Systems*, vol. 90, pp. 158–174, 2019.

[39] R. Dobbe, Y. Pu, J. Zhu, K. Ramchandran, and C. Tomlin, "Customized local differential privacy for multi-agent distributed optimization," 2018, https://arxiv.org/abs/1806.06035.

[40] C. Tekin, J. Yoon, and M. van der Schaar, "Adaptive ensemble learning with confidence bounds," *IEEE Transactions on Signal Processing*, vol. 65, no. 4, pp. 888–903, 2017.

[41] C. Tekin and E. Turgay, "Multi-objective contextual multi-armed bandit with a dominant objective," *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3799–3813, 2018.

[42] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[43] J. Wilkes, "More Google cluster data," 2011, http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4879557, 14 pages
https://doi.org/10.1155/2022/4879557

Research Article

Effective Data Optimization and Evaluation Based on Social Communication with AI-Assisted in Opportunistic Social Networks

Limiao Li,[1] Fangfang Gou,[2] Huiyun Long,[3] Keke He,[1] and Jia Wu[2]

[1]*School of Computer Engineering and Applied Mathematics, Changsha University, Changsha 410003, China*
[2]*School of Computer Science and Engineering, Central South University, Changsha 410083, China*
[3]*State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China*

Correspondence should be addressed to Huiyun Long; hylong@gzu.edu.cn and Jia Wu; jiawu5110@163.com

Received 22 February 2022; Revised 9 May 2022; Accepted 14 June 2022; Published 1 July 2022

Academic Editor: Muhammad Ismail

Billions of people around the world send and receive data over online networks daily. Sufficient and redundant data are transmitted over social platforms with AI-assisted in 5G networks. In opportunistic social networks, the main challenge faced by traditional methods is that numerous user nodes participate in data transmission, causing a lot of message copy redundancy and node cache consumption. As a result, the transmission delay of the algorithm is high, the node energy consumption is too large, and even information is lost. To solve these problems, this study establishes an artificial intelligence-based optimization multiple evaluation method. The main purpose of this method is to avoid information loss caused by data loss when reducing data noise, reasonably select communication nodes in opportunistic social network scenarios, optimize data transmission performance, and avoid network congestion. Moreover, our method can effectively identify and exclude potential malicious nodes, reducing the situation that packets are intercepted and discarded. The experiment confirms that the optimized transmission evaluation scheme can effectively reduce routing overheads and energy consumption of a user node, improve the delivery ratio of node data transmission, and ensure the reliability and security of data transmission.

1. Introduction

With the development of social media, online social networks have become an essential platform for information dissemination. For the support of 5G networks with high reliability and low latency, billions of users share information on social networking sites such as Facebook, Twitter, and Google+ every day [1]. The 5G network has created abundant conditions for the rapid development of massive data on the Internet [2]. People exchange ideas, share photo albums, upload videos, and live stream videos in real time through these platforms anytime and anywhere [3, 4]. All of this information will be converted into data that can be stored, sent, and received by users [5]. With people's activities, everyone could become a source of data center and social communication.

When the amount of data is large, the communication link of the traditional social network method is easily interrupted, and a stable end-to-end path cannot be maintained, which significantly affects the success rate of information delivery [6]. Opportunistic social networks are intermittently connected mobile ad hoc networks, which usually implement communication between nodes in a "storage-carry-forward" routing mode, which can provide strong support for the effective transmission of data [7, 8]. This transmission mechanism relies on node movement to achieve, and human mobility can be used to carry messages between separate parts of the network physically. Therefore, human mobility plays a vital role in the performance of forwarding protocols in the network. In addition, 5G realizes the interconnection of all things and make people in the network of intelligent interaction [9]. In modern times,

people's communication is carried out in such a social network environment, and each activity is affected by the social interaction with each other in the network [10].

In the application scenario of opportunistic social networking, when people transfer data through portable mobile devices, considering that there is no suitable transmission target within a certain transmission range, or the transmission target does not respond, it will occupy a large amount of buffer in the data storage area of the device, which affected the data transmission rate [11]. On the one hand, the information has not been received and responded for a long time, and it is always in a waiting state, which causes a large amount of reserved information to be stored in the device, resulting in transmission delay [12]. On the other hand, when there is a large amount of buffer space when new urgent data information is received, the information release sequence needs to be updated to achieve real-time release [13]. However, when a large amount of storage space is occupied, the information cannot be updated in time.

However, in algorithms commonly used to solve the problem of message transmission in opportunistic social networks, the transmission and screening methods of messages have been difficult to adapt to the complex big data environment and the variability of data-carrying mobile devices [14]. For flood-based methods or social perception methods, they may cause a massive amount of information stored in the network and may not be effectively dealt with in time [15]. In the flood-based routing algorithm, each node forwards the received data packets to all nodes that can meet by broadcasting. This can cause copies of the message to flood the network, blocking data transmission [16]. For the social awareness method, messages wait infinitely for transmission in the process of filtering nodes, occupying limited buffer space, and mobile nodes cannot adjust the storage and transmission of messages according to environmental pressure [17]. In addition, a large number of redundant replicas in the network will lead to increased node energy consumption, increased data overhead, and buffer overflow [18]. This in turn leads to problems such as network congestion and packet loss. We can conclude that facing numerous communication nodes in a big data environment will bring significant pressure to the entire network, thus resulting in a decrease in transmission success rate, transmission delay, and data packets. It may even cause information leakage and security problems due to the existence of pseudo nodes.

In fact, the era of big data is also an era of "big noise." The information transmitted by people is increasing exponentially, but the useful information is minimal, causing information "noise" to increase much faster than useful information [19]. The pending information in the process of social network message transmission is noise. Although it may become an information resource, the preservation of its potential state also requires resources [20]. In other words, the value of data is contingent, but the loss of data processing is inevitable. In addition, data loss may result in the loss of important messages, creating risks such as information leakage [21]. Therefore, we must establish appropriate models and algorithms to solve a series of problems such as low transmission rate and increased data noise caused by massive data communication.

In this study, a multievaluation method for noise reduction optimization is established with AI-assisted. The nodes must be transmitted under the opportunistic network scenario in 5G, and the transmission must be optimized. This process is convenient for finding the law of research. The design of an effective data optimization and evaluation algorithm (EDOE) effectively divides the information. The noise optimization model ensures reliable transmission of valid data and solves the problem of important information loss when data is lost. Simultaneously, the task of information transmission is conducted using a community node, which effectively reduces the overhead of the user node and improves the performance of the algorithm. In addition, we provide privacy protection for nodes in the form of public key encryption, and exclude malicious nodes through signatures, thereby resisting routing attacks and ensuring data transmission security. The EDOE method proposed in this paper effectively avoids network congestion and packet loss and improves the overall operational efficiency of the network.

The contributions of this study are presented as follows:

(1) A function optimization model in a noisy environment is established using social user nodes to measure user noise with AI-assisted. And different noise reduction strategies are determined according to the data transmission situation, which effectively prevents the loss of important information in the data

(2) Various effective methods for optimizing the measured noise are established. The use of effective evaluation methods adjusts the adaptability between user nodes, effectively improves the communication ability between users, reduces the influence of noise, and improves the reliability of data transmission

(3) The experiment shows that the optimized transmission evaluation scheme can effectively reduce the routing overhead and energy consumption of the user nodes and improve the delivery ratio of node data transmission

2. Related Work

In recent years, in opportunistic social networks and how to use the least resources, the least delay to transmit the sent messages to their desired destinations has been the core of research. This section mainly briefly introduces some research work related to data transmission optimization in the current opportunity social network, from which we can learn about different ways to deal with transmission problems for different application scenarios.

In literature [22], Xiao et al. proposed a distributed optimal community-aware opportunistic routing-CAOR algorithm, which uses a family-aware community model. This algorithm, based on community information, transforms the routing problem between mobile nodes into a routing

problem between static communities to reduce information communication costs and maintenance costs.

Zhou et al. [23] proposed an effective opportunistic mobile network data forwarding strategy based on time closeness and centrality, called TCCB (temporal closeness and centrality-based). The key innovation is to capture data within the valid time and use temporal correlation to infer possible future social contact. This strategy aims to improve the data transmission rate when the transmission cost is similar. In literature [24], Cheol Jeong et al. introduced a hierarchical collaboration (HC) protocol of network decomposition to maximize the throughput-latency trade-off. In a dense network, compared with the case of a network without social relations, as the density of social groups increases, the performance of throughput delay trade-off can be significantly improved.

Abderrahmen Mtibaa [25] and others have proposed a set of social-based trust filters that can use explicit social information and real people movement trajectories to establish reliable communication between relay nodes. The real-time tracking-driven method is used to achieve a fair trade-off between trust and a successful delivery rate. That way, the trusted filter can achieve trusted communication in the opportunity network based on a successful delivery rate of more than 35%. Kr. Sharma et al. [26] proposed a routing protocol-Ant Router based on ant colony optimization algorithm, which stores the relevant information in the buffer of each node and then uses the "ant" algorithm to find a reasonable next hop of the data packet. It reaches the final destination through the shortest channel from the source. The protocol overcomes the problems of frequent disconnection between nodes and low routing efficiency and ensures effective and reliable message delivery.

In literature [27], Xia et al. proposed an interest-based socially aware networking message forwarding scheme BEEINFO (BEE-colony-inspired INterest-based Forwarding). It consists of BEEINFO-D (community density), BEEINFO-S (social tie), and BEEINFO-D&S. Classify communities into specific categories based on interests, introduce message scheduling strategies and buffer management algorithms, and use swarm intelligence to enhance adaptability to dynamic environments and improve forwarding performance. In literature [28], Lin Yao et al. proposed to incorporate social trust into the routing decision-making process and design trust routing based on the social similarity (TRSS) scheme based on the common interests of nodes or social similarity movements. Based on direct and recommended trust, untrusted nodes are removed from the next jump sender, and only data packets from trusted nodes are forwarded.

Recently, much AI-related wireless communication work has been discussed by researchers. To realize the rapid response of node communication and meet the requirements of the bandwidth and response time of network transmission for a large amount of data and applications, Pratik Goswami et al. [29] proposed a method for implementing distributed artificial intelligence using neural networks. The method realizes node prediction by considering the number of observations of node power consumption when executing a specific application and finally effectively saves the response time of routing. Amrit Mukherjee [30] et al. established a method for high-speed communication of large capacity data in wireless multimedia sensor networks. The main idea is to distribute power through distributed artificial intelligence and analyze the real-time spectrum sensing output system. This method not only achieves fast communication between nodes, but also guarantees cost benefits.

In opportunistic social networks, many user nodes participate in data transmission, leading to message replication redundancy, information noise growth, and node cache consumption. Previous research mostly relies on user nodes' social attributes to find a suitable next hop to improve transmission efficiency, without considering the transmission delay and information loss caused by user information noise. To solve these problems, we have established a multiple evaluation method for noise reduction optimization in opportunistic social networks, reasonably communicating nodes in an opportunistic social network scenario, optimizing transmission, and increasing users' transmission rate.

3. Data Decision and Transmission Model Design

The community, between the media and social [31–34], satisfies the connection between a person and a group. From the BBS when the Internet started to the current social networks (Facebook, WeChat Moments, etc.) [35–37], social media (Instagram, Twitter, etc.), and some vertical communities (Tiger, Blued, etc.), the community has always existed around the users, and the users are related to each other through various communities [38, 39], as shown in Figure 1. Social networks take people as the core and communicate with each other in the 5G network through smart terminals and wireless networks. It is obvious from Figure 1 that in the same community, people usually have similar hobbies and social characteristics, so people are connected more frequently. According to this property of opportunistic social networks, we can apply the identified communities to various forms of social applications such as collaborative recommendation, information dissemination, and knowledge sharing [40–42].

Now, the continuous development of the Internet and mobile devices has led to more and more people joining online social networks. Unlike social interaction, the online community we propose reduces physical distance and time limitations and has more vital interaction and timeliness. When users use social networks, they show different social attributes, such as their interests and hobbies, thereby forming multiple communities. Users with more of the same social attributes are usually divided into the same online community and become nodes in the community, which relies on the analysis of the relationship between people [43, 44]. The nodes in the community can communicate with each other, which makes communication more convenient and faster.

With the increase of social network users, data transmission between users will retain a large amount of information, and data noise will also increase exponentially, which interferes with transmission quality [45–47]. Certain information

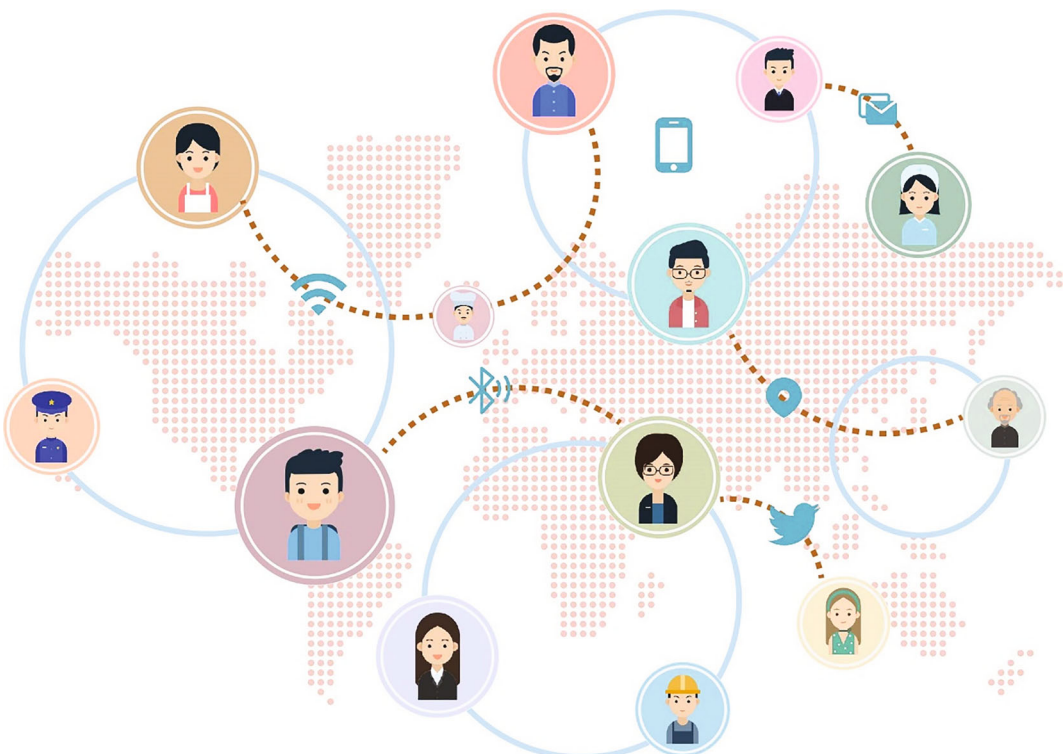


Figure 1: Social relationships between community nodes in social networks.

may be in a state of no user acceptance and response for a long time. When new data information is received, and urgent information needs to be released in real-time, the information transmission sequence cannot be updated due to a large amount of cache space, resulting in data delay. Reducing the noise between the user data and improving the transmission success rate is the key [48–50]. To solve these problems, we built an optimization function to simulate the noise problem in the social network environment.

Social networks are constantly evolving with time. Many interference factors will affect the evaluation of node relationships. In the opportunistic social network, the process of data transmission is affected by normal random noise. Reducing the impact of noise in the network helps to evaluate node relationships better. Therefore, we must establish a function optimization model in a noisy environment. Model sets:

$$f_\sigma(X) = f(X) + k \times \delta_x, \delta_x \sim N\left(0, \sigma^2\right), \tag{1}$$

where $f_\sigma(X)$ is the function value after noise interference and is the original multimodal function value,X is a random variable, δ_x is the Gaussi $f(X)$ an noise with a mean value of 0 and variance of σ^2, and k is the noise intensity.

For any two users u_1 and u_2, we can set two functions; X_1 and X_2 are the two solutions of the function $f(X_1) > f(X_2)$. When noise interference occurs, $f_\sigma(X_1) = f(X_1) + k \times \delta_2$, and its probabilities are $P(f_\sigma(X_1))$ and $P(f_\sigma(X_2))$.

$X_1 > X_2$ is better than X_1 for X_2. When no noise interference occurs, the probability of $X_1 > X_2$ is $p(X_1 > X_2) = 1$. When noise interference occurs and if δ_1 and $\delta_2 \sim N(0, \sigma^2)$ are independent of each other and $(\delta_1 - \delta_2) \sim N(0, 2\sigma^2)$, the probability of $X_1 > X_2$ is:

$$p_\sigma(X_1 > X_2) = P[f_\sigma(X_1) > f_\sigma(X_2)] = \left[P(\delta_1 - \delta_2) > \frac{f(X_1) - f(X_2)}{k}\right]. \tag{2}$$

If X_1 and X_2 are deterministically competitive in the opportunistic social network, then no noise interference occurs when $p(X_1 > X_2) = 1$. When noise interference occurs, the probability of X_1 remaining in the next user group is less than 1. Noise affects the selection operation of the relay node of the network change algorithm and reduces the search performance of the algorithm.

A common problem in AI-assisted social networks is that users are subject to noise interference, which reduces transmission quality. The fitness evaluation is divided into two processes, namely, objective function sampling and fitness calculation, to analyze the influence of normal random noise on fitness evaluation. Obtaining the value of $f_\sigma(X)$ is a sampling process AI-assisted, and converting one or more sample values into node fitness is a fitness calculation process.

3.1. One Evaluation and One Sample (OO). In opportunistic social network, the traditional network change algorithm's fitness evaluation uses a one-time evaluation sampling method. This method samples a node's objective function and evaluates the fitness once. The user's large population

size AI-assisted is the method of fitness evaluation, as expressed in

$$f_\sigma^{\mathrm{OO}}(X) = f(X) + k \times \delta, \delta \sim N\left(0, \sigma^2\right), \tag{3}$$

where $f_\sigma^{\mathrm{OO}}(X)$ is the fitness of X in a noisy environment, $f(X)$ is the fitness of X in the environment, and δ is the sampling noise when evaluating nodes that obey normal distribution with a mean of 0 and a variance of σ^2.

3.2. One Evaluation and Multisamples (OM). Resampling user data is a method for evaluating multiple sampling times. This method evaluates the fitness of each node only once, and the evaluation process samples the objective function AI-assisted multiple times, as defined in

$$f_\sigma^{\mathrm{OM}}(X) = \frac{1}{N}\sum_{i=1}^{N}(f(X) + k \times \delta_i), \delta_i \sim N\left(0, \sigma^2\right), \tag{4}$$

where $f_\sigma^{\mathrm{OM}}(X)$ is the fitness of X in a noisy environment; $f(X)$ is the fitness of X in the environment; δ_i is the i-th sampling noise; its mean is 0 and variance is σ^2; and N is the number of samples. One evaluation multiple sampling method reduces the standard deviation of noise to $\sigma/\sqrt{N}$.

3.3. Evaluation of every Generation and One Sample Test (EO). For the opportunistic social network, multiple evaluations of the sampling method reevaluate the fitness of nodes in each generation of social networks, and each objective evaluates the objective function once. A node only evaluates once. When a low fitness node is misjudged as a high fitness node, the node will be widespread in the user group, and the search time of the misleading algorithm will be extended. When two nodes in the network compete for survival opportunities at the same time, the sampling method is evaluated multiple times to reevaluate the adaptability of the selectable nodes in a noisy environment. The probability of the inferior node being evaluated as excellent is small, thus eliminating this node from the AI-assisted user group. The sampling method is expressed in

$$f_\sigma^{\mathrm{EO}}(X, t) = f(X) + k \times \delta(t), \delta(t) \sim N\left(0, \sigma^2\right), \tag{5}$$

where $f_\sigma^{\mathrm{EO}}(X, t)$ is the fitness of X at time t in a noisy environment, $f(X)$ is the fitness of X in the environment, and $\delta(t)$ is the sampling noise at time t and obeys normal distribution with a mean of 0 and a variance of σ^2.

Assuming no noise interference, $f(X_1) > f(X_2)$, and let $f(X_1) - f(X_2) = \Delta > 0$. When using the OO evaluation method, the probability of $X_1 > X_2$ in a noisy environment is:

$$\begin{aligned} &P_\sigma^{\mathrm{OO}}[(X_1 > X_2)|f(X_1) > f(X_2)] \\ &\quad = P\left[f_\sigma^{\mathrm{OO}}(X_1) > f_\sigma^{\mathrm{OO}}(X_2)\right] = P[(\delta_1^\sigma - \delta_2^\sigma) < (-\Delta/k)]. \end{aligned} \tag{6}$$

When using the OO evaluation method AI-assisted, the probability of $X_2 > X_1$ in a noisy environment is:

$$\begin{aligned} P_\sigma^{\mathrm{OO}}[(X_2 > X_1)|f(X_1) > f(X_2)] &= P\left[\left(f_\sigma^{\mathrm{OO}}(X_2) > f_\sigma^{\mathrm{OO}}(X_1)\right]\right. \\ &= P[(\delta_1^\sigma - \delta_2^\sigma) < (-\Delta/k)]. \end{aligned} \tag{7}$$

Therefore, when using the OO evaluation method AI-assisted in a noisy environment, the probability difference between the correct and the incorrect judgments is:

$$\begin{aligned} &P_\sigma^{\mathrm{OO}}[(X_1 > X_2)|f(X_1) > f(X_2)] - P_\sigma^{\mathrm{OO}}[(X_2 > X_1)|f(X_1) > f(X_2)] \\ &\quad = P[\Delta/k > (\delta_1^\sigma - \delta_2^\sigma) > -\Delta/k]. \end{aligned} \tag{8}$$

Similarly, when using the OM evaluation method in the noise environment, the probability difference between the correct and the incorrect judgments is:

$$\begin{aligned} &P_\sigma^{\mathrm{OM}}[(X_1 > X_2)|f(X_1) > f(X_2)] - P_\sigma^{\mathrm{OM}}[(X_2 > X_1)|f(X_1) > f(X_2)] \\ &\quad = P\left[\Delta/k > \left(\delta_1^{\sigma/\sqrt{N}} - \delta_2^{\sigma/\sqrt{N}}\right) > -\Delta/k\right]. \end{aligned} \tag{9}$$

Let the t-th generation be $P(t)$. Moreover, let X_1^* and X_2^* be the optimal solution, and $|f(X_1^*) - f(X_2^*)| = \varepsilon > 0$.

When using the EO evaluation method to evaluate the competition of nodes in two generations of social networks, the nodes in the previous generation network must be reevaluated. Therefore, $f_\sigma^{\mathrm{EO}}[X_1^*(t+1)] = f(X_1^*) + k \times \delta_1^*(t+1)$, and $f_\sigma^{\mathrm{EO}}[X_2^*(t+1)] = f(X_2^*) + k \times \delta_2^*(t+1)$.$\delta_1^*(t+1)$ and $\delta_2^*(t+1)$ are the sampling noises of the $t+1$ generation. Therefore, $\delta_2^*(t+1) - \delta_1^*(t+1) \sim N(0, 2\sigma^2)$.

When using the EO evaluation method, the probability that the optimal solution is correctly updated (i.e., $f(X_1^*) < f(X_2^*)$) is:

$$P_\sigma^{\mathrm{EO}}[(X_2^* > X_1^*)|f(X_1^*) < f(X_2^*)] = P[\delta_2^*(t+1) - \delta_1^*(t+1) > -\varepsilon/k]. \tag{10}$$

When using the EO evaluation method, the probability of the optimal solution error update (i.e., $f(X_1^*) > f(X_2^*)$) is:

$$P_\sigma^{\mathrm{EO}}[(X_2^* > X_1^*)|f(X_1^*) > f(X_2^*)] = P[\delta_2^*(t+1) - \delta_1^*(t+1) > \varepsilon/k]. \tag{11}$$

Therefore, when using the EO evaluation method, the probability difference between a correct and an incorrect update of the optimal solution is:

$$\begin{aligned} &P_\sigma^{\mathrm{EO}}[(X_2^* > X_1^*)|f(X_1^*) < f(X_2^*)] - P_\sigma^{\mathrm{EO}}[(X_2^* > X_1^*)|f(X_1^*) > f(X_2^*)] \\ &\quad = P\{\varepsilon/k > [\delta_2^*(t+1) - \delta_1^*(t+1)] > -\varepsilon/k\}. \end{aligned} \tag{12}$$

```
INPUT u1, u2, δ1, δ2, σ
OUTPUT Tr_method
Begin
        While (user != NULL)
                {u1-> user; u2-> user;
                Establish evaluation f_σ(X) = f(X) + k × δ_x, δ_x ~ N(0, σ^2)
                Calculation P(f_σ(X1)), P(f_σ(X2))
                If (X1>X2),(δ1 − δ2) ~ N(0, 2σ^2)
                        Else (δ2 − δ1) ~ N(0, 2σ^2) }
        End While
        While (Tr_method != NULL)*transmission will start and not interrupt */
                If (X1>X2),P((δ1^σ − δ2^σ) < −Δ/k, optimal solution X1*
                        f_σ^EO(X1*(t + 1)) = f(X1*) + k × δ1*(t + 1)
                        Tr_method-> f_σ^OM(X1*)
                Else optimal solution X2*
                        f_σ^EO(X2*(t + 1)) = f(X2*) + k × δ2*(t + 1)
                                Tr_method-> f_σ^OM(X2*)
                End While
End
```

Algorithm 1: Effective Data Optimization and Evaluation (EDOE) Algorithm.

The OM evaluation method does not need to reevaluate the nodes in the previous generation of social networks. Thus, $f_\sigma^{\mathrm{OM}}(X_1^*) = f(X_1^*) + k \times \delta_1^*$, where δ_1^* is the sampling noise of X_1^* and δ_1^* is a relatively large value that already exists given the role of the selection operator. Furthermore, $f_\sigma^{\mathrm{OM}}[X_2^*(t+1)] = f(X_2^*) + k \times \delta_2^*(t+1)$, where $\delta_2^*(t+1)$ is the sampling noise in the $t+1$-th generation. Therefore, $\delta_2^*(t+1) \sim N(0, \sigma^2)$. When using the OM evaluation method, the probability that the optimal solution is correctly updated (i.e., $f(X_1^*) < f(X_2^*)$) is:

$$P_\sigma^{\mathrm{OM}}[(X_2^* > X_1^*)|f(X_1^*) < f(X_2^*)] = P[\delta_2^*(t+1) > \delta_1^* - \varepsilon/k]. \tag{13}$$

When using the OM evaluation method, the probability of the optimal solution error update (i.e., $f(X_1^*) > f(X_2^*)$) is:

$$P_\sigma^{\mathrm{OM}}[(X_2^* > X_1^*)|f(X_1^*) > f(X_2^*)] = P[\delta_2^*(t+1) > \delta_1^* + \varepsilon/k]. \tag{14}$$

Therefore, when using the OM evaluation method, the probability difference between a correct and an incorrect update of the optimal solution is:

$$\begin{aligned} &P_\sigma^{\mathrm{OM}}[(X_2^* > X_1^*)|f(X_1^*) < f(X_2^*)] - P_\sigma^{\mathrm{OM}}[(X_2^* > X_1^*)|f(X_1^*) > f(X_2^*)] \\ &\quad = P[\delta_1^* + \varepsilon/k > \delta_2^*(t+1) > \delta_1^* - \varepsilon/k]. \end{aligned} \tag{15}$$

When using the OM evaluation method, the probability difference between a correct and an incorrect update of the optimal solution is $[\varepsilon_L, \varepsilon_R]$. $\varepsilon_R = \delta_1^* + \varepsilon/k$, and $\varepsilon_L = \delta_1^* - \varepsilon/k$. When $\delta_1^* = \delta_{eq}$, ε/k is very small.

$$P[\delta_1^* + \varepsilon/k > \delta_2^*(t+1) > \delta_1^* - \varepsilon/k] = P\{\varepsilon/k > [\delta_2^*(t+1) - \delta_1^*(t+1)] > -\varepsilon/k\}, \tag{16}$$

$\lim_{t \longrightarrow \infty} P(\delta_1^* > \delta_{eq}) = 1$. Therefore,

$$P\{\varepsilon/k > [\delta_2^*(t+1) - \delta_1^*(t+1) > -\varepsilon/k]\} > P[\delta_1^* + \varepsilon/k > \delta_2^*(t+1) > \delta_1^* - \varepsilon/k]. \tag{17}$$

In opportunistic social networks, the EO method can be used to reduce noise when data transmission must clarify the impact of previous generation historical data information. When the current data transmission does not need to consider the influence of the historical data information of the previous generation, the OM method can effectively improve the normal randomness. By adjusting the adaptability between user nodes, the impact of noise on data transmission is effectively reduced. This study has good noise reduction performance in the noisy environment of opportunistic social network. When dealing with data noise, it solves the problem of important message loss caused by data loss and ensures the security and effectiveness of data transmission.

We can effectively reduce user data transmission noise and improve data transmission efficiency through the evaluation method. We can establish the EDOE algorithm in accordance with the evaluation method.

Effective data optimization and evaluation algorithm are shown in Algorithm 1.

To improve the comprehensibility of our proposed algorithm, as shown in Algorithm 1, we have further explained the algorithm. Since the total time for the node to complete

the task must be less than or equal to the time for the node to exist in the online community, the subtasks will be allocated to the node to power from strong to weak. Then, all the nodes that have been assigned to the task perform the task and return the result to X, and X evaluates a suitable relay node for data transmission based on the evaluation result. The time complexity of this process is $O(n)$. Besides, the time complexity of spray and wait is $O(nlogn)$, and the time complexity in the epidemic routing algorithm is $O(n^2)$. Therefore, our proposed algorithm has low time complexity.

Suppose that the nodes defined in this article meet the following conditions. When using social networks, users will show different social attributes such as their interests and hobbies, and users with more similar social attributes are usually divided into the same online community. Through the "storage-carry-forward" routing model, communication can be achieved between nodes in an opportunity social network. To analyze the nodes in the opportunistic social network, we first need to understand the characteristics of the nodes.

We use a representation called a module network, which divides variables into modules. Each module represents a set of variables with the same statistical behavior. By implementing this constraint on the learned network, we can significantly reduce the model's complexity and the number of parameters. These reductions allow for more reliable estimates and better generalizations of the data. Thereby, the utilization rate of resources in the network and the overall operation efficiency are guaranteed.

Defined at the time p, we express the modularity of the community as:

$$B(p) = \frac{h_m}{H} - \frac{{r_n}^2}{4H^2}, \tag{18}$$

where B is the modularity of the community, h_m indicates the edge' total weight in community M, r_n is the node n' total level, and H is the total edge weight.

Condition 1. If any node belongs to a node in the opportunistic social network, with the increase of edge weight formed with other nodes in the network, the community's total edge weight h_m will increase. It will lead to a concomitant increase in the relevance of the community in opportunistic social networks.

Proof. At time p, the modularity $B(p)$ in the community is:

$$B(p) = \frac{h_m + \Delta h}{H + \Delta h} - \frac{(r_n + 2\Delta h)^2}{4(H + \Delta h)^2}. \tag{19}$$

Then, the modular change after time $p+1$ can be expressed as:

$$\begin{aligned} B(p+1) - B(p) &= \frac{h_m + \Delta h}{H + \Delta h} - \frac{(r_n + 2\Delta h)^2}{4(H + \Delta h)^2} - \left(\frac{h_m}{H} - \frac{{r_n}^2}{4H^2}\right) \\ &\geq \frac{4H^3\Delta h - 6H^2 r_n \Delta h + 2H^2 r_n \Delta h - 2H^2 r_n \Delta h + (r_n \Delta h)^2}{4H^2(H + \Delta h)^2} \\ &= \Delta h \frac{(2H^2 - 2Hr_n - r_n\Delta h) \times (2H - r_n)}{4H^2(H + \Delta h)^2}. \end{aligned} \tag{20}$$

We can get $\Delta h > 0$. So, we only need to prove $(2H^2 - 2Hr_n - r_n\Delta h) \times ((2H - r_n)$, and then, we can get $B(p+1) - B(p) > 0$.

In other words,

$$\begin{cases} 2H^2 - 2Hr_n - r_n\Delta h > 0 \\ 2H - r_n > 0 \end{cases}$$

$$\begin{cases} 2H^2 - 2Hr_n - r_n\Delta h > 0 \\ 2H - r_n > 0 \\ \Delta h > 0 \end{cases} \Rightarrow \begin{cases} 0 < \Delta h < 2H\left(\frac{H}{r_n} - 1\right) \\ r_n < 2H \end{cases} \Rightarrow \begin{cases} 0 < \Delta h < 2H\left(\frac{H}{r_n} - 1\right) \\ 2H\left(\frac{H}{r_n} - 1\right) > 0 \\ r_n < 2H \end{cases} \Rightarrow \begin{cases} 0 < \Delta h < 2H\left(\frac{H}{r_n} - 1\right) \\ r_n < H \end{cases}. \tag{21}$$

In summary, it can be concluded that the increase in the weight of nodes in an opportunity social network can increase its relevance to the community. However, since $2B$ is the sum of the network nodes, no community in the network can be more saturated than $2B$. Besides, we can confirm that when a node belongs to this opportunistic social network, its edge node weights will affect the relevance of the community in the opportunistic social network.

Condition 2. In the opportunistic social network, if node N_α satisfies condition $r_\alpha r_\beta / 2H < h_{\alpha\beta} < \Delta h + (r_\alpha r_\beta + r_n \Delta h + \Delta h^2 / 2(H + \Delta h))$, it will then be separated from the community of node N_β.

Proof. Firstly, supposing that community M is divided into two subcommunities M_α and M_β, where nodes N_α and N_β belong to different communities. With the increase of the

Table 1: Simulation parameters.

Parameter	Value	Parameter	Value
Simulation	12 hours	Area in networks	4,500∗3,400m^2
Communication method	Bluetooth	Transmission pattern	Broadcast
Transmit area	10m^2	Initial data packet	10
Frequency of data packet	25-35 s	Number of nodes	50-400
Size of data packet	100-200 KB	Speed of transmitting	150 k bit/s
Speed of node movement	0.5-1.5 m/s	Cache	5 MB
Transmission pattern of nodes	Social model	Initial energy	100 J

community' total weight, then

$$\begin{cases} H_\alpha + H_\beta < H \\ \dfrac{b_\alpha}{H} - \dfrac{{r_\alpha}^2}{4H^2} + \dfrac{b_\beta}{H} - \dfrac{{r_\beta}^2}{4H^2} < \dfrac{r_\alpha + r_\beta + h_{\alpha\beta}}{H} - \dfrac{(r_\alpha + r_\beta)^2}{4H^2}. \\ h_{\alpha\beta} > \dfrac{r_\alpha r_\beta}{2H} \end{cases} \tag{22}$$

When the total weight decreases, the formula can be expressed as follows:

$$\begin{aligned} {H_\alpha}^* + {H_\beta}^* > H^*, \\ h_{\alpha\beta} < \Delta h + \frac{r_\alpha r_\beta + r_n \Delta h + \Delta h^2}{2(H + \Delta h)}. \end{aligned} \tag{23}$$

Therefore, when the communities M_α and M_β where the two nodes are located satisfy the condition $r_\alpha r_\beta / 2H < h_{\alpha\beta} < \Delta h + (r_\alpha r_\beta + r_n \Delta h + \Delta h^2 / 2(H + \Delta h))$, the communities are divided. □

Condition 3. If the unique edge of node N_α in the opportunistic network is connected to node N_β, then, node N_α exists. When the weight between nodes N_α and N_β drops, node N_β will still not be separated from the community.

Proof. If community M is divided, then it must meet the following three conditions:

$$\begin{cases} H_\alpha + H_\beta < H \\ \dfrac{b_\alpha}{H} - \dfrac{{r_\alpha}^2}{4H^2} + \dfrac{b_\beta}{H} - \dfrac{{r_\beta}^2}{4H^2} < \dfrac{r_\alpha + r_\beta + h_{\alpha\beta}}{H} - \dfrac{(r_\alpha + r_\beta)^2}{4H^2}. \\ h_{\alpha\beta} > \dfrac{r_\alpha r_\beta}{2H} \end{cases} \tag{24}$$

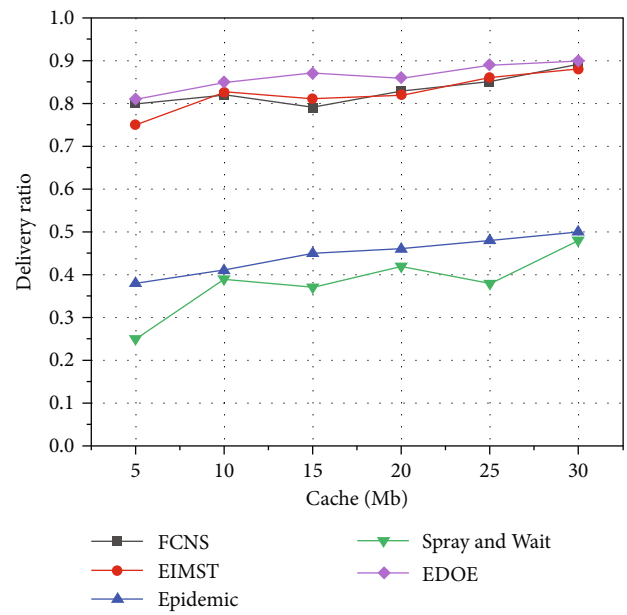


Figure 2: Delivery ratio comparison of five algorithms with different node cache.

As the weight changes, the formula can be expressed as:

$$\begin{aligned} {H_\alpha}^* + {H_\beta}^* > H^*, \\ h_{\alpha\beta} < \Delta h + \frac{r_\alpha r_\beta + r_n \Delta h + \Delta h^2}{2(H + \Delta h)}. \end{aligned} \tag{25}$$

It can be understood as:

$$\frac{r_\alpha r_\beta}{2H} < h_{\alpha\beta} < \frac{r_\alpha (r_\beta + \Delta h)}{2(H + \Delta h)} = \frac{r_\alpha r_\beta + r_\alpha \Delta h}{2(H + \Delta h)}, \tag{26}$$

As:

$$\frac{r_\alpha r_\beta}{2H} < h_{\alpha\beta} < \frac{r_\alpha (r_\beta + \Delta h)}{2(H + \Delta h)} = \frac{r_\alpha r_\beta + r_\alpha \Delta h}{2(H + \Delta h)} < 0 (\Delta h < 0). \tag{27}$$

Then,

$$\frac{r_\alpha r_\beta + r_\alpha \Delta h}{2(H + \Delta h)} < \frac{r_\alpha r_\beta}{H}. \tag{28}$$

So, we can get $r_\alpha r_\beta / 2H < h_{\alpha\beta} < \Delta h + (r_\alpha r_\beta + r_n \Delta h + \Delta h^2 / 2(H + \Delta h))$ is wrong in the end. □

In conclusion, we can prove that a node belonging to the opportunistic social network will not be separated from the community if its only edge is connected to another node when the weight between the two nodes drops.

In the previous analysis, we designed efficient algorithms. Below, we consider the routing security issues caused by the message forwarding mechanism.

In the process of data transmission, if the private information of the node is exposed to the untrusted node, or the malicious node attacks the routing, the security of the routing algorithm will not be guaranteed. Public key encryption is one of the important means to protect the confidentiality of information [51]. Therefore, to ensure the security of the routing algorithm, on the one hand, we use the public key to encrypt the message, so as to ensure the confidentiality of the message and protect the privacy information from being exposed to untrusted nodes. On the other hand, we use the threshold signature method to avoid malicious nodes in the network from attacking the routing and improve the security of the routing algorithm.

We first set the corresponding trapdoor for each node and encrypt the information of the destination node. The relay node can match the information of the destination node, but cannot obtain any attribute information of the destination node, so as to protect the privacy of the node during data transmission. In addition, a public-private key pair is generated from the information of the node, and only the destination node can decrypt the information to ensure the confidentiality of the message. Among them, the private key generator is responsible for generating and maintaining the private key of the property, only distributing its trapdoor to the nodes. It avoids the private key exposure problem that may be caused by the capture of a single node [52].

In addition, due to the lack of security mechanisms, the authenticity of node information cannot be detected, and there are malicious nodes in opportunistic social networks. By forging and publishing false information, malicious nodes deceive the encountering nodes to forward data packets to them, so as to tamper and delete the data packets, thus forming a black hole attack in the network [53–55]. To resist the attack of malicious nodes, the nodes jointly generate corresponding private keys for the social attributes they possess and distribute these private keys to each node. Among them, the generation and distribution of all private keys are based on the verifiable secret sharing (VSS) architecture [56]. The newly added node obtains part of the signature service through the chance to meet the initial node and then reconstructs the signature certificate of its own attributes. All mobile nodes do not need to authenticate the identity of the encountering node when forwarding messages. They judge the validity of the social attributes of the encountering nodes by verifying whether the neighbor nodes have valid signatures, thereby resisting potential routing attacks in opportunistic social networks.

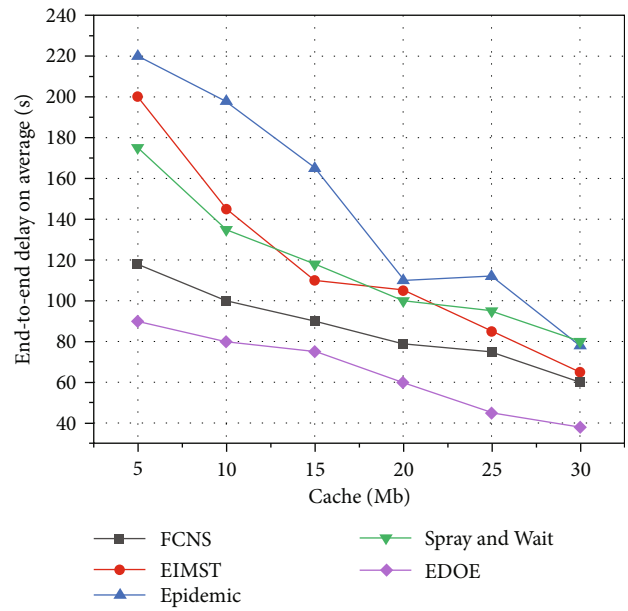


Figure 3: End-to-end delay on average comparison of five algorithms with different node cache.

Our approach can effectively identify and exclude potentially malicious nodes, reducing the number of packets being intercepted and dropped, thereby protecting the network against opportunistic social network routing attacks.

4. Performance Evaluations

4.1. Datasets. We use real datasets, the Cambridge experimental dataset of haggle for simulation experiments to evaluate the algorithm performance [57]. The dataset is the collection and statistics of at least one month's user groups' social activities carrying portable mobile devices in environments such as cities, conferences, and offices. The simulation experiment is carried out in a computer laboratory.

4.2. Simulation Setup. This article uses one simulator to perform simulation training on the proposed algorithm and compares it with some message transmission algorithms applied to opportunistic social networks, such as EIMST algorithm [46], epidemic [58], FCNS [47], and spray and wait [59]. In the simulation, the mobile nodes in the map are based on the shortest path movement (SPMBM); that is, the actual map data is collected through the minimum distance coordinates. We determine the simulation parameters through the random model in the opportunistic network. Specifically, the movement parameters used in the experiment are shown in Table 1.

4.3. Experiments and Discussion. In the evaluation experiment, the experimental results mainly focus on the impact of a node's cache space and the number of nodes on the

performance of five opportunistic network routing algorithms. Therefore, this chapter constantly changes these two variables to observe the changes in delivery ratio, end-to-end delay on average, and overhead on average.

Figure 2 shows the delivery ratio comparison results of different algorithms with different node cache. As shown in the figure, the five algorithms' delivery ratio shows an upward trend as the node cache increases. This is because the task can obtain the target data from the node cache and return it, thereby effectively improving the response speed and allowing limited resources to serve more user nodes. It can be seen that the transmission rate of the EDOE algorithm is better than the other four algorithms in most cases, even as high as 0.91. Because the EDOE algorithm measures user noise through social user nodes, adjusts the adaptability between user nodes, and reduces node noise information. In opportunistic networks, popular algorithms and Spray and Wait algorithms are typical routing algorithms based on flooding strategies. So they generate many message copies to forward messages during the information transmission process, resulting in a reduction in network transmission efficiency. It is not difficult to see from the figure that these two algorithms' transmission rate is always lower than 0.55. Besides, the EIMST algorithm conducts message transmission through the collaboration of multiple nodes. When the buffer space of the node is limited, the efficiency is low. When the nodes of the FCNS algorithm transfer information, due to the nodes' low movement efficiency and the insufficient computing power of the message carrier, the transmission efficiency of this algorithm is not high. Based on the above analysis, it can be concluded that the transmission rate of the EDOE algorithm is better than the other four algorithms when the cache is increased.

Figure 3 compares the end-to-end delay on average of the five algorithms with different node cache. As the node cache space increases from 5 Mb to 30 Mb, the end-to-end delay of various algorithms shows an inevitable downward trend. As shown in the figure, the end-to-end delay decline curve of the spray and wait algorithm and EIMST algorithm tends to be similar, and the epidemic algorithm delay is higher than the other four algorithms. This is because its information diffusion capability is not as good as the other four algorithms, and a large amount of copied information is generated when transmitting on the community, which causes a high transmission delay. The end-to-end delay on average of the EDOE algorithm and the FCNS algorithm decreases to a small degree, and the delay is always less than 120. The FCNS algorithm can analyze the transmission preference before data transmission by comparing node movement similarity, so the end-to-end delay on average is lower than traditional routing algorithms. Our proposed algorithm can reduce inefficient nodes that are not conducive to the transmission process through community reduction strategies and information screening, reducing the end-to-end delay on average. In summary, among these five algorithms, the end-to-end delay on the average performance of EDOE is the best.

Figure 4 shows the comparison of the overhead on average of the five algorithms with different node cache spaces.

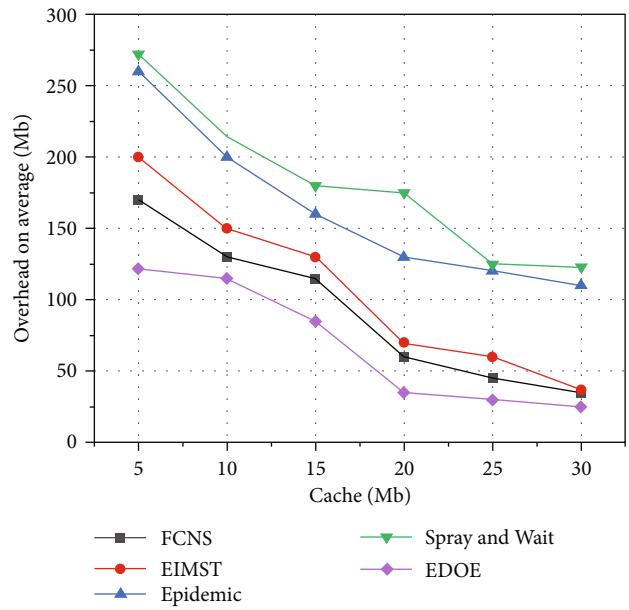


Figure 4: Overhead on average comparison of five algorithms with different node cache.

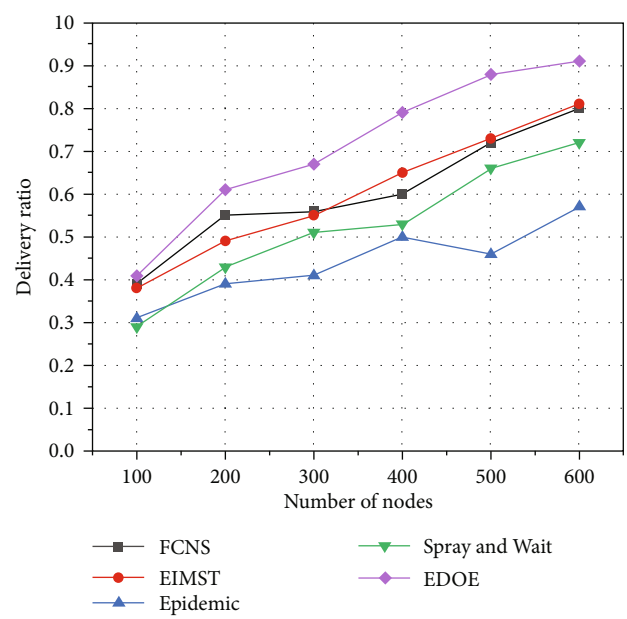


Figure 5: Delivery ratio comparison of five algorithms with different number of nodes.

As the node's cache space increases, the spray and wait algorithm and the epidemic algorithm of the five algorithms have more massive network overhead, with the highest values approaching 280 and 260, respectively. Because in the epidemic algorithm and the spray and wait algorithm, a large amount of repeated data information will occupy the node's cache space, and the node with a large amount of data is likely to consume more energy during the movement. Besides, the network overhead of the EIMST algorithm and the FCNS algorithm are both in the midstream

position. As the node's cache space increases from 5 Mb to 30 Mb, the network overhead drops by nearly 150. The EIMST algorithm can effectively manage the memory space and node messages and realize the reasonable allocation of resources. For the FCNS algorithm, the carry storage required for information transmission is relatively small, so it is relatively stable in network overhead performance. The fluctuation range of the network overhead of the EDOE algorithm is the smallest. In message transmission, the algorithm can effectively limit the number of message copies in the network through community optimization and neighbor node prediction, reduce the number of message hops from the source node to the destination node, and thus control the unnecessary energy consumption of the node.

Figure 5 compares the delivery ratio of the five algorithms when the number of nodes is different. The delivery ratio of the EIMST algorithm, the FCNS algorithm, and the spray and wait algorithm are in the midstream position, while the epidemic algorithm's delivery ratio is at a relatively low level. The EIMST algorithm and the FCNS algorithm use cache management and fuzzy inference mechanisms to effectively evaluate the social relationship between the relay node and the destination node, thereby improving message routing and forwarding efficiency. Also, it may take a long time for the relay node to meet the destination node in the second stage of the spray and wait algorithm, which will cause an inevitable decrease in the transmission rate in the network. The average delivery ratio of the EDOE algorithm is about 0.70, which is nearly 30% higher than the epidemic algorithm. The EDOE algorithm can effectively improve the relay node's probability of meeting the destination node in the communication area through reasonable detection and division of the node community and information noise evaluation.

Figure 6 shows the comparison results of the end-to-end delay on average of five algorithms with different numbers of nodes. As the number of nodes increases, the average end-to-end delay performance of all algorithms shows a downward trend overall. The flooding mechanism often leads to a higher end-to-end delay on average in the network environment, so the epidemic algorithm's delay is always higher than the other four algorithms. The EIMST algorithm can maximize the rational use of each node's cache space through node collaboration and cache management mechanisms, thereby reducing the end-to-end delay on average. As shown in the figure, the end-to-end delay on average of the EDOE algorithm and the FCNS algorithm is always between 20 and 65. As a routing algorithm based on traditional fuzzy inference, the FCNS algorithm increases the success rate of data routing and forwarding by comparing the social and mobile similarities between nodes, thereby making reasonable data transmission decisions. Besides, the EDOE algorithm can also predict the probability that the relay node will reach the destination node's community at the next moment, evaluate the mobility relationship between the relay node and the destination node, and reduce the number of message routing hops in the network, thereby significantly reducing the end-to-end delay on average. In summary, the EDOE algorithm and the FCNS algorithm perform better in terms of the end-to-end delay on average performance among the five algorithms.

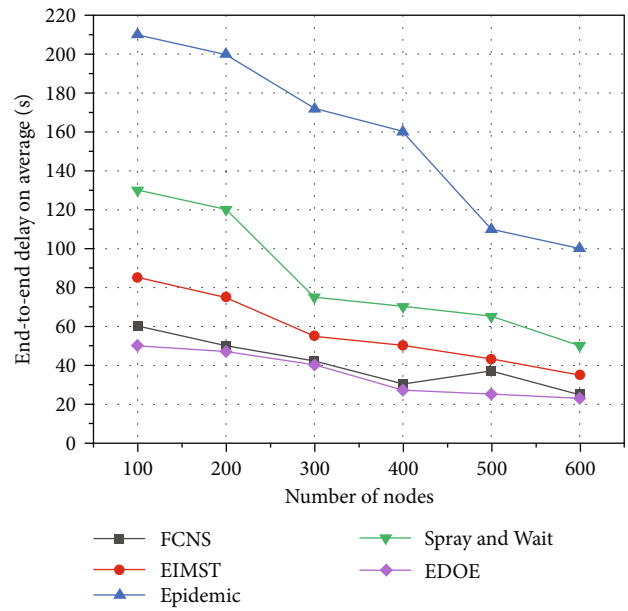


Figure 6: End-to-end delay on average comparison of five algorithms with different number of nodes.

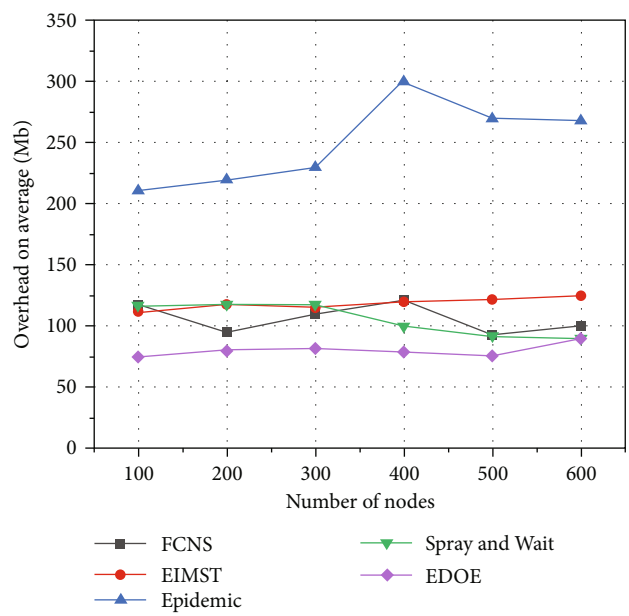


Figure 7: Overhead on average comparison of five algorithms with different number of nodes.

Figure 7 shows the performance comparison of the five algorithms' overhead on average under different numbers of nodes. As the number of nodes increases from 100 to 600, the epidemic algorithm's overhead on average has shown an upward trend. This is because too many participants generate duplicate data information and may cause relatively high network overhead during the movement. As shown in the figure, the other four algorithms' overhead

on average fluctuates little, tending to be flat or slightly rising, because these algorithms are relatively more stable. The network overhead of the EDOE algorithm is always kept to the lowest level because it is committed to storing data information and transmitting messages through the cache cooperation between multiple nodes, reducing the energy consumption of nodes.

5. Conclusions

In this study, a multiple evaluation method (EDOE) for noise reduction optimization is established. Through effective data optimization and evaluation algorithms to deal with noisy data, the problem of important information loss when data noise is lost is avoided. In addition, the information transmission task is performed by community nodes, which effectively reduces the overhead of user nodes and improves the performance of the algorithm. The EDOE method effectively avoids network congestion and packet loss and improves the overall operational efficiency of the network.

In the future, with the development of network communication technology and artificial intelligence, we will further study the trusted interaction of nodes and analyze network rules through machine learning methods according to the characteristics of nodes and the characteristics of social networks. The trust routing table is established according to the selected nodes, and the time stamp mechanism is used to prevent the routing table from being tampered with by malicious nodes during the feedback process. By preventing the competition of malicious nodes and rationally allocating network resources, the security and reliability of data transmission are further improved.

Data Availability

"Data Availability" statement data used to support the findings of this study are currently under embargo, while the research findings are commercialized. Requests for data, 12 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The Natural Science Foundation of Hunan Province (Grant No. 2020JJ4647, Grant No. 2020JJ6064) supports this work. The outstanding youth project of the Hunan Education Department (Grant No. UQ19B056) supports this work. The general project of Hunan Education Department (Grant No. 19C0171) supports this work. The general project of Changsha Technology Bureau (Grant No. KC1809015) provides this work with support.

References

[1] A. Halevy, C. Canton-Ferrer, H. Ma et al., "Preserving integrity in online social networks," *Communications of the ACM*, vol. 65, no. 2, pp. 92–98, 2022.

[2] M. Aliapoulios, E. Bevensee, J. Blackburn et al., "An early look at the parler online social network," 2021, https://arxiv.org/abs/2101.03820.

[3] F. Wilhelmi, M. Carrascosa, C. Cano, A. Jonsson, V. Ram, and B. Bellalta, "Usage of network simulators in machine-learning-assisted 5G/6G networks," *IEEE Wireless Communications*, vol. 28, no. 1, pp. 160–166, 2021.

[4] L. Li, F. Gou, and J. Wu, "Modified data delivery strategy based on stochastic block model and community detection in opportunistic social networks," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 5067849, 16 pages, 2022.

[5] F. Gou and J. Wu, "Triad link prediction method based on the evolutionary analysis with IoT in opportunistic social networks," *Computer Communications*, vol. 181, pp. 143–155, 2022.

[6] J. Xia, J. Xia, and F. Gou, "Information transmission mode and IoT community reconstruction based on user influence in opportunistic s ocial networks," *Peer-to-Peer Networking and Applications*, vol. 15, no. 3, pp. 1398–1416, 2022.

[7] M. N. Mahdi, A. R. Ahmad, Q. S. Qassim, H. Natiq, M. A. Subhi, and M. Mahmoud, "From 5G to 6G technology: meets energy, internet-of-things and machine learning: a survey," *Applied Sciences*, vol. 11, no. 17, article 8117, 2021.

[8] K. Graffi and N. Masinde, "LibreSocial: a peer-to-peer framework for online social networks," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 8, article e6150, 2021.

[9] N. Xiao, R. Xinyi, Z. Xiong et al., "A diversity-based selfish node detection algorithm for socially aware networking," *Journal of Signal Processing Systems*, vol. 93, no. 7, pp. 811–825, 2021.

[10] F. Liu, F. Gou, and J. Wu, "An attention-preserving network-based method for assisted segmentation of osteosarcoma MRI images," *Mathematics*, vol. 10, no. 10, article 1665, 2022.

[11] Y. Tan, F. Gou, and Y. Tan, "A staging auxiliary diagnosis model for nonsmall cell lung cancer based on the intelligent medical system," *Computational and Mathematical Methods in Medicine*, vol. 2021, Article ID 6654946, 15 pages, 2021.

[12] S. Sultan, Q. Javaid, A. J. Malik, F. Al-Turjman, and M. Attique, "Collaborative-trust approach toward malicious node detection in vehicular ad hoc networks," *Environment, Development and Sustainability*, vol. 24, no. 6, pp. 7532–7550, 2022.

[13] J. Wu, F. Gou, and X. Tian, "Disease control and prevention in rare plants based on the dominant population selection method in opportunistic social networks," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 1489988, 16 pages, 2022.

[14] W. Li, K. Zhong, J. Wang, and D. Chen, "A dynamic algorithm based on cohesive entropy for influence maximization in social networks," *Expert Systems with Applications*, vol. 169, article 114207, 2021.

[15] Y. Lu, L. Chang, J. Luo, and J. Wu, "Routing algorithm based on user adaptive data transmission scheme in opportunistic social networks," *Electronics*, vol. 10, no. 10, p. 1138, 2021.

[16] S. Yang, Z. Zhou, P. Xie, N. Xu, and Z. Dai, "Intelligent segmentation medical assistance system for MRI images of osteosarcoma in developing countries," *Computational and Mathematical Methods in Medicine*, vol. 2022, Article ID 7703583, 2022.

[17] F. Gou and J. Wu, "Message transmission strategy based on recurrent neural network and attention mechanism in IoT system," *Journal of Circuits, Systems and Computers*, vol. 31, no. 7, 2022.

[18] Y. Shen and Z. Dai, "Osteosarcoma MRI image-assisted segmentation system base on guided aggregated bilateral network," *Mathematics*, vol. 10, no. 7, p. 1090, 2022.

[19] C. Durán, A. Muscoloni, and C. V. Cannistraci, "Geometrical inspired pre-weighting enhances Markov clustering community detection in complex networks," *Applied Network Science*, vol. 6, no. 1, p. 29, 2021.

[20] W. Yang and J. Luo, "Application of information transmission control strategy based on incremental community division in IoT platform," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21968–21978, 2021.

[21] M. Zhou, H. Jin, Q. Wu, H. Xie, and Q. Han, "Betweenness centrality-based community adaptive network representation for link prediction," *Applied Intelligence*, vol. 52, no. 4, pp. 3545–3558, 2022.

[22] M. Xiao, J. Wu, and L. Huang, "Community-aware opportunistic routing in mobile social networks," *IEEE Transactions on Computers*, vol. 63, no. 7, pp. 1682–1695, 2014.

[23] H. Zhou, V. C. M. Leung, C. Zhu, S. Xu, and J. Fan, "Predicting temporal social contact patterns for data forwarding in opportunistic mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10372–10383, 2017.

[24] C. Jeong and W. Shin, "Network-decomposed hierarchical cooperation in ad hoc networks with social relationships," *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7606–7619, 2018.

[25] A. Mtibaa and K. A. Harras, "Social-based trust in mobile opportunistic networks," in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, Maui, HI, 2011.

[26] D. K. Sharma, S. Singh, V. Gautam, S. Kumaram, M. Sharma, and S. Pant, "Ant router: an efficient routing protocol for social opportunistic networks using ant routing," *IET Networks*, vol. 9, no. 2, pp. 83–93, 2020.

[27] F. Xia, L. Liu, J. Li, A. M. Ahmed, L. T. Yang, and J. Ma, "BEE-INFO: interest-based forwarding using artificial bee colony for socially aware networking," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1188–1200, 2015.

[28] L. Yao, Y. Man, Z. Huang, J. Deng, and X. Wang, "Secure routing based on social similarity in opportunistic networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 594–605, 2016.

[29] P. Goswami, A. Mukherjee, R. Hazra et al., "AI based energy efficient routing protocol for intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1670–1679, 2022.

[30] A. Mukherjee, P. Goswami, and L. Yang, "DAI based wireless sensor network for multimedia applications," *Multimedia Tools and Applications*, vol. 80, no. 11, pp. 16619–16633, 2021.

[31] Y. Deng, F. Gou, and J. Wu, "Hybrid data transmission scheme based on source node centrality and community reconstruction in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 6, pp. 3460–3472, 2021.

[32] J. Luo, J. Wu, and Y. Wu, "An efficient data transmission strategy for edge-computing-based opportunistic social networks," in *Network and Parallel Computing. NPC 2020*, X. He, E. Shao, and G. Tan, Eds., vol. 12639 of Lecture Notes in Computer Science, pp. 323–335, Springer, Cham, 2021.

[33] Y. Dong, L. Chang, J. Luo, and J. Wu, "A routing query algorithm based on time-varying relationship group in opportunistic social networks," *Electronics*, vol. 10, no. 13, p. 1595, 2021.

[34] J. Zhou, H. Wu, Y. Lin, W. Liang, and Q. Liu, "Multi-community opportunistic routing algorithm based on machine learning in the internet of vehicles," in *2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pp. 194–199, New York, USA, 2021.

[35] J. Zhao, X. You, Q. Duan, and S. Liu, "Multiple ant colony algorithm combining community relationship network," *Arabian Journal for Science and Engineering*, pp. 1–16, 2022.

[36] X. Zhou, W. Liang, Z. Luo, and Y. Pan, "Periodic-aware intelligent prediction model for information diffusion in social networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 894–904, 2021.

[37] B. Yi, X. Wang, and M. Huang, "Content delivery enhancement in vehicular social network with better routing and caching mechanism," *Journal of Network and Computer Applications*, vol. 177, article 102952, 2021.

[38] S. Yang, U. Adeel, Y. Tahir, and J. A. McCann, "Practical opportunistic data collection in wireless sensor networks with mobile sinks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1420–1433, 2017.

[39] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2022.

[40] L. Yuan, Q. He, F. Chen et al., "CSEdge: enabling collaborative edge storage for multi-access edge computing based on blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1873–1887, 2021.

[41] Y. Chen, F. Zhao, Y. Lu, and X. Chen, *Dynamic task offloading for mobile edge computing with hybrid energy supply*, Tsinghua Science and Technology, 2019.

[42] J. Wu, Z. G. Chen, and M. Zhao, "Weight distribution and community reconstitution based on communities communications in social opportunistic networks," *Peer-to-Peer Networking and Applications*, vol. 12, no. 1, pp. 158–166, 2019.

[43] L. Qi, H. Song, X. Zhang, G. Srivastava, X. Xu, and Y. Shui, "Compatibility-aware web API recommendation for mashup creation via textual description mining," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 1s, pp. 1–19, 2021.

[44] X. Zhang, L. Chang, J. Luo, and J. Wu, "Effective communication data transmission based on community clustering in opportunistic social networks in IoT system," *Journal of Intelligent & Fuzzy Systems.*, vol. 41, no. 1, pp. 2129–2144, 2021.

[45] Q. Xu, Z. Su, K. Zhang, P. Ren, and X. S. Shen, "Epidemic information dissemination in mobile social networks with opportunistic links," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, pp. 399–409, 2015.

[46] Y. I. N. Sheng and Y. U. Genghua, "Low energy consumption routing algorithm based on message importance in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 2, pp. 948–961, 2021.

[47] X. Wang, J. Hu, H. Lin et al., "QoS and privacy-aware routing for 5G-enabled industrial internet of things: a federated reinforcement learning approach," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4189–4197, 2022.

[48] Z. Kuang, Z. Ma, Z. Li, and X. Deng, "Cooperative computation offloading and resource allocation for delay minimization in mobile edge computing," *Journal of Systems Architecture*, vol. 118, article 102167, 2021.

[49] L. Qi, Y. Yang, X. Zhou, W. Rafique, and J. Ma, "Fast anomaly identification based on multi-aspect data streams for intelligent intrusion detection toward secure industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6503–6511, 2022.

[50] Y. Zhao, W. Song, and Z. Han, "Social-aware data dissemination via device-to-device communications: fusing social and mobile networks with incentive constraints," *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 489–502, 2019.

[51] A. Braeken, "Public key versus symmetric key cryptography in client–server authentication protocols," *International Journal of Information Security*, vol. 21, no. 1, pp. 103–114, 2022.

[52] M. J. Hossain, C. Xu, Y. Zhang, X. Zhang, and W. Li, "LAMA: a secure lattice-based authentication scheme for cloud storage against misbehaved private key generator," *Journal of Ambient Intelligence and Humanized Computing*, 2022.

[53] Y. Shen, F. Gou, and J. Wu, "Node screening method based on federated learning with IoT in opportunistic social networks," *Mathematics*, vol. 10, no. 10, p. 1669, 2022.

[54] X. Li, H. Qi, and J. Wu, "Node social nature detection OSN routing scheme based on IoT system," *IEEE Internet of Things Journal*, p. 1, 2022.

[55] W. Xiong and X. Zhou, "A reputation value-based task-sharing strategy in opportunistic complex social networks," *Complexity*, vol. 2021, Article ID 8554351, 16 pages, 2021.

[56] J. Baek and Y. Zheng, "Identity-based threshold signature scheme from the bilinear pairings (extended abstract)," in *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004*, pp. 124–128, Beijing, China, 2004.

[57] "Website of CRAWDAD project," http://crawdad.org/cambrige/haggle/20090529/.

[58] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.

[59] Z. G. Chen and M. Zhao, "Information cache management and data transmission algorithm in opportunistic social networks," *Wireless Networks*, vol. 25, pp. 2977–2988, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8805416, 13 pages
https://doi.org/10.1155/2022/8805416

Research Article

Deep Reinforcement Learning-Based UAV Data Collection and Offloading in NOMA-Enabled Marine IoT Systems

Yanpeng Dai, Ziyi Liang, Ling Lyu, and Bin Lin

School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China

Correspondence should be addressed to Ling Lyu; jidalvling@126.com

Received 29 January 2022; Revised 22 April 2022; Accepted 26 April 2022; Published 12 May 2022

Academic Editor: Qiang Ye

The rapid growth of maritime wireless communication demand and the complex offshore wireless communication environment have brought challenges to ensure the real-time and reliability of data transmission in the marine Internet of Things (MIoT). Unmanned aerial vehicles (UAVs) have great advantages in enhancing coverage and channel quality. Hence, we investigate a UAV-assisted data collection and data offloading system based on nonorthogonal multiple access (NOMA) technology in this paper. We jointly optimize the buoy-UAV association relationship, transmit powers, and the UAV trajectory to minimize the total mission completion time while ensuring data transmission requirements. We first propose a UAV trajectory optimization algorithm based on deep reinforcement learning (DRL). Then, we design a heuristic algorithm to effectively solve the subproblem of power control and the association relationship. Finally, we propose a joint optimization scheme to solve the minimization problem. Simulation results show the effectiveness of the proposed scheme.

1. Introduction

Marine environmental monitoring is indispensable with the continuous increase of human marine activities. A large amount of meteorological and hydrological data leads to an increase in the demand for maritime wireless communication [1, 2]. Buoys are widely deployed in the ocean due to their low cost and flexible deployment. With the development of technology, buoys can be used for marine environment monitoring with a variety of sensors and communication equipment and can also be powered by power supply methods such as lithium-ion batteries and solar energy [3, 4]. However, the transmit power of the buoy is limited. Traditional maritime wireless communication methods, such as land base stations and satellites, have disadvantages such as limited coverage and long transmission distance, which seriously affect the real-time and reliability of information transmission [5]. For the current five-generation (5G) and the upcoming six-generation (6G) era, it is of great significance to build an efficient and dynamic maritime communication network [6]. Therefore, the UAV-assisted wireless communication system (UWCS) has received widespread attention.

Unmanned aerial vehicles (UAVs) have the advantages of maneuverability and easy manipulation, which can be deployed on demand and enlarge coverage [7]. It is easier to establish a line-of-sight (LoS) channel and a stronger communication link with the target device, which can better deal with the variable ocean environment [8, 9]. In the marine Internet of Things (MIoT), aiming at the problem of the large number and wide distribution of buoys, UAV can act as a mobile base station, collecting data collected by buoys from the target area and offloading the data to the OBS [10, 11]. Furthermore, the limited spectrum resources of MIoT also pose a challenge to the reliability and efficiency of data transmission. Nonorthogonal multiple access (NOMA) technology is considered a promising technology in the 5G era [12]. Compared with orthogonal multiple access (OMA) technology, NOMA greatly improves the spectrum efficiency in the presence of limited spectrum resources by allowing multiple users to access simultaneously in the same channel and relying on power domain

multiplexing and successive interference cancellation (SIC) decoding technology [13–15].

Recently, much research has applied NOMA technology to UAV-assisted wireless communication system. Zhao et al. in [13] investigated a NOMA-assisted UAV large-scale IoT data collection system and proposed a data collection optimization algorithm. The results show that, compared with the traditional UWCS, the NOMA-based UWCS has better performance in data collection. W. Chen et al. in [16] maximized the sum rate of the UAV-assisted uplink NOMA system by jointly optimizing the UAV location, buoy sensor grouping, and power control. The simulation results show the performance gain of NOMA in the sum rate of the system. Tang et al. in [17] investigated the scenario of a UAV-assisted marine wireless communication downlink in which the UAV hovers continuously to provide services to multiple groups of ships. Obviously, the NOMA-based UWCS has great advantages in enhancing coverage and strengthening communication links.

In the above work, the optimization problem is usually formulated as a mixed-integer nonconvex problem. They can usually be divided into several subproblems, which can be solved by traditional optimization techniques and iterative algorithms [18]. However, the above solutions may have high computational complexity. Furthermore, the buoys associated with the UAV and NOMA cochannel interference vary with UAV position. The complex dynamic changes bring great challenges to the traditional convex optimization technology as well. With the development of machine learning technology, reinforcement learning (RL) is considered to be an effective solution to the high-dynamic environment [19–21]. Deep reinforcement learning (DRL) solves the continuous state space problem that RL cannot solve by introducing deep neural network (DNN), such as deep Q-learning (DQN) and deep deterministic policy gradient (DDPG) [22]. At present, a lot of work has focused on the research of UWCS based on DRL. L. Wang et al. in [23] minimized the energy consumption of all user equipment by jointly optimizing UAV trajectories, user associations, and resource allocation. Two algorithms are proposed to effectively solve the minimization problem based on convex optimization and DRL technology, respectively. The results show that the DRL-based method is better than the convex optimization method. Zhang et al. in [24] studied the UAV lineup and user distribution change scenarios and developed a DDPG-based proactive self-regulation method for UAV networks, which is based on the proposed asynchronous parallel computing architecture. Wang et al. in [25] studied a UAV-assisted mobile edge computing system. They minimized the maximum processing delay and proposed a DDPG-based algorithm to solve the high-dimensional state space and continuous action space. However, the flight action space of actual UAV is continuous and high dimensional, which may bring dimensional disaster to traditional reinforcement learning methods (such as DQN) [23, 26]. DDPG exists the problem of overestimation. To solve the above problems, Fujimoto et al. in [27] proposed the twin-delayed deep deterministic (TD3) algorithm based on DDPG. In [28], Sun et al. considered the age of information (AoI) and energy consumption and proposed an AoI-energy-aware UAV trajectory optimization algorithm based on TD3.

In this paper, we investigate a UAV-assisted data collection and data offloading system in MIoT. Specifically, buoys are used to collect marine environment information in the sensing layer. The UAV collects the sensing information from buoys based on NOMA technology and offloads the collected data to the OBS. Our goal is to minimize the total mission completion time of the UAV by jointly optimizing the UAV trajectory, the buoy-UAV association relationship, the UAV transmit power, and the buoys transmit power. The main contributions of our paper are listed as follows.

(i) We jointly consider the UAV trajectory, buoy-UAV association relationship, and transmit powers to investigate the UAV's total mission completion time minimization problem. The above minimization problem is a mixed-integer nonconvex problem. Accordingly, we divide the total mission process of UAV into data collection stage and data offloading stage for analysis

(ii) We propose a UAV trajectory optimization algorithm based on TD3 to solve the UAV trajectory coupling of data collection and data offloading since the minimization problem is a mixed-integer nonconvex problem. Furthermore, we design a heuristic algorithm to effectively solve the above problem due to the coupling between the buoy-UAV association relationship and the buoys transmit power

(iii) We propose a joint TD3-based trajectory optimization, power control, and buoy-UAV association relationship scheme that effectively solves the mixed-integer nonconvex problem. The simulation results show that the proposed scheme can effectively shorten the UAV's total mission completion time while ensuring that the data transmission requirements are met

The remainder of this paper is organized as follows. Section 2 presents the system model and the problem formulation. Section 3 briefly introduces the TD3 algorithm and proposes the TD3-based UAV trajectory optimization algorithm. Then, we design a heuristic algorithm to solve the subproblem of power control and buoy-UAV association relationship. Finally, we propose a joint optimization scheme for the minimization problem. Simulation results and conclusion are given in Sections 4 and 5.

2. System Model and Problem Formulation

2.1. Network Model. We consider a UAV-assisted MIoT system as shown in Figure 1, which includes a UAV base station, M buoys, and an OBS. Each buoy senses and stores hydrometeorological data and is powered by a lithium-ion battery to ensure that it has sufficient energy to transmit data. The total mission time of the UAV is denoted as T_{total} and divided into K time slots. The time slot length is

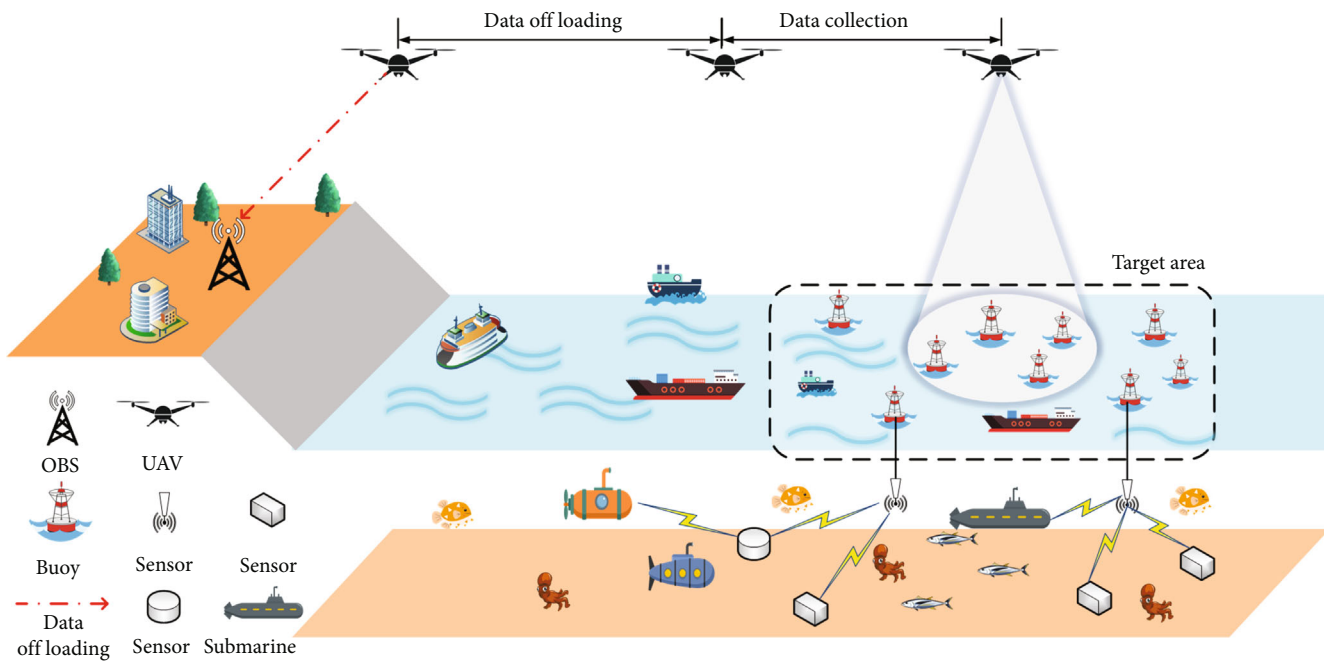


Figure 1: UAV-assisted marine IoT system.

δ. The total mission process of UAV consists of two stages. The first stage is that the UAV utilizes NOMA to collect data from M buoys, and the number of time slots of this stage is K_{co}. The UAV is allowed to collect data from at most U buoys in each time slot where $U \leq M$. The second stage is that the UAV offloads all the collected data to the OBS after completing the first stage, and the number of time slots of this stage is K_{of}. Therefore, T_{total} can be expressed as

$$T_{\text{total}} = K\delta = (K_{\text{co}} + K_{\text{of}})\delta. \tag{1}$$

Let $\mathscr{M} = \{1, 2, \cdots, M\}$ denote the set of all buoys. We denote that the horizontal coordinate of the UAV in the k-th time slot is $q_k = (x_k, y_k)$, where $k \in \mathscr{K} = \{1, \cdots, K_{\text{co}}, K_{\text{co}} + 1, \cdots, K\}$. Let $\mathscr{K}_{\text{co}} = \{1, 2, \cdots, K_{\text{co}}\}$ denote the set of the data collection time, and $\mathscr{K}_{\text{of}} = \{K_{\text{co}} + 1, \cdots, K\}$ denote the set of the data offloading time.

The fixed flight height of UAV is H, and the flight velocity of UAV in the k-th time slot is V_k. Then, the UAV should follow the maximum flight velocity constraints, which are expressed as

$$\begin{aligned} V_{k+1} &= V_k + \Delta_k\delta, \forall k \in \mathscr{K}, \\ V_{\min} \leq \|V_k\| &\leq V_{\max}, \forall k \in \mathscr{K}, \\ \|\Delta_k\| &\leq \Delta_{\max}, \forall k \in \mathscr{K}, \\ q_{k+1} &= q_k + V_k\delta + \frac{1}{2}\Delta_k\delta^2, \forall k \in \mathscr{K}, \end{aligned} \tag{2}$$

where $V_{\max}$ and $\Delta_{\max}$ are the maximum flight velocity and acceleration, respectively, and $V_{\min}$ is the minimum flight velocity.

The horizontal coordinate of the m-th buoy is $D_m = (x_m, y_m)$. The horizontal coordinate of OBS is $D_0 = (x_0, y_0)$. If the time slot δ is small enough, the motion of the UAV in each time slot can be regarded as static approximately. Hence, the distance between UAV and the m-th buoy at the k-th time slot is $d_{m,k} = \sqrt{\|q_k - D_m\|^2 + (H - H_m)^2}$, $\forall k \in \mathscr{K}_{\text{co}}$, $\forall m \in \mathscr{M}$. In the stage of data offloading, the distance between UAV and OBS at the k-th time slot is $d_{0,k} = \sqrt{\|q_k - D_0\|^2 + (H - H_0)^2}$, $\forall k \in \mathscr{K}_{\text{of}}$. H_m and H_0 denote the antenna heights of the m-th buoy and OBS, respectively.

2.2. Transmission Model

2.2.1. Channel Model. We adopt the model of the air-to-ground channel and the two-ray path loss model [29] and give the LoS and NLoS path loss models of the buoy-UAV and UAV-OBS links, respectively.

Specifically, the channel gain of buoy-UAV and UAV-OBS links is expressed as $h_{i,k} = 1/L_{i,k}$, $\forall k \in \mathscr{K}, \forall i \in \{0, 1, \cdots, M\}$, where $L_{i,k}$ denotes the average path loss in the k-th time slot, expressed as

$$\begin{aligned} L_{i,k} &= P_{i,k}^{\text{LoS}} L_{i,k}^{\text{LoS}} + \left(1 - P_{i,k}^{\text{LoS}}\right) L_{i,k}^{\text{NLoS}}, \\ L_{i,k}^{\text{LoS}} &= \left(\frac{4\pi d_{i,k}}{\lambda}\right)^2 \mu_{i,k}\xi_{\text{LoS}}, \\ L_{i,k}^{\text{NLoS}} &= \left(\frac{4\pi d_{i,k}}{\lambda}\right)^2 \mu_{i,k}\xi_{\text{NLoS}}, \end{aligned} \tag{3}$$

where $L_{i,k}^{\text{LoS}}$ and $L_{i,k}^{\text{NLoS}}$ are the average path loss for LoS and NLoS, $\mu_{i,k} = 1$. ξ_{LoS} and ξ_{NLoS} are the excessive path loss for LoS and NLoS paths, respectively, λ is the wavelength, and $P_{i,k}^{\text{LoS}}$ denotes the probability of LoS link which is expressed as

$$P_{i,k}^{\mathrm{LoS}} = \frac{1}{1 + a \exp\left(-b\left(\psi_{i,k} - a\right)\right)}, \tag{4}$$

where a and b are two constant values depending on the environment and $\psi_{i,k}$ denotes the elevation angle between the m-th buoy (or OBS) and UAV, which is given by $\psi_{i,k} = (180^\circ/\pi) \times \arcsin\left(H - H_i/\sqrt{\|q_k - D_{i,k}\|^2}\right)$.

2.2.2. UAV Data Collection from Buoys. In this stage that $\forall k \in \mathscr{K}_{\mathrm{co}}$, let $\alpha_{m,k}$ denote the association indicator between m-th buoy and UAV. $\alpha_{m,k} = 1$ means that the m-th buoy is associated with the UAV in the k-th time slot. Otherwise, $\alpha_{m,k} = 0$.

In uplink NOMA system, the UAV is regarded as a receiver to receive signals from multiple buoys at the same time and allows multiple buoys to share the same channel. The SIC decoding technique is used to demodulate the received signals with different received power levels. The successfully demodulated signal is deleted from all received signals, and the later decoded signal receives less cochannel interference. Therefore, the buoy with high channel gain is usually demodulated first, and its interference comes from the buoys with worse channel gain [30, 31]. The cochannel interference of uplink transmission between the m-th buoy and UAV in the k-th time slot can be given by

$$I_{m,k} = \sum_{i \in \mathscr{M}_k} \alpha_{i,k} P_{i,k} h_{i,k}, \forall k \in \mathscr{K}_{\mathrm{co}}, \tag{5}$$

where $\mathscr{M}_k = \{i | i \in \mathscr{M}, h_{m,k} > h_{i,k}\}$ is the set of the buoy whose channel gain is worse than the m-th buoy in the k-th time slot.

Hence, the signal-to-interference-noise-ratio (SINR) between the m-th buoy and UAV at the k-th time slot is expressed as

$$g_{m,k} = \frac{P_{m,k} h_{m,k}}{\sigma^2 + I_{m,k}}, \forall m \in \mathscr{M}, \forall k \in \mathscr{K}_{\mathrm{co}}, \tag{6}$$

where $P_{m,k}$ is the transmit power of the m-th buoy in the k-th time slot and σ^2 is the noise power.

The transmission rate of m-th buoy in the k-th time slot is expressed as

$$R_{m,k} = \alpha_{m,k} B \log_2\left(1 + g_{m,k}\right), \forall m \in \mathscr{M}, \forall k \in \mathscr{K}_{\mathrm{co}}, \tag{7}$$

where B is the spectrum. In order for the received signal to be demodulated successfully, the SIC demodulation condition that SINR needs to meet is as [32]

$$\frac{\alpha_{m,k} P_{m,k} h_{m,k}}{\sum_{i \in \mathscr{M}_k} \alpha_{i,k} P_{i,k} h_{i,k} + \sigma^2} \geq \eta_{\mathrm{SIC}}, \forall m \in \mathscr{M}, \tag{8}$$

where η_{SIC} denotes the SIC threshold.

2.2.3. UAV Data Offloading to OBS. In this stage that $\forall k \in \mathscr{K}_{\mathrm{of}}$, let β_k denote the association indicator between UAV and OBS. $\beta_k = 1$ means that the UAV can offload the data to the OBS at the k-th time slot. Otherwise, $\beta_k = 0$.

The signal-to-noise-ratio (SNR) between the UAV and OBS at the k-th time slot needs to satisfy the following condition:

$$g_{0,k} = \frac{P_k h_{0,k}}{\sigma^2} \geq \bar{g}_0, \forall k \in \mathscr{K}_{\mathrm{of}}, \tag{9}$$

where P_k is the transmit power of the UAV and $\bar{g}_0$ is the SNR threshold.

The transmission rate between the UAV and OBS at the k-th time slot is given by

$$R_{0,k} = \beta_k B \log_2\left(1 + g_{0,k}\right), \forall k \in \mathscr{K}_{\mathrm{of}}. \tag{10}$$

2.3. Problem Formulation. Our goal is to minimize the UAV's total mission completion time by jointing optimization of the buoy-UAV association relationship, UAV transmit power, buoys transmit power, and UAV trajectory. Let $A_m = \{\alpha_{m,k}, \forall m \in \mathscr{M}, \forall k \in \mathscr{K}_{\mathrm{co}}\}$ denote the buoy-UAV-associated variables, $P = \{P_k, \forall k \in \mathscr{K}_{\mathrm{of}}\}$ denote the UAV transmit power during data offloading, $P_m = \{P_{m,k}, \forall m \in \mathscr{M}, \forall k \in \mathscr{K}_{\mathrm{co}}\}$ denote the buoys transmit power, and $Q = \{q_k, \forall k \in \mathscr{K}\}$ denote the UAV trajectory. The total mission completion time minimization problem can be formulated as

$$\begin{aligned}
(P1): &\min_{\mathbf{A}_m, \mathbf{P}, \mathbf{P}_m, \mathbf{Q}} T_{\mathrm{total}}, \\
\mathrm{s.t.} C1: &\, 0 \leq P_{m,k} \leq P_{m_{\max}}, \\
C2: &\, 0 \leq P_k \leq P_{\max}, \\
C3: &\, \alpha_{m,k} \in \{0, 1\}, \forall m \in \mathscr{M}, \forall k \in \mathscr{K}_{\mathrm{co}}, \\
C4: &\, \beta_k \in \{0, 1\}, \forall k \in \mathscr{K}_{\mathrm{of}}, \\
C5: &\, \sum_{m=1}^{M} \alpha_{m,k} \leq U, \forall k \in \mathscr{K}_{\mathrm{co}}, \\
C6: &\, \sum_{k=K_{\mathrm{co}}+1}^{K} R_{0,k} \delta \geq \sum_{m=1}^{M} C_m, \\
C7: &\, \sum_{k=1}^{K_{\mathrm{co}}} R_{m,k} \delta \geq C_m, \forall m \in \mathscr{M}, \\
C8 - C11: &\, (3), (4), (13), (14)
\end{aligned} \tag{11}$$

In problem $P1$, $C1$, and $C2$ restrict the maximum transmit power of UAV and buoy, respectively. $C5$ limits the maximum number of buoys that can be associated with the UAV in each time slot. Let C_m denote the data size that needs to be collected in m-th buoy. $C6$ ensures that all data collected by the UAV is offloaded to OBS. $C7$ ensures that the data collection requirements of each buoy are met. $C8$ and $C9$ are the UAV maximum velocity and maximum acceleration constraints, respectively. $C10$ is the SIC demodulation constraint that SINR needs to meet in the data collection stage. $C11$ is the SNR constraint in the data offloading. Problem $P1$ is a mixed-integer nonconvex

problem since it contains binary relational variables, which makes it difficult to be solved effectively.

3. Proposed Scheme

In order to solve problem $P1$, we first propose a TD3-based UAV trajectory optimization algorithm (TTO). Then, we design a heuristic algorithm to solve the power control problem while determining the buoy-UAV association relationship (PCAR). Finally, the above two algorithms are combined to effectively solve the problem $P1$.

3.1. TD3-Based UAV Trajectory Optimization. In our system, the start position of data offloading stage is similar to the end position of data collection, which is named as the transition position (TP) between two stages. Therefore, the UAV trajectories of these two stages are coupled, and the TP cannot be determined in advance. Furthermore, the UAV trajectory changes dynamically according to the requirements of data collection and data offloading. The traditional deterministic optimization method is difficult to solve the above problems [19]. Therefore, in this paper, we use an advanced DRL method, TD3, to solve the UAV trajectory optimization subproblem with the given transmit powers and association relationships. In the following, we first give the definitions of state, action, and reward and then briefly introduce the TD3.

3.1.1. State Definition. In the data collection stage, the UAV's action is closely related to the remaining data size of buoys and the buoy-UAV association relationship. Similarly, in the data offloading stage, the UAV's action is related to the remaining data size of UAV and the UAV-OBS association relationship. Furthermore, the UAV only collects data from buoys in the target area. Hence, we define the state space as

$$S_k = \{\alpha_k, c_k, x_k, y_k, \beta_k, c_k^{\text{uav}}, \rho_k\}, \tag{12}$$

where the variables contained in the above expression are defined as

(i) $\alpha_k = \{\alpha_{1,k}, \alpha_{2,k}, \cdots, \alpha_{M,k}\}$ denotes the set of the buoy-UAV association relationship in the k-th time slot

(ii) $c_k = \{c_{1,k}, c_{2,k}, \cdots, c_{M,k}\}$ denotes the remaining data size of each buoy in the k-th time slot

$$c_{m,k} = C_m - \sum_{j=1}^{k-1} R_{m,j}\delta \tag{13}$$

(iii) c_k^{uav} denotes the remaining data size of UAV

$$c_k^{\text{uav}} = \sum_{k=1}^{K_{\text{co}}} \sum_{m=1}^{M} R_{m,k}\delta - \sum_{j=K_{\text{co}}+1}^{k-1} R_{0,j}\delta \tag{14}$$

(iv) ρ_k denotes the boundary penalty information of the UAV to judge whether the UAV position exceeds the target area in the k-th time slot

3.1.2. Action Definition. Based on the above state and environment information, the UAV's action is defined as

$$A_k = \{\varphi_k, V_k\}, \tag{15}$$

where $\varphi_k \in (0, 2\pi]$ denotes the flight angle of UAV in the k-th time slot and $V_k \in [0, V_{\max}]$.

3.1.3. Reward Definition. Let $K_{\max}$ denote the number of maximum total mission completion time slots. $K^* = K_{\max} - K$ if the UAV completes the mission in K time slots. Then, we design the reward function as

$$r_k = \begin{cases} R_k\delta + \rho_k + K^*, & \text{if C6 and C7 are satisfied,} \\ C_k(\text{or } R_k\delta) + \rho_k, & \text{otherwise.} \end{cases} \tag{16}$$

It can be seen from (16) that the shorter the time it takes for the UAV to complete the mission, the greater the reward it will eventually obtain.

Note that for better performance, we reduce the order of magnitude of B and C_m by n_1 orders of magnitude to be less than or equal to the order of magnitude of $K_{\max}$ when calculating the state and reward.

3.1.4. TD3. The TD3 has the following advantages [27]:

(i) Clipped double Q-learning for actor-critic: TD3 contains two critic networks. For the two target Q-values generated by the two critic target networks, the minimum of them is selected to suppress the overestimation problem caused by high variance, expressed as

$$y = r_k + \gamma \min_{i=1,2} Q_{\theta_{i'}}\left(S', \tilde{A}\right) \tag{17}$$

where y is the target value which is used to update the two critic networks, r is reward, γ is the reward discount factor, $\tilde{A}$ is the target policy, $\theta_{i'}$ is the target network parameter, and S' is the next state

(ii) Target networks and delayed policy updates: the TD3 algorithm updates the actor and its target network after a fixed number of updates to the critic network, expressed as

$$\theta_{i'} \longleftarrow \tau\theta_i + (1-\tau)\theta_{i'}, i = 1, 2 \tag{18}$$

where τ is the update parameter

(iii) Target policy smoothing regularization: TD3 smoothes the estimate and reduces the error by adding a small amount of random noise to the target actor network and averaging over minibatch, expressed as

$$\tilde{A} \longleftarrow \pi_{\phi'}(S_k) + \varepsilon, \varepsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\omega}), -c, c) \tag{19}$$

It can be seen from (12) that most of the state dimensions are related to buoys. Only two dimensions are related to the UAV's position, two dimensions are related to OBS, and one dimension is related to UAV boundary penalty information. Therefore, there is a problem of dimension imbalance. Dimension spread technology can effectively solve this problem. We spread the above state dimensions. For example, we connect the position state dimension of UAV to a spread network composed of M neurons and spread its dimension to M [33, 34]. Furthermore, we set a termination flag l to indicate whether UAV has completed its mission. l is applied to the target value function. Hence, the Q-value of the target value function is 0 after the UAV completes the mission, so as to make the critic learning performance more stable [34].

In summary, our proposed algorithm TTO is shown in Algorithm 1. The update process of TD3 is shown in Figure 2. In lines 1-3, we first initialize the network parameters and the experience replay buffer $\mathscr{B}$. In lines 5-17, we initialize the environment, obtain the initial state information, and set a time variable k to represent the time spent by UAV to perform the mission. Moreover, a termination mark l is set. $l = 0$ indicates that UAV has not completed the mission. Then, UAV makes an action selection according to the observed state and environmental information. Specifically, UAV constantly interacts with the environment and updates the actor and critic networks. The actor network outputs the action A_k to be executed by the UAV according to the state information S_k. Then, the transition information (S_k, A_k, r_k, S', l) is stored in $\mathscr{B}$. If the UAV flies beyond the target area, the action will be canceled and the boundary penalty information will be given. In the data collection stage, we remain $\beta_k = 0$. The data offloading stage begins when UAV completes the data collection mission. Then, we remain $\alpha_k = 0$. $l = 1$ if the UAV completes the data offloading mission. The episode is terminated when $l = 1$ or $k = K_{\max}$.

In lines 18-28, N transition information is randomly selected from $\mathscr{B}$ to form a minibatch, which is input into the actor and critic network. The actor network calculates the corresponding $\tilde{A}$ according to S'. After selecting the target Q-value and smoothing the target policy according to S' and $\tilde{A}$, the critic network minimizes the loss function and updates the critical network by the following way:

$$\theta_i \longleftarrow \arg \min_{\theta_i} N^{-1} \sum \left(y - Q_{\theta_i}(S_k, A_k)\right)^2. \tag{20}$$

Then, the actor network is updated in the way of delayed update by the deterministic policy gradient, expressed as

$$\nabla_\phi J(\phi) = N^{-1} \sum \nabla_{A_k} Q_{\theta_1}(S_k, A_k)\Big|_{A_k = \pi_\phi(S_k)} \nabla_\phi \pi_\phi(S_k). \tag{21}$$

Finally, the optimal trajectory of UAV is obtained by cyclic iteration until the maximum number of episodes $E_{\max}$.

3.2. Power Control and Buoy-UAV Association Relationship. Given the UAV trajectory, the problem $P1$ can be written as

$$\begin{aligned} (\text{P2}): \min_{\mathbf{A}_m, \mathbf{P}, \mathbf{P}_m, \mathbf{Q}} \quad & T_{\text{total}}, \\ \text{s.t.} \quad & \text{C1} - \text{C7}, \text{C10}, \text{C11} \end{aligned} \tag{22}$$

Obviously, the problem $P2$ is still a mixed-integer nonconvex problem. Given the mission completion time and the association relationships of buoy-UAV and UAV-OBS, $P2$ can be transformed into a problem of maximizing the total transmission data size in the k-th time slot, which can be divided into two parts.

First, in the data collection stage, it can be seen from problem $P2$ that the SINR between UAV and buoys is related not only to the transmit power of buoys but also to the buoy-UAV association relationship. Therefore, in order to determine the buoy-UAV association relationship, we first introduce Lemmas 1 and 2.

Lemma 1. *The UAV must be associated with the first U^* $(U^* \le U)$ buoys with the larger channel gain in the k-th time slot.*

Proof. As can be seen from $P2a$, the total transmission data size C_k depends on the summation term $\sum_{m=1}^{U} P_m h_m$. Therefore, we might as well assume that the transmit power of all buoys is the maximum transmission power $P_{m_{\max}}$. The channel gain between M buoys and UAV is expressed in descending order as $\{h_1 \longrightarrow h_2 \longrightarrow \cdots \longrightarrow h_M\}$. Obviously, selecting the first U^* buoys with the largest channel gain can maximize the total transmission data size. □

Lemma 2. *The transmit power of the buoy with the largest channel gain among the buoys associated with the UAV in the k-th time slot must be $P_{m_{\max}}$.*

Proof. Except for the buoys transmit power, other assumptions are the same as Proof. If the UAV is associated with U buoys, for the buoy with the largest channel gain, the SINR between it and the UAV is expressed as

$$g_1 = \frac{P_1 h_1}{\sum_{i=2}^{U} P_i h_i + \sigma^2}. \tag{23}$$

In order to meet the constraints $C10$ and $C12$ and maximize the total transmission data size, the value of

Initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ with random parameters θ_1, θ_2, and ϕ.
Initialize target networks $\theta_1{}' \longleftarrow \theta_1$, $\theta_2{}' \longleftarrow \theta_2$, and $\phi' \longleftarrow \phi$.
Initialize experience replay buffer $\mathscr{B}$.
for episode =0 **to** $E_{\max}$ **do**
  Initialize the environment and state S_0, and the terminated flag $l = 0$.
  for epoch $k = 1$ **to** $K_{\max}$ **do**
    Select action $A_k = \pi_\phi(S_k) + \varepsilon$, $\varepsilon \sim \mathscr{N}(0, \omega)$, and observe reward r_k and next state S'.
    if the UAV flies beyond the target area **then**
      $\rho_k = 1$. Then cancel the UAV's action and update
      r_k, S' based on the current state.
    end if
    if $\sum_{k=1}^{K_{\text{co}}} R_{m,k}\delta \geq C_m, \forall m \in \mathscr{M}$ **then**
      let $\alpha_k = 0$, and start the data offloading.
    else
      Let $\beta_k = 0$, and continue the data collection.
    end if
    if $\sum_{k=k^*}^{K} R_k\delta \geq \sum_{m=1}^{M} C_m$ **then**
      $r_k = R_k\delta + \rho_k + K^*$, and let $l = 1$.
    end if
    Store transition tuple (S_k, A_k, r_k, S', l) in $\mathscr{B}$.
    if $\mathscr{B} > 2{,}000$ **then**
      Sample mini-batch of N transitions (S_k, A_k, r_k, S', l) from $\mathscr{B}$.
      $\tilde{A} \longleftarrow \pi_{\phi'}(S') + \varepsilon$, $\varepsilon \sim$ clip$(\mathscr{N}(0, \tilde{\omega}), -c, c)$.
      $y \longleftarrow r_k + (1 - l) \cdot \gamma \min_{i=1,2} Q_{\theta_i}{}'(S', \tilde{A})$.
      Update critics:
      $\theta_i \longleftarrow \arg\ \min_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(S_k, A_k))^2$.
      Update the actor policy ϕ by the deterministic policy gradient:
      $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_{A_k} Q_{\theta_1}(S_k, A_k)|_{A_k = \pi_\phi(S_k)} \nabla_\phi \pi_\phi(S_k)$.
      Update target networks:
      $\theta_i{}' \longleftarrow \tau\theta_i + (1 - \tau)\theta_i{}'$.
      $\phi' \longleftarrow \tau\phi + (1 - \tau)\phi'$.
    end if
  end for
end for
return The UAV trajectory **Q**

Algorithm 1: TD3-based trajectory optimization algorithm (TTO).

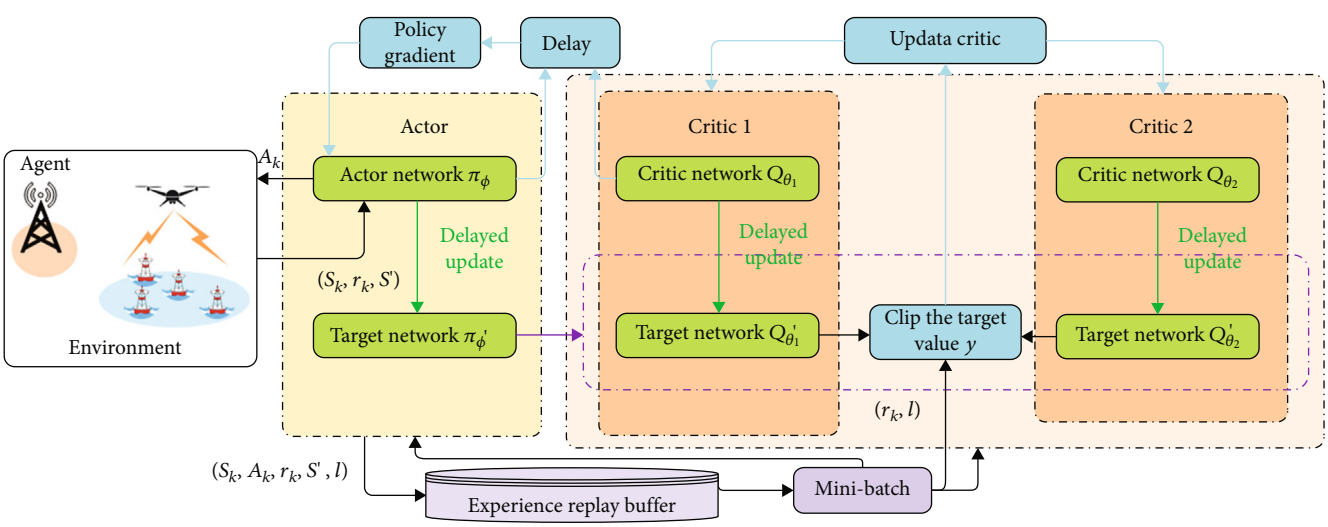


Figure 2: TD3-based UAV trajectory optimization.

1. Input the UAV's current position q_k, η_{SIC} and U.
2. Input current channel gain $g_{m,k}$, and sort it in descending order to obtain $g'_{m,k}$.
3. According to U, get initial NOMA group and the number of initial associated buoys $U^* = U$.
4. **repeat**
5. Solve P2a to obtain the optimal solution $\mathbf{P}_m{}^*$ and the optimal value C_k.
6. **if** the solution state is not optimal **then**
7. Remove the one with the worst channel gain in the currently associated buoy.
8. $U^* \longleftarrow U^* - 1$.
9. **end if**
10. **until** $U^* = 1$.
11. **return** $\mathbf{P}_m{}^*$ and $\mathbf{A}_m$;

Algorithm 2: Power control and buoy-UAV association relationship algorithm (PCAR).

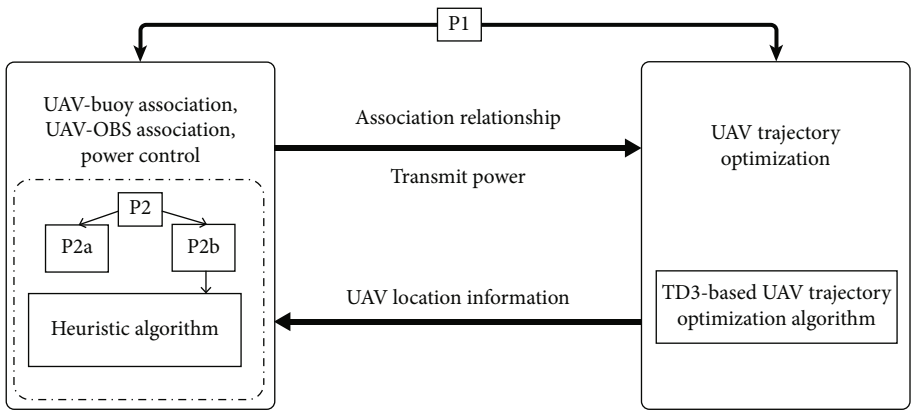


Figure 3: Joint TD3-based trajectory optimization, power control, and buoy-UAV association relationship scheme.

Input: The UAV's initial position q_1, the buoys' position D_m, the OBS's position D_0;
Output: $\mathbf{A}_m$, $\mathbf{P}$, $\mathbf{P}_m$, $\mathbf{Q}$;
1. **for** episode =0 **to** $E_{\max}$ **do**
2. **for** epoch $k = 1$ **to** $K_{\max}$ **do**
3. /∗ *Lines* 11-15 of Algorithm 1 ∗/
4. **if** $\sum_{k=1}^{K_{\mathrm{co}}} R_{m,k}\delta \geq C_m, \forall m \in \mathscr{M}$ **then**
5. Let $\alpha_k = 0$, and obtain current channel gain $g_{0,k}$.
6. Set $\mathbf{P} = P_{\max}$.
7. **if** $g_{0,k} \geq \bar{g}_0$ **then**
8. Let UAV-OBS association relationship $\beta_k = 1$.
9. **end if**
10. **else**
11. Let $\beta_k = 0$, and obtain current channel gain $g_{m,k}$.
12. Update $P_{m,k}, \forall m \in \mathscr{M}$ and α_k with given q_k by performing Algorithm 2.
13. /∗ *Lines* 7-10 of Algorithm 1 ∗/
14. Update q_k with given tranmist power and association relationship.
15. **end if**
16. **end for**
17. **end for**

Algorithm 3: Joint TD3-based trajectory optimization, power control, and buoy-UAV association relationship scheme (TTO-PCAR).

Table 1: Simulation parameters.

Parameter	Value
Maximum acceleration, $\Delta_{\max}$	25 m/s^2
Environment parameter, a, b	9.61, 0.16
Excessive path loss, ξ_{LoS}, ξ_{NLoS}	1, 20
Channel parameter, $\mu_{m,k}$	1
The antenna heights of buoys and OBS, H_m, H_0	0 m, 0 m
Wavelength, λ	0.15 m
Noise power, σ^2	-94 dBm
SINR and SNR threshold, $\bar{g}_{\mathrm{co}}$, $\bar{g}_{\mathrm{of}}$	10 dB, 3 dB
SIC threshold, η_{SIC}	10 dB
Order of magnitude parameter, n	6

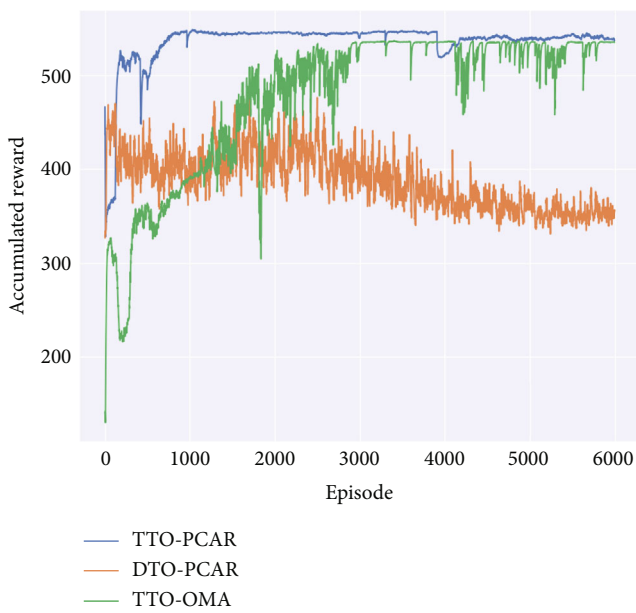


Figure 4: Accumulative reward.

denominator term $P_1 h_1$ should be as large as possible, so as to maximize the value of molecular term and make more buoys connected to UAV. Therefore, P_1 should be $P_{m_{\max}}$.

The total transmission data size of U^* buoys in the k-th time slot is expressed as

$$C_k = \sum_{m=1}^{U^*} B \log_2\left(1 + \frac{P_{m,k} h_{m,k}}{\sum_{i \in \mathcal{M}_k} P_{i,k} h_{i,k} + \sigma^2}\right)\delta. \tag{24}$$

Therefore, problem $P2$ can be transformed into

$$\begin{aligned} &(\mathrm{P2a}):\ \max_{\mathbf{P}_m} C_k, \\ &\text{s.t.}\quad \mathrm{C1}: 0 \le P_m \le P_{m_{\max}}, \\ &\mathrm{C10}: \frac{P_{m,k} h_{m,k}}{\sum_{i \in \mathcal{M}_k} P_{i,k} h_{i,k} + \sigma^2} \ge \eta_{SIC} \end{aligned} \tag{25}$$

Due to the existence of cochannel interference between buoys, $P2a$ is still nonconvex. Therefore, we convert C_k into the following form:

$$C_k = B\delta \log_2\left(1 + \frac{\sum_{m=1}^{U^*} P_{m,k} h_{m,k}}{\sigma^2}\right). \tag{26}$$

Therefore, $P2a$ is a convex problem that can be solved by a standard convex optimization solver (such as cvxpy). □

The algorithm PCAR is shown in Algorithm 2. In Algorithm 2, the channel gains are first sorted in descending order. It is stipulated that the UAV is associated with U buoys at most, and the first U buoys with the largest channel gain are selected to form the initial NOMA group. Then, we solve $P2a$. If $P2a$ has an optimal solution, the optimal solution P_m and the optimal value C_k are obtained. If $P2a$ has no optimal solution, the buoy with the worst channel gain is removed from the current NOMA group to form a new NOMA group. The above process is repeated until $P2a$ has an optimal solution. Note that the UAV is associated with at least one buoy in each time slot.

Second, in the data offloading stage, the total transmission data size of UAV in the k-th time slot is expressed as

$$C_k^{\mathrm{uav}} = B\delta \log_2\left(1 + \frac{P_k h_{0,k}}{\sigma^2}\right). \tag{27}$$

Then, $P2$ can be transformed into the following form:

$$\begin{aligned} &(\mathrm{P2b}):\ \max_{\mathbf{P}} C_k^{\mathrm{uav}}, \\ &\text{s.t.}\mathrm{C2}: 0 \le P_k \le P_{\max}, \\ &\mathrm{C11}: \frac{P_k h_{0,k}}{\sigma^2} \ge \bar{g}_0 \end{aligned} \tag{28}$$

Problem $P2b$ is a standard convex problem, the optimal solution of which is $P_{\max}$.

In summary, we propose a joint TTO and PCAR scheme (TTO-PCAR) to solve the problem $P1$. The TD3 agent is deployed in the OBS, and OBS maintains the communication with the UAV. During training, UAV collects data from buoys through traffic channel. Meanwhile, the UAV receives the states information of buoys through the control channel and feeds back the states information of itself and buoys to OBS. The OBS operates the proposed scheme with the above states information and sends the operation results to UAV in each time slot. Then, the UAV forwards relevant signalling (such as transmit power of buoys and buoy-UAV association relationship) to the buoy through the control channel. The specific process is shown in Figure 3 and Algorithm 3. Specifically, the UAV initial position is first given. In lines 3-12, P and A_m are obtained by Algorithm 2 according to the UAV current position q_k in the data collection stage. β_k is obtained according to q_k and $\bar{g}_0$ in the data offloading stage. Then, the UAV next position information is updated by lines 7-10 of Algorithm 1. Finally, the above process is repeated until $E_{\max}$.

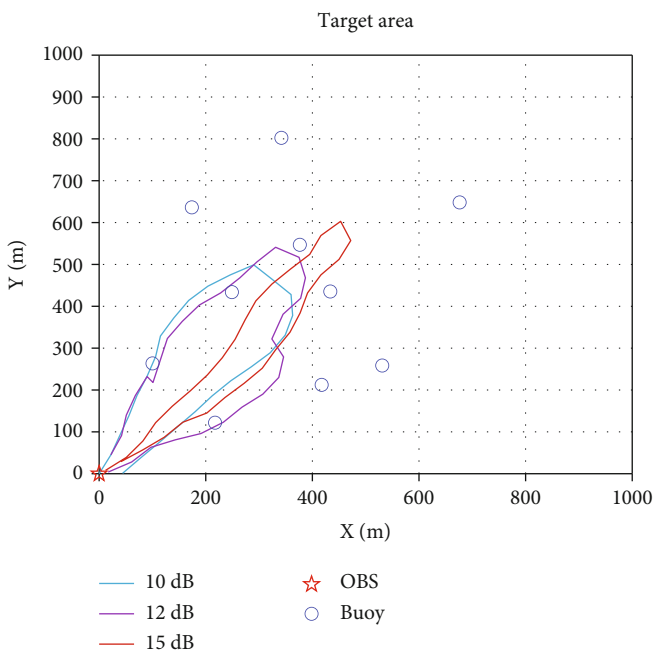


Figure 5: UAV trajectory with different SIC thresholds.

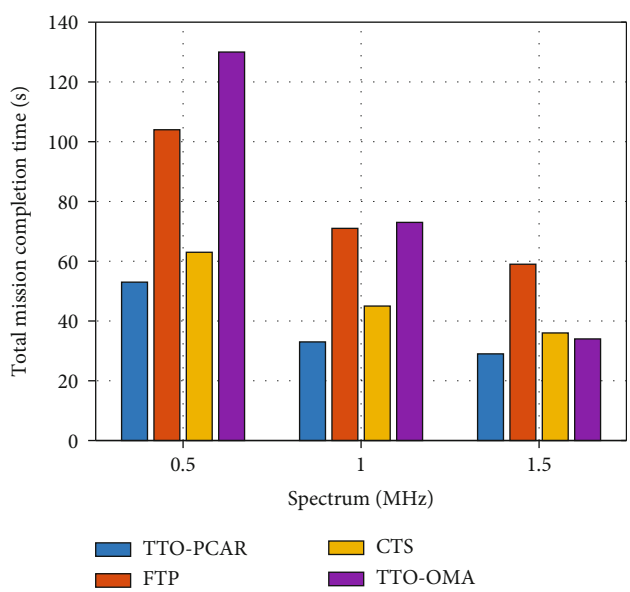


Figure 6: Total mission completion time with different schemes.

3.3. Complexity Analysis. TD3 contains two actor networks and four critic networks. Hence, the computational complexity of Algorithm 1 is $\mathcal{O}(2\sum_{e=1}^{E^a} n^a{}_e n^a{}_{e-1} + 4\sum_{e=1}^{E^c} n^c{}_e n^c{}_{e-1})$. E^a is the number of the fully connected layers of actor network. E^c is the number of the fully connected layers of critic network. n^a and n^c are the unit numbers in the e-th layer of the actor network and the critic network, respectively. Since the UAV is associated with at most U buoys in each time slot, the computational complexity of Algorithm 2 is $\mathcal{O}(U)$. Hence, the computational complexity of Algorithm 3 is $\mathcal{O}(E_{\max}K_{\max}(\sum_{e=1}^{E^a} n^a{}_e n^a{}_{e-1} + \sum_{e=1}^{E^c} n^c{}_e n^c{}_{e-1} + U))$.

4. Simulation Results

In simulation, the considered target area is 1000 m × 1000 m where $M = 10$ buoys are randomly distributed. A UAV is used to collect data with a fixed height $H = 100$ m from the target area. The flight velocity of UAV is $V_{\max} = 50$ m/s and $V_{\min} = 0$ m/s. The maximum transmit power of UAV and buoys is $P_{\max} = 0.1$ W and $P_{m_{\max}} = 24$ dBm, respectively. The position of OBS is $D_0 = (0, 0)$ m. The UAV is allowed to associate up to 3 buoys in each time slot, i.e., $U = 3$. The time slot length is $\delta = 1$ s. The data size range of each buoy is $C_m \in [10, 20]$ Mbits. The spectrum is $B = 1$ MHz. Furthermore, our proposed algorithm TTO is based on Pytorch. For actor and critic networks, we use a fully connected DNN with two hidden layers of 400 neurons. The learning rate is 0.0001. The experience memory buffer size is 100000. The minibatch size N is 256. The discount factor γ is 0.99. $\tilde{\omega} = 0.36$. $\tau = 0.005$. $K_{\max} = 300$. Other simulation parameters are shown in Table 1.

In order to compare performance, we use the following scheme as the comparison algorithm.

(i) UAV trajectory based on Fermat point (FTP) [35]: this scheme first regards each user as the vertex of a triangle to form multiple triangles. Then, the Fermat points of each triangle are taken as the hovering points of the UAV. The UAV hovers at the points in turn to collect data

(ii) UAV trajectory based on circle scheme (CTS): this scheme first finds the geometric center of all users as the center of the circle and then averages the distance from all users to the center of the circle to determine the radius of the UAV trajectory

(iii) UAV data collection based on OMA (TTO-OMA): this scheme refers to the UAV using OMA technology for data collection. The proposed TTO algorithm is still used to determine the UAV trajectory

(iv) DRL scheme based on DQN (DTO-PCAR): this scheme uses DQN instead of TD3 in our proposed algorithm

Figure 4 shows the comparison of accumulative reward for different schemes. For the convenience of observation, we smoothed the curves. It can be seen that the proposed TTO-PCAR scheme could be convergent after 1000 episodes, while the compared TTO-OMA scheme needs 3000 episodes to be convergent. Moreover, the compared DTO-PCAR scheme cannot be convergent after 6000 episodes. Therefore, the performance of our proposed scheme is significantly better than the other two schemes. Figure 5 shows the UAV trajectory comparison with TTO-PCAR scheme under different SIC thresholds. The SIC thresholds are 10 dB, 12 dB, and 15 dB, respectively. We find that the average total mission completion time of UAV is basically the same, which is 33 s, 36 s, and 37 s, respectively. However, the UAV trajectory is closer to the farther buoy with the increase of SIC threshold. This is because when the UAV

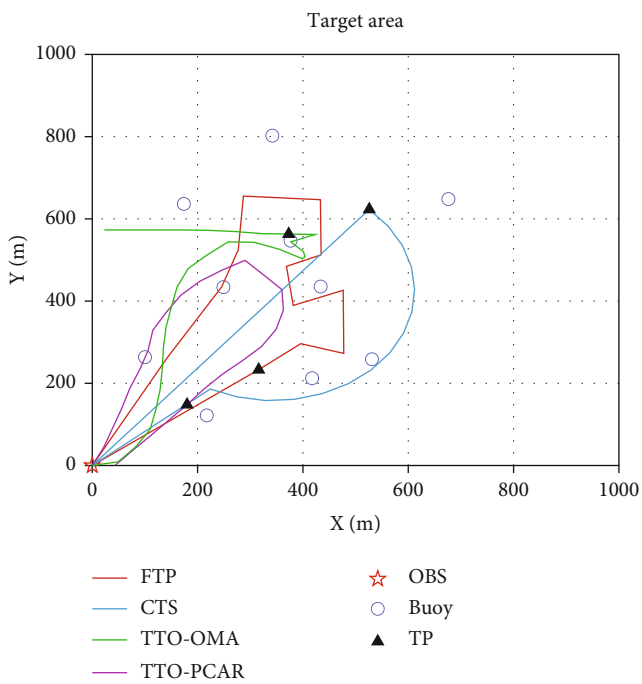


Figure 7: UAV trajectory of different schemes.

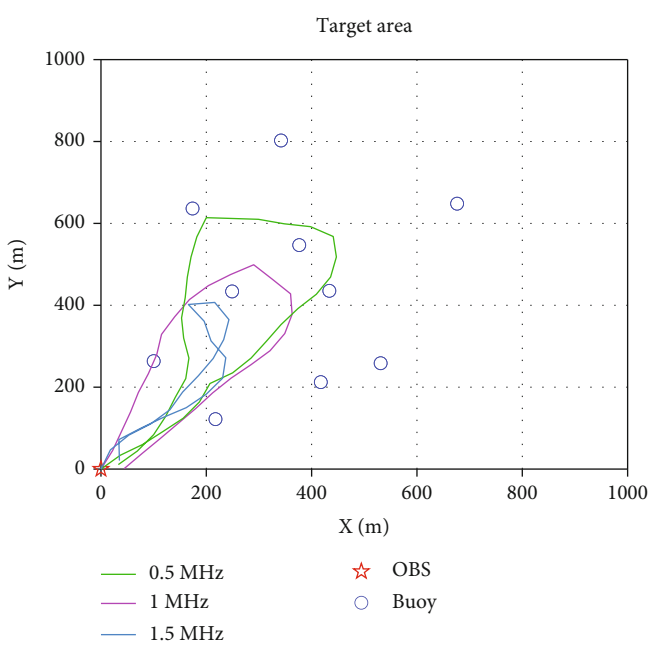


Figure 8: UAV trajectory with different spectrums.

uses NOMA technology for data collection, the channel gain of the farther buoy is poor. Therefore, in order to meet the SIC constraint, the UAV will gradually fly to the farther buoys whose data has not yet been collected.

Figure 6 shows the comparison of the total mission completion time of under different spectrums. Figure 7 shows the UAV trajectory obtained by our proposed scheme with $M = 10$ buoys and compares it with the other three schemes. It can be seen that the total mission completion time of

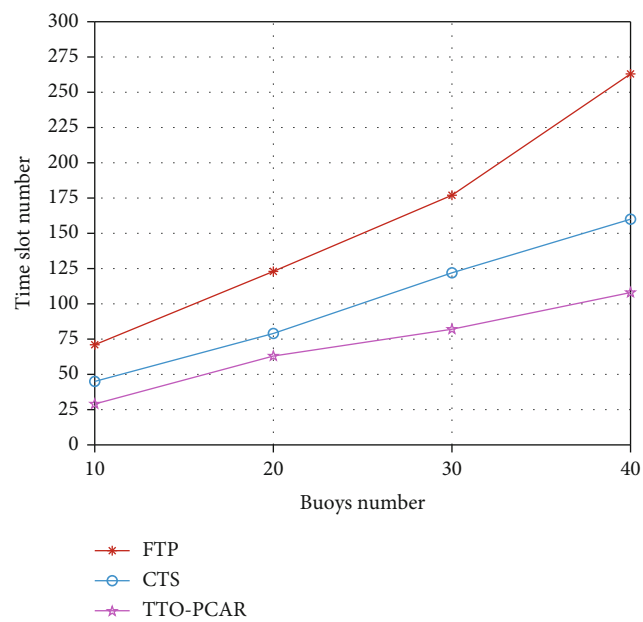


Figure 9: Total mission completion time with different buoy numbers.

TTO-PCAR is significantly lower than that of other schemes. In particular, the data collection time of our proposed scheme is 20 s with $B = 1$ MHz and that of TTO-OMA is 33 s; thus, NOMA is more efficient in data collection than OMA. This is because the designed reward (shown in Equation (16)) is related to the total transmission rate in each time slot. Second, the trajectory of UAV data collection process is fixed with the FTP and CTS scheme, resulting in the UAV trajectory of data offloading process longer. TTO-PCAR scheme takes the coupling of two stages into the consideration of UAV trajectory optimization; thus, the time of data offloading process is less. The total flight distance based on TTO-PCAR is also significantly lower than that of the other two schemes.

Figure 8 shows the UAV trajectory based on TTO-PCAR with different spectrums. It can be seen from Figure 8 that the flight distance of UAV decreases with the increase of spectrum. This is because the transmission rate of buoys is reduced with the reduction of spectrum. If the data of the buoy far from the OBS has not been collected, the agent chooses to make the UAV closer to the buoy in order to increase the transmission rate and obtain greater reward according to (16).

Figure 9 shows the total mission completion time of different schemes with different buoy numbers. FTP scheme is to find the hover points to collect data and classify the problem as a travelling salesman problem, so as to traverse the hover points. Hence, FTP takes a lot of time on UAV flight. Although CTS scheme can collect data in each time slot, it does not consider the data collection requirements of different buoys, because the UAV just flies based on circle. The proposed scheme TTO-PCAR dynamically adjusts the UAV trajectory according to the data collection requirements of

different buoys. Therefore, the total mission completion time of TTO-PCAR is significantly lower than that of FTP and CTS.

5. Conclusion

This paper has investigated the joint optimization problem of the buoy-UAV association relationship, transmit powers, and the UAV trajectory for NOMA-enabled UAV data collection and offloading in MIoT. First, we propose a TD3-based UAV trajectory optimization algorithm to solve the UAV trajectory subproblem. Second, we design a heuristic algorithm to solve the subproblem of power control and buoy-UAV association relationship. Finally, we propose a joint TD3-based trajectory optimization, power control, and buoy-UAV association relationship scheme. The proposed scheme can effectively solve the mixed-integer nonconvex problem. Simulation results show that the proposed scheme significantly shortens the total mission completion time of UAV. In future work, we will investigate the problem of UAV trajectory optimization based on NOMA to shorten the time for UAV to perform mission in the MIoT.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China under Grant 2019YFE0111600, in part by the National Natural Science Foundation of China under Grant Nos. 62101089, 62002042, 51939001, and 61971083, in part by the China Postdoctoral Science Foundation under Grant Nos. 2021M700655 and 2021M690022, in part by the Liaoning Revitalization Talents Program under Grant No. XLYC2002078, and in part by the Fundamental Research Funds for the Central Universities under Grant No. 3132022231.

References

[1] T. Wei, W. Feng, Y. Chen, C.-X. Wang, N. Ge, and J. Lu, "Hybrid satellite-terrestrial communication networks for the maritime Internet of Things: key technologies, opportunities, and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8910–8934, 2021.

[2] Y. Li, Y. Zhang, W. Li, and T. Jiang, "Marine wireless big data: efficient transmission, related applications, and challenges," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 19–25, 2018.

[3] A. Laun and E. Pittman, "Development of a small, low-cost, networked buoy for persistent ocean monitoring and data acquisition," in *OCEANS 2018 MTS/IEEE*, pp. 1–6, Charleston, 2018.

[4] Y. Huo, X. Dong, and S. Beatty, "Cellular communications in ocean waves for maritime Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9965–9979, 2020.

[5] X. Li, W. Feng, Y. Chen, C.-X. Wang, and N. Ge, "Maritime coverage enhancement using UAVs coordinated with hybrid satellite-terrestrial networks," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2355–2369, 2020.

[6] Y. Wang, W. Feng, J. Wanga, and T. Q. Quek, "Hybrid satellite-UAV-terrestrial networks for 6g ubiquitous coverage: a maritime communications perspective," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 11, pp. 3475–3490, 2021.

[7] H. Shen, Q. Ye, W. Zhuang, W. Shi, G. Bai, and G. Yang, "Drone-small-cell-assisted resource slicing for 5G uplink radio access networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 7071–7086, 2021.

[8] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: a connectivity-constrained trajectory optimization perspective," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2580–2604, 2019.

[9] C. Zhan and Y. Zeng, "Aerial–ground cost tradeoff for multi-UAV-enabled data collection in wireless sensor networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1937–1950, 2020.

[10] X. Li, W. Feng, Y. Chen, C.-X. Wang, and N. Ge, "UAV-enabled accompanying coverage for hybrid satellite-UAV-terrestrial maritime communications," in *2019 28th Wireless and Optical Communications Conference (WOCC)*, pp. 1–5, Beijing, China, 2019.

[11] D. S. Lakew, A. Masood, and S. Cho, "3D UAV placement and trajectory optimization in UAV assisted wireless networks," in *2020 International Conference on Information Networking (ICOIN)*, pp. 80–82, Barcelona, Spain, 2020.

[12] Y. Dai, J. Liu, M. Sheng, N. Cheng, and X. Shen, "Joint optimization of BS clustering and power control for NOMA-enabled comp transmission in dense cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1924–1937, 2021.

[13] J. Zhao, Y. Wang, Z. Fei, X. Wang, and Z. Miao, "Noma-aided UAV data collection system: trajectory optimization and communication design," *IEEE Access*, vol. 8, pp. 155843–155858, 2020.

[14] D. Hu, Q. Zhang, Q. Li, and J. Qin, "Joint position, decoding order, and power allocation optimization in UAV-based NOMA downlink communications," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2949–2960, 2020.

[15] N. Senadhira, S. Durrani, X. Zhou, N. Yang, and M. Ding, "Uplink NOMA for cellular-connected UAV: impact of UAV trajectories and altitude," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 5242–5258, 2020.

[16] W. Chen, S. Zhao, R. Zhang, Y. Chen, and L. Yang, "UAV-assisted data collection with nonorthogonal multiple access," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 501–511, 2021.

[17] R. Tang, W. Feng, Y. Chen, and N. Ge, "NOMA-based UAV communications for maritime coverage enhancement," *China Communications*, vol. 18, no. 4, pp. 230–243, 2021.

[18] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4576–4589, 2019.

[19] X. Liu, M. Chen, Y. Liu, Y. Chen, S. Cui, and L. Hanzo, "Artificial intelligence aided next-generation networks relying on UAVs," *IEEE Wireless Communications*, vol. 28, no. 1, pp. 120–127, 2021.

[20] X. Shen, J. Gao, W. Wu et al., "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 45–66, 2020.

[21] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2021.

[22] R. Zhong, X. Liu, Y. Liu, and Y. Chen, "Multi-agent reinforcement learning in noma-aided uav networks for cellular offloading," 2021, https://arxiv.org/abs/2010.09094.

[23] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and A. Nallanathan, "Deep reinforcement learning based dynamic trajectory control for UAV assisted mobile edge computing," 2021, https://arxiv.org/abs/1911.03887.

[24] R. Zhang, M. Wang, L. X. Cai, and X. Shen, "Learning to be proactive: self-regulation of UAV based networks with UAV and user dynamics," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4406–4419, 2021.

[25] Y. Wang, W. Fang, Y. Ding, and N. Xiong, "Computation offloading optimization for UAV-assisted mobile edge computing: a deep deterministic policy gradient approach," *Wireless Networks*, vol. 27, no. 4, pp. 2991–3006, 2021.

[26] Q. Ye, W. Shi, K. Qu, H. He, W. Zhuang, and X. Shen, "Joint ran slicing and computation offloading for autonomous vehicular networks: a learning-assisted hierarchical approach," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 272–288, 2021.

[27] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," 2018, http://arxiv.org/abs/1802.09477.

[28] M. Sun, X. Xu, X. Qin, and P. Zhang, "Aoi-energy-aware UAV-assisted data collection for iot networks: a deep reinforcement learning method," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17275–17289, 2021.

[29] J. Zhang, F. Liang, B. Li, Z. Yang, Y. Wu, and H. Zhu, "Placement optimization of caching UAV-assisted mobile relay maritime communication," *China Communications*, vol. 17, no. 8, pp. 209–219, 2020.

[30] Y. Dai, M. Sheng, J. Liu, N. Cheng, X. Shen, and Q. Yang, "Joint mode selection and resource allocation for d2d-enabled NOMA cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6721–6733, 2019.

[31] Z. Yang, Z. Ding, P. Fan, and N. Al-Dhahir, "A general power allocation scheme to guarantee quality of service in downlink and uplink NOMA systems," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7244–7257, 2016.

[32] Y. Wang, Z. Gao, J. Zhang et al., "Trajectory design for UAV-based Internet-of-Things data collection: a deep reinforcement learning approach," 2021, https://arxiv.org/abs/2107.11015.

[33] R. Duan, J. Wang, C. Jiang, H. Yao, Y. Ren, and Y. Qian, "Resource allocation for multi-UAV aided IoT NOMA uplink transmission systems," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7025–7037, 2019.

[34] R. Ding, F. Gao, and X. S. Shen, "3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: a deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7796–7809, 2020.

[35] L. Lyu, Z. Chu, B. Lin, Y. Dai, and N. Cheng, "Fast trajectory planning for UAV-enabled maritime IoT systems: a Fermat-point based approach," *IEEE Wireless Communications Letters*, vol. 11, no. 2, pp. 328–332, 2022.