Online Incentive Mechanism for Task Offloading with Privacy-Preserving in UAV-assisted Mobile Edge Computing

Ruiting Zhou[1] Renli Zhang[1] Yufeng Wang[1] Haisheng Tan[2,*] Kun He[1]
[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University [2]LINKE Lab, University of Science and Technology of China
{ruitingzhou,zhang_rl,yufengwang,hekun}@whu.edu.cn,hstan@ustc.edu.cn

ABSTRACT

Unmanned aerial vehicles (UAVs) have emerged as a promising technology to provide low-latency mobile edge computing (MEC) services. To fully utilize the potential of UAV-assisted MEC in practice, both technical and economic challenges need to be addressed: how to optimize UAV trajectory for online task offloading and incentivize the participation of UAVs without compromising the privacy of user equipment (UE). In this work, we consider unique features of UAVs, *i.e.,* high mobility as well as limited energy and computing capacity, and propose a privacy-preserving auction framework, *Ptero*, to schedule offloading tasks on the fly and incentivize UAVs' participation. Specifically, *Ptero* first decomposes the online task offloading problem into a series of one-round problems by scaling the UAV's energy constraint into the objective. To protect UE's privacy, *Ptero* calculates UAV's coverage based on subset-anonymity. At each round, *Ptero* schedules UAVs greedily, computes remuneration for working UAVs, and processes unserved tasks in the cloud to maximize the system's utility (*i.e.,* minimize social cost). Theoretical analysis proves that *Ptero* achieves truthfulness, individual rationality, computational efficiency, privacy preserving and a non-trivial competitive ratio. Trace-driven evaluations further verify that *Ptero* can reduce the social cost by up to 116% compared with four state-of-the-art algorithms.

CCS CONCEPTS

• **Theory of computation** → *Design and analysis of algorithms.*

KEYWORDS

UAV-assisted MEC, Task offloading, Online algorithm, Incentive mechanism

ACM Reference Format:
Ruiting Zhou[1] Renli Zhang[1] Yufeng Wang[1] Haisheng Tan[2,*] Kun He[1]. 2022. Online Incentive Mechanism for Task Offloading with Privacy-Preserving in UAV-assisted Mobile Edge Computing. In *The Twenty-third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '22), October 17–20, 2022, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3492866.3549715

1 INTRODUCTION

With the development of 5G, computing-intensive applications running on user equipment (UE), such as online games, VR/AR, [18], require low latency and high computation power [17]. To support such applications, mobile edge computing (MEC) which processes offloading requests from UEs in edge servers at base stations (BSs) has recently emerged as a new computing paradigm [1, 27]. Unfortunately, the infrastructure-based MEC generates high deployment cost, and may not work effectively in rural areas without sufficient infrastructures or urban areas during peak hours/ natural disasters [5]. In recent years, Unmanned Aerial Vehicle (UAV) has received extensive attention due to its high mobility and flexible deployment. UAVs can effectively improve the computing capacity and extend the coverage of MEC [20]. For example, on July 21, 2021, Mihe town in China was submerged due to heavy rainfall. In its communication interruption area, the mobile base station enabled by UAVs provided a mobile signal coverage of about 50 square kilometers for five hours. A total of 3572 users were connected, and the maximum number of users accessing at one time was 648 [19].

To fully realize the potential of UAV-assisted MEC in practice, both *technical* and *economic* challenges need to be addressed. *First*, on the technical side, one UAV cannot provide continuous computing services in a long time because the amount of energy obtained from one charge is limited [20]. Therefore, it is essential to optimize UAV trajectory to serve dynamic offloading requests from UEs under the UAVs' battery capacity constraint. Furthermore, when an emergency occurs, hundreds of UAVs from different organizations need to cooperate to support long-term and continuous services. Collaborative task offloading between UAVs is non-trivial, because UAVs from different organizations have heterogeneous computing and energy capacities. For example, In the Henan rescue operation in July 2021, the DJI emergency rescue alliance mobilized 37 drone rescue teams, including 255 drones, to provide aerial sensing and lighting services [6]. *Second*, on the economic side, UAVs process offloading tasks with their own cost, *e.g.,* energy and computing resource consumption [21]. It is difficult to quickly gather enough *voluntary UAVs* in an emergency situation at the same time. Therefore, incentive mechanisms which pay remuneration to working UAVs are necessary for enabling UAV-assisted MEC. *Furthermore*, one UE's private information (*e.g.,* the location, offloading requests) infers the UE owner's hobbies or preferences [26]. This information

This work is supported in part by the NSFC Grants (62072344, 62172303, 62132009, and U20A20177).
* corresponding author .

MobiHoc '22, October 17–20, 2022, Seoul, Republic of Korea

ACM ISBN 978-1-4503-9165-8/22/10…$15.00
https://doi.org/10.1145/3492866.3549715

should be hidden from UAVs to prevent the risk of privacy leakage [1]. Hence, both technical and economic challenges should be addressed without compromising the privacy of UEs.

There have been some efforts to solve the above two challenges. For the technical side, existing studies on UAV-assisted MEC focus on minimizing the overall UEs'/UAVs' energy consumption [29] or the system delay [11]. For the economic side, there are only a few related studies adopting fixed pricing [5]. However, fixed pricing cannot capture the changing relationship between the supply from UAVs and the demand from UEs. Consequently, overpricing and underpricing routinely occur, jeopardizing the UAV's profit as well as the system's utility. Other related papers are either to maximize the individual UAV's profit [21] or long-term average revenue [2] and fail to optimize the trajectories of UAVs. We will discuss it in detail in Sec. 2. To the best of our knowledge, none of them consider the limited capacity of the UAV's battery in the long term and protect the UE's private information.

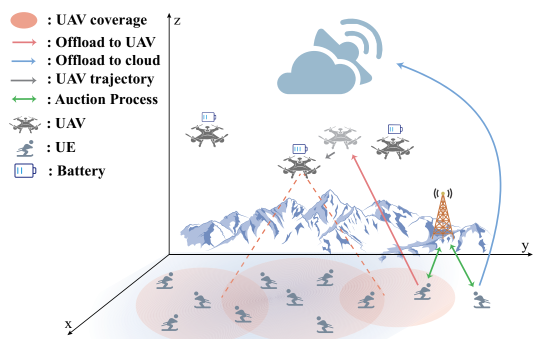

Figure 1: An illustration of UAV-assist offloading.

To overcome the aforementioned challenges, we capture important features of UAVs, *i.e.,* mobility as well as limited energy and computing capacity, and study the online incentive mechanism design for task offloading with privacy-preserving in UAV-assisted MEC. In particular, we consider an emergency situation, *e.g.,* sports event (as shown in Fig. 1), where a base station (BS) needs to hire UAVs to assist in offloading. We utilize auction technique to incentivize UAVs since it is a natural approach to balance supply and demand and automatically discover the right price so that BS can select the most cost-efficient UAVs. We optimize the system's utility in the long run (*i.e.,* social cost), and decide the trajectories of UAVs on the fly under UAVs' battery capacity constraints. Our main contributions are summarized as follows.

- We formulate the social cost minimization problem of task offloading in UAV-assisted MEC as an integer linear programming problem, which determines the two types of decisions (offloaded to UAVs or the remote cloud) under UAVs' battery capacity constraints. The problem is NP-hard even in offline settings. The challenge further escalates when tackling the online scenario and privacy-preserving.
- We design a privacy-preserving auction framework *Ptero*. *Ptero* first decomposes the online problem into a series of one-round winner determination problems by scaling the UAV's energy constraint into the objective. To protect the location information of UEs, *Ptero* next calculates each UAV's service scheme based on subset-anonymity. At each round, *Ptero* calls A_{winner} to schedule UAVs based on a greedy approach and process leftover tasks in the cloud to minimize the total cost. A payment strategy is proposed to calculate the remuneration for UAVs based on the critical value.
- We apply the primal-dual theory to prove that *Ptero* achieves a bounded competitive ratio with a small loss in online decomposition. We also prove that the one-round auction A_{winner} is α-approximate. We further prove that *Ptero* guarantees truthfulness, individual rationality, privacy-preserving, and computational efficiency.
- We evaluate the performance of *Ptero* based on real-world traces. Some promising results are observed: i) A_{winner} achieves a low approximation ratio (≈ 1.2), and can reduce the social cost by up to 12% and 51% compared to *Trac* [8] and *ODSH* [4]; ii) in the online settings, *Ptero* achieves a small competitive ratio (≈ 1.4) and reduces the social cost by up to 116%, 17.8%, and 4.5% compared with two state-of-art scheduling strategies [4, 21] and a naive version of *Ptero*; iii) A_{winner} makes decisions in less than 80ms while *Ptero* executes the online scheduling decisions in less than 160s for 45 rounds.

In the rest of the paper, we review related work in Sec. 2. The auction model for UAV-assisted MEC is introduced in Sec. 3. We present our online auction framework, *Ptero*, in Sec. 4 and analyze its theoretical performance in Sec. **??**. Sec. 6 shows the trace-driven simulation results and Sec. 7 concludes the paper.

2 RELATED WORK

UAV-assisted MEC. The UAV-assisted task offloading in MEC has been studied under different objectives. Li *et al.* [11] consider the popularity of cached contents in the UAVs to optimize their flight strategy, aiming to minimize the system service delay. Zhao *et al.* [29] focus on the total length of trajectories of UAVs and propose a multi-agent stochastic game to reach the optimal route. Zhang *et al.* [28] propose a game theory based scheme to jointly optimize the weighted values of time delay and energy consumption. In [27], UAV energy consumption and service delay are iteratively optimized by SCA-based algorithms. There is also other literature focusing on the cover rate of users [20], the number of deployed UAVs [15], and average throughput [22]. All of these papers focus on technical challenges and ignore the problem of how to incentivize the participation of UAVs. In addition, we optimize the trajectory of UAVs with the information of UEs protected, which is ignored by all of these papers.

Incentive Mechanisms. Incentive mechanism has been widely studied through auction [9, 17], game theory [2], and contract theory [12]. Chen *et al.* [5] propose an iterative matching algorithm to maximize UAV's utility in disaster rescuing. Wang *et al.* [21] design a dynamic pricing strategy for UAV profit-maximizing. Ng *et al.* [17] use UAV swarms to perform federated learning (FL) tasks, where UAVs aim to maximize their individual profit. Targeting on crowdsourcing, Jaimes *et al.* [9] propose an incentive mechanism for maximizing the coverage of UAVs with a budget constraint. In [2], the authors consider a content delivery network cached by UAVs and try to maximize the average revenue of the content provider. Lim *et al.* [12] consider UAVs and model owners in FL and propose a contract matching algorithm to maximize the owner's profit. However, the above studies mainly focus on maximizing the profit and ignore the efficiency of the system. Furthermore, we schedule the trajectories of UAVs online while protecting the individual location information of UEs.

Privacy-preserving in Offloading. To protect the communication channels between UAVs and the BS, Xu *et al.* [25] consider a dual UAV-assisted MEC system, where the additional UAV acts as a jammer to suppress the eavesdroppers. To protect the information of UAVs, Wei *et al.* [23] propose a differential privacy-based deep reinforcement learning method in the UAV-assist system. Xu *et al.* [24] build a distributed ledger based on blockchain to resist the invasion of the malicious UAV for UAV swarm. To protect the privacy of offloaded data, Liu *et al.* [13] develop a secure homomorphic encryption framework for processing private data. In [26], the authors propose an efficient asynchronous UAV-assisted FL framework to protect the raw data in training. However, these studies mainly focus on protecting the offloading process and ignore the protection of UEs. We optimize trajectories of UAVs, without compromising the privacy of UEs

3 SYSTEM MODEL

3.1 System Overview

UAV-assisted Offloading Scenario. As shown in Fig. 1, we consider a UAV-assisted MEC system including a number of user equipments (UEs), a base station (BS), and a mobile network of rotary-wing UAVs. The system involves a time-slotted fashion over T time slots (a.k.a., rounds). Let $\mathcal{X}$ denote the integer set $\{1, 2, \ldots, X\}$. Assume there are K UEs and U UAVs. each slot t, there are M_t UEs (where $M_t < K$) requesting to offload tasks for data processing to the BS. All tasks must be processed in one slot due to the delay requirement. In an emergency situation (*e.g.,* sports event, power failure, natural disaster), the BS may be overloaded by too many requests and needs to hire UAVs to assist offloading. To incentivize the UAVs, a procurement auction is applied where the BS acts as the auctioneer and each UAV submits a series of bids for participation. With all bids collected in a round, the BS determines and pays the selected winners, and schedules UAVs to process tasks.

Offloading Requirement. The offloading tasks from UEs come on the fly. The tasks can be offloaded to UAVs only when these two conditions are satisfied: i) the UE is connected with a UAV; 2) the connected UAV has sufficient computation capacity to execute the task. If some tasks cannot be handled by UAVs, the BS will upload those tasks to a remote cloud with a higher cost. Let $c(\chi)$ be the price function of processing χ units of data in the remote cloud.

UE Privacy-preserving. We assume some UAVs in the swarm are malicious and curious about individual information of UEs.[1] The UE's private information (*e.g.,* the location, offloading requests) infers the UE owner's hobbies or preferences. Once obtained by the malicious UAVs, they can send advertisements to UEs or even threaten the owner's safety. Note that the data of UEs can be protected by other existing methods (*e.g,* homomorphic encryption), and it is out of the scope in this paper. We aim to protect the location information of UEs so that it is not obtained by the other UAVs except the scheduled one.

[1] To get a more specific user profile, the malicious UAVs would honestly conduct the auction process.

3.2 Auction Model

Bid Information. We first introduce the bidding language in a traditional scenario without privacy-preserving for UEs. The privacy-preserving auction process is discussed in Sec. 4.3. At the beginning of each round, UAVs receive the current information of UEs from the BS to compute the feasibility between themselves and UEs. Let d_m^t be the data size that UE m wants to offload in slot t, G_m^t be the current position of UE m in slot t (*i.e.,* $G_m^t = (x_m^t, y_m^t)$). F_u^{max} and R_u^{max} are the computation capacity and the maximum coverage radius of UAV u respectively. Each UAV calculates its feasible service sets and submits up to J bids. We use $S_u^{t,j}$ to represent the feasible service set that UAV u can covers in its j-th bid at slot t. Each service set is a subset of $\mathcal{M}_t$. UAV u's j-th bid in round t ($B_u^{t,j}$) is expressed as $B_u^{t,j} = \{b_u^{t,j}, S_u^{t,j}, G_u^{t,j}\}$, where $b_u^{t,j}$ is the claimed cost that UAV u serves the UE set $S_u^{t,j}$ at the location $G_u^{t,j}$.

UAV Properties. The capacity of UAV u's battery is limited in a single flight, denoted as E_u^{max}. When UAV u serves set $S_u^{t,j}$, there are three types of energy costs generated at slot t: hovering cost, computation cost, and additional propulsion cost[2], denoted as $E_u^{h,t}$, $E_{u,j}^{s,t}$, $E_{u,j}^{f,t}$ respectively. The overall energy cost for UAV u to serve set $S_u^{t,j}$ at slot t is denoted as $E_u^{t,j}$. The hovering cost $E_u^{h,t}$ is a constant for UAV u at every slot. The detailed expressions of computation cost and propulsion cost are $E_{u,j}^{s,t} = p_u^s \sum_{m \in S_u^{t,j}} d_m^t$ and $E_{u,j}^{f,t} = p_u^f \|G_u^{t,j} - G_u^{t-1}\|$, where p_u^s, p_u^f, and G_u^{t-1} are the unit energy cost of computation, propulsion for UAV u, and the location of UAV u at slot $t-1$ respectively. Meanwhile, the moving speed of UAV u is limited. Therefore, the flying distance of UAV u between two adjacent slots is upper bounded by D_u^{max}.

Decision Variables. After all bids received, the BS determines the following variables: i) $x_u^{t,j} \in \{0, 1\}$, which equals 1 if the BS accepts UAV u's j-th bid $B_u^{t,j}$; ii) $\mathcal{Z}_t \subseteq \mathcal{M}_t$, the set of UEs offloaded to the remote cloud; iii) p_u^t, the payment to UAV u.

Auction Preliminaries. We introduce some definitions in the auction design. Let $v_u^{t,j}$ denote the true cost for UAV u to perform tasks in set $S_u^{t,j}$. The utility of this bid with bidding price $b_u^{t,j}$ is $p_u^t - v_u^{t,j}$ if $x_u^{t,j} = 1$, and 0 otherwise. The BS's utility (u_{BS}) equals $-\sum_{u,j,t} p_u^t x_u^{t,j} - \sum_t c(\sum_{m \in \mathcal{Z}_t} d_m^t)$.

Definition (Truthful Auction). A *truthful* auction guarantees that a different bidding price other than the true value $v_u^{t,j}$ would not increase the utility, *i.e.,* $\forall b_u^{t,j} \neq v_u^{t,j}, u_u^{t,j}(v_u^{t,j}) \geq u_u^{t,j}(b_u^{t,j})$.

Definition (Individual Rationality). An auction is *individual rational* if each UAV's utility is non-negative, *i.e.,* $u_u^{t,j}(b_u^{t,j}) \geq 0$.

Definition (Social Welfare, Social Cost). The social welfare equals UAVs' utilities plus the BS's utility. The social welfare is $-\sum_{u,j,t} v_u^{t,j} x_u^{t,j} - \sum_t c(\sum_{m \in \mathcal{Z}_t} d_m^t)$ when payment cancel themselves. Maximizing the social welfare equals to minimizing the social cost $\sum_{u,j,t} v_u^{t,j} x_u^{t,j} + \sum_t c(\sum_{m \in \mathcal{Z}_t} d_m^t)$.

In general, UAVs are rational but egocentric. They tend to maximize their own utilities by lying about their true cost. Maximizing social welfare and maximizing the BS's profit are both natural and interesting research problems in real-world auction design.

[2] Hovering is a special condition of propulsion. Additional propulsion cost is denoted as the extra flying energy cost except hovering cost. We assume that the UAV flies from G_i to G_j without any stop and keeps a constant speed in propulsion for simplicity.

Social welfare maximization prompts the efficiency of the whole eco-system and everyone has a higher probability to be happy in the long run. Thus, we target the utilities of the entire system (social welfare), which needs to elicit truthful bidding from UAVs.

3.3 Social Cost Minimization Problem

Problem Formalization. Under truthful bidding ($b_u^{t,j} = v_u^{t,j}$), the social cost minimization problem is formulated by the following integer linear problem (ILP):

$$\text{minimize} \sum_{t\in\mathcal{T}}\sum_{u\in\mathcal{U}}\sum_{j\in\mathcal{J}} b_u^{t,j}x_u^{t,j} + \sum_{t\in\mathcal{T}} c\Big(\sum_{m\in\mathcal{Z}_t} d_m^t\Big) \tag{1}$$

$$\text{s.t} \sum_{j\in\mathcal{J}}\sum_{u\in\mathcal{U}}\sum_{m\in S_u^{t,j}} x_u^{t,j} \geq 1, \forall t, \forall m \in \mathcal{M}_t - \mathcal{Z}_t, \tag{1a}$$

$$\sum_{j\in\mathcal{J}} x_u^{t,j} \leq 1, \forall u, \forall t, \tag{1b}$$

$$\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}} (E_u^{h,t} + E_{u,j}^{f,t}x_u^{t,j} + E_{u,j}^{s,t}x_u^{t,j}) \leq E_u^{max}, \forall u, \tag{1c}$$

$$x_u^{t,j} \in \{0,1\}, \mathcal{Z}_t \subseteq \mathcal{M}_t, \forall u, \forall t, \forall j, \forall m \in \mathcal{M}_t. \tag{1d}$$

Constraint (1a) ensures that all UEs' requests are served. We adopt the XOR-bidding strategy in constraint (1b) which specifies that each UAV wins at most one bid at each round. Constraint (1c) indicates the energy capacity of each UAV for the whole time span.

Table 1. List of Notations

U	# of UAVs	K	# of user equipments
T	system time-span	$\mathcal{X}$	integer set $\{1, 2, \ldots, X\}$
J	# of service sets	M_t	# of active UEs at slot t
E_u^{max}	battery capacity of UAV u		
D_u^{max}	maximum distance between two slots for UAV u		
R_u^{max}	maximum coverage radius of UAV u		
F_u^{max}	computation capacity of the u		
d_m^t	data size that UE m wants to offload at slot t		
$G_m^t(G_u^t)$	horizontal coordinate of UE m (UAV u) at slot t		
$S_u^{t,j}$	service set in UAV u's jth bid at slot t		
$x_u^{t,j}$	whether or not accept UAV u's j'th bid at slot t		
$y_{u,m}^{t,j}$	whether to serve UE m in slot t by UAV u in set j		
$\mathcal{Z}_t$	set of UEs processed in the remote cloud at slot t		
$b_u^{t,j}$	claim cost of UAV u to serve tasks in $S_u^{t,j}$		
$E_u^{h,t}$	hovering energy of UAV u at slot t		
$E_{u,j}^{s,t}$	computation energy of UAV u's j'th bid at slot t		
$E_{u,j}^{f,t}$	propulsion energy of UAV u's j'th bid at slot t		

Challenges. i) *Intractability.* Even a simplified version of ILP (1) without constraints (1b), (1c) and variable $\mathcal{Z}_t$ is still NP-hard, which is equivalent to the set cover problem [10]. ii) *Online decision-making.* Furthermore, we consider a dynamic offloading scenario. The BS should schedule UAVs on the fly based on the current UE demands and their valuations, which are not known as a priori. Constraint (1c) couples the energy consumption, which is affected at all time slots. iii) *Privacy-preserving.* Protecting the location of UEs is incapable if we permit all UAVs to access information of UEs from bidding process. An effective mechanism is required to adjust the bidding process with privacy-preserving. Important notations are listed in Table. 1 for easy reference.

4 AUCTION DESIGN

4.1 Overview of Auction Designing

Algorithmic Idea. We design an auction framework, *Ptero*, which solves ILP (1) with a privacy-preserving approach. *Ptero* determines the winning bids and the set of UEs for cloud processing to minimize the social cost with a bounded competitive ratio. The algorithmic idea of *Ptero* is shown in Fig. 2.

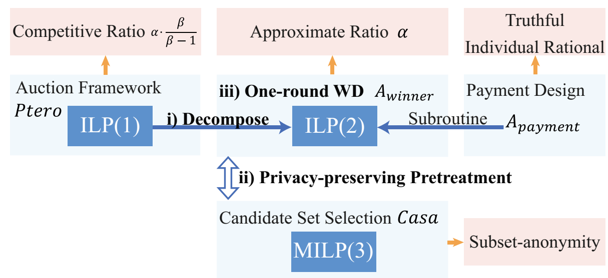

Figure 2: Main idea of UAV-assisted auction *Ptero*.

i) **Problem Decompose.** *Ptero* first decomposes ILP (1) into a set of one-round winner determination problems (WDP) (2) by scaling the energy constraints into the objectives by a carefully designed factor λ_u^t. The detailed implementation of problem decomposition is introduced in Sec. 4.2.
ii) **Privacy-preserving Pretreatment.** In Sec. 4.3, we show how to protect the individual information of UEs by subset-anonymity. We formulate MILP (3) to characterize the matching process between UAVs and UEs. To solve it, *Ptero* modifies the bidding process and provides a privacy-preserving pretreatment algorithm *Casa* based on reverse matching.
iii) **One-round Winner Determination.** In Sec. 4.4, we show how to determine the winners at each time slot t efficiently. We treat the remote cloud as a special case of UAV for ease of analysis. To solve each WDP (2), *Ptero* calls A_{winner} based on a greedy approach to minimize the total cost at each round with a bounded approximation ratio. The payment is calculated by a subroutine of A_{winner}, $A_{payment}$, to guarantee the truthfulness and individual rationality in the bidding process.

Algorithm Details. We present our auction framework *Ptero* in Alg. 1. Line 1 first initializes relative variables for decomposing and winner determination. At time slot t, *Ptero* executes the following steps. In Line 4, *Ptero* calls *Casa* to find the feasible service sets ($S_u^{t,j}$), service locations ($G_u^{t,j}$) for each UAV and the unreachable UAV set $\mathcal{B}_t$. In Line 6, *Ptero* computes the propulsion distance for each service scheme to hide UE's individual information. In Line 7, UAVs prepare bid information. More specifically, once receiving the service sets as well as the propulsion distance from the BS, UAVs claim their costs including bidding price ($b_u^{t,j}$) and additional energy cost ($E_{u,j}^{s,t} + E_{u,j}^{f,t}$). *Ptero* exams the feasibility of each service set in the view of battery capacity and form the grand set $\mathcal{F}^t$, candidate set C^t, and scale their bidding price according to the λ_u^t (which is related to the remaining battery level, and will be discussed in Sec. 4.2) in Lines 9-14. The BS decides the winner set ($\mathcal{A}^t$), the payment ($\mathcal{P}^t$), and cloud processing set ($\mathcal{Z}_t$) by A_{winner} in Line 15. According to the result returned from A_{winner}, *Ptero* records the best solution which achieves the minimum social cost, and announces the scheduling decision as well as the payment in Line 18. *Ptero* updates the time-energy coupled parameter λ_u^t in Line 17-24. UEs processed in the cloud are announced in Line 26.

Algorithm 1 UAV-assisted offloading auction framework *Ptero*

Input: $G_m^t, d_m^t, G_u^{t-1}, \forall u, m, t$

Output: $\mathcal{A}^t, \mathcal{P}^t, \mathcal{Z}_t \forall t$

1: Initialize $\lambda_u^t = 0, \eta_u = 0, C^t = \mathcal{F}^t = \mathcal{Z}_t = \phi, \forall t, u$
2: **for** $1 \leq t \leq T$ **do**
3:   **for** $u \in \mathcal{U}, m \in \mathcal{M}_t$ **do**
4:     $\{S_u^{t,j}, G_u^{t,j}, \mathcal{B}_t\} = Casa\{G_m^t, d_m^t, G_u^{t-1}\}$
5:     **for** $j \in \mathcal{J}$ **do**
6:       $D_u^{t,j} = \|G_u^{t,j} - G_u^{t-1}\|$
7:       $\{b_u^{t,j}, E_{u,j}^{s,t} + E_{u,j}^{f,t}\} = bidprocess\{S_u^{t,j}, D_u^{t,j}\}$
8:     **end for**
9:     $\mathcal{F}^t = C^t = \cup_{u \in \mathcal{U}, j \in \mathcal{J}}(u, j)$
10:     **if** $\eta_u + E_{u,j}^{s,t} + E_{u,j}^{f,t} > E_u^{max} - \sum_t E_u^{h,t}$ **then**
11:       $w_u^{t,j} = b_u^{t,j}; C^t = C^t \backslash (u, j)$
12:     **else**
13:       $w_u^{t,j} = b_u^{t,j} + \lambda_u^{t-1}(E_{u,j}^{s,t} + E_{u,j}^{f,t})$
14:     **end if**
15:     $\{\mathcal{A}^t, \mathcal{P}^t, \mathcal{Z}_t\} = A_{winner}\{w_u^{t,j}, S_u^{t,j}, \mathcal{F}^t, C^t, \mathcal{B}_t\}$
16:   **end for**
17:   **for** $(u, j) \in \mathcal{A}^t$ **do**
18:     Accept UAV u's j'th bid and schedule u according to G_u^t, S_u^t; Pay $p_u^{t,j} \in \mathcal{P}^t$ to the UAV u
19:     $\lambda_u^t = \lambda_u^{t-1}\left(1 + \frac{E_{u,j}^{s,t} + E_{u,j}^{f,t}}{\alpha(E_u^{max} - \sum_t E_u^{h,t})}\right) + \frac{b_u^{t,j}(E_{u,j}^{s,t} + E_{u,j}^{f,t})}{\alpha(E_u^{max} - \sum_t E_u^{h,t})^2}$
20:     $\eta_u = \eta_u + E_{u,j}^{s,t} + E_{u,j}^{f,t}$
21:   **end for**
22:   **for** $u \notin \mathcal{A}^t$ **do**
23:     $\lambda_u^t = \lambda_u^{t-1}$
24:   **end for**
25:   **for** $m \in \mathcal{Z}_t$ **do**
26:     offload UE m's request to the cloud
27:   **end for**
28:   $\lambda_u = \lambda_u^T, \forall u$
29: **end for**
    return $\mathcal{A}^t, \mathcal{P}^t, \mathcal{Z}_t \forall t$

4.2 Problem Decomposition

One-round Problem Formulation. The one-round WDP is defined as follows, which includes the same constraints related to the current time slot from (1), and excludes the time-coupled constraint (1c). We boost the cost $b_u^{t,j}$ into $w_u^{t,j}$ according to the remaining battery capacity of each UAV by a well-designed factor λ_u^t. Thus, the constraint (1c) is considered in the objective function as a scaled cost.

$$\text{minimize} \sum_{u \in \mathcal{U}} \sum_{j \in \mathcal{J}} w_u^{(t),j} x_u^{(t),j} + c\left(\sum_{m \in \mathcal{Z}_{(t)}} d_m^{(t)}\right) \tag{2}$$

$$\text{s.t} \sum_{j \in \mathcal{J}} \sum_{u \in \mathcal{U}} \sum_{m \in S_u^{(t),j}} x_u^{(t),j} \geq 1, \forall m \in \mathcal{M}_{(t)} - \mathcal{Z}_{(t)}, \tag{2a}$$

$$\sum_{j \in \mathcal{J}} x_u^{(t),j} \leq 1, \forall u, \tag{2b}$$

$$x_u^{(t),j} \in \{0, 1\}, \mathcal{Z}_{(t)} \subseteq \mathcal{M}_{(t)}, \forall u, \forall j, \forall m \in \mathcal{M}_{(t)}. \tag{2c}$$

The battery capacity limits the service sets that a UAV can cover over T rounds, leading to a different overall social cost when the battery is spent at different rounds. The over-consumption of energy at the early stage may narrow future scheduling decision space. The BS may be consequently forced to choose service sets from higher biddings, which further increases the social cost in the whole time span. The intuition in designing the online auction framework *Ptero* to achieve lower social cost is to reserve a certain amount of battery capacity for future demands. Thus, the scaled cost $w_u^{(t),j} = b_u^{(t),j} + \lambda_u^{t-1}(E_{u,j}^{s,(t)} + E_{u,j}^{f,(t)})$ increases with the decrease of a UAV's remaining battery capacity, reducing its chance to win. The adjustment in Line 19 of *Ptero* is carefully computed, such that the energy constraint (1c) is guaranteed over the T rounds of online auctions with a bounded ratio (See Theorem 1 and Theorem 7 for details).

4.3 Auction Process Modification

To protect UE's information, the auction process is modified.

Definition (Subset-anonymity). Assuming that a set of UEs forms k subsets. The subset-anonymity ensures that an adversary can not identify whether a UE is in one exact subset of these k subsets. That is, the adversary cannot distinguish an individual UE with a probability higher than $1/k$.

Specifically, we adopt the location subset-anonymity based strategy to reformulate the bidding service set, which obscures the individual location information of UEs in multiple service areas. The BS calculates the *service set* $S_u^{t,j}$, and the *UAV service location* $G_u^{t,j}$ for UAV u's serving scheme j. Furthermore, The BS only tells the distance between the service location and the UAV's location ($\|G_u^{t,j} - G_u^{t-1}\|$). With enough service sets, the location of UEs are hidden into the service sets. Thus, the UAV cannot infer the location of a UE. The request tuple provided to UAV u at time t by the BS is expressed as: $\Gamma_u^t = \{\sum_{m \in S_u^{t,j}} d_m^t, \|G_u^{t,j} - G_u^{t-1}\|\}_{\forall j \in \mathcal{J}}$. The UAV considers its current condition and claims a cost $b_u^{t,j}$ if it wants to accept the set $\Gamma_u^{t,j}$. After jointly considering information of both UEs and UAVs, the BS decides the winners of each round.

Pretreatment Formulation. Let $\mathcal{B}_t$ denotes the inaccessible UEs caused by distance limitation in slot t (*i.e,* $\|G_u^{(t-1)} - G_m^{(t)}\| \geq D_u^{max} + R_u^{max}, \forall m \in \mathcal{B}_t, \forall u$), which is a determined in each round. The BS calculates the feasible service sets $S_u^{t,j}$ and the service location $G_u^{t,j}$ for all UAVs in every round. A binary variable $y_{u,m}^{t,j}$ equals 1 if UE m's task is offloaded to UAV u by scheme j in slot t. The privacy-preserving pretreatment problem in each round is formulated by the following mixed integer linear problem (MILP):

$$\text{maximize } \{S_u^{(t),j}\} \tag{3}$$

$$\text{s.t} \sum_{j \in \mathcal{J}} \sum_{u \in \mathcal{U}} y_{u,m}^{(t),j} \geq 1, \forall m \in \mathcal{M}_{(t)} - \mathcal{B}_{(t)}, \tag{3a}$$

$$\sum_{m \in S_u^{(t),j}} y_{u,m}^{(t),j} d_m^{(t)} \leq F_u^{max}, \forall u, \forall j, \tag{3b}$$

$$\|G_u^{(t),j} - G_u^{(t-1)}\| \leq \|D_u^{max}\|, \forall u, \forall j, \tag{3c}$$

$$y_{u,m}^{(t),j} \|G_u^{(t),j} - G_m^{(t)}\| \leq \|R_u^{max}\|, \forall u, \forall j, \forall m, \tag{3d}$$

$$y_{u,m}^{(t),j} \in \{0, 1\}, G_u^{(t),j} \in \mathbb{R}, S_u^{(t),j} = \{m | y_{u,m}^{(t),j} = 1\}, \forall u, j, m. \tag{3e}$$

Constraint (3a) indicates that all the accessible UEs should be selected at least once. Constraint (3b) confines the offloading capacity of each service set. The flying distance between two slots for

UAV u and service coverage between UAVs and UEs are specified in constraints (3c) and (3d) respectively.

Algorithm Idea. To get the feasible solutions of MILP (3), we design an efficient algorithm *Casa* which selects active UEs based on reverse matching. For each UAV, *Casa* first finds out *accessible* UEs for the UAV and calculates the service area that covers these UEs as the possible service set. Then, *Casa* exams the total offloading data size for each set. If the offloading data size of one set exceeds the UAV's computation capacity, *Casa* merges this set to its neighbor sets when the capacity constraint is satisfied. Finally, *Casa* outputs the candidate sets and points their service location where the flying distance is the minimum.

Algorithm 2 Candidate Sets Selection Algorithm *Casa*

Input: $G_m^{(t)}, d_m^{(t)}, G_u^{t-1}, \forall u, m$

Output: $S_u^{(t),j}, G_u^{(t),j}, \forall u, j$

1: Initialize $R_u, \mathcal{M}_u = \phi, \mathcal{B}_{(t)} = \phi, \forall u$
2: **for** $1 \le u \le \mathcal{U}$ **do**
3:   **for** $1 \le m \le \mathcal{M}_{(t)}$ **do**
4:     **if** $\|G_u^{(t-1)} - G_m^{(t)}\| \le D_u^{max} + R_u^{max}$ **then**
5:       $m \rightarrow \mathcal{M}_u$
6:       r_m^u=$circle(G_m^{(t)}, R_u^{max})$
7:     **end if**
8:   **end for**
9:   c_{ij}=$circlecross(r_i^u, r_j^u) \forall i, j \in \mathcal{M}_u$
10:   label served UEs in c_{ij}
11:   **while** $\sum_{m \in c} d_m^{(t)} \ge F_u^{max}$ **do**
12:     roll back the served UEs in this area to its neighbors'
13:   **end while**
14:   merge c_{ij} with the same served UEs as $\hat{c}$
15:   **for** qualified $\hat{c}$ **do**
16:     $G_u^{(t),j}$=$mindist(u, \hat{c})$
17:     **if** $\|G_u^{(t),j} - G_u^{(t-1)}\| \le \|D_u^{max}\|$ **then**
18:       $S_u^{(t),j}$=$\{m \in \hat{c}\}$
19:     **end if**
20:   **end for**
21: **end for**
22: $\mathcal{B}_{(t)} = \mathcal{M}_{(t)} \backslash \{m \in S_u^{(t),j}, \forall u, j\}$
    return $S_u^{(t),j}, G_u^{(t),j}, \mathcal{B}_{(t)}, \forall u, j$

Algorithm Details. The candidate sets selection algorithm *Casa* is presented in Alg. 2. Here, $\mathcal{M}_u$ is the *accessible* UEs for UAV u. r_m^u is the valid service area if a UAV u wants to serve UE $m \in \mathcal{M}_u$. c_{ij} is the combined area that covers UE i, j or both. For each UAV u, Lines 4-7 calculate the valid service area r_m^u for accessible UEs. Lines 9-10 finds the combined area between UEs and label the covered UEs. Lines 11-13 check the offloading data size for each UE sets in the combined area, and record the UE sets in the combined area the same as that of their qualified neighbors. Since an area may roll back to multiple qualified neighbors, line 14 merges these area with the same UEs for de-duplication. Lines 15-21 determines the service set and the service location for each qualified area $\hat{c}$. Line 17 checks the feasibility of flying distance between the service location and current location for the UAV. The inaccessible UE set $\mathcal{B}_{(t)}$ is calculated in line 22.

Example. We illustrate the process of *Casa* by a simple example shown in Fig. 3. Suppose there are 3 UEs within UAV's coverage, each submits a request (d_m^t) of 1 unit data. The offloading capacity of UAV u is 2 units. *Casa* first circles the valid service area for each UE, *i.e.,* these three colored circles. *Casa* then calculates the combined area with different combination of served UEs, *i.e.,* the 7 numbered areas. *Casa* next finds out that the requests in area *7* exceed the capacity of UAV u. Area *7* is rolled back to its neighbors whose service set is just one less than its, *i.e.,* 4, 5, 6. With three feasible service set choice in *7*, *Casa* merges area *7* into its neighbors. After calculating the minimum distance between these areas, the final service set is determined as listed in the figure.

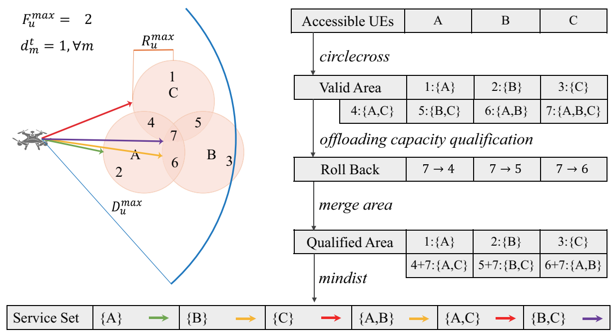

Figure 3: An example of *Casa*.

4.4 Winner Determination Algorithm

Problem Reformulation. We mark the remote cloud as a special case of UAV with the price of introducing the exponential number of service schemes. We mark the remote cloud as the UAV u_0, which has infinite energy and its service schemes provided in the bidding process are the combination of all the UEs in current round, *i.e.,* $|\mathcal{J}_{u_0}| = 2^{|M_{(t)}|}$. The additional energy consumption for offloading a request is 0 so that the bidding price is not scaled. The price of service set $S_{u_0}^{(t),j}$ is $w_{u_0}^{(t),j} = b_{u_0}^{(t),j} = c(\sum_{m \in S_{u_0}^{(t),j}} d_m^t)$. We reformulate the problem (2) as the following ILP:

$$\text{minimize} \sum_{u \in \mathcal{U}} \sum_{j \in \mathcal{J}} w_u^{(t),j} x_u^{(t),j} \tag{4}$$

$$\text{s.t} \sum_{j \in \mathcal{J}} \sum_{u \in \mathcal{U}} \sum_{m \in S_u^{t,j}} x_u^{t,j} \ge 1, \forall m \in \mathcal{M}_{(t)}, \tag{4a}$$

$$\sum_{j \in \mathcal{J}} x_u^{(t),j} \le 1, \forall u, \tag{4b}$$

$$x_u^{(t),j} \in \{0, 1\}, \forall u, \forall j. \tag{4c}$$

Constraints (4a) and (4b) are similar to constraints (2a) and (2b) respectively. It is clear that a feasible solution to ILP (4) can be transferred into the feasible solution to the WDP (2).

Dual Problem. To analyze the performance of A_{winner}, we formulate the dual problem by relaxing the integrality constraint (4c) into $0 \le x_u^{(t),j} \le 1$, and introduce dual variables $h_m^{(t)}, g_u^{(t)}, k_u^{(t),j}$ to constraints (4a), (4b), and $x_u^{(t),j} \le 1$, respectively. The dual problem is formulated as follows:

$$\text{maximize} \sum_{m \in \mathcal{M}_{(t)}} h_m^{(t)} - \sum_{u \in \mathcal{U}} g_u^{(t)} - \sum_{u \in \mathcal{U}} \sum_{j \in \mathcal{J}} k_u^{(t),j} \tag{5}$$

$$\text{s.t} \sum_{m \in S_u^{(t),j}} h_m^{(t)} - g_u^{(t)} - k_u^{(t),j} \le w_u^{(t),j}, \forall u, \forall j, \tag{5a}$$

$$h_m^{(t)}, g_u^{(t)}, k_u^{(t),j} \geq 0, \forall u, \forall j, \forall m \in \mathcal{M}_{(t)}. \tag{5b}$$

Algorithm 3 Winner Determination Algorithm A_{winner}

Input: $\{w_u^{(t),j}, S_u^{(t),j}\}, C^{(t)}, \mathcal{F}^{(t)}, \mathcal{B}_{(t)}, \forall u, m$

Output: $\mathcal{A}^{(t)}, \mathcal{P}^{(t)}$,

1: Initialize $\mathcal{A}^{(t)} = \mathcal{P}^{(t)} = \mathcal{L}^{(t)} = \phi$, $\mathcal{Z}_{(t)} = \mathcal{B}_{(t)}, x_u^{(t),j} = 0, h_m^{(t)} = 0, g_u^{(t)} = 0, k_u^{(t),j} = 0, \forall, m, j, u$ in real
2: **while** $U(\mathcal{A}^{(t)}) + |\mathcal{Z}_{(t)}| < |\mathcal{M}_{(t)}|$ **do**
3: (u^*, j^*)=argmin$_{(u,j)\in C^{(t)}} \frac{w_u^{(t),j}}{U_{u,j}(\mathcal{A}^{(t)})}$;
4: $\varphi(m, S_{u^*}^{(t),j^*}) = \frac{w_{u^*}^{(t),j^*}}{U_{u^*,j^*}(\mathcal{A}^{(t)})}, \forall m \in \mathcal{L}_{u^*,j^*}^{(t)}$
5: **if** $\sum_{m\in\mathcal{L}_{u^*,j^*}^{(t)}} \varphi(m, S_{u^*}^{(t),j^*}) \leq \Delta c(\sum_{m\in\mathcal{L}_{u^*,j^*}^{(t)}} d_m^t)$ **then**
6: $x_{u^*}^{(t),j^*} = 1$
7: $p_{u^*}^{(t)} = A_{payment}(C^{(t)}, (u^*, j^*), U_{u,j}(\mathcal{A}^{(t)}))$
8: **else**
9: $\mathcal{Z}_{(t)} = \mathcal{Z}_{(t)} \cup \{m \in \mathcal{L}_{u^*,j^*}^{(t)}\}$
10: **end if**
11: (u^+, j^+)=argmin$_{(u,j)\in\mathcal{F}^{(t)}} \frac{w_u^{(t),j}}{U_{u,j}(\mathcal{A}^{(t)})}$
12: $\varphi(m, S_{u^+}^{(t),j^+})' = \frac{w_{u^+}^{(t),j^+}}{U_{u^+,j^+}(\mathcal{F}^{(t)})}, \forall m \in \mathcal{L}_{u^+,j^+}^{(t)}$
13: $C^{(t)} = C^{(t)} \backslash (\cup_{j\in\mathcal{J}}(u^*, j))$
14: $\mathcal{F}^{(t)} = \mathcal{F}^{(t)} \backslash (u^*, j^*)$
15: $\mathcal{P}^{(t)} = \mathcal{P}^{(t)} \cup p_{u^*}^{(t)}$; $\mathcal{A}^{(t)} = \mathcal{A}^{(t)} \cup (u^*, j^*)$
16: **end while**
17: $x_{u_0}^{(t),j^\#} = 1$, where $j^\#$ exactly serves $m \in \mathcal{Z}_{(t)}$
18: $\max(m), \min(m) = \max, \min_{S_u^{(t),j}} \{\{\varphi(m, S_u^{(t),j})\} \cup \{\varphi(m, S_u^{(t),j})'\}\}, \forall m, \forall u$ in real
19: $\epsilon_m = \frac{\max(m)}{\min(m)}$; $\epsilon = \max_{m\in\mathcal{M}} \epsilon_m$
20: $n_\varphi(m) = \max_{S_u^{(t),j}} \{\varphi(m, S_u^{(t),j})\} \forall u$ in real; $h_m^{(t)} = n_\varphi(m)/(H_m\epsilon)$
21: **for** all $x_u^{(t),j} == 1$ **do**
22: $k_u^{(t),j} = \sum_{m\in\mathcal{L}_{u,j}^{(t)}} (n_\varphi(m) - \varphi(m, S_u^{(t),j}))/(H_m\epsilon)$
23: **end for**
return $\mathcal{A}^{(t)}, \mathcal{P}^{(t)}, \mathcal{Z}_{(t)}$

Main Idea. The winner determination algorithm A_{winner} selects schedules iteratively based on a greedy strategy to solve the problem (4). we form $\mathcal{Z}_{(t)}$ with the selection of other UAVs, and decide $x_{u_0}^{(t),j}$ after the UAV selection. We claim a UE is *available* if all these selected UAV service sets do not cover this UE. At each round, A_{winner} selects a UAV with a schedule j which can cover *available* UEs with the lowest *average cost.* If the cost is lower than processing these available UEs in the remote cloud, A_{winner} adds the selected UAV with its corresponding schedule to the winner set. Otherwise, these UEs are chosen for the cloud. This process terminates until all active UEs are served by the winners or the cloud.

Average Cost Let $\mathcal{A}^{(t)} = \{(u_1, j_1), (u_2, j_2), \dots)\}$ be a subset of bids submitted at round t, where (u_1, j_1) is UAV u_1's j_1'th bidding scheme, $\gamma_m^{A^{(t)}} = \sum_{(u,j)\in\mathcal{A}^t : m\in S_u^{(t),j}} 1$ denotes the total number of candidate UAVs to serve UE m among all bids in $\mathcal{A}^{(t)}$. The utility of set $\mathcal{A}^{(t)}$ is the number of UEs that has been served in $\mathcal{A}^{(t)}$, defined as $U(\mathcal{A}^{(t)}) = \sum_{m\in\mathcal{M}_t} \min(\gamma_m^{A^{(t)}}, 1)$. The increased number of served UEs of adding UAV u's j'th bid to $\mathcal{A}^{(t)}$ is:

$$\begin{aligned} U_{u,j}(\mathcal{A}^{(t)}) &= U(\mathcal{A}^{(t)} \cup (u, j)) - U(\mathcal{A}^{(t)}) \\ &= \sum_{m\in\mathcal{M}_{(t)}} (\min(\gamma_m^{A^{(t)}\cup(u,j)}, 1) - \min(\gamma_m^{A^{(t)}}, 1)). \end{aligned} \tag{6}$$

The *average cost* of schedule j is $\varphi(m, S_u^{(t),j}) = \frac{w_u^{(t),j}}{U_{u,j}(\mathcal{A}^{(t)})}$. Let $\mathcal{F}^{(t)}$ be the grand set of all bids in round t, and $C^{(t)}$ be the candidate set of all *valid* bids, such that satisfy XOR-bidding (1b) and UAV battery capacity (1c). At the beginning of A_{winner}, $\mathcal{A}^{(t)}$ is an empty set. Let $\mathcal{L}_{u,j}^{(t)}$ be the current *available* UEs in UAV u's j'th bid while selecting bid winners. Then $U_{i,j}(\mathcal{A}^{(t)})$ denotes the number of *available* UEs in that bid.

Algorithm Details. The winner determination algorithm is presented in Alg. 3. Line 1 initializes sets and variables. The *while* loop in Line 2-16 iteratively selects UAV bids until all UEs are covered. Specifically, Line 3-4 determine the possible winning bid with the lowest average cost at the current iteration and record corresponding average cost $\varphi(m, S_{u^*}^{(t),j^*})$. Line 5 compares the price for covering the newly added UEs in $\mathcal{L}_{u^*,j^*}^{(t)}$ with the additional cost for adding them to the cloud set. If the UAV is cheaper, the scheme j^* is the winning bid, $x_{u^*}^{(t),j^*}$ is updated to 1. Line 7 calls $A_{payment}$ to calculate the payment of these selected schedules while ensuring truthful bidding. Line 9 updates the remote cloud service set $\mathcal{Z}_{(t)}$ if the cloud is cheaper. Line 11-12 compute the winner average cost $\varphi(m, S_{u^*}^{(t),j^*})'$ if we relax the XOR-bidding constraint and energy restriction to decide the value of dual variables. Line 13 removes all the biddings from winner u^* in the candidate set to satisfy XOR-bidding rule. Line 14-15 removes the winning bid from grand set $\mathcal{F}^{(t)}$, and adds it to the winner set $\mathcal{A}^{(t)}$. Its relative payment is also added into $\mathcal{P}^{(t)}$. After $\mathcal{Z}_{(t)}$ is decided by the *while* iterations, the winner scheme for cloud u_0 is decided in Line 17. Lines 18-20 update the dual variables based on the payment to bound approximate ratio, where $H_m = \sum_{m=1}^{\mathcal{M}} \frac{1}{m}$ is the Mth harmonic number. The *for* loop in line 21-23 update the dual variable $k_u^{(t),j}$ if the primal constraint is satisfied. (*i.e.,* $x_u^{(t),j} = 1$).

Payment Design. It is vital to ensure truthful bidding for minimizing social welfare. The basic idea of $A_{payment}$ is to calculate the payment based on the critical bid, *i.e.,* the schedule that has the second smallest bidding value.

$A_{payment}$ is presented in Alg. 4. Line 1 finds out the critical bid (u^-, j^-). Line 2 computes the payment for the selected schedule (u^*, j^*) according to the critical value rule [16].

Algorithm 4 Payment Algorithm $A_{payment}$

Input: $C^{(t)}, (u^*, j^*), U_{u,j}(\mathcal{A}^{(t)})$

Output: $p_{u^*}^{(t)}$

1: (u^-, j^-)=argmin$_{(u,j)\in C^{(t)}:(u,j)\neq(u^*,j^*)} \frac{w_u^{(t),j}}{U_{u,j}(\mathcal{A}^{(t)})}$
2: $p_{u^*}^{(t)} = U_{u^*,j^*}(\mathcal{A}^{(t)}) \cdot \frac{w_{u^-}^{(t),j^-}}{U_{u^-,j^-}(\mathcal{A}^{(t)})}$
return $p_{u^*}^{(t)}$

5 THEORETICAL ANALYSIS

In this section, we analyze the properties of *Ptero* with its sub-algorithms *Casa*, A_{winner}, and $A_{payment}$. The correctness and time complexity are analyzed in Sec. 5.1. We prove the truthfulness and individual rationality of *Ptero*, A_{winner}, and $A_{payment}$ in Sec. 5.2. The privacy preserving of *Casa* is shown in Sec. 5.3. We prove the approximate ratio of A_{winner} is α in Sec. 5.4. The competitive ratio of *Ptero* is proven to be $\alpha\frac{\beta}{\beta-1}$ in Sec. 5.5. Due to the space limitation, the proofs for all Lemmas and Theorems can be found in the technical report [30].

5.1 Correctness and Time Complexity

LEMMA 1. *Casa computes a feasible solution to MILP (3) in polynomial time.*

LEMMA 2. *A_{winner} computes a feasible solution to ILP (2), ILP (4) and its dual (5) in polynomial time.*

THEOREM 1. *Ptero computes a feasible solution to MILP (1) in polynomial time.*

5.2 Truthfulness and Individual Rationality

THEOREM 2. *A_{winner} is a truthful auction in bidding price.*

THEOREM 3. *Ptero is a truthful auction.*

THEOREM 4. *Ptero achieves individual rationality.*

5.3 Privacy Preserving

THEOREM 5. *The proposed privacy-preserving online auction framework Ptero satisfies subset-anonymity.*

Proof: According to Lemma 1, the number of subsets for active UEs in each round is proven to be $O(M^2)$. The UAV could only get the distance and total offloading size of each set from the BS before the winner determination process, and each UAV could win at most one bidding. Thus, the adversary cannot identify the exact location of all the UEs at each round with a probability higher than $\frac{1}{O(M^2)}$. As for these *inaccessible* UEs which are offloaded to the remote cloud, the BS does not tell their existence to the UAVs, so that the probability of leaking their exact location is 0. In different rounds, The UEs move randomly, so that the previous information is outdated. The adversary could not trace a certain UE due to the uncertainty of winners in bidding. □

5.4 Approximation Ratio

Definition (Approximation Ratio): The approximation ratio of A_{winner} is the upper bound ratio of the objective value of ILP (2) returned by A_{winner} to the optimal objective value of (2) achieved by an optimal algorithm.

THEOREM 6. *Let p and d be the objective values of problem (4) and its dual (5) returned by A_{winner} respectively. $\alpha d \geq p$ with $\alpha = H_m\epsilon$, where $H_m = \sum_{m=1}^{\mathcal{M}} \frac{1}{m}$ and ϵ is defined in Line 19 of A_{winner}. A_{winner} is α-approximate algorithm to problem (2).*

5.5 Competitive Ratio

Definition (Competitive Ratio): The competitive ratio is the upper-bound ratio of the social cost achieved by *Ptero* to the optimal objective value of the offline WDP in MILP (1).

LEMMA 3. *Let $\Delta P^{(t)}$ and $\Delta D^{(t)}$ represent the incremental increase of the primal and dual objective values after t-th round, i.e., $\Delta P^{(t)} = P^{(t)} - P^{(t-1)}, \Delta D^{(t)} = D^{(t)} - D^{(t-1)}$. For all $t \in \mathcal{T}, \Delta P^{(t)} \leq \alpha\frac{\beta}{\beta-1}\Delta D^{(t)}$, where $\beta = \min_{u,j,t} \frac{E_u^{max} - \sum_t E_u^{h,t}}{E_{u,j}^{s,t} + E_{u,j}^{f,t}}$.*

Let $P^{(t)}$ and $D^{(t)}$ be the primal and dual objective value in problem (1) and it dual returned by *Ptero* after t-th round. P^T and D^T are the final primal and dual value achieved by *Ptero*. Let P^* denotes the optimal objective value of (1).

THEOREM 7. *The competitive ratio of Ptero is $\alpha\frac{\beta}{\beta-1}$.*

Proof: $P^{(0)} = D^{(0)} = 0$, and $\Delta P^{(t)} \leq \alpha\frac{\beta}{\beta-1}\Delta D^{(t)}$. We obtain $P^{(T)} = \sum_{t=0}^{T}(P^{(t)} - P^{(t-1)}) \leq \alpha\frac{\beta}{\beta-1}\sum_{t=0}^{T}(D^{(t)} - D^{(t-1)}) = \alpha\frac{\beta}{\beta-1}D^{(T)}$. According to LP duality[14], $P^* \geq D^{(T)}$. Thus, $\frac{P}{P^*} \leq \frac{P^T}{D^T} \leq \alpha\frac{\beta}{\beta-1}$. The competitive ratio of *Ptero* is $\alpha\frac{\beta}{\beta-1}$. □

6 PERFORMANCE EVALUATION

6.1 Evaluation Setup

Parameter Settings. We consider a UAV-assisted MEC network with 15 UAVs (*U=15*) and 55 UEs (*K=55*), and the percent of active UEs in each slot (M_t) is set to 80%. We utilize a recently-published bike trace [3], which includes GPS coordinates of a million shared bicycles in Shanghai, China. We select a specific region in 3 km*5 km and assume that a UE is carried by the bike. The length of each slot is 2 minutes and the entire system lasts for 45 slots (T). In each slot, the task size (d_m^t) of an active UE is set within [5,10] Mb. The configuration of UAVs is set according to the properties of DJI Mavic 2 Pro [7]. Specifically, the battery capacity (B_u^{max}) is 60 Wh. The maximum flying speed is 20 m/s. Since the trace updates coordinates in every 40 seconds, the maximum travel distance (F_u^{max}) between slots is 800m. The hovering energy cost ($E_u^{h,t}$) is 4 kJ in a slot. The additional unit propulsion energy cost (p_u^f) is 4 J/m. The maximum offloading capacity (D_u^{max}) is 40 Mb/slot, and the transmit power (p_u^s) is 20 dBm. To stabilize the offloading process, we limit the radiation radius (R_u^{max}) to 400 m. Suppose that the initial coordinates of UAVs are randomly distributed in this square. The bidding price of offloading a unit size of the data for UAV is uniformly distributed in [10,16]. Since the remote cloud has large computation capacity, the price would hardly change dynamically, we set the price function of the remote cloud to grow linearly with a unit price of 30.

Benchmark Algorithms: We compare *Ptero* with A_{winner} to four benchmark algorithms.

- $A_{pricing}$ [21]: which scales the price according to its overall expectation of profit in choosing to serve a task or stay still. For the one-round choosing strategy, we adopt the same as that in A_{winner}.
- **Greedy:** a naive algorithm of *Ptero* that selects winners using their original bidding price $b_u^{t,j}$ rather than $w_u^{t,j}$. The unqualified bids are also excluded in the current round. Its one-round choosing strategy is the same as A_{winner}.
- **ODSH** [4]: UAVs are scheduled to their nearest service set for energy saving in each round. If UAVs have no power, they are excluded in the future rounds. We compare its one-round performance with A_{winner} and the overall result with *Ptero*.

- **Trac** [8]: a one-round algorithm adopted in location-aware crowdsourcing without the XOR-bidding constraint. It calculates the effectiveness of a service set based on the total covered UEs rather than the number of *available* UEs, and selects the most effective service sets iteratively.

6.2 Performance of One-round Auction A_{winner}

Approximation Ratio. Fig. 4 shows the approximation ratio of A_{winner}, $Trac$, and $ODSH$ under different numbers of UEs. All of them perform well with the increase of the number of UEs. However, A_{winner} always outperforms the other two, because A_{winner} is able to select more cost-efficient service schemes to cover the UEs.

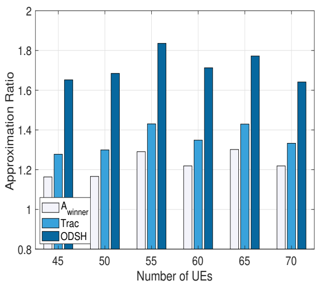

Figure 4: Approximate ratio of A_{winner}, $Trac$ and $ODSH$ under different numbers of UEs.

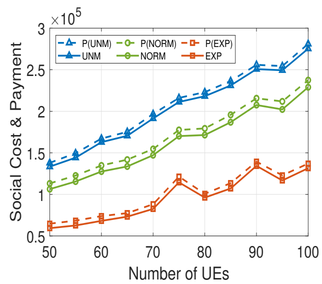

Figure 5: Social cost & payment with different numbers of UEs and price distributions.

Social Cost and Payment. In Fig. 5, we compare the social cost and payment under different bidding price distributions: UNI (a uniform distribution with unit cost in [10,16]), NORM (a normal distribution with the mean unit cost and standard deviation of 13 and 3 respectively), and EXP (an exponential distribution with a mean cost of 16). When the number of UEs increases, the BS must hire more UAVs, incurring a higher social cost. The social cost of the exponential distributions is lower than the other two because it generates a lower unit price with larger probability. The payment in these three distributions is always a bit higher than the social cost as $A_{payment}$ computes the payment no smaller than its cost.

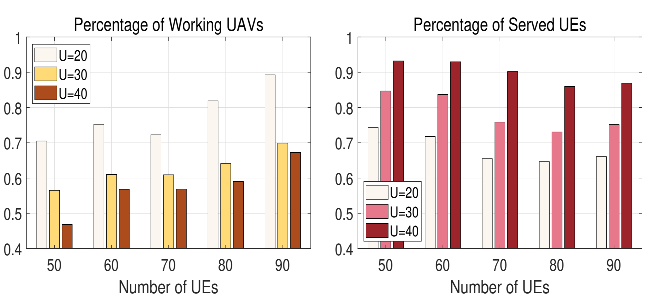

Figure 6: Percentage of working UAVs and served UEs under different numbers of UAVs and UEs.

System Satisfaction and Time Complexity. In fig. 6, we examine the percentage of working UAVs and served UEs by UAVs in various conditions. The percentage of working UAVs increases with more UEs and sharply decreases when more UAVs participate. The percentage of served UEs is always larger than 60% in all conditions and changes dynamically. When there are more UEs distributed in the area, the UAV cannot cover many of them due to its limited service area, which leads to the decrease in served UEs and the increase of working UAVs. Fig. 7 illustrates the complexity of A_{winner} under different input scales. We apply the *tic* and *toc* functions in MATLAB to measure the main body of the program. We repeat 20 times on a laptop (Intel i7-9750H/16GB RAM) and plot the average value. The running time of A_{winner} is only 72ms even with input on a large scale (U=25, M=75). It can be observed that A_{winner} grows linearly with the growing number of UEs and UAVs. We also plot the running time of the optimal algorithm *intlinprog* in MATLAB, which is significantly larger than ours.

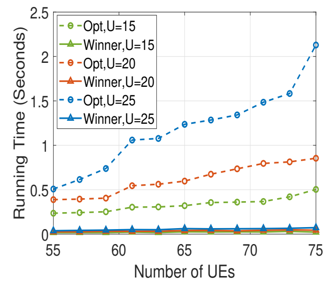

Figure 7: The average running time of A_{winner} and *Optimal* under different numbers of UEs and UAVs.

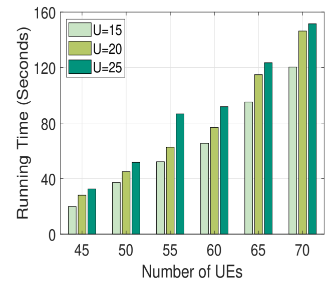

Figure 8: The average execution time of *Ptero* under different numbers of UEs and UAVs.

6.3 Performance of Online Auction *Ptero*

Execution Time. Fig. 8 shows the average execution time of *Ptero* for 45 slots. The execution time is longer than the sum of A_{winner} plotted in Fig. 7, since *Ptero* also includes *Casa* and variable updating process. The time is still acceptable even when we include 70 UEs and 25 UAVs in the system. The running time grows linearly with the increase of the number of UAVs and UEs, which confirms our time complexity analysis.

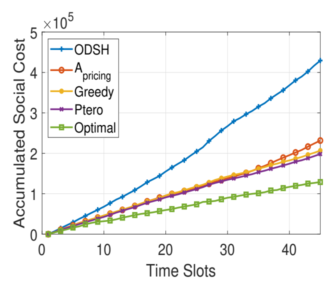

Figure 9: Social cost of five algorithms accumulated in slots.

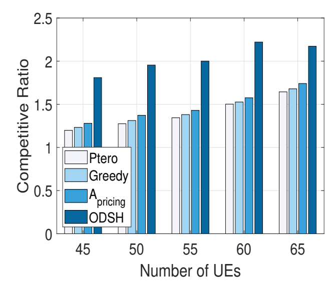

Figure 10: Competitive ratio of four algorithms under different numbers of UEs.

Social Cost. We analyze the accumulation of social cost in the entire time span of these five algorithms in Fig. 9. All of them grow near-linear as time goes on. *ODSH* performs the worst since it only considers the shortest path to reserve energy. The gap between *Ptero*, *greedy*, and $A_{pricing}$ start to increase in around slot 30. At that time, some UAVs may have run out of energy under the schedule of *greedy* and $A_{pricing}$, leading less number of available UAVs to schedule. Although the final gap between *greedy* and *Ptero* is around 3%, *Ptero* has always outperformed in all slots, proofing the efficiency of decomposition.

Competitive Ratio. Fig. 10 illustrates the competitive ratio of *Ptero*, *Greedy*, $A_{pricing}$, and *ODSH* with different numbers of UEs. All of them perform worse with the increase of UEs. The rank is the same as that in Fig. 9, where *Ptero* performs the best in these four online algorithms. we observed that there is only a small performance loss in *Ptero* compared with A_{winner} in Fig. 4, which confirms the analysis of decomposition loss in Theorem 7.

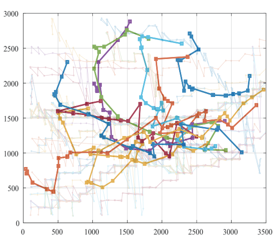

Figure 11: 10 UAV traces with 55 UEs in 45 slots.

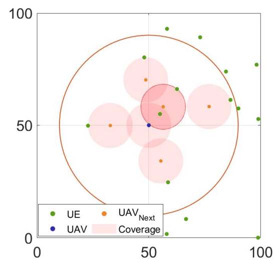

Figure 12: An example of *Casa* with 1 UAV and 15 UEs.

Performance of UAVs. We plot the traces of UEs and the trajectory traces of UAVs returned by *Ptero* in Fig. 11. Specifically, the 55 lighter lines indicate the trace information of UEs and the 10 darker lines are these UAVs. The UAVs tend to fly to areas where more UEs are active. Since its coverage is 400m, most of these UEs are served by UAVs. Most of these UAVs travel around the map instead of hovering in a certain area, which avoids the local optimum decision for a single UAV. Fig. 12 shows the result of *Casa* with 1 UAV and 20 UEs distributed in 100*100 squares. The large orange circle is the maximum service area for the UAV. These small pink circles and the orange dots are the coverage area and service location for a service set. Note that the darker circle in the northeast of the UAV covers more than one UE with minimum trajectory distance. The UAV stays still if a UE has been within its service coverage.

7 CONCLUSION

UAV complements MEC by providing flexible and low-latency services for UEs in the era of 5G. In this paper, we study online mechanism design for task offloading in UAV-assisted MEC, for scheduling offloading tasks on the fly and incentivizing UAV's participation. A reserve auction framework, *Ptero*, is proposed, with a goal of social cost minimization. Different from existing literature, *Ptero* determines UAV's trajectory and UE-UAV association for task offloading under battery capacity constraint, while protecting UE's information. Through rigorous theoretical analysis, we prove that *Ptero* is truthful, individual rational, computational efficient, privacy preserving and achieves a good competitive ratio. Trace-driven simulations further show that *Ptero* outperforms four baselines and has near-to-optimal performance.

REFERENCES

[1] Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. 2018. Mobile Edge Computing: A Survey. *IEEE Internet of Things Journal* 5, 1 (2018), 450–465.
[2] Alia Asheralieva and Dusit Niyato. 2019. Game Theory and Lyapunov Optimization for Cloud-Based Content Delivery Networks With Device-to-Device and UAV-Enabled Caching. *IEEE Transactions on Vehicular Technology* 68, 10 (2019), 10094–10110.
[3] Jie Bao, Tianfu He, Sijie Ruan, Yanhua Li, and Yu Zheng. 2017. Planning Bike Lanes based on Sharing-Bikes' Trajectories. In *Proc. of SIGKDD*. ACM.
[4] Shuaibing Chen, Lei Shi, Xu Ding, Zengwei Lv, and Zhehao Li. 2021. Energy Efficient Resource Allocation and Trajectory Optimization in UAV-Assisted Mobile Edge Computing System. In *Proc. of BIGCOM*. IEEE.
[5] Weiwei Chen, Zhou Su, Qichao Xu, Tom H. Luan, and Ruidong Li. 2020. VFC-Based Cooperative UAV Computation Task Offloading for Post-disaster Rescue. In *Proc. of INFOCOM*. IEEE.
[6] DJI. 2021. *emergency-response*. https://enterprise.dji.com/cn/public-safety
[7] DJI. 2022. *DJI Mavic 2 Pro*. www.dji.com/cn/mavic-2-enterprise-advanced
[8] Zhenni Feng, Yanmin Zhu, Qian Zhang, Lionel M. Ni, and Athanasios V. Vasilakos. 2014. TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *Proc. of INFOCOM*. IEEE.
[9] Luis Gabriel Jaimes and Juan M. Calderón. 2020. An UAV-based incentive mechanism for Crowdsensing with budget constraints. In *Proc. of CCNC*. IEEE.
[10] Jon Kleinberg and Eva Tardos. 2006. *Algorithm design*. Pearson Education India.
[11] Lixin Li, Meng Wang, Kaiyuan Xue, Qianqian Cheng, Dawei Wang, Wei Chen, Miao Pan, and Zhu Han. 2021. Delay Optimization in Multi-UAV Edge Caching Networks: A Robust Mean Field Game. *IEEE Transactions on Vehicular Technology* 70, 1 (2021), 808–819.
[12] Wei Yang Bryan Lim, Jianqiang Huang, Zehui Xiong, Jiawen Kang, Dusit Niyato, Xian-Sheng Hua, Cyril Leung, and Chunyan Miao. 2021. Towards Federated Learning in UAV-Enabled Internet of Vehicles: A Multi-Dimensional Contract-Matching Approach. *IEEE Transactions on Intelligent Transportation Systems* 22, 8 (2021), 5140–5154.
[13] Tianyuan Liu, Hongpeng Guo, Claudiu Danilov, and Klara Nahrstedt. 2020. A Privacy-preserving Data Collection and Processing Framework for Third-party UAV Services. In *Proc. of TrustCom*. IEEE.
[14] David G. Luenberger. 1984. *Introduction to linear and nonlinear programming*. Addison-Wesley Reading.
[15] Jiangbin Lyu, Yong Zeng, Rui Zhang, and Teng Joon Lim. 2017. Placement Optimization of UAV-Mounted Mobile Base Stations. *IEEE COMMUNICATIONS LETTERS* 21, 3 (2017), 604–607.
[16] Roger B. Myerson. 1981. Optimal Auction Design. *Mathematics of Operations Research* 3 (1981), 58–73.
[17] Jer Shyuan Ng, Wei Yang Bryan Lim, Hong-Ning Dai, Zehui Xiong, Jianqiang Huang, Dusit Niyato, Xian-Sheng Hua, Cyril Leung, and Chunyan Miao. 2021. Joint Auction-Coalition Formation Framework for Communication-Efficient Federated Learning in UAV-Enabled Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems* 22, 4 (2021), 2326–2344.
[18] Sukhmani Sukhmani, Mohammad Sadeghi, Melike Erol-Kantarci, and Abdulmotaleb El Saddik. 2019. Edge Caching and Computing in 5G for Mobile AR/VR and Tactile Internet. *IEEE MultiMedia* 26, 1 (2019), 21–30.
[19] Global Times. 2021. *Chinese drones provide lifeline network services for flood-stranded residents*. https://www.globaltimes.cn/page/202107/1229363.shtml
[20] Kai Wang, Xiao Zhang, Lingjie Duan, and Jun Tie. 2021. Multi-UAV Cooperative Trajectory for Servicing Dynamic Demands and Charging Battery. *IEEE Transactions on Mobile Computing* (2021), early access.
[21] Xuehe Wang and Lingjie Duan. 2019. Dynamic Pricing and Capacity Allocation of UAV-provided Mobile Services. In *Proc. of INFOCOM*. IEEE.
[22] Zhe Wang, Lingjie Duan, and Rui Zhang. 2019. Adaptive Deployment for UAV-Aided Communication Networks. *IEEE Transactions on Wireless Communications* 18, 9 (2019), 4531–4543.
[23] Dawei Wei, Ning Xi, Jianfeng Ma, and Lei He. 2021. UAV-Assisted Privacy-Preserving Online Computation Offloading for Internet of Things. *Remote Sensing* 13, 23 (2021), 4853.
[24] Xiaobin Xu, Hui Zhao, Haipeng Yao, and Shangguang Wang. 2021. A Blockchain-Enabled Energy-Efficient Data Collection System for UAV-Assisted IoT. *IEEE Internet of Things Journal* 8, 4 (2021), 2431–2443.
[25] Yu Xu, Tiankui Zhang, Dingcheng Yang, Yuanwei Liu, and Meixia Tao. 2021. Joint Resource and Trajectory Optimization for Security in UAV-Assisted MEC Systems. *IEEE Transactions on Communications* 69, 1 (2021), 573–588.
[26] Helin Yang, Jun Zhao, Zehui Xiong, Kwok-Yan Lam, Sumei Sun, and Liang Xiao. 2021. Privacy-Preserving Federated Learning for UAV-Enabled Networks: Learning-Based Joint Scheduling and Resource Management. *IEEE Journal on Selected Areas in Communications* 39, 10 (2021), 3144–3159.
[27] Cheng Zhan, Han Hu, Xiufeng Sui, Zhi Liu, and Dusit Niyato. 2020. Completion Time and Energy Optimization in the UAV-Enabled Mobile-Edge Computing System. *IEEE Internet of Things Journal* 7, 8 (2020), 7808–7822.
[28] Kaiyuan Zhang, Xiaolin Gui, Dewang Ren, and Defu Li. 2021. Energy-Latency Tradeoff for Computation Offloading in UAV-Assisted Multiaccess Edge Computing System. *IEEE Internet of Things Journal* 8, 8 (2021), 6709–6719.
[29] Chenxi Zhao, Junyu Liu, Min Sheng, Wei Teng, Yang Zheng, and Jiandong Li. 2021. Multi-UAV Trajectory Planning for Energy-Efficient Content Coverage: A Decentralized Learning-Based Approach. *IEEE Journal on Selected Areas in Communications* 39, 10 (2021), 3193–3207.
[30] Ruiting Zhou. 2022. *Technical Report*. www.jianguoyun.com/p/DXG6g-cQ46PsCRj26rAE