Auction-Based Crowdsourced First and Last Mile Logistics

Yafei Li, Yifei Li, Yun Peng, Xiaoyi Fu, Jianliang Xu, and Mingliang Xu

Abstract—The booming of mobile internet and crowdsourcing technology has offered great opportunities for first and last mile logistics (FLML) service. Unlike the traditional FLML service that separates the parcel collection in the first mile from the parcel delivery in the last mile, a new type of crowdsourced FLML service integrates parcel collection and parcel delivery services as a whole, which can significantly improve the efficiency of FLML service. Briefly, in a crowdsourced FLML service, the platform assigns the customers' triggered pick-up parcels to the couriers who are delivering drop-off parcels in terms of the real-time status of couriers (e.g., capacity, location, and schedule). Existing works solving the crowdsourced FLML problem only consider the utility maximization for the platform but ignore the incentive to the utilities of couriers. Inspired by this, in this paper, we investigate a novel type of crowdsourced FLML problem, namely *Auction-based Crowdsourced FLML (ACF)*, where the platform assigns the couriers with suitable pick-up parcels based on the preferences of couriers with the goal of maximizing the social welfare of the platform and couriers. To solve the ACF problem, we present a novel auction model named *Multi-attribute Reverse Vickrey (MRV)*, where the couriers bid on parcels according to their preferences for parcels. Based on the MRV model, we present three efficient assignment algorithms to assign parcels to couriers. In addition, we give theoretical analysis for our proposed algorithms. Extensive experiments examine the efficiency and effectiveness of our solutions.

Index Terms—Auction, crowdsourcing, location-based services, logistics, optimization

1 Introduction

The booming of mobile internet and crowdsourcing technology has offered great opportunities for the first and last-mile logistics (also known as FLML) service. The FLML service refers to two important aspects of the logistics process: the first mile of transporting parcels from the customer to the transit station and the last mile of transporting parcels from the transit station to the customer [7]. It is considered a critical service to manage the booming e-commerce and supply chains seamlessly. As reported in a recent survey [34], the cost of the FLML service accounts for 53 percent of the total shipment cost. In a classical FLML service, the pick-up service and the drop-off service for parcels are two independent services that are performed by different couriers. Obviously, such a separate approach is not efficient. Recently, a new type of crowdsourced FLML service, where the platform assigns the customers' triggered pick-up parcels to the couriers on their way to deliver drop-off parcels according to the real-time status of couriers (e.g., capacity, location, and schedule), significantly improves the crowdsourced FLML service's efficiency and leads to increasing interest from academia and industry [4], [28]. In this paper, we focus on the crowdsourced FLML service.

The crowdsourced FLML service integrates the pick-up service and the drop-off service for parcels, in which couriers can collect the dynamically arrived pick-up parcels during parcel delivery. More specifically, the couriers full of drop-off parcels depart from the transit stations and deliver the drop-off parcels to customers before a given deadline. During the delivery process, the platform assigns suitable couriers to collect the pick-up parcels that arrive dynamically. The pioneering works [23], [43] have studied the crowdsourced logistics service that finds the best assignment for couriers and parcels to maximize the number of served parcels. For commercial platforms, Amazon [1], Cainiao [5], and JD [12] have deployed efficient crowdsourced logistics services to cope with the massive increase in demand for parcel delivery. However, the existing works have two issues:

- First, the platform assigns parcels to couriers in a one-way manner. That is, they only consider the time constraints of pick-up parcels but ignore the preferences of the couriers for pick-up parcels. For example, the couriers usually have different willingness to the weight

- *Yafei Li is with the School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China, and also with the Yunnan Key Laboratory of Blockchain Application Technology, Beihang Yunnan Innovation Institute, Kunming 650001, China. E-mail: ieyfli@zzu.edu.cn.*
- *Yifei Li and Mingliang Xu are with the School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China. E-mail: lyfzzu@outlook.com, iexumingliang@zzu.edu.cn.*
- *Yun Peng is with the Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China. E-mail: yunpeng@gzhu.edu.cn.*
- *Xiaoyi Fu and Jianliang Xu are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, China. E-mail: {xiaoyifu, xujl}@comp.hkbu.edu.hk.*

Manuscript received 7 June 2022; revised 23 October 2022; accepted 1 November 2022. Date of publication 7 November 2022; date of current version 5 December 2023.
This work was supported in part by the Project of Science and Technology Major Project of Yunnan Province under Grant 202105AG070005 (YNB202202), in part by the NSFC under Grants 61972362, 62036010, 61602420, 62002074, and 62102107, in part by the HNSF under Grant 202300410378, in part by the HK-RGC under Grants C2004-21GF and 172922, and in part by the GDNSF under Grant 2019B1515130001.
(Corresponding authors: Jianliang Xu and Mingliang Xu.)
This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2022.3219881, provided by the authors.
Digital Object Identifier no. 10.1109/TMC.2022.3219881

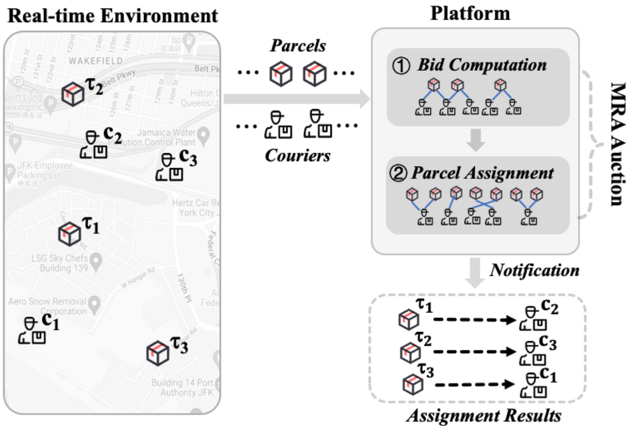


Fig. 1. Auction-based crowdsourced FLML services.

and detour of the pick-up parcels at different delivery stages. Particularly, in the early stage, the couriers resist detour and prefer to accept on-the-road pick-up parcels such that they can complete the carried drop-off parcels in time. In the late stage, the couriers prefer to collect the heavy pick-up parcels when they complete most of drop-off parcels and have free time and unoccupied capacity, because their service fees are usually positively related to the parcel's weight.

- Second, for commercial platforms, revenue is a critical business benefit that drives the platform to offer convenient logistics services. However, there is no existing work on the problem of revenue optimization for parcel assignment under the consideration of the courier's preference.

Therefore, the convergence for assigning couriers with desirable pick-up parcels and maximizing the platform's revenue is a good incentive to enhance the platform and couriers to participate in the crowdsourced FLML service. Since the auction mechanism is a widely used transaction mechanism that defends the social welfare of all parties involved in the transaction of goods or services [29], in this paper, we attempt to use the auction mechanism to solve the above issues.

Inspired by the above, in this paper, we investigate a novel type of crowdsourced FLML problem, namely *Auction-based Crowdsourced FLML (ACF)*, in which the platform assigns the couriers with suitable pick-up parcels based on the preferences of couriers for pick-up parcels. As illustrated in Fig. 1, the ACF services framework contains three entries: a set of pick-up parcels entering the platform in a stream fashion, a set of couriers full of drop-off parcels, and the platform. In general, customers submit their requests for pick-up parcels; couriers send in their preferences for the expected pick-up parcels; the platform is in charge of bid computation based on the couriers' preferences for parcels, and conducts the assignment for pick-up parcels and couriers in terms of the bids. The ACF problem aims to assign pick-up parcels for suitable couriers to maximize the social welfare of the platform and couriers.

While assisting couriers in bidding for ideal pick-up parcels efficiently, the ACF problem brings two new technical challenges. First, since the measure of the courier's preferences for pick-up parcels is manifold, it is non-trivial for us to define a reasonable auction model to support the preferences of couriers on different attributes of pick-up parcels. Moreover, similar to many existing works [14], [26], [47], we should ensure the proposed auction mechanism follows four auction properties including computational efficiency, individual rationality, budget balance, and truthfulness. Second, the ACF problem is NP-hard as proved later, it means that in the worst case the time complexity for finding the optimal assignment is exponential to the number of parcels and couriers. To address the first challenge, we propose a novel auction model, namely *Multi-attribute Reverse Vickrey (MRV)* auction, which is a variant of *Vickrey auction* [16], [36], [44]. In our auction model, couriers place their bids to pick-up parcels with sealed-bid based on their preferences of parcels (e.g., weight and detour). Given a pick-up parcel, the platform selects the courier that places the lowest bid as the winner and gives the winner the second lowest bid as the payment. Based on the MRV auction model, we propose three efficient parcel assignment algorithms to solve the second challenge. Specifically, we first present a greedy algorithm to find the assignment of couriers and parcels efficiently. Thereafter, we develop a multi-round assignment algorithm and its variant to improve the social welfare further. Moreover, we provide theoretical analysis for these three algorithms. Finally, we conduct extensive experiments to evaluate our proposed algorithms. In brief, our key contributions can be summarized as follows:

- We formally define a novel ACF problem which assigns proper pick-up parcels to couriers and aims to maximize the social welfare for the platform and couriers. We also establish the hardness of the ACF problem.
- We present a novel MRV auction model based on the Vickrey auction in which couriers bid on a pick-up parcel based on their preferences, the courier with the lowest bid wins the pick-up parcel and gets the payment equal to the second lowest bid.
- We propose a novel bid computation algorithm to compute the bid of each courier and present three efficient assignment algorithms to solve the ACF problem.
- We prove that the MRV auction model implemented by our proposed algorithms is computational efficient, individual rational, budget balanced, and truthful.
- Extensive experiments conducted on real-world datasets confirm that our proposed algorithms outperform the baselines.

We organize the rest of this paper as follows. We introduce the related work in Section 2. In Section 3, we present the models and problem formulation. Then, we propose our solutions to solve the ACF problem and give the theoretical analysis for the proposed auction mechanism in Section 4. After conducting the empirical study in Section 5, we conclude this paper in Section 6.

2 Related Work

To the best of our knowledge, we are the first to study the ACF problem. In this section, we review a selected set of related works on the first and last mile logistics and the auction mechanism.

2.1 Crowdsourced Logistics

Crowdsourced logistics is seen as a way to reduce the expensive cost of current FLML services [6], [35]. Existing works on crowdsourced logistics can be classified into two types. On the one hand, several existing works focus on the route planning of crowdsourced logistics. Wang et al. [38] study route planning for couriers to minimize the total detour cost of last mile delivery in crowdsourced logistics. They use the network min-cost flow to model route planning problem and propose several pruning techniques to improve the processing efficiency. Zeng et al. [42] propose an efficient kLMD operation framework to schedule the routes of multiple couriers under a certain optimization objective. Chen et al. [8] propose a two-stage solution to solve the problem of ridesharing and logistics at the same time. They predict the orders of the passengers by multivariate gauss distribution and Bayesian inference and plan parcel delivery routes in crowdsourced logistics. In addition, the other important study is to estimate the arrival time for stay points based on route planning in crowdsourced logistics. Ruan et al. [33] plan the couriers' routes based on the most potential stay points in order to automatically infer the delivery time of each task. On the other hand, a line of studies focus on task assignment of crowdsourced logistics. Zhang et al. [43] study the problem of task assignment where they assign the tasks based on couriers' shortest incurred distance (i.e., minimum detour cost) and aim to maximize the number of completed tasks. Li et al. [23] study a novel task assignment problem in a city crowdsourced logistics scenario. They present a reinforcement learning solution that relocates the couriers to suitable areas to complete as many tasks as possible. Liu et al. [24] construct a food delivery network of crowdsourced logistics. They propose an adaptive large neighborhood search algorithm based on simulated annealing to complete the food delivery tasks while minimizing the number of selected taxis. In summary, despite the potential to provide significant efficiency, these studies cannot reasonably support the auction of diverse preferences for couriers and thus cannot solve the ACF problem well.

2.2 Auction Mechanism

The auction mechanism refers to a market mechanism that handles resource allocation. There exists a line of existing works studying the auction mechanisms [2], [10], [11], [16], [20], [25], [40], [41], [47]. Asghari et al. [2] and Zheng et al. [47] study the first price sealed-bid auction in ridesharing, in which the platform selects the driver or requester placing the highest sealed-bid as winner and treats the highest bid as final payment. However, the first price auction mechanism often results in utility loss of the participants. To address this problem, Kleiner et al. [16] study the second price auction (i.e., vickrey auction) in crowdsourcing, they determine the bidder placing the highest price as the winner and pay the winner the second price as payment. Note that the first price auction and the second price auction are forward auction in which the participants consist of one seller and many buyers. In contrast, several existing works have studied the reverse auction where there is one buyer and multiple sellers participating in an auction. Lee et al. [20] propose a novel incentive mechanism of dynamic pricing based on reverse auction to reduce the operating and incentive costs while improving the fairness of incentive distribution in economic models. Xiao et al. [41] propose a secure reverse auction model to assign tasks by protecting the sensitive information of workers. Besides, several recent works have studied the double auction in which bidding is not only carried out between the buyers and the sellers but also between the buyers or between the sellers. For example, Wei et al. [40] present a novel framework for truthful double auction to establish the truthfulness of auction in complex scenario. However, the above mentioned mechanisms cannot fit the features of reverse and multi-attribute preference in ACF service.

Although there are some latest existing works that inherit the advantages of both multi-attribute and reverse auction [10], [11], [25], they still cannot solve the ACF problem because they do not meet the truthfulness and real-time requirements. Specifically, Huang et al. [11] match the risk aversion of buyers and suppliers based on the multi-attribute reverse auction, but they cannot guarantee truthfulness. Ma et al. [25] present a margin bidding solution that transforms multi-attribute reverse auction into single-attribute auction and proved that the proposed margin bidding meets the truthfulness. However, their auction model requires multiple manual interventions from buyers and suppliers, which is inefficient and thus is not suitable for real-time scenarios. Hu et al. [10] propose a truthfulness incentive mechanism based on the multi-attribute reverse auction. They adopt the concept of "trust degree" to reduce the impact of malicious competition but they still cannot avoid them.

TABLE 1
Summary of Notations

Notation	Description
λ, Λ	a drop-off parcel and a set of drop-off parcels
τ, Γ	a pick-up parcel and a set of pick-up parcels
l_τ, t_τ	the pick-up point and deadline of τ
w_τ, f_τ	the weight and fare of τ
l_λ, w_λ	the drop-off point and weight of λ
c, C	a courier and a sets of couriers
l_c, w_c	the current point and maximum capacity of c
t_c	the deadline returning to the station of c
α_c, β_c	the preference coefficients of c
$\mathcal{S}_c$	the schedule of c
r_0, μ	the base revenue and the sharing rate
$bid(c, \tau)$	the bid of c to τ
$\pi(\cdot, \cdot)$	the shortest distance between two points
$\Delta w_\tau, \Delta d_\tau$	the unoccupied rate and detour rate
$pay(c, \tau)$	the payment of c completing τ
u_c, u_p	the courier utility and platform utility
$\mathcal{U}(C, \Gamma)$	the social welfare of C and Γ

3 Models and Problem Formulation

In this section, we first introduce our system model and auction model. Then, we formally define the ACF problem and establish its hardness. We summarize the frequently used notations throughout this paper in Table 1.

3.1 System Model

In modern logistics, the commercial platform typically establishes a set of stations $\mathcal{H}$ in charge of the FLML service for the whole city. Each station $h \in \mathcal{H}$ is denoted as a tuple $h = \langle l_h, C_h, \mathcal{R}_h \rangle$ where the station h located at point l_h manages a set of couriers C_h to deliver and collect parcels within the service region $\mathcal{R}_h$. In order to clearly define our system model, we first describe the process of the crowdsourced FLML service as follows. In brief, the platform arranges trucks to deliver parcels to corresponding stations according to the destinations of parcels in several batches each working day, and the stations arrange for the couriers to deliver the parcels to customers. During the delivery process, the platform assigns couriers to collect some of pick-up parcels requested by customers. Finally, the courier needs to complete all the drop-off parcels and bring all the pick-up parcels back to the stations on or before the deadline. We clarify that the platform always arranges some couriers to collect the parcels that have not been picked up for a long time, in order to avoid hurting the platform's reputation and customer service quality [42], [43]. In what follows, we formally define the drop-off parcel, the pick-up parcel, and the courier that are closely related to the ACF problem.

Definition 1 (Pick-up parcel). *A pick-up parcel $\tau \in \Gamma$ is denoted as a four-entry tuple $\tau = \langle l_\tau, t_\tau, w_\tau, f_\tau \rangle$ where a parcel with weight w_τ and fare f_τ should be collected at pick-up point l_τ before deadline t_τ.*

Definition 2 (Drop-off parcel). *A drop-off parcel $\lambda \in \Lambda$ is denoted as a three-entry tuple $\lambda = \langle l_\lambda, t_\lambda, w_\lambda \rangle$ where a parcel with weight w_λ should be delivered at drop-off point l_λ before deadline t_λ.*

Note that the customer may not be at home for some unexpected reasons. Therefore, following the settings of the previous works [32], [33], we assume that the parcel's pick-up and drop-off points can be at the customer's home or express cabinet, and the courier can deliver (or collect) the parcels into (or from) a nearby express cabinet when the target customer is not at home. In addition, we clarify that the fare is not explicitly defined in the drop-off parcel because, in reality, the couriers should finish a certain amount of drop-off parcels every day to earn a basic salary; the platform encourages the couriers to earn the additional bonus by collecting pick-up parcels; the platform and the courier typically share the revenue for completing the pick-up parcel in proportion. Therefore, the revenues of the platform and the courier are both dominated by pick-up parcels. For this reason, in this paper, we only focus on the revenue brought by completing the pick-up parcels.

Definition 3 (Courier). *A courier $c \in C$ is denoted as a seven-entry tuple $c = \langle l_c, w_c, t_c, \alpha_c, \beta_c, \Lambda_c, \Gamma_c \rangle$, where l_c is the current point of c, w_c is the maximum capacity, t_c is the deadline returning to station, α_c and β_c are the preference coefficients to the weight and detour of pick-up parcel ($\alpha_c + \beta_c = 1$), Λ_c is a set of drop-off parcels, and Γ_c is a set of pick-up parcels.*

We clarify that the deadlines of drop-off parcels Λ_c and pick-up parcels Γ_c are earlier than the deadline t_c of the courier, because the courier should return to the station on time to start the next batch of parcel delivery (i.e., $t_\tau \le t_c$ and $t_\lambda \le t_c$). In addition, for ease of presentation, we use $\overline{w_c}$ to denote the unoccupied capacity of courier c where $\overline{w_c} = w_c - \sum_{\lambda \in \Lambda_c} w_\lambda - \sum_{\tau \in \Gamma_c} w_\tau$. Next, we formally define the schedule for a courier to deliver and collect parcels in Definition 4.

Definition 4 (Schedule). *A schedule of a courier c is represented as a time-based point sequence $\mathcal{S}_c = \langle l_1, l_2, \ldots, l_m \rangle$, where l_i is a pick-up or drop-off point of a parcel in Γ_c and Λ_c. A valid schedule $\mathcal{S}_c$ should satisfy the constraints: (i) the total weight of parcels in Γ_c and Λ_c should be less than or equal to w_c at any time, (ii) all the parcels in Γ_c and Λ_c and the courier c should satisfy their deadline constraints.*

Following the existing works [9], [27], [46], we assume that the courier always selects the shortest path between two points in the schedule, and the order of pick-up parcels in the current schedule does not need to be changed when inserting a new pick-up parcel. As explained in existing works [9], [27], reordering all the existing points in the current schedule will greatly increase the time cost, and that is not suitable for the real-time scenarios where the operation of parcel insertion is invoked frequently. In contrast, it is more practical to insert the new pick-up parcel at a suitable location of the current schedule to minimize the increase of total travel distance. Besides, we assume each courier full of drop-off parcels has an initial delivery schedule based on his/her delivery experience when leaving the station. How to reasonably plan the couriers' initial delivery schedule so that they have more chances to receive desirable pick-up parcels is another interesting topic that will be investigated in our future works.

Definition 5 (Parcel assignment). *In ACF problem, the platform periodically assigns the pick-up parcels to suitable couriers in a batch fashion. Each courier can collect one or more pick-up parcels as long as there exists a valid schedule to collect them. If there exists an unassigned pick-up parcel in the current batch, it will be handled in the next batch.*

As mentioned above, the platform periodically conducts parcel assignments in a batch fashion. Therefore, following the existing works [30], [46], we divide a stream of pick-up parcels into a time-aware batch sequence $\mathcal{B} = \langle b_1, b_2, \ldots, b_m \rangle$ by a sliding window (e.g., 15s). The pick-up parcels that are not assigned in the current batch b_i will be handled in the next batch b_{i+1}. Moreover, we clarify that the platform and all couriers belong to the same service provider in this paper.

In what follows, we would like to elaborate on data acquisition and processing for our ACF problem. As illustrated in Fig. 1, there are three parties in the framework, including couriers, customers, and the platform. For data acquisition, when delivering and collecting parcels, couriers report their status information (such as location, capacity, schedule, and preferences) to the platform in real time via their GPS-enabled mobile devices. For data processing, the platform will periodically (i.e., every few seconds) assign the parcel pick-up request received online to the appropriate couriers for execution based on the couriers' real-time status information. Note that some couriers may change their preference coefficients in an assignment period, but the platform completes the assignment only based on the couriers' latest submitted coefficients. After conducting parcel assignments, the platform will update the new route schedules for couriers immediately.

3.2 Auction Model

In this section, we introduce the mechanism of how to assign pick-up parcels to suitable couriers, which is achieved by an auction model. In existing auction models, vickrey auction is a well-known sealed auction in which bidders bid at the highest price that they can truly accept, they submit their bids to an item without knowing the bids of others in the auction, and the highest bidder wins the item but only needs to pay for the second-highest bid. Inspired by the above, in this paper, we present a novel auction model, namely *multi-attribute reverse vickrey (MRV)*, in which couriers bid on a pick-up parcel based on their true preference on the attributes of a pick-up parcel, the courier with the lowest bid wins the pick-up parcel and receives a payment equal to the second lowest bid. Compared with the vickrey auction model [15], [39], [45], our MRV auction model supports the reverse bidding of the couriers under multi-attribute preferences, which is very suitable for the crowdsourced FLML service. Similar to the vickrey auction model, our MRV auction model assumes that the couriers bid according to their true preferences, and their bids are the lowest payment they can accept.

Next, we present how to compute the bid based on the courier's preference on the attributes of a pick-up parcel. Since the parcel's weight and the detour caused by collecting a parcel are the two attributes that the courier cares mostly, in our bid model, we focus on calculating the bid of the courier on the two attributes of detour and weight. Without loss of generality, we will discuss later how to extend our bid model to bid the parcel with more than two attributes. Given a courier c and a pick-up parcel τ, the bid of c to τ is defined as follows:

$$bid(c,\tau) = \begin{cases} r_0 + \mu f_\tau, & \textbf{if } sum(\tau) = 1 \\ r_0 + (\alpha_c \Delta w_\tau + \beta_c \Delta d_\tau)\mu f_\tau, & \textbf{if } sum(\tau) \geq 2 \end{cases} \quad (1)$$

Here, $sum(\tau)$ denotes the number of couriers who bid for τ. If $sum(\tau)$ is 1, the bid is $r_0 + \mu f_\tau$ where r_0 and μ are base price and sharing rate for collecting τ, respectively. For example, assume $f_\tau = 20$, $r_0 = 2$, and $\mu = 0.2$, then we have $bid(c,\tau)$ = 2 + 0.2×20 = 6. In this case, since there is no other courier biding for this parcel, the only bidder c will receive a standard revenue for collecting τ. If $sum(\tau)$ is no less than 2, the bid metric is $r_0 + (\alpha_c \Delta w_\tau + \beta_c \Delta d_\tau)\mu f_\tau$, where α_c and β_c are two preference coefficients to balance the capacity unoccupied ratio Δw_c and the detour ratio Δd_τ, respectively. In this case, the courier with the closest unoccupied capacity to the weight of τ and the smallest detour from collecting τ will win τ with the lowest bid, i.e., the value of $(\alpha_c \Delta w_\tau + \beta_c \Delta d_\tau)$ is closest to 1. Here,

$$\Delta w_\tau = 1 - \frac{w_\tau}{\overline{w}_c}, \quad (2)$$

where Δw_τ means that the closer the weight w_τ of τ is to the unoccupied capacity $\overline{w}_c$ of c, the more likely c wins τ. For the computation of Δd_τ, we do not consider reordering the existing parcels in a schedule when inserting a new pick-up parcel τ, where the reason has been stated when we define the schedule. Thus, we define that,

$$\Delta d_\tau = \min_{1 \leq i \leq |S_c|-1} 1 - \frac{\pi(l_i, l_{i+1})}{\pi(l_i, l_\tau) + \pi(l_\tau, l_{i+1})}, \quad (3)$$

where $\pi(\cdot,\cdot)$ is the shortest path length between two points. Δd_τ indicates the detour rate incurred by c collecting τ.

Note that our bid model can be easily extended to involve more preference attributes. For example, one possible extension of the bid model is that we can replace $\alpha_\tau \Delta w_\tau + \beta_\tau \Delta d_\tau$ with $\vec{\alpha} \cdot \vec{\Delta}$ where $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ and $\vec{\Delta} = (\Delta_1, \Delta_2, \ldots, \Delta_m)$. Here, α_i and Δ_i denote the courier's preference coefficients and preference attributes, respectively. In practical settings, for ease of operation, the preference coefficients can be expressed as a set of options for couriers to select. Note that a centralized service platform is assumed in this paper, where the couriers only need to provide the preference for the detour and weight of the expected pick-up parcel, and they do not need to know the bids of other couriers.

From above discussion, given a pick-up parcel τ and a set of couriers C_τ ($|C_\tau| > 1$) who bid for τ, let c_{win} be the winner with the lowest bid to τ, the final payment $pay(c_{win}, \tau)$ of τ is the second lowest bid and that can be computed as follows:

$$pay(c_{win}, \tau) = \mathbf{min}\{bid(c,\tau) | c \in C_\tau - \{c_{win}\}\}. \quad (4)$$

We would like to clarify the courier's bid includes the courier's cost. For example, when a courier picks up a parcel, the courier's detour rate and the remaining capacity filling ratio are the measurements of the cost of a courier. The cost of the platform is generally the payment paid by the platform to a courier for the courier completing each pick-up parcel. The above cost descriptions of the platform and couriers are widely used in many spatial crowdsourcing studies [14], [47].

Example 1. Given a pick-up parcel τ and three couriers c_1, c_2, and c_3, we assume their bids to τ are $bid(c_1,\tau)$, $bid(c_2,\tau)$, and $bid(c_3,\tau)$ ($bid(c_1,\tau) < bid(c_2,\tau) < bid(c_3,\tau)$), respectively. Based on the MRV auction, c_1 with the lowest bid wins the parcel τ, the platform will assign τ to c_1 and pay c_1 with the second lowest bid $bid(c_2,\tau)$.

When we use the auction mechanism to solve crowdsourced logistics problems, the first (i.e., highest/lowest) price auction among many auction mechanisms can theoretically achieve the optimal revenue. However, the first price auction cannot satisfy the truthful property of the auction. As mentioned in related work [26], if an auction mechanism fails to satisfy the truthful property, then participants will be motivated to obtain additional revenue through unrealistic bids, which will harm the platform's long-term revenue. That is, couriers can increase their revenue by maliciously adjusting their preferences, leading to loss of platform revenue. As proved by Theorem 3 in our paper, the second-price auction satisfies the truthfulness property of the auction mechanism. That is, couriers cannot increase their own benefits by maliciously changing their preferences. Therefore, we adopt the second-price auction in this paper.

3.3 Problem Formulation

According to the above system and auction models, in this subsection, we first define the utilities of auction participants including courier and platform, followed by the objective function of social welfare for the ACF problem. Finally, we formally define the ACF problem.

Definition 6 (Courier utility). *Considering a pick-up parcel τ, the utility of the winner c_{win} to τ in an auction is,*

$$u_c(c_{win}, \tau) = pay(c_{win}, \tau) - bid(c_{win}, \tau). \tag{5}$$

Definition 7 (Platform utility). *Considering a pick-up parcel τ, the utility of the platform to assign τ to the winner c_{win} in an auction is*

$$u_p(c_{win}, \tau) = f_\tau - pay(c_{win}, \tau). \tag{6}$$

We clarify that the courier utility is used to measure how much the courier's actual payment exceeds his/her expected payment, while the platform utility reflects that the platform's profit gained from collecting a pick-up parcel. That is, the greater the courier's utility, the more interested the courier; the greater the platform's utility, the more profit it earns.

Next, we give the objective function of social welfare $\mathcal{U}(C, \Gamma)$ in Definition 8 to defend the interests of all participants and ensure they are willing to participate in an auction. Following the existing works [13], [18], [37], [47], we define the social welfare as the sum of all utilities in an auction.

Definition 8 (Social welfare). *Given a set of pick-up parcels Γ and a set of couriers C, the social welfare is defined as the sum of all utilities of participants in the auction, i.e.,*

$$\begin{aligned}\mathcal{U}(C, \Gamma) &= \sum_{\tau \in \Gamma, c_{win} \in C} u_p(c_{win}, \tau) + u_c(c_{win}, \tau) \\ &= \sum_{\tau \in \Gamma, c_{win} \in C} f_\tau - bid(c_{win}, \tau),\end{aligned} \tag{7}$$

From Equation (7), we observe that the payment $pay(c_{win}, \tau)$ does not affect the social welfare. Based on the above definitions, we formally define the ACF problem as follows:

Definition 9 (ACF problem). *Given a stream of pick-up parcels Γ and a set of couriers C, the ACF problem aims to find the best assignment for couriers and pick-up parcels such that the social welfare $\mathcal{U}(C, \Gamma)$ is maximized.*

In this paper, we adopt the auction mechanism to ensure that the courier truly gives preferences, which is guaranteed by the four auction properties we have proved in this paper. In such an auction mechanism, we conduct the assignments for couriers and parcels with the goal of maximizing the utility of both couriers and the platform as a whole. In what follows, we theoretically analyze the complexity of the ACF problem in Theorem 1.

Theorem 1. *The ACF problem is NP-hard.*

Proof. The proof can be found in Appendix A.1 (available online). □

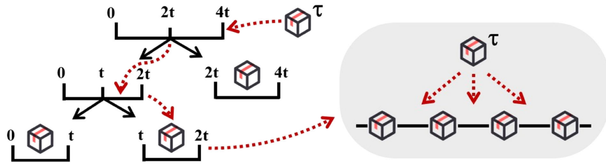


Fig. 2. An example of TS-tree structure.

4 Solutions for ACF Problem

In this paper, our proposed ACF platform provides a centralized service including three entries such as customers, couriers, and the platform. More specifically, the platform collects the status information of couriers (e.g., locations, preferences, and schedules) and parcel pick-up requests submitted by customers (e.g., locations, deadlines, and weights) in real time. Then, the platform assigns pick-up parcels to appropriate couriers based on our proposed auction mechanism which is implemented by two modules of bid computation and parcel assignment (see Fig. 1). For the bid computation module, it is served as a basic component to calculate the bid of a courier to a pick-up parcel based on the auction model illustrated in Section 3.2. For the parcel assignment module, it is invoked when there are pick-up parcels streaming into the platform. Next, we first present an efficient algorithm to quickly determine the bids of couriers. Then, we propose three efficient assignment algorithms to solve the ACF problem. Finally, we conduct a theoretical analysis for our MRV auction model.

4.1 Bid Computation

Bid computation is the core component in an auction mechanism. As described in our auction model, the platform selects the courier who places the lowest bid as the winner, and pays the winner the second lowest bid as the payment. Therefore, in this section we mainly introduce the method to calculate the bids efficiently. Given a courier c and a pick-up parcel τ, we can derive that computation overhead of $bid(c, \tau)$ is dominated by Δd_τ based on our bid model, because Δw_τ can be efficiently computed by Equation (2). A straightforward method to compute Δd_τ is to insert the pick-up parcel between any two consecutive points in current schedule to find the minimum detour ratio. However, this brute force approach is very time-consuming, especially when the number of pick-up parcels is huge. To solve this problem, we propose an efficient algorithm that integrates a *time-aware segment tree (TS-Tree)* to improve bid computation efficiency.

We construct the TS-Tree for each courier and maintain it over time when adding a new pick-up parcel or removing a completed pick-up parcel. Fig. 2 illustrates the basic structure of a TS-Tree. Briefly, the TS-Tree is a variant of B-Tree. Each internal node e is associated with a time interval $\xi_e = [\xi_e^-, \xi_e^+]$, where the bounds ξ_e^- and ξ_e^+ indicate the earliest time and latest time for collecting a pick-up parcel in all the child nodes respectively, while each leaf node points to a set of pick-up parcels. Considering the schedule $\mathcal{S}_c$ for a courier c, we adopt a dual search algorithm presented in [21], [22] to find the earliest and latest time to collect or deliver a parcel. In the beginning, the TS-tree only includes the drop-off parcels when a courier departs from the station. We insert a parcel into the TS-Tree according to whether the time interval of the parcel overlaps that of the tree node (i.e., the union of the time intervals of child nodes). Adopting the TS-Tree, we can quickly find a minimum set of parcels to insert a pick-up parcel. Thereafter, we retrieve the position with the minimum detour cost to insert a pick-up parcel into the minimum set of parcels (e.g., the shadow in Fig. 2). Finally, the bid computation is completed.

Algorithm 1. Bid Computation

Input: *A pick-up parcel* τ*, a courier* c*, a TS-Tree* tr
Output: *The* $bid(c, \tau)$ *of* c *to* τ
1: $bid(c, \tau) = \infty$;
2: **if** *c can reach l_τ before t_τ and $w_\tau < \overline{w_c}$* **then**
3: Calculate the time interval ξ_τ when c arrives at l_τ by dual search algorithm;
4: $\mathcal{S}'_c \leftarrow$ *GetParcels*(ξ_τ, $\mathcal{S}_c$, tr);
5: Insert τ into $\mathcal{S}'_c$ with minimum Δd_τ;
6: Invoke Equation (1) to calculate $bid(c, \tau)$;
7: **return** $bid(c, \tau)$;

Algorithm Details. The pseudo code of bid computation is illustrated in Algorithm 1. In the beginning, we check if the courier c can reach the pick-up point of a pick-up parcel τ in time and the unoccupied capacity $\overline{w_c}$ is greater than the parcel weight w_τ (line 2). If yes, we calculate the time interval ξ_τ when the courier c arrives at the pick-up point l_τ. After that, we invoke the function *GetParcels(·)* to find the sub-schedule $\mathcal{S}'_c$ affected by the insertion of τ. Due to the space limitation, we brief the idea of *GetParcels(·)*, which retrieves the parcels over the TS-tree tr in a top-down approach and returns a small set $\mathcal{S}'_c$ of parcels whose time intervals overlap the time interval ξ_τ (line 4). Then we find the best position between two connected points in $\mathcal{S}'_c$ such that the detour ratio Δd_τ is minimized (line 5). Finally, we calculate the $bid(c, \tau)$ by calling Equation (1) and return it as the bid for c to τ (lines 6–7).

Based on the bid computation, we can calculate all couriers' bids to a given pick-up parcel τ. Among these couriers' bids, we select the courier who places the lowest bid as the winner in the auction, and treat the second lowest bid as the payment.

Complexity Analysis. The time complexity of Algorithm 1 is $O(log|\mathcal{S}_c| + |\mathcal{S}'_c|)$, where $log|\mathcal{S}_c|$ is the time cost of finding the sub-schedule of $\mathcal{S}_c$ affected by inserting a pick-up parcel, and $|\mathcal{S}'_c|$ is the cost of finding the insertion in $\mathcal{S}'_c$ with minimum cost.

4.2 Parcel Assignment

Parcel assignment is the other important component in our MRV auction mechanism in addition to bid computation. In this section, we propose three efficient parcel assignments to solve the ACF problem. In what follows, we introduce them in details.

4.2.1 Greedy Algorithm

In this subsection, we present an efficient greedy algorithm to solve the ACF problem. The general idea of the greedy algorithm is as follows. Given a set of couriers C and a set of pick-up parcels Γ, for each pick-up parcel, we first find all couriers who have unoccupied capacity and can collect the pick-up parcel in time. Among these couriers, we assign the pick-up parcel to the courier with the lowest bid. Then, we iteratively repeat the above process until there are no couriers or pick-up parcels left. Finally, the courier-parcel assignment $\mathcal{M}$ is returned as the result. The details of the greedy algorithm are presented below.

Algorithm 2. Greedy Algorithm

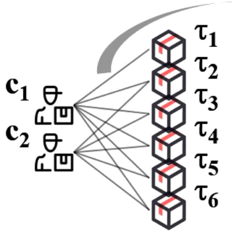
Input: *A courier set* C*, a pick-up parcel set* Γ
Output: *A courier-parcel assignment* $\mathcal{M}$
1: Initialize an assignment $\mathcal{M}$;
2: **for** *each* $\tau \in \Gamma$ **do**
3: $bid_m \leftarrow \infty$;
4: **for** *each* $c \in C$ **do**
5: $bid(c, \tau) \leftarrow$ call Algorithm 1;
6: **if** $bid_m > bid(c, \tau)$ **then**
7: $bid_m = bid(c, \tau)$;
8: **if** *bid_m is not ∞* **then**
9: Assign τ to c with $pay(c_{win}, \tau)$;
10: Update $\mathcal{M}$;
11: **return** $\mathcal{M}$;

Algorithm Details. The pseudo code of greedy algorithm is shown in Algorithm 2. For each pick-up parcel τ, we calculate the bids for the couriers who can complete the pick-up parcel by invoking Algorithm 1 (line 5). Then we select the courier with the lowest bid as the winner and set the second lowest bid as the winner's payment. If bid_m is not ∞, we assign the pick-up parcel to the winner and give the final payment $pay(c_{win}, \tau)$, and update the assignment $\mathcal{M}$ (lines 8–10). Finally, we return the assignment $\mathcal{M}$ (line 11).

Complexity Analysis. The time complexity of the greedy algorithm is $O(mn(log|\mathcal{S}_c| + |\mathcal{S}'_c|))$, where m is the number of couriers, n is the number of pick-up parcels, and $O(log|\mathcal{S}_c| + |\mathcal{S}'_c|)$ denotes the time complexity of bid computation.

4.2.2 Multi-Round Assignment Algorithm

As discussed above, the greedy algorithm can solve the ACF problem efficiently. However, the social welfare of the final assignment is not high. In this subsection, we present a novel *Multi-round Assignment (MRA)* algorithm to improve the social welfare of the assignment. The general idea of MRA algorithm is as follows. Given a set of pick-up parcels Γ and a set of couriers C, we first calculate the bids of the couriers in C to the pick-up parcels in Γ by Algorithm 1. Then, we construct a bipartite graph based on the bids between couriers and pick-up parcels, in which an edge connecting a courier and a pick-up parcel exists if the courier can serve the pick-up parcel and the weight of an edge is the bid of the courier to the pick-up parcel. Fig. 3 shows the basic structure of the bipartite graph. Utilizing the bipartite graph, we compute an optimal assignment in a multi-round assignment fashion. For each round, we process the assignments for couriers and pick-up parcels according to the weights of their linking edges in an ascending order, i.e., the courier with the lowest bid to the parcel is processed first. After that, we update the bipartite graph based on the assigned pick-up parcels and couriers. Specifically, we delete the edges linking the assigned pick-up parcels and



	c_1	c_2
τ_1	2.33	-
τ_2	3.62	3.32
τ_3	2.78	2.91
τ_4	3.84	2.93
τ_5	2.62	2.51
τ_6	3.61	2.21

Fig. 3. The structure of bipartite graph.

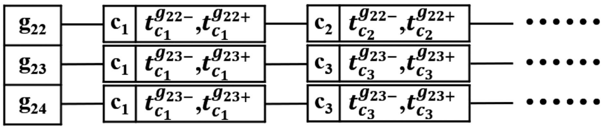


Fig. 4. The structure of the TIL index.

update the weights of the edges of the assigned couriers. Note that the rounds of assignment repeat until there is no remaining edge in the updated bipartite graph.

In each round, the most time-consuming operations are the bipartite graph construction and updating, since we should invoke Algorithm 1 to calculate the bid for each pair of courier and pick-up parcel. Actually, due to the time constraints of pick-up parcels and the capacity constraints of couriers, each courier can only serve a limited number of pick-up parcels. Based on this consideration, we present a novel index structure, namely *Time-bounded Inverted List (TIL)*, to improve the pruning efficiency for invalid courier-parcel pairs. More specifically, we first divide the whole road network into a set of continuous grid cells. Then, we construct our proposed TIL index based on these grid cells. For each grid cell, we index the couriers who can pass through the grid cell and record the time period they pass. Fig. 4 shows the structure of the TIL index. To construct the TIL index, we first formally define the reachable area of a courier as Definition 10.

Definition 10 (Reachable Area). *The reachable area of a courier c, denoted as ${}_c$, is a set of grid cells where the courier c may pass through the grid cell $g \in_c$ during the time interval $\Omega_c^g = [t_c^{g-}, t_c^{g+}]$ under the time constraints of c.*

Next, we introduce how to calculate the reachable area of a courier. Given a courier c, we can calculate the reachable area ${}_c$ based on the schedule $\mathcal{S}_c$ of c and the returning deadline t_c of c. Specifically, for each point l on the schedule $\mathcal{S}_c$, the grid cells where the courier c can collect the pick-up parcels are covered by the circle with the point l as the center and the maximum distance that the courier c can reach before the deadline t_c as the radius. For example, the shaded grid cells shown in Fig. 6 are the reachable area of a courier. In addition, we also record the courier's earliest arrival time t_c^{g-} and the latest departure time t_c^{g+} regarding any reachable grid cell c. Here, the earliest arrival time t_c^{g-} can be calculated by the current time t_n and the minimum travel time $t_{min}(l_c, g)$ from the current location l_c to the grid cell g,

$$t_c^{g-} = t_n + t_{min}(l_c, g), \tag{8}$$

and the latest departure time t_c^{g+} can be calculated by the returning deadline t_c and the minimum travel time $t_{min}(l_h, g)$ from the grid cell g to the transit hub's location l_h,

$$t_c^{g+} = t_c - t_{min}(l_h, g). \tag{9}$$

Making use of the TIL index, we can quickly find the couriers who are likely to collect a given pick-up parcel. More specifically, considering a pick-up parcel τ, we first identify the earliest pick-up time of τ by $t_\tau^- = t_n + t_{min}(l_{c'}, l_\tau)$ where t_n is the current time and $t_{min}(l_{c'}, l_\tau)$ is the minimum time that the nearest courier c' of τ moves to the pick-up point l_τ. Then, we identify the grid cell g where the pick-up point l_τ locates in and get the couriers who are associated with the grid cell g by the TIL index. For each courier c linked with g, we check whether the time interval for picking up τ from t_τ^- to t_τ and the time interval Ω_c^g of c overlaps. If not, c cannot collect the pick-up parcel in time. From the TIL index, we can quickly find all valid candidate courier-parcel assignments and only need to compute bids for them, which greatly improves the processing performance.

Next, we consider the updates of the TIL index in two cases. First, if a courier has collected or delivered some parcels as some grid cells, the courier should be removed from the linked cells of the TIL index. Second, If a courier wins to collect a new pick-up parcel, the reachable area of this courier (i.e., the radius of the circle in Fig. 6) will decrease, since the courier may spend additional time cost to collect the new pick-up parcel. As such, we need to update the TIL index based on the courier's shrinking available area.

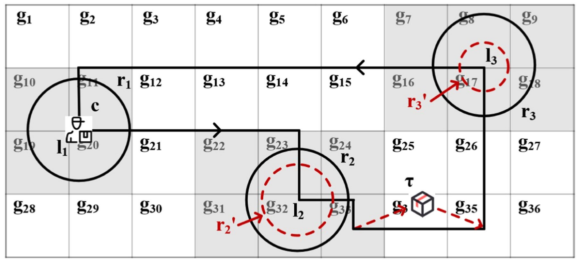
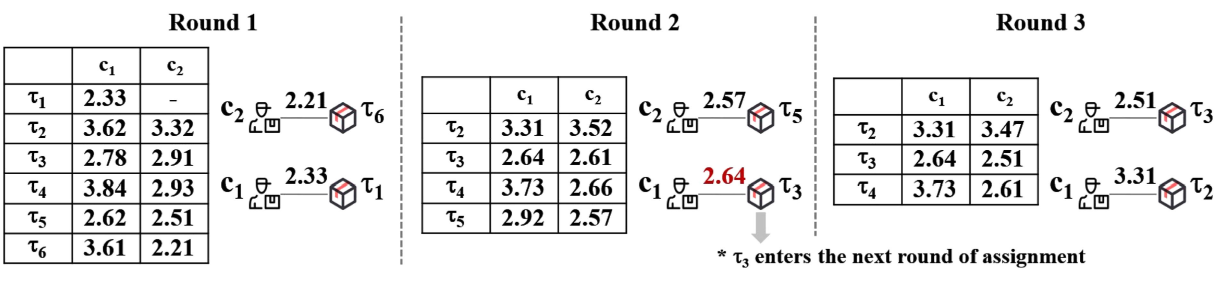


Fig. 6. An example of a courier's reachable area.



Round 1

	c_1	c_2
τ_1	2.33	-
τ_2	3.62	3.32
τ_3	2.78	2.91
τ_4	3.84	2.93
τ_5	2.62	2.51
τ_6	3.61	2.21

Round 2

	c_1	c_2
τ_2	3.31	3.52
τ_3	2.64	2.61
τ_4	3.73	2.66
τ_5	2.92	2.57

Round 3

	c_1	c_2
τ_2	3.31	3.47
τ_3	2.64	2.51
τ_4	3.73	2.61

Fig. 5. A running example of the MRA algorithm.

Meanwhile, we calculate the reachable area of the new pick-up parcel and update the TIL index accordingly. In what follows, we present the MRA algorithm in details.

Algorithm 3. MRA Algorithm

Input: *A courier set* C, *a pick-up parcel set* Γ
Output: *A courier-parcel assignment* $\mathcal{M}$
1: Initialize an assignment $\mathcal{M}$ and a bipartite graph $\mathcal{G}$ by $\emptyset$;
2: Find the candidate courier-parcel assignments $\mathcal{M}'$ over TIL;
3: **for** *each* $(c, \tau) \in \mathcal{M}'$ **do**
4: $\quad bid(c, \tau) \leftarrow$ call Algorithm 1;
5: $\quad$ **if** $bid(c, \tau)$ *is not* ∞ **then**
6: $\quad\quad$ Add the entry $\langle c, \tau, bid(c, \tau)\rangle$ into $\mathcal{G}$;
7: **while** $\mathcal{G}$ *is not empty* **do**
8: $\quad \mathcal{L} \leftarrow$ the entries of $\mathcal{G}$ in ascending order of $bid(c, \tau)$;
9: $\quad$ **while** $\mathcal{L}$ *is not empty* **do**
10: $\quad\quad e \leftarrow$ the first entry in $\mathcal{L}$;
11: $\quad\quad$ **if** $e.bid(c, \tau)$ *is the lowest bid of* $e.\tau$ *in* $\mathcal{G}$ **then**
12: $\quad\quad\quad$ Add e into $\mathcal{M}$;
13: $\quad\quad\quad$ Remove the entries of $e.\tau$ in $\mathcal{L}$;
14: $\quad\quad$ Remove the entries of $e.c$ in $\mathcal{L}$;
15: $\quad$ Update $\mathcal{G}$ based on $\mathcal{M}$;
16: **return** $\mathcal{M}$;

Algorithm Details. Algorithm 3 illustrates the pseudo code of the MRA algorithm, where the input contains a courier set C and a pick-up parcel set Γ, the output is an optimal courier-parcel assignment $\mathcal{M}$. In the beginning, we initialize an assignment $\mathcal{M}$ and a bipartite graph $\mathcal{G}$ with empty. Then, we find the candidate courier-parcel assignments $\mathcal{M}'$ by traversing the inverted list index *TIL*. For each candidate courier-parcel assignment (c, τ) in $\mathcal{M}'$, we invoke Algorithm 1 to calculate its bid $bid(c, \tau)$. If $bid(c, \tau)$ is not infinity, we add the entry $\langle c, \tau, bid(c, \tau)\rangle$ into $\mathcal{G}$. The above steps construct a bipartite graph $\mathcal{G}$. After that, we iteratively assign pick-up parcels to couriers in a round fashion until $\mathcal{G}$ is empty. In each round of assignment, if $\mathcal{G}$ is not empty, we initialize an entry list $\mathcal{L}$ with the entries of $\mathcal{G}$ which are sorted in ascending order of the bids. Then, if the bid $e.bid(c, \tau)$ of the first entry e is the lowest bid placed to τ in $\mathcal{G}$, we assign the pick-up parcel $e.\tau$ to the courier $e.c$. Meanwhile, the entry e is added into the assignment $\mathcal{M}$, the entries of $e.\tau$ or $e.c$ are removed from $\mathcal{L}$. According to $\mathcal{M}$, we update $\mathcal{G}$ by removing the entries that contain the assigned pick-up parcels and recalculating the bids for the entries that contain the couriers who are assigned new pick-up parcels, since the newly assigned pick-up parcels may change their bids (lines 7–14). Moreover, we also remove the entries that contain the couriers who have no available capacity or place no bids for any pick-up parcels (line 15). Finally, the assignment $\mathcal{M}$ is returned (line 16).

Example 2. A running example of the MRA algorithm is shown in Fig. 5, where there are two couriers c_1 with capacity 2 and c_2 with capacity 3, and six pick-up parcels $\tau_1 \sim \tau_6$ with weight 1 and fare 6. In Fig. 5, the whole assigning process contains three rounds. In Round 1, τ_6 is first assigned to c_2, because after sorting all entries, the entry $\langle c_2, \tau_6, 2.21\rangle$ owns the lowest bid 2.21 which is also the lowest bid placed to τ_6. Then, τ_1 is assigned to c_1 since after we remove all entries of τ_6, the entry $\langle c_1, \tau_1, 2.33\rangle$ has the lowest bid 2.33 which is also the lowest bid placed to τ_1. We update the bids of c_1 and c_2 and delete τ_1 and τ_6 based on the assignment in Round 1. In Round 2, we first assign τ_5 to c_2 because the entry $\langle c_2, \tau_5, 2.57\rangle$ has the lowest bid 2.57 which is also the lowest bid placed to τ_5. After assigning τ_5 to c_2 and removing all entries of τ_5, since the $\langle c_1, \tau_3, 2.64\rangle$ has the lowest bid 2.64 but it is not the lowest bid placed to τ_3, τ_3 enters the next round of assignment. Accordingly, we update the bids of c_2 and delete τ_5. Similar to the assignments in Round 1&2, in Round 3, we assign τ_3 and τ_2 to c_2 and c_1, respectively. After Round 3, all other pick-up parcels are assigned except for τ_4, because there are no available capacities for c_1 and c_2 to collect τ_4. Finally, the assignments of these three rounds are returned as the final results.

Complexity Analysis. The time complexity of the MRA algorithm is $O(nglogm + mn(log|\mathcal{S}_c| + |\mathcal{S}'_c|) + n|\mathcal{L}|log|\mathcal{L}|)$, where g is the number of grids, m is the number of couriers, n is the number of pick-up parcels, and $|\mathcal{L}|$ is the number of entries in $\mathcal{G}$. $O(nglogm)$ is the cost of finding candidate courier-parcel assignment (line 2). $O(mn(log|\mathcal{S}_c| + |\mathcal{S}'_c|))$ is the time complexity of constructing G (lines 3–6). $O(n|\mathcal{L}|log|\mathcal{L}|)$ is the time complexity of the multi-round assignment based on G (lines 7–16).

4.2.3 Packing-Based Optimization

Compared with the greedy algorithm, the MRA algorithm can refine the quality of assignment, but it also increases the processing time. Although the TIL index can filter out many invalid courier-parcel assignments quickly and accelerate the search speed, the time cost is still extremely high when a large number of pick-up parcels enter the platform on a short notice. In practice, many pick-up parcels are more concentrated (e.g., neighborhood or the same street). If these pick-up parcels can be packed as one to process, the computation time cost can be dramatically reduced. Motivated by this, we propose an efficient *packing-based optimization (PBO)* algorithm with social welfare loss guarantees to solve this issue. In brief, the idea of packing pick-up parcels is that two pick-up parcels τ_i and τ_j can be packed into a package Γ^g if it satisfies the following condition:

$$\pi(l_{\tau_i}, l_{\tau_j}) \leq \delta, \quad i \neq j \tag{10}$$

where δ is the threshold that indicates the maximum distance between any two parcels of a package. Then, we define the bid of a courier c to a package Γ^g as follows:

$$bid(c, \Gamma^g) = \kappa \cdot r_0 + (\alpha_c \Delta w_{\Gamma^g} + \beta_c \Delta d_{\Gamma^g})\mu f_{\Gamma^g} \tag{11}$$

where κ is the number of pick-up parcels in Γ^g, Δw_{Γ^g} and Δd_{Γ^g} denote the unoccupied ratio and the detour ratio of picking up a package, respectively. f_{Γ^g} is the sum of all fares of parcels in a package. In the unpacking mode, we need to calculate the bid for each pick-up parcel. While in the packing mode, we only need to calculate once for the whole package. In a package, if κ pick-up parcels are packed up, we need to select the shortest route from $\kappa!$ routes. Obviously, the computational cost of finding the shortest route is huge. To reduce the time cost, we herein select an arbitrary route instead of the shortest route. In Theorem 2, we establish the upper bound of social welfare loss between any route and the shortest one.

Algorithm Details. Algorithm 4 shows the pseudo code of PBO algorithm. In the beginning, we initialize a package set Γ^G and a bipartite graph $\mathcal{G}$ (line 1). For each pick-up parcel $\tau \in \Gamma$, we attempt to add τ into a Γ^g in Γ^G by packing policy (see Equation (10)). If τ cannot be added into any package in Γ^G, we create a new package Γ^g_{new} for τ and add Γ^g_{new} into Γ^G (lines 2–9). Then we find the candidate courier-parcel assignments $\mathcal{M}'$ from TIL and add all the valid courier-parcel assignments into $\mathcal{G}$ (lines 10–14). Finally, we execute the same processing like the MRA algorithm on $\mathcal{G}$ to find the best assignment $\mathcal{M}$ (lines 15–24).

Algorithm 4. PBO Algorithm

Input: *A courier set C, a pick-up parcel set Γ*
Output: *A courier-parcel assignment $\mathcal{M}$*
1: Initialize an assignment $\mathcal{M}$, a package set Γ^G, a bipartite graph $\mathcal{G}$ by $\emptyset$;
2: **for** *each* $\tau \in \Gamma$ **do**
3:   **for** *each package* $\Gamma^g \in \Gamma^G$ **do**
4:     **if** τ *can be added into* Γ^g *by packing policy* **then**
5:       Add τ into Γ^g;
6:       **break**;
7:   **if** τ *cannot be added into any package of* Γ^G **then**
8:     Initialize a new package Γ^g_{new} for τ;
9:     Add Γ^g_{new} into Γ^G;
10: Find the candidate assignment $\mathcal{M}'$ over TIL;
11: **for** *each* $(c, \Gamma^g) \in \mathcal{M}'$ **do**
12:   $bid(c, \Gamma^g) \leftarrow$ call Algorithm 1;
13:   **if** $bid(c, \Gamma^g)$ *is not* ∞ **then**
14:     Add the entry $\langle c, \Gamma^g, bid(c, \Gamma^g)\rangle$ into $\mathcal{G}$;
15: **while** $\mathcal{G}$ *is not empty* **do**
16:   $\mathcal{L} \leftarrow$the entries of $\mathcal{G}$ in ascending order of $bid(c, \Gamma^g)$;
17:   **while** $\mathcal{L}$ *is not empty* **do**
18:     $e \leftarrow$ the first entry in $\mathcal{L}$;
19:     **if** $e.bid(c, \Gamma^g)$ *is the lowest bid of* $e.\Gamma^g$ *in* $\mathcal{G}$ **then**
20:       Add e into $\mathcal{M}$;
21:       Remove the entries of $e.\Gamma^g$ in $\mathcal{L}$;
22:     Remove the entries of $e.c$ in $\mathcal{L}$;
23:   Update $\mathcal{G}$ based on $\mathcal{M}$;
24: **return** $\mathcal{M}$;

Complexity Analysis. The time complexity of PBO algorithm is $O(|\Gamma^G|(n+glogm)+m|\Gamma^G|(log|\mathcal{S}_c|+|\mathcal{S}'_c|)+|\Gamma^G||\mathcal{L}|log|\mathcal{L}|)$ where g is the number of grids, m is the number of couriers, n is the number of pick-up parcels, and $|\Gamma^G|$ is the number of packages. $O(|\Gamma^G|(n+glogm))$ is the time complexity of package generation and finding candidate assignments (lines 2–10). $O(m|\Gamma^G|(log|\mathcal{S}_c|+|\mathcal{S}'_c|))$ is the time complexity of constructing G (lines 11–14). $O(|\Gamma^G||\mathcal{L}|log|\mathcal{L}|)$ is the time complexity of the multi-round assignment (lines 15–24).

Theorem 2. *The upper bound on social welfare loss of the PBO algorithm is* $\frac{\beta_c \mu f_{\Gamma^g}}{\phi}(\kappa+1)\delta$*, where* ϕ *is the distance between the two adjacent points before and after the package insertion point.*

Proof. The proof can be found in Appendix A.2, available in the online supplemental material. □

4.3 Auction Mechanism Analysis

So far we have proposed several efficient algorithms to find the best parcel assignment for the ACF problem. Following the existing works [3], [17], we should ensure the proposed auction mechanism implemented by our proposed assignment algorithms follows the below properties:

- *Computational efficiency.* An auction mechanism is computational efficient if the time complexity of assignment algorithm is polynomial.
- *Individual rationality.* An auction mechanism is individual rational if each courier's utility is non-negative.
- *Budget balance.* An auction mechanism is budget balanced if the platform's utility is non-negative.
- *Truthfulness.* An auction mechanism is truthful if couriers place their bids based on their true preferences and cannot increase their utilities by placing false bids.

In what follows, we prove our solutions follow the auction properties in Theorem 3 and 4.

Theorem 3. *The auction mechanism implemented by Algorithm 2 is computational efficiency, individual rationality, budget balance, and truthfulness.*

Proof. The proof can be found in Appendix A.3, available in the online supplemental material. □

Theorem 4. *The auction mechanisms implemented by Algorithm 3 and Algorithm 4 are computational efficiency, individual rationality, budget balance, and truthfulness.*

Proof. Since the proofs on the computational efficiency, individual rationality, budget balance, and truthfulness of Algorithms 3 and 4 are similar to those of Algorithm 2 and can be easily proved. Therefore, we omit the proofs here. □

5 Empirical Study

In this section, we first present the experimental settings, and then evaluate the efficiency and effectiveness of our solutions on two real-world datasets by varying the settings of parameters in a wide range.

5.1 Experimental Settings

Data and Parameters. The two datasets used in our experiments are collected from Cainiao [5] and NYCTaxi [31], respectively. More specifically, the dataset Cainiao consists of one-day parcel orders of Cainiao logistics in Shanghai. We randomly select these parcel orders as the pick-up parcels and drop-off parcels. The dataset NYCTaxi contains one month's taxi trajectories placed in New York. Since the pick-up and drop-off points are typically the residential or commercial areas that correspond to the pick-up and drop-off points of the parcels, we extract the origins with pick-up time and the destinations with drop-off time from these trajectories and treat them as the points and time of pick-up parcels and drop-off parcels. We simulate the arrival of pick-up parcels based on the occurrence order of parcels in the open public datasets. For the road networks, we extract the road networks of Shanghai (216,231 edges and 14,945 nodes) and New York (8,625,942 edges and 142,428 nodes) from OpenStreetMap[1]. Since the parcel's weight is not

1. http://www.openstreetmap.org

TABLE 2
Parameter Settings

Parameter	Value
# of couriers $\vert C\vert$ (Cainiao)	0.1K,0.2K,**0.3K**,0.4K,0.5K
# of couriers $\vert C\vert$ (NYTaxi)	1K,2K,**3K**,4K,5K
# of parcels $\vert\Gamma\vert$ (Cainiao)	1.25K,2.5K,**5K**,10K,20K
# of parcels $\vert\Gamma\vert$ (NYTaxi)	12.5K,25K,**50K**,100K,200K
Capacity of courier w	25,50,**75**,100,125
Batch size b	5s,10s,**15s**,20s,30s
Packing threshold δ	10,25,**50**,100,200

available in both datasets, we randomly generate the weight from (0,10] for each parcel. Besides, the preference coefficients α_c and β_c for each courier are generated in a uniform distribution [19], and the platform reads the status information of couriers in every few seconds. All the parameter settings and the default values in bold font are listed in Table 2.

Compared Algorithms. We compare our proposed algorithms (GA in Section 4.2.1, MRA in Section 4.2.2, PBO in Section 4.2.3), the SIDF* [43] that assigns parcels to couriers in terms of the shortest incurred distance, and the ALNS [24] that selects a certain proportion of the largest matching to destroy in each round to avoid the assignment result falling into a local optimum. All proposed algorithms are implemented in Python and tested on a machine with Intel i7-9700K 3.6GHz CPU and 16GB of main memory.

Metrics. We test the effectiveness of all algorithms by the average completion ratio of pick-up parcels (CR) and the average total social welfare (SW) and evaluate the efficiency by the average processing time in each batch (BT).

5.2 Experimental Results

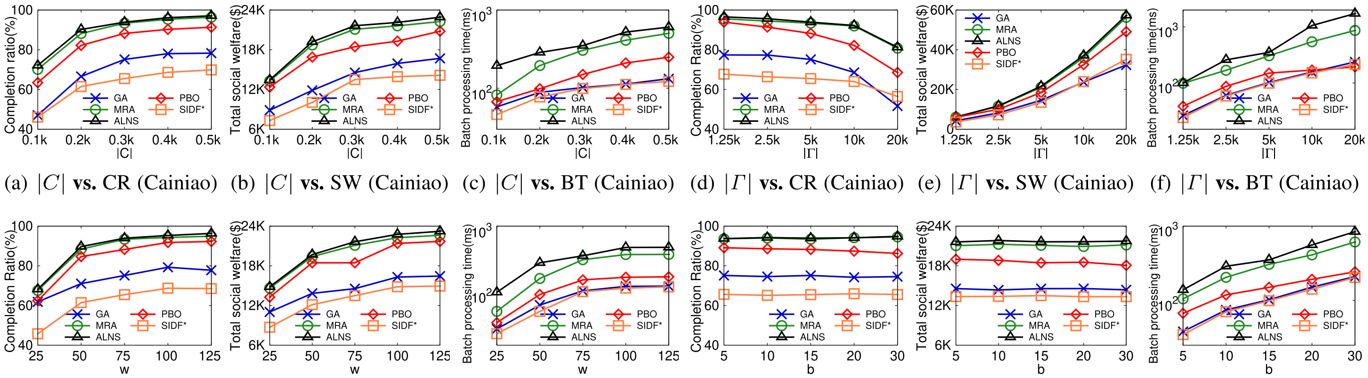
5.2.1 Evaluation on Cainiao Dataset

Effect of the Number of Couriers $|C|$. As shown in Figs. 7a and 7b, we observe that the completion ratio and social welfare of all algorithms increase when increasing the number of couriers, because the more the number of couriers, the more pick-up parcels that can be served, leading to a higher completion ratio and social welfare. Compared with GA and SIDF*, other algorithms (MRA, PBO, and ALNS) perform better on the completion ratio and social welfare, since they consider the assignment of all couriers and parcels in one batch, while GA and SIDF* do the assignment in the first-come-first-serve approach. Fig. 7c shows that GA and SIDF* have faster processing time than MRA, PBO, and ALNS, since they need more time to find the best assignment in a wide range. In addition, compared with MRA and ALNS, PBO achieves 17.73%$\sim$ 49.98% performance improvement on processing time at the cost of 6.55%$\sim$ 6.94% loss on completion ratio and social welfare due to our packing-based optimization.

Effect of the Number of Parcels $|\Gamma|$. Figs. 7d, 7e, and 7f report the evaluation results of the compared algorithms by varying the number of parcels. It can be observed that with the increase of the number of parcels, the social welfare and processing time increase but the completion ratio decreases. The reason for the decrease in completion rate is that the capability to serve more pick-up parcels is limited by the maximum unoccupied capacity of a fixed number of couriers. MRA and ALNS achieve the best effectiveness in completion ratio and social welfare among all compared algorithms but have the highest processing time (reaching 0.98s and 1.28s). Due to the packing-based optimization, the completion ratio and social welfare of PBO algorithm are close to those of MRA and ALNS. Meanwhile, the processing time of PBO algorithm is comparable to that of GA and SIDF*.

Effect of the Maximum Capacity w. The effect of the maximum capacity is reported in Figs. 7g, 7h, and 7i. Intuitively, a larger maximum capacity allows couriers to be potentially qualified for more pick-up parcels (i.e., as the capacity w grows, the number of completed parcels raises, leading to an incremental social welfare and processing time). As expected, the completion ratio, social welfare, and processing time of all algorithms increase gradually. It is worth noting that the PBO algorithm performs poorly when the capacity is too small (w=25) due to the heavier weight of the parcels after packing. Nevertheless, PBO algorithm still has a similar processing time with the SIDF* but greater social welfare than that of GA and SIDF*.

Effect of the Batch Size b. We evaluate the effect of the batch size in Figs. 7j, 7k, and 7l. As the increase of the batch size, more pick-up parcels are assigned in each batch, while the total number of the served pick-up parcels has not changed. Therefore, the completion ratio and social welfare of MRA, ALNS, GA, and SIDF* are almost no change at all. However, the processing times of MRA and ALNS are much higher than the other three algorithms. An interesting observation



(a) $|C|$ **vs.** CR (Cainiao) (b) $|C|$ **vs.** SW (Cainiao) (c) $|C|$ **vs.** BT (Cainiao) (d) $|\Gamma|$ **vs.** CR (Cainiao) (e) $|\Gamma|$ **vs.** SW (Cainiao) (f) $|\Gamma|$ **vs.** BT (Cainiao)

(g) w **vs.** CR (Cainiao) (h) w **vs.** SW (Cainiao) (i) w **vs.** BT (Cainiao) (j) b **vs.** CR (Cainiao) (k) b **vs.** SW (Cainiao) (l) b **vs.** BT (Cainiao)

Fig. 7. Performance of $|C|$, $|\Gamma|$, w, b, **vs. CR**, **SW**, **and BT** on Cainiao dataset.

TABLE 3
Results on Varying Packing Threshold δ

δ	10	25	50	100	200
CR(%)	94.52	91.34	88.24	78.32	66.08
SW($)	21,158	20,402	18,459	16,930	14,207
BT(ms)	302	192	150	110	72

is that the completion ratio and social welfare of PBO slightly decrease with the expansion of the batch size. This is because more parcels are packed, and the requirement for the remaining unoccupied capacity of the couriers becomes higher. Meanwhile, we observe there are more pick-up parcels to be processed when the batch size increases, leading to the increase of processing time. Overall, our proposed algorithms are still efficient for processing ACF problem.

Effect of the Packing Threshold δ. We examine the effect of δ under the default parameter settings. From Table 3, we can observe that as δ expands in PBO, the social welfare and completion ratio decrease to a certain extent. However, the processing time continues to decrease, and we trade a slight loss of social welfare for improving performance.

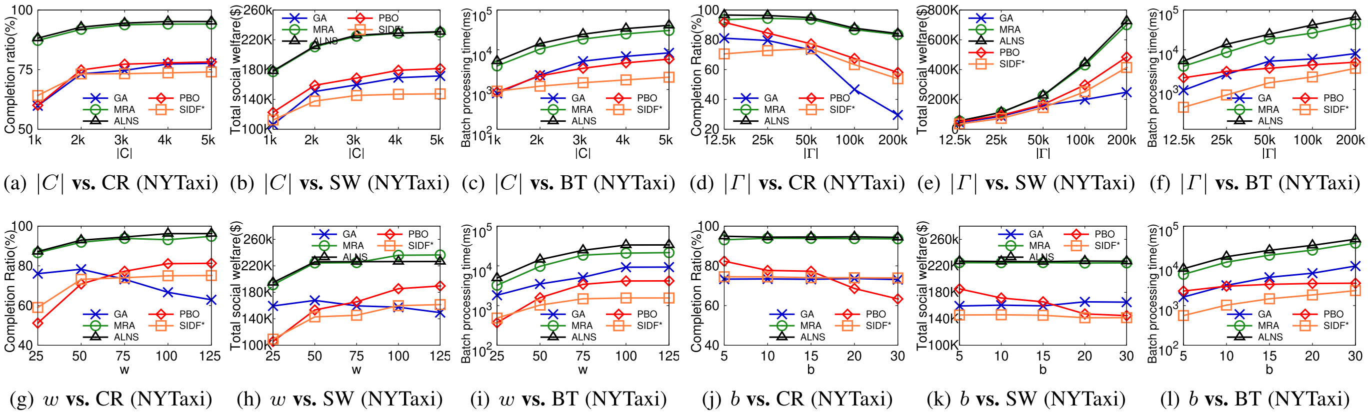
5.2.2 Evaluation on NYTaxi Dataset

Effect of the Number of Couriers $|C|$. Figs. 8a, 8b, and 8c plot the effect of the number of couriers on completion ratio, social welfare, and processing time. It can be seen that MRA performs best in completion ratio and social welfare when varying the number of couriers. However, MRA and ALNS have poor scalability on the large-scale dataset New York, and this is because when $|C|$=$3k$, the processing time of MRA (ALNS) is $18.59s$ ($24.66s$), which is larger than the default batch size $15s$. In other words, MRA and ALNS cannot complete the assignment between couriers and parcels in a batch. Compared with MRA and ALNS, PBO dramatically reduces the processing time by the packing-based optimization, and the processing time of PBO is even slightly lower than GA and is comparable with SIDF*. While in most cases, the completion ratio and social welfare of PBO are notably better than those of GA and SIDF*. SIDF* has the shortest processing time because it only finds the courier with the shortest detour distance, but this processing approach also causes its social welfare to be much lower than the other three algorithms.

Effect of the Number of Parcels $|\Gamma|$. Figs. 8d, 8e, and 8f depict the effect of varying the number of parcels. In brief, when $|\Gamma|$ increases, the completion ratio of the compared algorithms decreases, but the social welfare and processing time rise. When $|\Gamma| \leq 50k$, MRA and ALNS have the best completion ratio and social welfare comparing with GA, SIDF*, and PBO. As discussed above, however, MRA and ALNS cannot complete the assignment between couriers and pick-up parcels in a batch when $|\Gamma| > 50k$ due to the poor scalability. PBO is the runner-up in terms of completion ratio and social welfare, and the processing time of PBO is slightly less than that of GA in most cases. In addition, we can observe that SIDF* has the best processing time, but the performance in completion ratio and social welfare is rather poor. In conclusion, PBO has better scalability and achieves the best performance comparing with other algorithms.

Effect of the Maximum Capacity w. The effect of the maximum capacity is shown in Figs. 8g, 8h, and 8i. It can be seen that the completion ratio and social welfare of ALNS perform best when varying the maximum capacity. For PBO, lots of packed parcels cannot be served by the couriers due to their small capacity. Therefore, the completion ratio and social welfare of PBO are the lowest when $w = 25$. However, with the increase of the maximum capacity, the completion and social welfare of PBO are better than those of GA and SIDF*.

Effect of the Batch Size b. As shown in Fig. 8j, we report the effect of varying the batch size in the completion ratio. It can be observed that the completion ratio of MRA, ALNS, GA, and SIDF* change small when enlarging the batch size. The reason is that with the increasing of batch size, the number of parcels in each batch increases but the number of parcels of all batches remains the same. However, the completion ratio of PBO decreases when the batch size increases, because more pick-up parcels are added into a package in a large batch, this may cause some large packages not to be served by the couriers due to their capacity limitations. From Figs. 8k and 8l, we can observe that the processing time of PBO is closer to that of SIDF* when b=30, because the packing-based optimization enables many pick-up parcels to be processed at the same time, which greatly improves the processing efficiency.



(a) $|C|$ **vs.** CR (NYTaxi) (b) $|C|$ **vs.** SW (NYTaxi) (c) $|C|$ **vs.** BT (NYTaxi) (d) $|\Gamma|$ **vs.** CR (NYTaxi) (e) $|\Gamma|$ **vs.** SW (NYTaxi) (f) $|\Gamma|$ **vs.** BT (NYTaxi)

(g) w **vs.** CR (NYTaxi) (h) w **vs.** SW (NYTaxi) (i) w **vs.** BT (NYTaxi) (j) b **vs.** CR (NYTaxi) (k) b **vs.** SW (NYTaxi) (l) b **vs.** BT (NYTaxi)

Fig. 8. Performance of $|C|$, $|\Gamma|$, w, b, **vs. CR, SW, and BT** on NYTaxi dataset.

TABLE 4
Index Performance on Different Datasets

Datasets	Cainiao	NYTaxi
Memory Cost (MB)	23.36	653.46
Update Cost (MS)	3.12	13.05

Evaluation of the TIL Index. In the last set of experiments, we evaluate the performance of the TIL index in terms of memory cost and update cost under the default settings of parameters, since the TIL index plays an important role in the optimization of MRA and PBO. As shown in Table 4, we can observe that memory cost and update cost on the Cainiao dataset are much less than those on the NYTaxi, because the dataset scale of NYTaxi is larger than that of Cainiao. However, compared with the processing time of MRA and PBO, the update cost of the TIL index is much smaller and thus can be ignored.

6 Conclusion and Future Work

In this paper, we propose a novel ACF problem and prove the hardness of this problem. To efficiently and effectively solve this problem, we present a bid computation algorithm and three assignment algorithms to assign pick-up parcels to suitable couriers. We also prove that the proposed algorithms satisfy the auction properties. Extensive experiments validate that our proposed algorithms are practical to solve the ACF problem.

As for the future work, we plan to further extend our work in three aspects. First, since the threshold for parcel packing greatly affects the quality of parcel assignment, we intend to study how to adaptively determine the best packing threshold through learning based approach. Second, the distribution of the pick-up parcels in real-world scenario is uneven, ensuring that each courier can collect parcels fairly is also an interesting problem to investigate. Third, since the initial delivery route planning affects the quality of the parcels received by the courier, we plan to design the initial delivery route based on the courier's travel preference such that the couriers have more chances to collect desirable parcels.

References

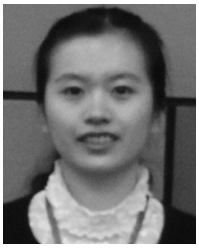
[1] Amazon, 2022. [Online]. Available: https://logistics.amazon.com/marketing
[2] M. Asghari, D. Deng, C. Shahabi, U. Demiryurek, and Y. Li, "Price-aware real-time ride-sharing at scale: An auction-based approach," in *Proc. ACM Int. Conf. Adv. Geographic Inf. Syst.*, 2016, pp. 3:1–3:10.
[3] M. Babaioff and W. E. Walsh, "Incentive-compatible, budget-balanced, yet highly efficient auctions for supply chain formation," *Decis. Support Syst.*, vol. 39, no. 1, pp. 123–149, 2005.
[4] F. M. Bergmann, S. M. Wagner, and M. Winkenbach, "Integrating first-mile pickup and last-mile delivery on shared vehicle routes for efficient urban e-commerce distribution," *Transp. Res. Part B: Methodol.*, vol. 131, pp. 26–62, 2020.
[5] Cainiao, 2021. [Online]. Available: https://www.cainiao.com
[6] I. D. Cardenas, W. Dewulf, J. Beckers, C. Smet, and T. Vanelslander, "The e-commerce parcel delivery market and the implications of home B2C deliveries vs pick-up points," *Int. J. Transport Econ.*, vol. 44, no. 2, pp. 253–256, 2017.
[7] A. Charisis, C. A. Iliopoulou, and K. L. Kepaptsoglou, "DRT route design for the first/last mile problem: Model and application to athens," *Public Transport*, vol. 10, no. 3, pp. 499–527, 2018.
[8] Y. Chen, D. Guo, M. Xu, G. Tang, T. Zhou, and B. Ren, "PPtaxi: Non-stop package delivery via multi-hop ridesharing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2684–2698, Nov. 2020.
[9] P. Cheng, H. Xin, and L. Chen, "Utility-aware ridesharing on road networks," in *Proc. ACM Int. Conf. Manage. Data*, 2017, pp. 1197–1210.
[10] Y. Hu, Y. Wang, Y. Li, and X. Tong, "An incentive mechanism in mobile crowdsourcing based on multi-attribute reverse auctions," *Sensors*, vol. 18, no. 10, 2018, Art. no. 3453.
[11] M. Huang, X. Qian, S. Fang, and X. Wang, "Winner determination for risk aversion buyers in multi-attribute reverse auction," *Omega*, vol. 59, pp. 184–200, 2016.
[12] JD, 2022. [Online]. Available: https://www.jdl.cn
[13] Y. Jiao, P. Wang, D. Niyato, and Z. Xiong, "Social welfare maximization auction in edge computing resource allocation for mobile blockchain," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.
[14] H. Jin, L. Su, D. Chen, H. Guo, K. Nahrstedt, and J. Xu, "Thanos: Incentive mechanism with quality awareness for mobile crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1951–1964, Aug. 2019.
[15] K. Chard, S. Caton, O. Rana, and K. Bubendorfer, "Social cloud: Cloud computing in social networks," in *Proc. IEEE Int. Conf. Cloud Comput.*, 2010, pp. 99–106.
[16] A. Kleiner, B. Nebel, and V. A. Ziparo, "A mechanism for dynamic ride sharing based on parallel auctions," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 266–272.
[17] Klemperer and Paul, "What really matters in auction design," *J. Econ. Perspectives*, vol. 16, pp. 169–189, 2002.
[18] V. Krishna, *Auction Theory.* Amsterdam, The Netherlands: Elsevier, 2009.
[19] I. Krontiris and A. Albers, "Monetary incentives in participatory sensing using multi-attributive auctions," *Int. J. Parallel Emergent Distrib. Syst.*, vol. 27, no. 4, pp. 317–336, 2012.
[20] J. Lee and B. Hoh, "Sell your experiences: A market mechanism based incentive for participatory sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2010, pp. 60–68.
[21] Y. Li, R. Chen, L. Chen, and J. Xu, "Towards social-aware ridesharing group query services," *IEEE Trans. Services Comput.*, vol. 10, no. 4, pp. 646–659, Jul./Aug. 2017.
[22] Y. Li et al., "Top-k vehicle matching in social ridesharing: A price-aware approach," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 1251–1263, Mar. 2021.
[23] Y. Li, Y. Zheng, and Q. Yang, "Efficient and effective express via contextual cooperative reinforcement learning," in *Proc. ACM Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 510–519.
[24] Y. Liu et al., "FooDNet: Toward an optimized food delivery network based on spatial crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1288–1301, Jun. 2019.
[25] B. Ma, Z. Zhou, M. F. Bashir, and Y. Huang, "A multi-attribute reverse auction model on margin bidding," *Asia Pacific J. Oper. Res.*, vol. 37, no. 6, pp. 2 050 032:1–2 050 032:21, 2020.
[26] L. Ma, X. Wang, X. Wang, L. Wang, Y. Shi, and M. Huang, "TCDA: Truthful combinatorial double auctions for mobile edge computing in industrial Internet of Things," *IEEE Trans. Mobile Comput.*, vol. 21, no. 11, pp. 4125–4138, Nov. 2022.
[27] S. Ma, Y. Zheng, and O. Wolfson, "T-share: A large-scale dynamic taxi ridesharing service," in *Proc. IEEE Int. Conf. Data Eng.*, 2013, pp. 410–421.
[28] E. Macioszek, "First and last mile delivery – problems and issues," in *Proc. Sci. Tech. Conf. Transport Sys. Theory Pract.*, 2017, pp. 147–154.
[29] A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani, "Adwords and generalized on-line matching," in *Proc. IEEE Symp. Found. Comput. Sci.*, 2005, pp. 264–273.
[30] T. Na, G. Li, T. Zhao, J. Feng, H. Ma, and Z. Gong, "An efficient ride-sharing framework for maximizing shared route," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 2, pp. 219–233, Feb. 2018.
[31] NYTaxi, 2021. [Online]. Available: http://www.nyc.gov
[32] S. Ruan et al., "Filling delivery time automatically based on couriers' trajectories," *IEEE Trans. Knowl. Data Eng.*, early access, Jul. 27, 2021, doi: 10.1109/TKDE.2021.3100116.
[33] S. Ruan et al., "Doing in one go: Delivery time inference based on couriers' trajectories," in *Proc. ACM Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 2813–2821.
[34] W. I. L. M. D. C. H. to Optimize, 2022. [Online]. Available: https://optimoroute.com/last-mile-delivery

[35] T. Vanelslander, L. Deketele, and D. V. Hove, "Commonly used e-commerce supply chains for fast moving consumer goods: Comparison and suggestions for improvement," *Int. J. Logistics Res. Appl.*, vol. 16, no. 3, pp. 243–256, 2013.
[36] N. Wang and T. A. Gulliver, "Distributed queue-aware relay node selection for cooperative wireless networks via Vickrey auction game," *IEEE Wireless Commun. Lett.*, vol. 4, no. 3, pp. 257–260, Jun. 2015.
[37] Y. Wang, Z. Cai, Z.-H. Zhan, Y.-J. Gong, and X. Tong, "An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 3, pp. 414–429, Jun. 2019.
[38] Y. Wang, D. Zhang, Q. Liu, F. Shen, and L. Lee, "Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions," *Transp. Res. Part E: Logistics Transp. Rev.*, vol. 93, pp. 279–293, 2016.
[39] A. Watanabe, F. Ishikawa, Y. Fukazawa, and S. Honiden, "Web service selection algorithm using Vickrey auction," in *Proc. IEEE Int. Conf. Web Serv.*, 2012, pp. 336–342.
[40] Y. Wei, Y. Zhu, H. Zhu, Q. Zhang, and G. Xue, "Truthful online double auctions for dynamic mobile crowdsourcing," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2074–2082.
[41] M. Xiao et al., "SRA: Secure reverse auction for task assignment in spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 782–796, Apr. 2020.
[42] Y. Zeng, Y. Tong, and L. Chen, "Last-mile delivery made practical: An efficient route planning framework with theoretical guarantees," *Proc. VLDB Endowment*, vol. 13, no. 3, pp. 320–333, 2019.
[43] S. Zhang, L. Qin, Y. Zheng, and H. Cheng, "Effective and efficient: Large-scale dynamic city express," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3203–3217, Dec. 2016.
[44] Y. Zhang, Q. Yang, W. Yu, D. An, D. Li, and W. Zhao, "An online continuous progressive second price auction for electric vehicle charging," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 2907–2921, Apr. 2019.
[45] H. Zhao and W. Chen, "Online second price auction with semi-bandit feedback under the non-stationary setting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 6893–6900. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/6171
[46] L. Zheng, L. Chen, and J. Ye, "Order dispatch in price-aware ridesharing," *Proc. VLDB Endowment*, vol. 11, no. 8, pp. 853–865, 2018.
[47] L. Zheng, P. Cheng, and L. Chen, "Auction-based order dispatch and pricing in ridesharing," in *Proc. IEEE Int. Conf. Data Eng.*, 2019, pp. 1034–1045.

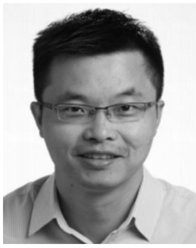
Yafei Li received the PhD degree in computer science from the Hong Kong Baptist University, in 2015. He is currently a professor with the School of Computer and Artificial Intelligence, Zhengzhou University. His research interests span mobile and spatial data management, location-based services, and urban computing. He has authored more than 20 journal and conference papers in these areas, including the *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Services Computing*, *ACM Transactions on the Web*, *ACM Transactions on Intelligent Systems and Technology*, PVLDB, IEEE ICDE, WWW, etc.

Yifei Li received the BEng degree in computer science and technology from Zhengzhou University, in 2019. He is currently working toward the MEng degree with the School of Computer and Artificial Intelligence, Zhengzhou University. His research interests include multi-agent computing, deep learning, and spatio-temporal data processing.

Yun Peng received the BSc degree in computer science from Shandong University, in 2006, the MPhil degree in computer science from the Harbin Institute of Technology, in 2008, and the PhD degree in computer science from the Hong Kong Baptist University, in 2013. His research interests include graph data processing, data outsourcing, view update, and incremental maintenance.



Xiaoyi Fu received the BEng degree in computer science and engineering from Zhejiang University, and the PhD degree in computer science from the Hong Kong Baptist University, in 2020. She currently works in the fields of databases, location-based services, geo-social computing, and geo-spatial recommendation.



Jianliang Xu received the BEng degree in computer science and engineering from Zhejiang University, Hangzhou, China, and the PhD degree in computer science from the Hong Kong University of Science and Technology. He is a professor with the Department of Computer Science, Hong Kong Baptist University. He held visiting positions with Pennsylvania State University and Fudan University. His research interests include Big Data management, mobile computing, and data security and privacy. He has published more than 200 technical papers in these areas. He has served as a program cochair/vice chair for a number of major international conferences including IEEE ICDCS 2012, IEEE CPSNA 2015, and APWeb-WAIM 2018. He is an associate editor of the *IEEE Transactions on Knowledge and Data Engineering* and the *Proceedings of the VLDB Endowment* 2018.

Mingliang Xu received the PhD degree from the State Key Lab of CAD&CG, Zhejiang University, China. He is a professor with the School of Computer and Artificial Intelligence, Zhengzhou University, China. His current research interests include computer graphics, multimedia, and artificial intelligence. He has authored more than 60 journal and conference papers in these areas, including the *ACM Transactions on Graphics*, *IEEE Transactions on Pattern Analysis and Machine Intelligence/IEEE Transactions on Image Processing/IEEE Transactions on Circuits and Systems for Video Technology*, ACM SIGGRAPH (Asia)/MM, ICCV, etc.