https://doi.org/10.1016/j.ijpe.2021.108067

Crowdsourcing mode evaluation for parcel delivery service platforms

Lu Zhen[a], Yiwei Wu[b], Shuaian Wang[b,*], Wen Yi[c]

[a] *School of Management, Shanghai University, Shanghai, China*
[b] *Department of Logistics & Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China*
[c] *School of Built Environment, College of Sciences, Massey University, Auckland, New Zealand*

Abstract

The fast-growing practice of e-commerce implies a strong increase in the urban parcel delivery, which in turn creates significant pressure on last-mile city logistics. Because the crowdsourced delivery offers greater flexibility and requires less capital investment than traditional delivery methods, it has been playing a more crucial role when faced with the growing demand for urban parcel delivery. With the increasing maturity of the crowdsourced delivery and the fierce competition among platforms, the evaluation of different crowdsourcing modes for the urban parcel delivery becomes important. This study proposes six mathematical models to evaluate different operation modes of the crowdsourced delivery in a quantitative way. Several realistic factors, such as the latest service time for each task, task cancellation rate and range distribution of tasks, are also analyzed in this paper. Numerical experiments are conducted to validate the effectiveness of the proposed models and to analyze the impact of different modes. Some managerial implications are also outlined on the

*Corresponding author. wangshuaian@gmail.com (S. Wang)

basis of the numerical experiments and sensitivity analysis to help crowdsourced companies to make scientific decisions.
Keywords: Crowdsourced delivery; crowdsourcing service platform; e-commerce; parcel delivery.

1. Introduction

Because of the improvement in a living standard and the convenience of online shopping, more and more customers prefer online shopping. In 2018, retail e-commerce sales of physical goods in the United States amounted to 501 billion USD and are projected to surpass 740 billion USD in 2023 (StatistaUS, 2019). E-commerce revenue in China is also expected to grow to 1,095.5 billion USD in 2023 (StatistaChina, 2019). The explosive growth of e-commerce and the ever-increasing desire for faster service lead to the huge demand for package delivery services, which urges the rapid development of logistics industry (Masson et al., 2017; Cheng et al., 2019). Recently, the evolution of a sharing economy and advances in communication and mobile device technologies give rise to new opportunities to improve distributing efficiency. One of those innovative opportunities is crowdsourced delivery (Devari et al., 2017).

Crowdsourced delivery entails the use of an available spare load capacity of private passenger vehicles on journeys that already takes place in order to support urban parcel delivery operations (Arslan et al., 2018). By using existing traffic flows, crowdsourced delivery could enable cheaper urban parcel delivery. Besides, it may help to reduce the negative consequences of the use of dedicated delivery vehicles on the urban environment, such as traffic emission reduction, and traffic congestion alleviation (Ren et al., 2019). A growing popularity of crowdsourced delivery can be seen in the real-world over the past few years. In 2018, Walmart tested a new last-mile grocery crowdsourced delivery platform, called Spark Delivery (Walmart, 2018). Amazon rolled out a crowdsourced service, called Amazon Flex, in 79 cities in the United States (Amazon, 2019). Over 50 start-ups materialized in IT platforms (websites and/or mobile applications) around the world in 2015 have been categorized as crowdsourced delivery service providers and engaging in local delivery, freight shipping, and freight forwarding activities (Carbone et al., 2017). These crowdsourced delivery platforms continue to gain greater popularity from the academia and industry (Huang and Ardiansyah, 2019).

Many scholars and industry managers have proposed many new operation modes for the rapid development of crowdsourced delivery platforms. Some crowdsourced delivery platforms that are in the beginning stage usually adopt the 'grabbing mode' which means an ad hoc driver will grab one delivery order when he (or she) thinks the order's destination is convenient. Before 2015, the order allocation model of Didi fast ride service (Didi is a leading mobile transportation company offering a full range of app-based transportation service for 550 million users across Asia (Didi, 2020b)) was the 'grabbing mode' (Didi, 2020c). In this situation, it is likely that the delivery order will be assigned to the first driver who grabs the order rather than the one who brings the maximal profit to the company, resulting in a mismatch between the ad hoc driver and the customer. For example, suppose that the crowdsourced delivery platform sends two tasks (e.g., task *1* and task *2*) simultaneously to two drivers (e.g., driver k and driver m). For driver k, two tasks are equally convenient to be undertook. But for driver m, task *1* is more convenient to be undertook than task *2*. In the 'grabbing mode', driver k grabs task *1* first, and at this point, driver m can only grab task *2*. But in the 'assignment mode', the crowdsourced delivery platform will assign task *1* to driver m and task *2* to driver k in the consideration of platform profit maximization. After converting to the ‘assignment mode’, these crowdsourced delivery platforms usually assign one delivery task to each driver once. However, if two tasks could be connected conveniently by a driver and the connected trip may also match the driver’s original journey very well, it is reasonable that the platform should allow a driver to take two tasks. In real life, some ‘unfavorable’ orders are often not picked up by any ad hoc driver, which could lead to decreases in customer satisfaction rate, task fulfillment rate, and the final profit of the company. Hence, some platforms want to adopt the bonus system that pays some extra bonus to incentivize some drivers to undertake these ‘unfavorable’ tasks. For example, Didi began offering ‘Spring Festival service fee’ to drivers in 2019 to encourage them to take orders during the holiday (CCTV, 2019). Moreover, a customer may cancel his (or her) task when waiting for a bit long time. A customer’s order cancellation may cause a ‘loss’ to the ad hoc driver, who has already headed to the pickup locations of the customer. Some crowdsourcing platforms may compensate these ad hoc drivers to attract them to continue using these platforms.

It is difficult to know the actual impact of different operation modes on crowdsourcing platforms’ profits. Hence, this study is motivated by these real-world problems encountered in the development of crowdsourced delivery platforms. Crowdsourced delivery makes use of idle resources to perform the delivery tasks that would otherwise have to be performed by the delivery companies in a traditional way. This concept continues to gain greater popularity due to the benefits associated with an increase in vehicle utilization, a decrease in vehicle mileage, and a cost saving especially for urban parcel delivery. These benefits also have great potential to reduce the economic, environmental, and social negative impacts caused by the booming e-commerce business. Hence, studying the concept of crowdsourced delivery has become more crucial for the academia and industry. However, few related studies provide scientific methods to analyze impacts of different operation modes, including 'grabbing mode', 'assignment mode', 'two tasks assignment mode', 'bonus mode', 'task cancellation mode', and 'mixed bonus-cancellation mode'. Since the above mentioned crowdsourced delivery issues are vital, this paper formulates six mathematical models to evaluate the six operation modes of crowdsourced delivery in a quantitative way to help crowdsourced companies to make scientific decisions.

Because our results indicate the commercial value of 'assignment mode' in terms of company profits, we recommend that crowdsourced companies use the 'assignment mode' rather than the 'grabbing mode'. Moreover, allowing a driver to undertake multiple tasks can bring a significant benefit to crowdsourced companies. We also find that the advantage of bonus system is comparatively small. Besides, we suggest that the company should adopt the 'mixed bonus-cancellation' operation strategy to attract more customers. We also consider several realistic factors, such as the latest service time for each task, task cancellation rate and range distribution of tasks, to make our models fit the realistic needs of the crowdsourced delivery. We summarize some managerial implications according to our sensitivity analysis. For instance, the increase in the latest service time has direct effect on the task assignment fulfillment. And when the latest service time is set to three minutes after the platform receives the task, the task assignment fulfillment rate is up to 68%. We also find less parcel tasks can be assigned successfully when task cancellation rate increases. In the end, given a collection delivery tasks, we also explored the minimum number of drivers needed to serve all the tasks. Some computational experiments are also conducted to investigate the performance of the ‘minimum fleet’ strategy by comparing it with the ‘maximum profit’ strategy. It is noted that the numbers of ad hoc drivers needed under ‘maximum profit’ strategy and ‘minimum fleet’ strategy are almost the same, but the profit under ‘minimum fleet’ strategy is significantly less than that under ‘maximum profit’ strategy in most cases. Besides, a larger service radius has a direct impact on the crowdsourced delivery platforms’ profit growth. However, this does not mean that constantly expanding the service radius will always lead to a profit growth. The quantitative methodology provided in this study can also be used for platform analyses of specific crowdsourced delivery patterns, not only for e-commerce platforms. We also take Walmart (Shanghai) as an example to verify the effectiveness of this methodology.

The remainder of the study is organized as follows. An overview of the related works is introduced in Section 2. Section 3 proposes two mathematical models to compare the ‘grabbing mode’ and the ‘assignment mode’. Section 4 evaluates the ‘two tasks assignment mode’. Section 5 investigates the advantage of the ‘bonus mode’. Section 6 evaluates the cost brought by ‘task cancellation mode’ to crowdsourcing platforms. Section 7 evaluates the advantage of the ‘mixed bonus-cancellation mode’. Section 8 reports the computational experiments with real-world data. Conclusions are then outlined in the last section.

2. Literature review and discussion

This study proposes a quantitative decision methodology to focus on the operation mode analysis of the crowdsourced delivery. The core part of the crowdsourced delivery is related to the pickup and delivery problem. Because the crowdsourcing mode is an emerging problem, many existing studies are qualitative studies and few studies provide a quantitative decision methodology for this important problem. Hence, the following three paragraphs review three streams of related literature: the pickup and delivery problem, the qualitative analysis of the crowdsourced delivery and the quantitative analysis of the crowdsourced delivery.

The first research stream of related works is the pickup and delivery problem. The pickup and delivery problem has been intensively studied over recent decades. It aims for a minimum cost route to distribute resources among nodes, including pickup nodes supplying resources and delivery nodes requiring resources (Ting and Liao, 2013). Cortés et al. (2010) proposed a generalization of the classical pickup and delivery problem. Baldacci et al. (2011) proposed a new exact algorithm for the pickup and delivery problem with time windows based on a set-partitioning-like integer formulation. Masson et al. (2013) presented an adaptive large neighborhood search method for the pickup and delivery problem with transfers. Karaoglan et al. (2012) proposed two polynomial-size mixed integer programming (MIP) models and a family of valid inequalities to strengthen models for the location routing problem with simultaneous pickup and delivery. Polat et al. (2015) proposed a perturbation based variable neighborhood search heuristic for solving the vehicle routing problem (VRP) considering the simultaneous pickup and delivery with time limits. Gschwind et al. (2018) examined a full-fledged branch-cut-and-price algorithm on the pickup and delivery problem with time windows.

Pickup and delivery problems reviewed in the first stream are for vocational drivers driving dedicated delivery vehicles. These vocational drivers don't need to consider their path planning before and after taking orders. Recently, the evolution of sharing economy and advances in communication and mobile device technologies give rise to new opportunities to improve distributing efficiency. One of those innovative opportunities is the crowdsourced delivery undertook by ad hoc drivers. Ad hoc drivers do need to consider their path planning before and after taking orders because they only want to undertake the task whose destination is convenient for them. Hence, the second research stream is related to the qualitative analysis of the crowdsourced delivery. There is an increasing amount of research studying the use of crowdsourcing to conduct the parcel delivery (Sadilek et al., 2013). Rougès and Montreuil (2014) examined 18 startups in the crowdsourcing industry based on available public documentation and proposed a paradigm change in this nascent industry. Kunze (2016) provided an overview of different existing and emerging transport logistic operations, and proposed a partial qualitative systemic model that shows how these different operations are influenced by global and logistics trends, and delivery service requirements. Cheah and Wang (2017) applied a deductive reasoning and case analysis method on manufacturing firms in China to validate the crowdsourcing mechanisms. Pan et al. (2015) proposed a crowdsourcing solution to collect e-commerce reverse flows in metropolitan areas by conducting qualitative and quantitative studies. In order to evaluate the nature and characteristics of the crowdsourcing mode, Frehe et al. (2017) provided initial insights into social changes in terms of drivers for the use of crowdsourced delivery services. Rai et al. (2017) systematically analyzed 42 papers, interviewed 11 logistics practitioners, identified a set of 18 characteristics that describe the variety of crowdsourcing, and found that 11 characteristics affect economic, social and/or environmental sustainability.

The last stream of studies explores the the quantitative analysis of the crowdsourced delivery. Devari et al. (2017) demonstrated the potential benefits of crowdsourcing the last mile delivery by exploiting a social network of customers. Setzke et al. (2017) illustrated an algorithm that automates and optimizes the assignment of drivers to transportation requests by matching them based on transportation routes and time constraints. Arslan et al. (2018) considered a service platform that automatically creates matches between parcel delivery tasks and ad hoc drivers, proposed a rolling horizon framework, and developed an exact solution approach to solve a dynamic crowdsourced delivery problem. Cheng et al. (2019) used the integer linear programming techniques to solve the city-wide package distribution problem using crowdsourced delivery and then proposed an efficient heuristic solution for this NP-hard problem. Huang and Ardiansyah (2019) formulated an MIP model to deal with the planning of the last-mile delivery with partial crowdsourcing integration. Li et al. (2020) demonstrated that an assignment matching strategy in a freight online to offline platform can help to assign orders effectively and efficiently to drivers and optimize the matching process in terms of the platform's profits. Yuan et al. (2020) built moderated mediation models to invest the impact of transaction attributes on crowdsourcing success. Some vehicle routing problems, such as the open VRP, are similar to the crowdsourced delivery. Like ad hoc drivers in crowdsourcing platforms who can go to their destinations after delivery, drivers in open VRP do not necessarily return to the initial depot after delivering parcels to the last customer. The newspaper distribution problem is modeled as an open VRP with time windows and zoning constraints (Chiang et al., 2009; Russell, 2013). Yu et al. (2016) created an open VRP to introduce a general example in retail wherein the capital expenditure necessary in the vehicle acquisition can become a burden for the retailer, who then needs to consider outsourcing its logistics service as a cost-effective option. Hosseinabadi et al. (2018) optimized the number of vehicles, the traveling distance and the traveling time of a vehicle to solve an open VRP, and developed a new combinatorial algorithm based on the gravitational emulation local search algorithm. Wang et al. (2018) considered a notion of stability for ride-share matches, and presented several mathematical programming methods to establish stable or nearly stable matches. Coindreau et al. (2019) considered a situation where workers can either walk or drive to work and where carpooling is enabled. In order to quantify the potential benefits offered by this new framework, a dedicated variable neighborhood search method is proposed to efficiently tackle the underlying synchronization and precedence constraints that arise in this extension of the VRP.

In summary, the majority of the existing studies on the crowdsourced delivery (Agatz et al., 2011; Archetti et al., 2016; Arslan et al., 2018; Huang and Ardiansyah, 2019) focus on creating matches between parcel delivery tasks and ad hoc drivers. Few studies consider the operation strategies of crowdsourced delivery service companies, such as different transport modes (Dupljanin et al., 2019) and the privacy preservation strategy (Tang et al., 2019), but they do not analyze or model different operation modes in a quantitative way. However, it is essential to evaluate operation modes for a crowdsourced delivery service company in the context of the spectacular growth of online sales. Therefore, this paper studies the concept of the crowdsourced delivery that aims to use the excess capacity on journeys that have already taken place. More specifically, we formulate six mathematical models to evaluate the 'grabbing mode', the 'assignment mode', the 'two tasks assignment mode', the 'bonus mode', the 'task cancellation mode', and the 'mixed bonus-cancellation mode'. Moreover, some other operating limits, such as the latest service time for each task, task cancellation rate and range distribution of tasks, have also been frequently ignored, even though these factors are crucial to the real-world delivery activities.

The proposed mathematical models are used to compare the effect of six operation modes on a crowdsourced delivery service company, and to analyze several realistic factors. These features make this paper significantly different from previous studies.

3. Evaluating the advantage of an assignment mode

Crowdsourced delivery entails the use of the excess capacity of private passenger vehicles on journeys that have already taken place to deliver parcels. Using existing traffic flows could potentially enable faster and cheaper deliveries (Arslan et al., 2018). In the beginning stage of the crowdsourced delivery industry, platforms usually adopt the mode of 'grabbing orders'. An order of the delivery service (named task in this study) is released to a set of available ad hoc drivers, whose locations are inside a circle with a certain radius (e.g., 3 km) and centered at the pickup position of the task; then these drivers apply to grab the task on their APPs if they think the task's destination is convenient for them. Among the drivers who apply to take (grab) the order, one of them will be chosen by the platform randomly or according to some implicit rules.

When the crowdsourced delivery industry becomes mature, the fierce competition among platforms forces them to think about improving performance (and their profits) through more precise matches between ad hoc drivers and delivery tasks rather than using the random-featured mode that drivers grab tasks. For example, in 2015, the order allocation model of Didi fast ride service (Didi is a leading mobile transportation company offering a full range of app-based transportation service for 550 million users across Asia (Didi, 2020b)) evolved from the 'grabbing mode' to the 'assignment mode' (Didi, 2020c). Compared with those under the 'grabbing mode', the hourly wages of ad hoc drivers who participated in the test increased by up to 50% and the empty driving rate decreased by up to 36% under the 'assignment mode' (Didi, 2020c). In addition, the response rate of customers was improved by more than 20%, thereby the 'assignment mode' has created great user value for Didi (Didi, 2020a). Hence, we try to compare the best case of the 'assignment model' and the worst case of the 'grabbing model' to quantify the advantage of the 'assignment model'.

The precise match should borrow supports from some advanced algorithms (mathematical models) as well as more well calibrated parameters on the basis of 'big data' related to drivers and customers who release tasks. For a platform's decision maker, the objective of the precise match could be to maximize the platform's profit. In this new mode of 'precise match', each task is 'optimally' assigned to one suitable driver.

When comparing the 'optimization-featured' mode that tasks are assigned to drivers and the 'random-featured' mode that drivers grab tasks, the first question is how to evaluate (quantify) the advantage of the 'optimization-featured' mode over the 'random-featured' mode. In order to investigate the above question, the key issue is to model the two modes in a quantitative way.

3.1. Investigating drivers' willingness to accept (grab) tasks

Before formulating models, it is necessary to investigate ad hoc drivers' willingness to accept or grab delivery tasks released on the platform. Suppose a crowdsourced delivery service company operates a delivery platform containing a set K of ad hoc drivers (indexed by k), who have registered on the platform and are available at the decision time point. Suppose the platform has a set P of delivery tasks (indexed by p), which are released by customers and need to be dispatched to the drivers at the decision time point.

As shown in Figure 1, the travel distance of task p is denoted by l_p. For each ad hoc driver k, his (or her) personal time cost (USD/h), the original trip's travel distance, the distance between the origins of the driver and the task p, and the distance between the destinations of the task p and the driver are defined as c_k, o_k, $d_{kp}^{\rightarrow}$, and $d_{kp}^{\leftarrow}$, respectively. Then if the driver k undertakes the task p, he (or she) will travel Δ_{kp} extra distance, where $\Delta_{kp} = d_{kp}^{\rightarrow} + l_p + d_{kp}^{\leftarrow} - o_k$.

In this paper, crowdsourced delivery platforms are used to undertake food

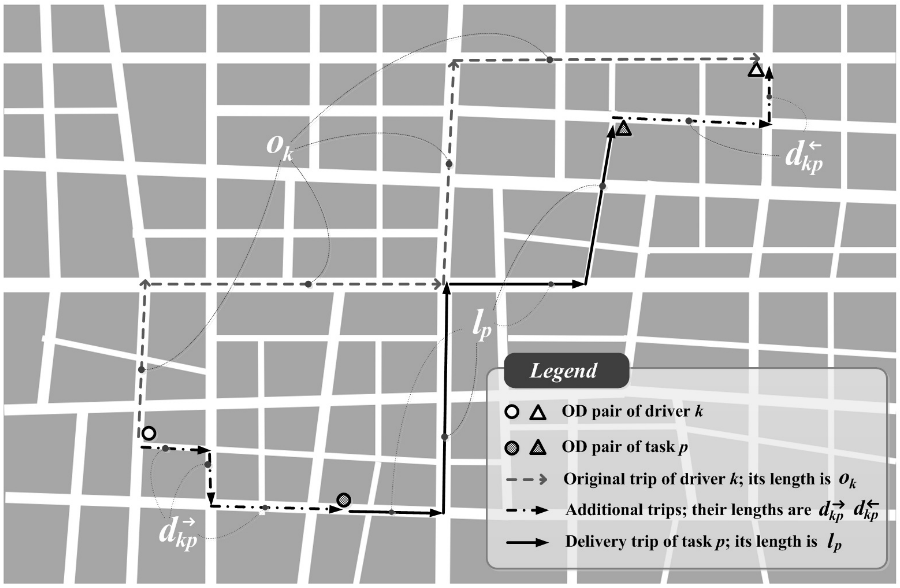


Figure 1: The traveling trip of a driver undertaking one task

or supermarket orders, or car-hailing platform orders. According to the policies of Meituan-Dianping Takeout, a takeout company that accounted for the most takeout transactions in China in the first half of 2018 (Meituan-Dianping, 2020b), order delivery fee is often affected by distance, category and other aspects (Meituan-Dianping, 2020a). Hence, this study assumes that the revenue of ad hoc drivers is increasing with the distance. The above defined extra distance has influence on the drivers' willingness for accepting the tasks on the platform. More specifically, the driver k's willingness to undertake the task p is denoted by w_{kp}, which is defined as the ratio of revenue of undertaking task p to time cost of driver k, $w_{kp} = \frac{l_p e}{c_k(\Delta_{kp}/v)} = \frac{l_p e v}{c_k \Delta_{kp}}$; here v and e are vehicle speed (km/h) and unit revenue per kilometer (USD/km) obtained by drivers, respectively. This study assumes: if $w_{kp} \geq 1$, driver k is willing to undertake task p; otherwise the task p should not be assigned to the driver k.

In reality, many ad hoc drivers on platforms gradually turn to 'full time' delivery drivers because they may be unemployed before or they can earn more than their previous jobs. These drivers actually do not have the above mentioned 'original trips'. Drivers travel from their corresponding location to the origin of the parcel to pick up the goods, and then to the destination of each parcel. In this case, extra travel distance of the driver k undertaking the task p is $\Delta_{kp} = d_{\vec{kp}} + l_p$, and their willingness values are calculated by $w_{kp} = \frac{l_p e}{c_k(\Delta_{kp}/v)} = \frac{l_p e v}{c_k \Delta_{kp}}$.

3.2. Basic models for matching drivers and tasks

Based on the above analysis, we formulate a mathematical model. We make the following assumptions:

(I) An ad hoc driver can be assigned at most one delivery task (Archetti et al., 2016).

(II) The vehicle of an ad hoc driver has sufficient capacity to accommodate the demand from customers (Archetti et al., 2016).

(III) An ad hoc driver only declares his (or her) willingness to make a delivery after delivery tasks are released to a crowdsourced delivery service company, which is reasonable because only after knowing the information of delivery tasks, including the origin and the destination, can an ad hoc driver evaluate whether the task's destination is convenient for him (or her) and declare the willingness to make this delivery.

(IV) Any ad hoc driver can be assigned with a delivery task when he (or she) is willing to undertake the task, which is based on the assumption allowing all vehicles to serve all tasks (Li and Lim, 2003; Ropke and Pisinger, 2006).

(V) Each customer has a unique location and corresponds to a single pickup and delivery demand (Kafle et al., 2017).

(VI) Ad hoc drivers are not expected to stop in route (Kafle et al., 2017).

(VII) Ad hoc drivers always start from the origin and end at the destination (Kafle et al., 2017).

(VIII) All delivery tasks are released to a set of available ad hoc drivers at the same time (Wang and Saksena, 1999).

Before formulating the mathematical model for this problem, we list the notation used in this paper as follows.

Indices and sets

K	set of ad hoc drivers, index $k, k = 1, 2, \cdots, \|K\|$.
P	set of delivery tasks, index $p, p = 1, 2, \cdots, \|P\|$.

Parameters

m_p	profit for the company if task p is fulfilled.
w_{kp}	willingness of driver k to undertake task p. If $w_{kp} \geq 1$, driver k is willing to undertake task p. Its definition is elaborated in previous section.

Variables

φ_{kp}	binary, equal to one if delivery task p is assigned to (or is grabbed by) driver k; otherwise zero.

Mathematical model

One model is formulated for each of the two modes. For the mode that tasks are assigned to one ad hoc driver by the platform (denoted by 'Asg' for short), a model

(M_1^{Asg}) is formulated for calculating the final profit earned by the platform. While, for the mode that drivers grab tasks (denoted by '*Grb*' for short), a model (M_1^{Grb}) is formulated for calculating the 'worst-case' profit earned by the platform. It is obvious that the 'best-case' profit for the mode '*Grb*' is the same as the profit for the mode '*Asg*'. Thus this study tries to investigate the relative advantage of the mode '*Asg*' to the mode '*Grb*' by calculating the gap between the best and worst cases.

$$[\boldsymbol{M_1^{Asg}}] \quad Z_1^{Asg} = \text{Maximize} \sum_{p \in P} m_p \sum_{k \in K} \varphi_{kp} \tag{1}$$

subject to:

$$\sum_{k \in K} \varphi_{kp} \leq 1 \quad \forall p \in P \tag{2}$$

$$\sum_{p \in P} \varphi_{kp} \leq 1 \quad \forall k \in K \tag{3}$$

$$\varphi_{kp} \leq \begin{cases} 1, & w_{kp} \geq 1 \\ 0, & w_{kp} < 1 \end{cases} \quad \forall k \in K, p \in P \tag{4}$$

$$\varphi_{kp} \in \{0, 1\} \quad k \in K, p \in P. \tag{5}$$

Objective (1) maximizes the total profit for the company that operates the crowdsourced delivery platform. Constraints (2) ensure that each task is fulfilled at most once. This problem allows some tasks are not assigned. Constraints (3) guarantee that each driver is assigned at most one task. Constraints (4) ensure a driver is assigned with a task which the driver is willing to undertake. Constraints (5) define the domain of decision variables.

Proposition 1. *This problem belongs to the assignment problem. The coefficient matrix of model* $[M_1^{Asg}]$ *is the totally unimodular.*

Proof. Let A be the coefficient matrix of Constraints (2) and (3). A^T is a $(|K|+|P|) \times |KP|$ matrix. Each element of A^T is either 0 or 1 and each column contains two 1. If we divide A^T into two matrices: the top $|P|$ rows, corresponding to Constraints (2), constitute one matrix and the bottom $|K|$ rows, corresponding to Constraints (3), constitute the other matrix, then each matrix has exactly one element of 1 in each column. Hence, A^T is totally unimodular and thereby A is totally unimodular. □

As Schrijver (1986) shows, if the coefficient matrix A of integer programming model is totally unimodular, this integer programming model has the totally unimodular property. Since the right-hand side coefficients of (2)–(4) are all integers, all the extreme point optimal solutions to model $[M_1^{Asg}]$ are integers. Hence, the integrality constraint in (5) can be dropped. In other words, model $[M_1^{Asg}]$ can be easily solved as a linear programming problem. We convert the above integer programming model $[M_1^{Asg}]$ to the linear programming relaxation of model $[M_1^{Asg'}]$.

Mathematical model

$[M_1^{Asg'}]$ Objective (1)

subject to: Constraints (2)–(4)

$$0 \leq \varphi_{kp} \leq 1 \quad k \in K, p \in P. \tag{6}$$

The integer programming model for 'grabbing mode' is summarized as follows:

$$[\boldsymbol{M_1^{Grb}}] \quad Z_1^{Grb} = \text{Minimize} \sum_{p \in P} m_p \sum_{k \in K} \varphi_{kp} \tag{7}$$

subject to: Constraints (2)–(5)

$$(1 - \sum_{p' \in P} \varphi_{kp'})(1 - \sum_{k' \in K} \varphi_{k'p}) \leq \begin{cases} 0, & w_{kp} \geq 1 \\ 1, & w_{kp} < 1 \end{cases} \qquad \forall k \in K, p \in P. \tag{8}$$

Constraints (8) guarantee that if driver k is willing to grab task p, at least one of the following two conditions holds: (1) driver k grabs one task (task p or another task); (2) task p is grabbed by one driver (driver k or another driver). Without Constraints (8), Objective (7) for the worst case of the 'grabbing mode' is zero, which is meaningless and disobeys the reality. Constraints (8) contain a nonlinear part '$(1 - \sum_{p' \in P} \varphi_{kp'})(1 - \sum_{k' \in K} \varphi_{k'p})$', which is the product of two binary variables. In order to linearize it, we convert (8) as follows:

$$1 - \sum_{p' \in P \backslash \{p\}} \varphi_{kp'} - \sum_{k' \in K \backslash \{k\}} \varphi_{k'p} - \varphi_{k'p} \leq 0 \quad \forall k \in K, p \in P, w_{kp} \geq 1. \tag{9}$$

Hence, the transformed model $[M_1^{Grb'}]$ is summarized below.

Mathematical model

$[M_1^{Grb'}]$ Objective: (7)

subject to: Constraints (2)–(5), (9).

Proposition 2. *The linear programming relaxation of model* $[M_1^{Grb'}]$ *does not always* *have an integer optimal solution.*

Proof. To prove the proposition, we just need to provide an example. Consider a simple case with two tasks (i.e., 1 and 2) and two ad hoc drivers (a and b). Suppose that each driver can grab any task, and the profit for the company if any task is fulfilled by any driver is 1. In this case, the minimal profit for the company is 2. However, if we solve the linear programming relaxation of model $[M_1^{Grb'}]$, we can find that the optimal solution is $\varphi_{a1} = \varphi_{a2} = \varphi_{b1} = \varphi_{b2} = \frac{1}{3}$, and the objective value is $\frac{4}{3}$. Therefore, the linear programming relaxation of model $[M_1^{Grb'}]$ does not always have an integer optimal solution. □

Then based on the objective values of the above two models, we quantify the advantage of the mode '*Asg*' to the mode '*Grb*' by calculating the following gap value.

$$Val_1^{A/G} = Z_1^{Asg} - Z_1^{Grb}. \tag{10}$$

4. Evaluating the advantage of a two tasks assignment mode

In the crowdsourced delivery industry, platforms usually assign one delivery task to each driver once. However, if two tasks could be connected conveniently by a driver and the connected trip may also match the driver's original journey very well, there is reasonability that the platform should allow a driver to take two tasks. This section investigates the potential advantage of the operation strategy that a driver can take two tasks.

4.1. Investigating a driver's willingness to accept two tasks

Section 3.1 analyzes the willingness of a driver to accept one task. Here we extend it to the case that a driver may take two tasks. Suppose an ad hoc driver fulfills two tasks (e.g., p and q) sequentially. As shown in Figure 2, we define b_{pq} as the distance between task p's destination and task q's origin. Then the extra distance Δ_{kpq} for the driver k undertaking tasks p and q sequentially is calculated by: $\Delta_{kpq} = d_{kp}^{\rightarrow} + l_p + b_{pq} + l_q + d_{kq}^{\leftarrow} - o_k$. Here, for tasks p and q, their travel distances are denoted by l_p and l_q, respectively; for each ad hoc driver k, his (or her) personal time cost (USD/h), the original trip's travel distance, the distance between the origins of the driver and the task p, and the distance between the destinations of the driver and the task q are defined as c_k, o_k, $d_{kp}^{\rightarrow}$, and $d_{kq}^{\leftarrow}$, respectively. The above defined extra travel distance Δ_{kpq} has influence on the drivers' willingness for accepting the orders on the platform. More specifically, driver k's willingness to undertake tasks p and q sequentially is denoted by $\hat{w}_{kpq}$, which is calculated as: $\hat{w}_{kpq} = \frac{(l_p+l_q)e}{c_k(\Delta_{kpq}/v)} = \frac{(l_p+l_q)ev}{c_k\Delta_{kpq}}$.

No matter whether these two tasks are assigned simultaneously or before the delivery of the first task, the willingness to accept tasks depends on the delivery routing problem. For instance, it is possible that the driver first visits task p's origin and task q's origin, and then to their destinations. We define O_p, D_p and $h_{O_pD_p}$ as the task p's origin, task p's destination and the distance between the task p's origin and task p's destination, respectively. As shown in Table 1, we define Δ_{kpqr} as the extra distance for driver k who takes delivery route r to deliver task p and task q (assuming that driver k visits task p's origin and then task q's origin), thereby there are three delivery routes to deliver two parcels. Hence, driver k's willingness to take

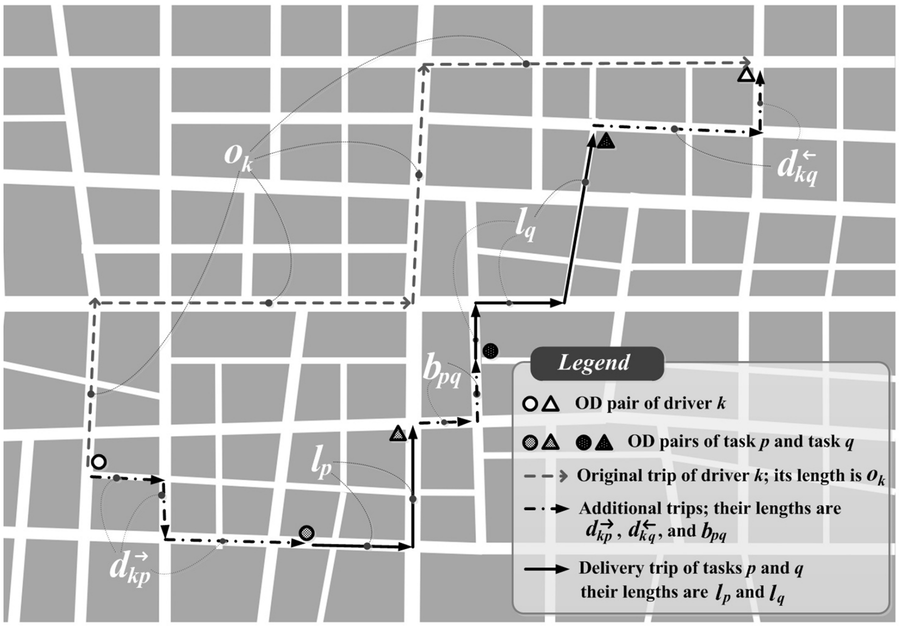


Figure 2: The traveling trip of a driver undertaking two tasks

delivery route r to deliver tasks p and q is denoted by $\hat{w}_{kpqr}$, which is calculated as: $\hat{w}_{kpqr} = \frac{(l_p+l_q)ev}{c_k\Delta_{kpqr}}$.

Table 1: Example of extra distance for three delivery routes

Route ID	Extra Distance
O_p-D_p-O_q-D_q	$\Delta_{kpq1} = d_{kp}^{\rightarrow} + h_{O_pD_p} + h_{D_pO_q} + h_{O_qD_q} + d_{kq}^{\leftarrow} - o_k$
O_p-O_q-D_p-D_q	$\Delta_{kpq2} = d_{kp}^{\rightarrow} + h_{O_pO_q} + h_{O_qD_p} + h_{D_pD_q} + d_{kq}^{\leftarrow} - o_k$
O_p-O_q-D_q-D_p	$\Delta_{kpq3} = d_{kp}^{\rightarrow} + h_{O_pO_q} + h_{O_qD_q} + h_{D_qD_p} + d_{kp}^{\leftarrow} - o_k$

* Notes: 'Route ID', the four characters denote the four places that the ad hoc driver passes by in sequence.

We define a job i as a set of one or two tasks. The set I denotes the collection of all jobs. Let parameter w'_{kir} denote the willingness of driver k to take the delivery route r to undertake job i. If driver k only undertakes task p (job i), the value of

w'_{kir} is equal to the value of w_{kp}. Else if driver k undertakes tasks p and q (job i), the value of w'_{kir} is equal to the value of the $\hat{w}_{kpqr}$.

4.2. Model considering 'one driver can undertake at most two tasks'

By extending the basic model M_1^{Asg}, a new model considering that one driver can undertake two tasks is formulated in this subsection. We make the following assumptions:

(II)–(VIII),

(IX) An ad hoc driver can be assigned at most two delivery tasks.

Before formulating the mathematical model for this problem, we list the notation used in this paper as follows.

Newly defined indices and sets

I	set of all jobs, i.e., combinations of tasks, index $i, i = 1, 2, \cdots, \lvert I \rvert$.
I_p	set of jobs that contain task p.
R	set of all delivery routes, index $r, i = 1, 2, \cdots, \lvert R \rvert$.

Newly defined parameters

m'_i	profit for the company if job i is fulfilled.
w'_{kir}	willingness of driver k to take delivery route r to undertake job i. If $w'_{kir} \geq 1$, driver k is willing to take delivery route r for job i.

Newly defined variables

γ'_{kir}	binary, equal to one if driver k is assigned to take delivery route r for job i; otherwise zero.

Mathematical model

$$[\boldsymbol{M_2^{Asg}}] \quad Z_2^{Asg} = \text{Maximize} \sum_{i \in I} m'_i \sum_{k \in K} \sum_{r \in R} \gamma'_{kir} \tag{11}$$

subject to:

$$\sum_{i \in I} \sum_{r \in R} \gamma'_{kir} \leq 1 \quad \forall k \in K \tag{12}$$

$$\sum_{k \in K} \sum_{i \in I_p} \sum_{r \in R} \gamma'_{kir} \leq 1 \quad \forall p \in P \tag{13}$$

$$\gamma'_{kir} \leq \begin{cases} 1, & w'_{kir} \geq 1 \\ 0, & w'_{kir} < 1 \end{cases} \quad \forall k \in K, i \in I, r \in R \tag{14}$$

$$\gamma'_{kir} \in \{0, 1\} \quad k \in K, i \in I, r \in R. \tag{15}$$

Objective (11) maximizes the total profit for the company that operates the crowdsourced delivery platform. Constraints (12) guarantee that each driver is assigned at most one job along one delivery route, which also means at most two delivery tasks. Constraints (13) make sure each delivery task is assigned to at most one driver. Constraints (14) guarantee that a driver is assigned to take one delivery route to deliver a job which the driver is willing to undertake. Constraints (15) define the domain of newly added decision variables.

The above model can also be extended to consider the case of delivering more than two tasks at the same time. If more than two tasks can be assigned to one driver, the modification of the model M_2^{Asg} focuses on the parameters w'_{kir}, besides replacing the constraints of the model M_2^{Asg}. The calculation of the new w'_{kir} is much more complex than the above defined one. In addition, for the case (e.g., 3 tasks) that a number of task p and other two tasks is assigned to a driver, we need to consider task p is the first, second, or the third one for the driver. All of the above further complicates the model significantly. This case is not discussed in this paper. However, the model structure is not changed when considering the case of 'more than two tasks for a driver'.

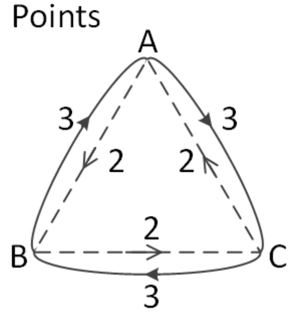
Proposition 3. *The linear programming relaxation of model* $[M_2^{Asg}]$ *does not always have an integer optimal solution.*

Proof. To prove the proposition, we just need to provide an example in Figure 3. Consider a simple case with three points (i.e., A, B and C), three tasks (i.e., 1, 2 and 3) and three ad hoc drivers (k, m and n). Suppose all routes are one-way, all of the distances from points A to B, from points B to C, and from points C to A are 2, and all of the distances from points A to C, from points B to A, and from points C to B are 3. The origins of tasks 1, 2, 3 and drivers k, m, n are points A, B, C, A, C and B, respectively, and the destinations of tasks 1, 2, 3 and drivers k, m, n are points B, C, A, C, B and A respectively. Personal time costs of all drivers are 1.1 and the profit for the company if any task is fulfilled by any driver is 1. The values of unit revenue per kilometer of all drivers and vehicle speed of all drivers are set to 1 and 1, respectively. In this case, an optimal solution of model $[M_2^{Asg}]$ is that task 1 and 2 are fulfilled by driver k, task 3 is not assigned successfully, and the objective value is 2. However, if we solve the linear programming relaxation of model $[M_2^{Asg}]$, we can find that the optimal solution is that driver k fulfills 0.5 task 1 and 0.5 task 2, driver m fulfills 0.5 task 1 and 0.5 task 2, driver n fulfills 0.5 task 1 and 0.5 task 2, and the objective value of the linear programming relaxation of model $[M_2^{Asg}]$ is 3. Therefore, the linear programming relaxation of model $[M_2^{Asg}]$ does not always have an integer optimal solution. $\square$

Tasks

ID	origin	→	destination
1	A	→	B
2	B	→	C
3	C	→	A

Points



Ad hoc drivers

ID	origin	→	destination
k	A	→	C
m	C	→	B
n	B	→	A

Figure 3: An example of the Proposition 3

Then based on the objective value of the two models M_2^{Asg} and $M_1^{Asg'}$, we quantify the advantage of the operation strategy that a driver can take two tasks by calculating the following gap value.

$$Val_{Asg}^{2/1} = Z_2^{Asg} - Z_1^{Asg}. \tag{16}$$

5. Evaluating the advantage of a bonus mode

In real life, there are often some ‘unfavorable’ orders not picked up by any ad hoc driver, which could lead to decreases in the customer satisfaction rate, the task fulfillment rate, and the final profit of the company. The feature of ‘unfavorable’ tasks is that no driver is willing to accept the task. For these 'unfavorable' tasks, the company may pay some extra bonuses so as to incentivize some drivers to undertake these tasks. This way could not only improve the task fulfillment rate, which reflects the service quality of the crowdsourced delivery platform, but also increase the final profit of the company although some extra 'costs' (bonuses) are paid.

Based on previously proposed model M_2^{Asg}, a new model considering 'company pays bonuses to drivers' is formulated as follows. We make the following assumptions: (II)–(IX).

Before formulating the mathematical model for this problem, we list the notation used in this paper as follows.

Newly defined parameters

l'_i	travel distance for job i. If job i consists of one task (e.g., p), $l'_i = l_p$; else if job i consists of two tasks (e.g., p and q), $l'_i = l_p + l_q$.
Δ'_{kir}	extra distance for driver k taking delivery route r to deliver job i. If job i consists of one task (e.g., p), $\Delta'_{kir} = \Delta_{kp}$; else if job i consists of two tasks (e.g., p and q), $\Delta'_{kir} = \Delta_{kpqr}$.

Newly defined variables

β_i	extra bonus given to job i to stimulate drivers to undertake tasks.
ω'_{kir}	willingness of driver k to take delivery route r to deliver job i. If $\omega'_{kir} \geq 1$, driver k is willing to take delivery route r to deliver job i.

Mathematical model

$$[\boldsymbol{M_3^{Asg}}] \quad Z_3^{Asg} = \text{Maximize} \sum_{i \in I} (m'_i - \beta_i) \sum_{k \in K} \sum_{r \in R} \gamma'_{kir} \tag{17}$$

subject to: Constraints (12), (13), (15)

$$\omega'_{kir} = \frac{(l'_i e + \beta_i)v}{c_k \Delta'_{kir}} \quad \forall k \in K, i \in I, r \in R \tag{18}$$

$$\gamma'_{kir} \leq \omega'_{kir} \quad \forall k \in K, i \in I, r \in R \tag{19}$$

$$\beta_i \geq 0 \quad \forall i \in I \tag{20}$$

$$\omega'_{kir} \geq 0 \quad \forall k \in K, i \in I, r \in R. \tag{21}$$

Objective (17), containing the nonlinear part '$\beta_i \gamma'_{kir}$', maximizes the total profit for the company that operates the crowdsourced delivery platform. Constraints (18) update the drivers' willingness values to undertake jobs along delivery routes considering the possible extra bonus. Constraints (19) replace the previous defined Constraints (14) with the updated values of drivers' willingness. Constraints (20)–(21) define the domain of newly added decision variables.

Proposition 4. *The following model* $[M_3^{Asg'}]$ *is equivalent to the previous nonlinear model* $[M_3^{Asg}]$.

Mathematical model

$$[\boldsymbol{M_3^{Asg'}}] \quad Z_3^{Asg'} = \text{Maximize} \sum_{i \in I} \sum_{k \in K} \sum_{r \in R} m'_i \gamma'_{kir} - \sum_{i \in I} \beta_i \tag{22}$$

subject to: Constraints (12), (13), (15), (18)–(21).

Proof. To prove the proposition, we just need to prove that Objective (17) is equivalent to Objective (22). First, we can convert Objective (17) as follows:

$$Z_3^{Asg} = \text{Maximize} \sum_{i \in I} m'_i \sum_{k \in K} \sum_{r \in R} \gamma'_{kir} - \sum_{i \in I} \beta_i \sum_{k \in K} \sum_{r \in R} \gamma'_{kir}. \tag{23}$$

Then in the optimal solution, if job i is not assigned to any driver, extra bonus β_i given to job i will be zero, and the value of '$\beta_i \sum_{k \in K} \sum_{r \in R} \gamma'_{kir}$' will be zero. Hence, the value of '$\sum_{i \in I} \beta_i \sum_{k \in K} \sum_{r \in R} \gamma'_{kir}$' equals the value of '$\sum_{i \in I} \beta_i$', and Objective (17) is equivalent to Objective (22). □

Proposition 5. *The linear programming relaxation of model* $[M_3^{Asg'}]$ *does not always have an integer optimal solution.*

Proof. To prove the proposition, we just need to provide an example. Consider a simple case with one job (i.e., 1) and one ad hoc driver (i.e., a). Suppose the values of travel distance for job 1 (l'_1), driver a's vehicle speed, driver a's personal time cost, driver a's unit revenue per kilometer, extra distance for the driver a undertaking job 1 are 0.3, 1, 1, 1, 1, respectively, and the profit for the company if job 1 is fulfilled by driver a is 0.8. In this case, the company will pay 0.7 as an extra bonus to incentivize driver a to undertake job 1, the maximal profit for the company of model $[M_3^{Asg'}]$ is 0.1. However, if we solve the linear programming relaxation of model $[M_3^{Asg'}]$, we can find that the optimal solution is $\gamma'_{a1} = 0.3$, and the objective value is 0.24. Therefore, the linear programming relaxation of model $[M_3^{Asg'}]$ does not always have an integer optimal solution. □

Based on the objective value of the two models $M_3^{Asg'}$ and M_2^{Asg} , we quantify the advantage of the bonus system by calculating the following gap value.

$$Val_{Asg}^{3/2} = Z_3^{Asg'} - Z_2^{Asg}. \tag{24}$$

6. Evaluating the cost brought by a task cancellation mode

In reality, a customer may cancel his (or her) task when waiting for a bit long time. For example, if the driver fails to arrive at the agreed pickup point within 10 minutes after receiving the order, Didi consumers can cancel the order at will without liability (Didi, 2020d). Suppose each task (e.g., task p) has its latest service time t_p. If task p is served by a driver after t_p, the task p's customer may cancel the order at a probability of q_p; otherwise, the customer cannot cancel the order (task). Here q_p can be estimated according to the historical data related to the order/cancel behavior of the task p's customer.

For the interest of simplicity, we consider the issue of 'order cancel' on the basis of the model M_1^{Asg}, i.e., the case of single task for each driver. In this case, the time when driver k arrives at the pickup point of task p is $d_{\vec{kp}}/v$. If $d_{\vec{kp}}/v > t_p$, the task may be cancelled at any time point τ in the interval $\tau \in [t_p, d_{\vec{kp}}/v]$ with a probability density function $f_p(\tau)$; here $q_p = \int_{t_p}^{d_{\vec{kp}}/v} f_p(\tau)d\tau$.

Customers' order cancel behavior will bring 'loss' for drivers because they have been heading to the pickup locations of the customers. Here the drivers' loss is defined as their extra time costs. Crowdsourcing platforms may compensate these ad hoc drivers and consider how to minimize the expected extra time cost of drivers who encounter order cancellations. For each pair of a driver k and a task p, the extra time cost depends on the time point τ of the cancellation, at which time the driver k will finish heading to the task p's pickup location and turn to the driver's own destination; the extra time cost can be calculated in advance and is defined as input data $c_k\tilde{\Delta}_{kp}(\tau)/v$; here $\tilde{\Delta}_{kp}(\tau)$ is a function denoting the extra distance if driver k undertakes task p but the task is cancelled at time τ. Then the expected loss for driver k undertaking task p is:

$$s_{kp} = \int_{t_p}^{d_{\overrightarrow{kp}}/v} c_k\tilde{\Delta}_{kp}(\tau)/v f_p(\tau)d\tau. \tag{25}$$

Here $\int_{t_p}^{d_{\overrightarrow{kp}}/v} f_p(\tau)d\tau = q_p$. For the interest of simplicity, we assume the probability function $f_p(\tau)$ is a uniform distribution in the following numerical experiments. Then $s_{kp} = \frac{c_k q_p}{d_{\overrightarrow{kp}} - t_p v} \int_{t_p}^{d_{\overrightarrow{kp}}/v} \tilde{\Delta}_{kp}(\tau)d\tau$.

Besides the above, another important parameter m_p (i.e., profit for the company if task p is fulfilled) in the objective depends on which driver is assigned, and is redefined as m''_{kp}.

$$m''_{kp} = \begin{cases} m_p(1-q_p) - s_{kp}, & t_p - \frac{d_{\overrightarrow{kp}}}{v} < 0 \\ m_p, & t_p - \frac{d_{\overrightarrow{kp}}}{v} \geq 0 \end{cases} \quad \forall k \in K, p \in P. \tag{26}$$

In terms of the 'loss' brought by customers' order cancel behavior to drivers, crowdsourcing platforms may compensate these ad hoc drivers and consider how to maximize the expected profit of company. Then the model for considering 'tasks may be cancelled' is formulated on the basis of the model M_1^{Asg}. We make the following assumptions:

(I)–(VIII),

(X) Customers are only allowed to cancel the order before the delivery.

Before formulating the mathematical model for this problem, we list the notation used in this paper as follows.

Newly defined parameters

m''_{kp} expected profit for the company if task p is fulfilled by driver k under the possibility that task p may be cancelled.

Mathematical model

$$[\boldsymbol{M_4^{Asg}}] \quad Z_4^{Asg} = \text{Maximize} \sum_{p \in P} \sum_{k \in K} m''_{kp} \varphi_{kp} \tag{27}$$

subject to: Constraints (2)–(5).

The above model is based on the case that one driver can undertake at most one task. If two tasks can be assigned to one driver, the modification of the model M_4^{Asg} focuses on the parameters s_{kp} and m''_{kp}, besides replacing the constraints with the model $M_4^{Asg'}$s constraints. The calculation of the new s_{kp} and m''_{kp} is much more complex than the above defined one because it should consider whether only task p is assigned to a driver or a pair of task p and another task is assigned to a driver, which makes the newly modified s_{kp} and m''_{kp} become decision variables related the 'task-driver' assignment decision. In addition, for the case that a pair of task p and another task is assigned to a driver, we need to consider task p is the first one or the second one for the driver. All of the above further complicates the model significantly. This case is not discussed in this paper. However, the model structure is not changed when considering the case of 'two tasks for a driver'. Besides, model

$[M_4^{Asg}]$ is similar to model $M_1^{Asg'}$, thereby the integrality constraint can be dropped and model $[M_4^{Asg}]$ can be easily solved as a linear programming problem. We convert the above integer programming model $[M_4^{Asg}]$ to the linear programming relaxation of model $[M_4^{Asg'}]$.

Mathematical model

$[M_4^{Asg'}]$ Objective (27)

subject to: Constraints (2)–(4), (6).

Then based on the objective values of this model and model $M_1^{Asg'}$, we evaluate the cost brought by 'tasks may be cancelled' to crowdsourcing platforms by calculating the following gap value.

$$Val_{Asg}^{1/4} = Z_1^{Asg} - Z_4^{Asg}. \tag{28}$$

7. Evaluating the advantage of a mixed bonus-cancellation mode

After studying a bonus model and a task cancellation model separately, this study also wants to investigate a 'mixed bonus-cancellation mode' to provide managerial implications. For the interest of simplicity, we consider the mixed bonus-cancellation mode on the basis of the model M_1^{Asg}, i.e., the case of single task for each driver. We make the following assumptions:

(I)–(VIII), (X).

Before formulating the mathematical model for this problem, we list the notation used in this paper as follows.

Newly defined variables

β'_p extra bonus given to task p to stimulate drivers to undertake the task.

ω''_{kp} willingness of driver k to undertake task p. If $\omega''_{kp} \geq 1$, driver k is willing to undertake task p.

Mathematical model

$$[\boldsymbol{M_5^{Asg}}] \quad Z_5^{Asg} = \text{Maximize} \sum_{p\in P}\sum_{k\in K}(m''_{kp} - \beta'_p)\varphi_{kp} \tag{29}$$

subject to: Constraints (2), (3), (5)

$$\omega''_{kp} = \frac{(l_p e + \beta'_p)v}{c_k \Delta_{kp}} \quad \forall k \in K, p \in P \tag{30}$$

$$\varphi_{kp} \leq \omega''_{kp} \quad \forall k \in K, p \in P \tag{31}$$

$$\beta'_p \geq 0 \quad \forall p \in P \tag{32}$$

$$\omega''_{kp} \geq 0 \quad \forall k \in K, p \in P. \tag{33}$$

Objective (29), containing the nonlinear part '$\beta'_p\varphi_{kp}$', maximizes the total profit for the company that operates the crowdsourced delivery platform with the mixed bonus-cancellation mode. Constraints (30) update the drivers' willingness values to undertake a task considering the possible extra bonus. Constraints (31) guarantee that a driver will be assigned to a task if this driver is willing to undertake this task. Constraints (32)–(33) define the domain of newly added decision variables.

As with the model M_4^{Asg}, the above model is based on the case that one driver can undertake at most one task. The integrality constraint can be dropped and model

$[M_5^{Asg}]$ can be easily solved as a linear programming problem. We convert the above integer programming model $[M_5^{Asg}]$ to the linear programming relaxation of model $[M_5^{Asg'}]$.

Mathematical model

$$[\boldsymbol{M_5^{Asg'}}] \quad Z_5^{Asg'} = \text{Maximize} \sum_{p \in P} \sum_{k \in K} m''_{kp} \varphi_{kp} - \sum_{p \in P} \beta'_p \tag{34}$$

subject to: Constraints (2), (3), (5), (30)–(33).

Then based on the objective values of the model $M_4^{Asg'}$ and model $M_5^{Asg'}$, we evaluate the benefit of the mixed bonus-cancellation mode to crowdsourcing platforms by calculating the following gap value.

$$Val_{Asg}^{5/4} = Z_5^{Asg'} - Z_4^{Asg}. \tag{35}$$

8. Computational experiments

In order to evaluate the proposed model and verify the efficiency of our 'optimization-featured' mode platform and different strategies, we perform several computational experiments on a PC (Intel Core i7, 2.6 GHz; Memory, 8 GB). Mathematical models proposed in this study are implemented by CPLEX 12.5.1 (Visual Studio 2015, C#).

8.1. Experimental setting

We first summarize the setting of our parameter values. In order to test the viability of the proposed crowdsourced delivery, we apply this model application to the Walmart shopping stores as a real-world case. Walmart investigated the possibility of in-store customers delivering goods as ad hoc drivers to online customers on the way home from stores in 2013 (Morphy, 2013). We develop a simulation environment that considers the proposed crowdsourced delivery in the Shanghai city, China. Shanghai represents a potentially interesting environment for the crowdsourced delivery since the working pace of people there is relatively fast and the delivery demand is huge. It also represents a challenging test case due to its large size and the large number of ad hoc drivers. As shown in Figure 4, there are 13 Walmart shopping stores in Shanghai. We set the origins of both delivery tasks and ad hoc drivers to follow the uniform distribution over these 13 Walmart shopping stores. The destinations of both delivery tasks and ad hoc drivers are uniformly distributed over the service area (30°23′–31°27′ N, 120°52′–121°45′ E). We use Euclidean distance to obtain the values of parameters l_p, o_k, $d_{kp}^{\rightarrow}$ and $d_{kp}^{\leftarrow}$. The average value of c_k is set to 10 USD/h, which is comparable to the US minimum wage rate (DOL, 2019) and the related work (Kafle et al., 2017). We set the value of v to 60 km/h, which is nearly in line with the setting used in related works (Agatz et al., 2011). The profit for the company m_p is set to 25% of all fares, which is consistent with realistic data from the Uber company (Uber, 2019). The values of e and q_p are set to 0.22 and 1, respectively. We set the value of the latest service time for each delivery task to 20 minutes after the platform receives the task (Arslan et al., 2018).

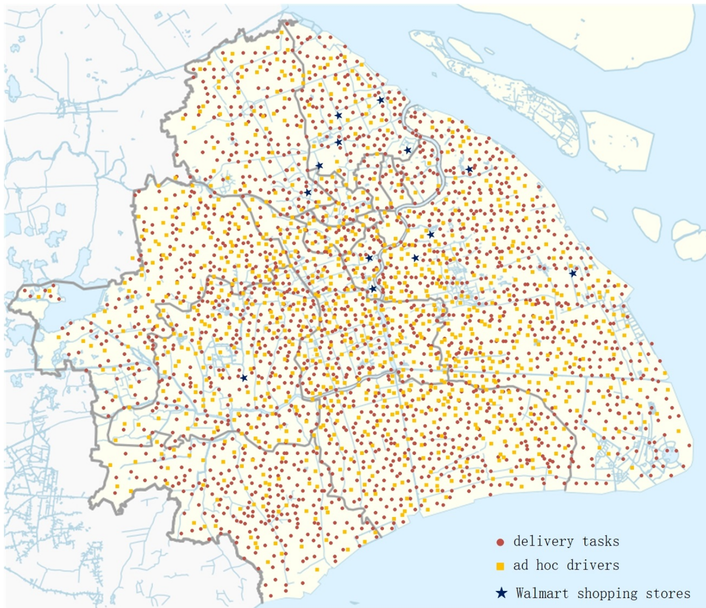


Figure 4: Walmart shopping stores and the destinations of delivery tasks and ad hoc drivers

8.2. Base analysis

For the base analysis, we present the results for different modes and compare the solutions with different problem scales.

(1) *Evaluating the advantage of 'Asg' mode to 'Grb' mode*

As shown in Table 2, the values of 'Z_1^{Asg}' and 'Z_1^{Grb}' represent the objective values of the '*Asg*' mode (recall the mode that tasks are assigned to one ad hoc driver by the platform) and the '*Grb*' mode (recall the mode that drivers grab tasks), respectively. 'GAP_{ABS}' represents the difference between objective function values of two modes, 'GAP_{REL}' makes the difference values more intuitive, and the 'GAP_{REL}' values can be calculated by $\frac{Z_1^{Asg}-Z_1^{Grb}}{Z_1^{Grb}} \times 100$. '$TIME^{A}$' and '$TIME^{G}$' represent the CPU running time of the '*Asg*' mode and '*Grb*' mode, respectively. '$\frac{TIME^{A}}{TIME^{G}}$' records the CPU running time ratio of the 'Asg' mode and 'Grb' mode. By calculating the difference values between the best and the worst cases, we can quantify the advantage of the mode 'Asg' to the mode 'Grb'. With the increasing number of ad hoc drivers and delivery tasks, although the CPU running time becomes longer, the advantage of 'Asg' mode becomes more obvious. It is clear that the 'Asg' mode leads to the company profit growth of more than 79%, even 345% in some cases, which demonstrates the significant advantage brought by the 'Asg' mode. We find that the company profit of 'Asg' mode is considerably higher than that of 'Grb' mode in all cases, which indicates the commercial value of 'Asg' mode again. Besides, for small-scale computational examples, the running speed of 'Asg' mode is relatively slow. As the computational scale increases, 'Asg' mode can match the ad hoc drivers with tasks well in a short time. Moreover, when the ratio of the number of ad hoc drivers to the number of delivery tasks is changed from $\frac{1}{2}$ to $\frac{1}{3}$, in most cases, crowdsourcing platforms may have more profits. Therefore, we recommend that crowdsourcing platforms focus on allocating ad hoc drivers to serve more customers.

(2) *Evaluating the advantage of undertaking two tasks for each driver*

This subsection tries to investigate the potential advantage of the operation strategy that a driver can take two tasks. As shown in Table 3, the values of 'Z_1^{Asg}' and 'Z_2^{Asg}' represent the objective values of the mode that a driver can take only one task assigned by the platform, and the mode that a driver can take two tasks assigned by the platform, respectively. 'GAP_{ABS}' represents the difference between objective function values of two modes that a driver can take two tasks and only one task. 'GAP_{REL}' makes the difference values more intuitive, and the 'GAP_{REL}'

Table 2: Evaluating results of mode ‘Asg’ and mode ‘Grb’ for the Walmart case

Case ID	$Val_1^{A/G}$						
	Z_1^{Asg} (USD)	Z_1^{Grb} (USD)	GAP_{ABS} (USD)	GAP_{REL} (%)	$TIME^A$ (s)	$TIME^G$ (s)	$\frac{TIME^A}{TIME^G}$
2-5	7.30	4.07	3.23	79.51	0.09	0.02	4.50
2-6	11.13	5.37	5.75	107.13	0.06	0.01	6.00
10-20	35.70	13.57	22.14	163.16	0.06	0.02	3.00
10-30	33.50	11.12	22.37	201.16	0.08	0.03	2.67
20-50	80.25	23.83	56.41	236.69	0.09	0.07	1.29
20-60	77.80	22.64	55.16	243.65	0.08	0.23	0.35
35-70	131.13	49.65	81.48	164.13	0.08	0.41	0.20
35-105	131.47	29.50	101.97	345.62	0.09	1.05	0.09
50-100	191.83	63.23	128.60	203.38	0.11	1.03	0.11
50-150	209.09	47.45	161.64	340.64	0.11	1.23	0.09
60-120	229.80	66.95	162.86	243.27	0.14	2.14	0.07
60-180	238.69	54.35	184.34	339.16	0.20	2.81	0.07
70-140	252.50	81.58	170.92	209.51	0.20	4.53	0.04
70-210	261.89	72.68	189.21	260.35	0.25	5.52	0.05

* Notes: In ‘Case ID’, the two values denote the number of ad hoc drivers and delivery tasks, respectively.

values can be calculated by $\frac{Z_2^{Asg} - Z_1^{Asg}}{Z_1^{Asg}} \times 100$. ‘$TIME_1^A$’ and ‘$TIME_2^A$’ represent the CPU running time of two modes that a driver can take only one task and two tasks, respectively. ‘$\frac{TIME_1^A}{TIME_2^A}$’ records the CPU running time ratio of these two modes. It is noted that with the increasing scale of computational experiments, the solution time becomes longer, but the effect of this operation strategy becomes better. The results in the ‘GAP_{REL}’ column also validate a significant benefit could be brought by allowing a driver to undertake multiple tasks. Besides, the operation strategy that a driver can take two tasks always corresponds to higher company profits (company profit growth of more than 38%).

(3) *Evaluating the advantage of the bonus system*

This subsection tries to evaluate the advantage of bonus system. In Table 4, we compute the ‘GAP_{ABS}’ values between the modes with and without the bonus

Table 3: Evaluating results of the modes that a driver can take one and two tasks for the Walmart case

Case ID	$Val_{Asg}^{2/1}$						
	Z_1^{Asg} (USD)	Z_2^{Asg} (USD)	GAP_{ABS} (USD)	GAP_{REL} (%)	$TIME_1^A$ (s)	$TIME_2^A$ (s)	$\frac{TIME_1^A}{TIME_2^A}$
2-5	7.30	10.08	2.78	38.05	0.06	0.04	1.50
10-20	35.70	63.69	27.98	78.38	0.06	0.14	0.43
20-50	80.25	131.63	51.38	64.03	0.08	1.66	0.05
35-70	131.13	237.33	106.20	80.99	0.11	7.79	0.01
50-100	191.83	292.92	101.09	52.70	0.14	19.20	<0.01
80-160	302.62	476.54	173.91	57.47	0.24	105.45	<0.01
100-200	380.18	584.66	204.48	53.78	0.31	306.69	<0.01
110-220	422.29	648.20	225.91	53.50	0.38	629.29	<0.01
120-240	457.86	666.56	208.70	45.58	0.42	1,230.90	–
160-320	626.35	–	–	–	0.72	$\geq$ 3,600	–
200-400	795.81	–	–	–	61.06	$\geq$ 3,600	–

* Notes: (1) In 'Case ID', the two values denote the number of ad hoc drivers and delivery tasks, respectively. (2) The en-dash means we did not find any solution within one hour or the solution is suspended by an 'out-of-memory' error.

system. Z_2^{Asg} and $Z_3^{Asg'}$ represent the objective values of model M_2^{Asg} and $M_3^{Asg'}$, respectively. '$TIME_2^A$' and '$TIME_3^A$' represent the CPU running time of two modes without and with the bonus system, respectively. Observing these gap values in Table 4, we find that the objective values of the mode with the bonus system are larger than those without the bonus system in $\frac{8}{10} \times 100\% = 80\%$ computational experiments. But the advantage of the bonus system is relatively small because the gap values between the objective values of the modes with and without the bonus system are small. Especially when the number of ad hoc drivers is large or the number of delivery tasks is small, although a certain delivery task is unfavorable for some drivers, it can attract other drivers to take this task. In this case, the advantage of bonus system is not obvious.

(4) *Evaluating the advantage of the policy that tasks can be cancelled*

Table 4: Evaluating results of the modes with and without the bonus system for the Walmart case

Case ID	$Val^{3/2}_{Asg}$					
	Z_2^{Asg} (USD)	$Z_3^{Asg'}$ (USD)	GAP_{ABS} (USD)	GAP_{REL} (%)	$TIME_2^A$ (s)	$TIME_3^A$ (s)
2-5-1	10.08	10.08	0.00	0.00	0.10	0.02
2-5-2	11.57	11.57	0.00	0.00	0.06	0.04
10-20-1	63.69	69.36	5.67	8.90	0.19	5.56
10-20-2	63.28	67.68	4.40	6.95	0.20	5.56
20-50-1	131.63	134.59	2.96	2.25	1.87	246.45
20-50-2	139.02	142.39	3.37	2.42	2.00	11.00
30-60-1	208.78	208.97	0.19	0.09	6.70	938.19
30-60-2	207.76	208.66	0.90	0.43	5.96	411.58
35-70-1	237.33	237.52	0.19	0.08	8.21	120.92
35-70-2	238.96	240.62	1.67	0.70	8.22	801.13
50-100-1	292.92	–	–	–	21.91	$\geq$ 3,600
50-100-2	300.41	–	–	–	20.43	$\geq$ 3,600

* Notes: (1) In 'Case ID', the two values denote the number of ad hoc drivers and delivery tasks, respectively. (2) The en-dash means we did not find any solution within one hour or the solution is suspended by an 'out-of-memory' error.

We also evaluate the cost brought by the 'tasks may be cancelled' to crowdsourcing platforms situation in Table 5. The values of 'Z_1^{Asg}' and 'Z_4^{Asg}' represent the objective values of the modes without and with the operation strategy 'tasks may be cancelled', respectively. 'GAP_{ABS}' represents the difference between objective function values of these two modes. 'GAP_{REL}' makes the difference values more intuitive, and the 'GAP_{REL}' values can be calculated by $\frac{Z_1^{Asg}-Z_4^{Asg}}{Z_1^{Asg}} \times 100$. '$TIME_1^A$' and '$TIME_4^A$' represent the CPU running time of two modes without and with the operation strategy 'tasks may be cancelled', respectively. By observing the 'GAP_{REL}' values, here Z_1^{Asg} and Z_4^{Asg} represent the objective values of model $M_1^{Asg'}$ and $M_4^{Asg'}$, respectively, we find that in most cases the operation strategy 'tasks may be cancelled' did not lead to a significant reduction in profits, and the profits in most cases fell less than 1%. At the same time, this operation strategy can attract more customers because it protects the rights and interests of customers. Because the reduction in company profits caused by the operation strategy 'tasks may be cancelled' is marginal in large-scale computational instances, we suggest that the company should adopt the operation strategy 'tasks may be cancelled' to attract more consumers.

Table 5: Evaluating results of the modes with and without the strategy 'tasks may be cancelled' for the Walmart case

Case ID	$Val_{Asg}^{1/4}$					
	Z_1^{Asg} (USD)	Z_4^{Asg} (USD)	GAP_{ABS} (USD)	GAP_{REL} (%)	$TIME_1^A$ (s)	$TIME_4^A$ (s)
2-5	7.30	7.30	0.00	0.00	0.09	<0.01
20-50	80.25	80.25	0.00	0.00	0.07	0.02
50-100	191.83	188.91	2.92	1.52	0.12	0.05
100-200	380.18	371.51	8.68	2.28	0.26	0.18
200-400	795.81	794.34	1.47	0.18	0.85	0.71
500-1000	1,953.14	1,941.65	11.49	0.59	6.04	5.52
1000-2000	4,083.40	4,074.59	8.81	0.22	45.37	31.52
1300-2600	5,251.65	5,233.81	17.84	0.34	91.82	61.15
1500-3000	6,061.92	6,044.69	17.23	0.28	140.51	97.79
1800-3600	7,263.90	7,246.69	17.22	0.24	190.92	169.84
2000-4000	8,050.76	8,029.50	21.26	0.26	370.20	237.60
2300-4600	9,291.20	9,261.04	30.16	0.32	452.09	552.98
2400-4800	9,610.87	9,576.57	34.29	0.36	590.63	865.90

* Notes: In 'Case ID', the two values denote the number of ad hoc drivers and delivery tasks, respectively.

(5) *Evaluating the advantage of the mixed bonus-cancellation mode*

We also evaluate the advantage of the 'mixed bonus-cancellation mode' to crowdsourcing platforms in Table 6. Since model $[M_4^{Asg'}]$ and model $[M_5^{Asg'}]$ assume that a driver undertakes at most one task, and model $[M_3^{Asg'}]$ assumes that a driver undertakes at most two tasks, this section of the experiment compares the results of model $[M_4^{Asg'}]$ and model $[M_5^{Asg'}]$. The values of 'Z_4^{Asg}' and '$Z_5^{Asg'}$' represent the objective values of the modes with the 'task cancellation mode' and the 'mixed bonus-cancellation mode', respectively. 'GAP_{ABS}' represents the difference between objective function values of these two modes. 'GAP_{REL}' makes the difference values more intuitive, and the 'GAP_{REL}' values can be calculated by $\frac{Z_5^{Asg'} - Z_4^{Asg}}{Z_5^{Asg'}} \times 100$. '$TIME_4^A$' and '$TIME_5^A$' represent the CPU running time of the 'task cancellation mode' and the 'mixed bonus-cancellation mode', respectively. By observing the 'GAP_{REL}' values, we find that in most cases the 'mixed bonus-cancellation mode' leaded to an increase in the profits. Hence, we suggest that the company should adopt the mixed bonus-cancellation mode.

Table 6: Evaluating results of the modes with and without the 'mixed bonus-cancellation' strategy for the Walmart case

Case ID	$Val_{Asg}^{5/4}$					
	Z_4^{Asg} (USD)	$Z_5^{Asg'}$ (USD)	GAP_{ABS} (USD)	GAP_{REL} (%)	$TIME_4^A$ (s)	$TIME_5^A$ (s)
2-5	7.30	7.30	0.00	0.00	0.08	0.01
20-50	80.25	90.71	10.46	11.53	0.07	0.05
50-100	188.91	208.91	20.00	9.57	0.15	0.14
100-200	371.51	407.16	35.65	8.76	0.30	0.50
200-400	794.34	817.12	22.78	2.79	1.01	1.98
500-1000	1,941.65	2,035.43	93.78	4.61	5.00	13.94
1000-2000	4,074.59	4,224.56	149.97	3.55	29.29	75.11
1300-2600	5,233.81	5,421.35	187.54	3.46	57.16	139.91
1500-3000	6,044.69	6,237.07	192.38	3.08	87.34	210.70
1800-3600	7,246.69	7,577.09	330.40	4.36	128.28	396.09
2000-4000	8,029.50	8,305.54	276.05	3.32	227.16	765.05

* Notes: In 'Case ID', the two values denote the number of ad hoc drivers and delivery tasks, respectively.

8.3. Sensitivity analysis and managerial insights

Some model parameters may affect the attractiveness of the crowdsourced delivery platform. In the remainder of this section, we examine the effect of the latest service time for each task, task cancellation rate and range distribution of tasks on the performance of the crowdsourcing mode. In addition, we also try to determine the minimum number of the needed ad hoc drivers given a set of delivery tasks.

We first discuss the impact of the value of the latest service time for each task on task assignment fulfillment. We take an example of 50 ad hoc drivers and 50 parcel tasks. The experimental setting of our parameter values is the same as the setting in Section 8.1. In Table 7, the columns 't_p', '$\sum_{p\in P}\sum_{k\in K}\varphi_{kp}$', and 'OBJ' record the values of the latest parcel service time, the number of drivers who are assigned successfully, and the objective values of the model $M_4^{Asg'}$, respectively. The results in Table 7 demonstrate that the increase in the latest service time has a direct effect on the task assignment fulfillment. Besides, when the latest service time is set to three minutes after the platform receives the task, the task assignment fulfillment rate is up to 68% (i.e., $\sum_{p\in P}\sum_{k\in K}\varphi_{kp} = 34$).

Table 7: Impact of the latest service time for each task on task assignment fulfillment

Case ID	t_p (min)	$\sum_{p\in P}\sum_{k\in K}\varphi_{kp}$	OBJ
Case 1	1	33	106.25
Case 2	2	33	106.25
Case 3	3	34	109.02
Case 4	4	34	109.02
Case 5	5	34	111.04
Case 6	7	37	119.95
Case 7	10	40	130.62

Then we analyze the impacts of the task cancellation rate on task assignment fulfillment. We take an example of 50 ad hoc drivers and 50 parcel tasks, and the latest service time is set to two minutes after the platform receives the task. As shown in Table 8, we find less parcel tasks can be assigned successfully when task cancellation rate increases. Although when the task cancellation rate is set to 1.0, which means customers will definitely cancel their tasks if they wait for a bit long time, the task assignment fulfillment rate can be $\frac{33}{50} \times 100\% = 66\%$.

Table 8: Impact of the task cancellation rate on task assignment fulfillment

Case ID	q_p	$\sum_{p\in P}\sum_{k\in K}\varphi_{kp}$	OBJ
Case 1	0.1	41	129.18
Case 2	0.2	41	124.47
Case 3	0.3	40	120.42
Case 4	0.4	40	116.72
Case 5	0.5	39	113.19
Case 6	0.6	39	110.07
Case 7	0.7	37	107.60
Case 8	0.8	34	106.46
Case 9	0.9	33	106.25
Case 10	1.0	33	106.25

Given a collection of delivery tasks (specified by the origin and destination), a decision maker may also be interested in determining the minimum number of drivers needed to serve all tasks. Based on the above computational experiments, we find that the economic benefits brought by the operation strategy that a driver can take two tasks are significant, hence this operation strategy is adopted in the following experiments. By extending the basic model M_2^{Asg}, a new model considering 'minimum fleet problem' is formulated in this subsection. It is noted that Constraints (13) are converted to Constraints (37). Without this conversion, Objective (36) for the minimal case is zero, which is meaningless and disobeys the reality.

Mathematical model

$$[\boldsymbol{M_6^{Asg}}] \quad Z_6^{Asg} = \text{Minimize} \sum_{i\in I}\sum_{k\in K}\sum_{r\in R}\gamma'_{kir} \tag{36}$$

subject to: Constraints (12), (14), (15)

$$\sum_{k\in K}\sum_{i\in I_p}\sum_{r\in R}\gamma'_{kir}=1 \quad \forall p\in P. \tag{37}$$

Objective (36) minimizes the number of ad hoc drivers needed to serve all tasks. Constraints (37) guarantee that each task is assigned to one ad hoc driver.

Some computational experiments are conducted to investigate the performance of the 'minimum fleet' strategy by comparing it with the 'maximum profit' strategy. Figure 5 shows the comparisons on the profit and the number of ad hoc drivers needed for each problem scale, which is featured by the number of ad hoc drivers and delivery tasks (i.e., the horizontal axis in Figure 5). The left and right vertical axes record the ratio of the total company profit under the 'minimum fleet' strategy (denoted by P_{min_fleet}) to that under the 'maximum profit' strategy (denoted by P_{max_profit}), and the ratio of the number of ad hoc drivers needed under the 'minimum fleet' strategy (denoted by D_{min_fleet}) to that under the 'maximum profit' strategy (denoted by D_{max_profit}), respectively. It is noted that the numbers of ad hoc drivers needed under 'maximum profit' strategy and 'minimum fleet' strategy are almost the same, but the profit under the 'minimum fleet' strategy is significantly less than that under the 'maximum profit' strategy in most cases. This comparison also validates the advantage of the 'optimization-featured' mode platform.

Crowdsourced delivery platforms have now been extended to many takeout platforms. Different cities have different order structures. For example, some big cities can serve a wider range of customers because they have more manpower and

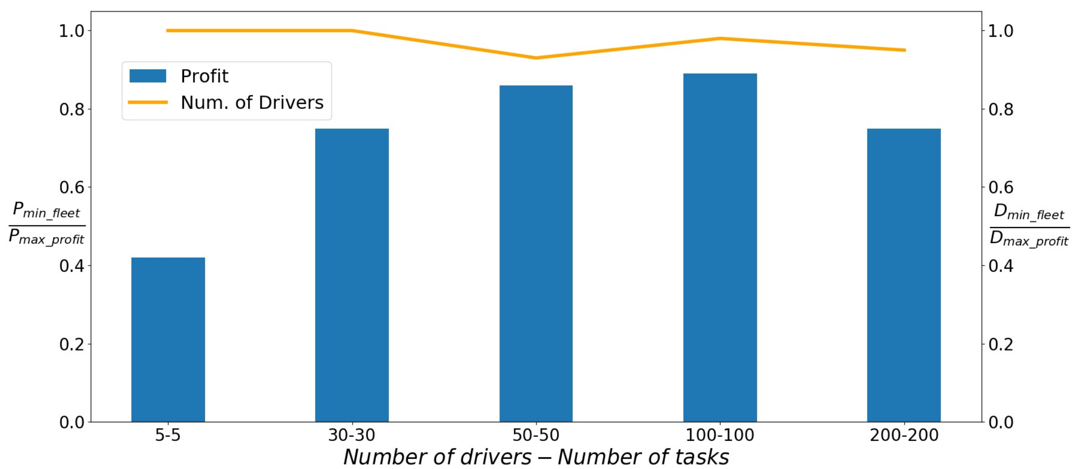


Figure 5: Comparison between two strategies on the profit and the number of drivers

customer demands. However, in some small cities, the service range is relatively small. Li et al. (2020) has shown that the properties of orders significantly affect matching performance and crowdsourced delivery platforms' profits, thereby this study investigates the impact of range distribution of orders on crowdsourced delivery platforms' profits. In this subsection, the destinations of orders are uniformly distributed inside a circle with a certain radius (e.g., 3 km) around 13 Walmart shopping stores in Shanghai. Values of other parameters are set the same as those in Section 8.1. We take an example of 50 ad hoc drivers and 50 orders, and the latest service time is set to two minutes after the platform receives the task. Table 9 records the objective values of the model $M_4^{Asg'}$, which represents the total profits for the crowdsourced delivery platform. And the 'Growth Rate' values of 'Case ID i' can be calculated by $\frac{OBJ_{i+1}-OBJ_i}{OBJ_i} \times 100$. The results in Table 9 shows that a larger service radius has a direct impact on the crowdsourced delivery platforms' profit growth. However, this does not mean that constantly expanding the service radius will always lead to profit growth, which is because as the service radius grows, the rate of profit growth decreases. As shown in Case ID 10, the growth rate is -19.14%.

Table 9: Impact of range distribution of orders on crowdsourced delivery platforms' profits

Case ID	Range	OBJ	Growth Rate(%)
Case 1	0-1 km	1.559	433.55
Case 2	0-2 km	8.318	76.73
Case 3	0-3 km	14.700	29.36
Case 4	0-4 km	19.016	26.19
Case 5	0-5 km	23.997	22.52
Case 6	0-6 km	29.402	41.80
Case 7	0-7 km	41.692	7.63
Case 8	0-8 km	44.873	13.97
Case 9	0-9 km	51.141	-19.14
Case 10	0-10 km	41.355	–

9. Conclusions

This study investigates a recently emerging business mode of crowdsourced delivery that utilizes the excess capacity of the existing traffic flow in urban areas for delivering parcels. We consider six operation modes of a crowdsourced delivery service company, including the 'grabbing mode', the 'assignment mode', the 'two tasks assignment mode', the 'bonus mode', the 'task cancellation mode', and the 'mixed bonus-cancellation mode'. This study proposes six mathematical models to analyze the effect of different operation modes on the crowdsourced delivery service company in a quantitative way. This study also investigates the impacts of several realistic factors, such as the latest service time for each task, task cancellation rate and range distribution of tasks. It is obvious that these factors make our quantitative methodology fit the realistic needs of the crowdsourced delivery in the background that the urban parcel delivery needs a revolutionary transformation because of the e-commerce explosion. This study has two contributions by comparing with the related works.

(1) This study introduces the evaluation of different crowdsourcing modes for the urban parcel delivery problem. We provide the quantitative methodology to create matches between parcel delivery tasks and ad hoc drivers who are willing to make a small detour in exchange for extra compensations. This study proposes six mathematical models to compare the effect of six operation modes on the crowdsourced delivery service company. Besides, we also consider some realistic operating limits, such as the latest service time for each task, task cancellation rate and range distribution of tasks, which have also been frequently ignored in existing studies even though these factors are crucial to the real-world parcel delivery.

(2) Based on the extensive computational experiments, including a real-world case and sensitivity analysis, we draw out some important managerial suggestions on the crowdsourced delivery for the urban parcel delivery. For instance, we indicate the commercial value of 'Asg' mode in terms of the company profits and CPU running time. We also find that the economic benefits brought by the operation strategy that a driver can take two tasks are significant and the advantage of bonus system is relatively small. Besides, we suggest that the company should adopt the 'mixed bonus-cancellation mode' to attract more customers.

However, this study also has limitations. Presently, we consider only single-hop delivery, i.e., when only one ad hoc driver can be involved in performing the crowdsourced delivery for one parcel. The possibility of multi-hop delivery, in which a parcel can be changed hands on its way to the target destination, can be analyzed in future studies. Allowing the parcel transfer between drivers will help to increase the utilization rate of ad hoc drivers, which means less system-wide distance. Secondly, the case study is kept simple for illustrative purposes. Although extensive sensitivity analyses have been conducted, data may not be sufficiently precise. More qualitative and quantitative studies investigating different concepts of crowdsourced delivery with various settings should be conducted to test the pros and cons of the crowdsourcing mode.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [Grant 72025103, 71831008, 72071173].

References

Agatz, N., Erera, A. L., Savelsbergh, M. W., Wang, X., 2011. Dynamic ride-sharing: A simulation study in metro atlanta. Procedia-Social and Behavioral Sciences 17, 532–550.

Amazon, 2019. Current amazon flex cities. (accessed on 17 September 2019). URL https://amazonflexbusiness.com/flex-cities.

Archetti, C., Savelsbergh, M., Speranza, M. G., 2016. The vehicle routing problem with occasional drivers. European Journal of Operational Research 254 (2), 472–480.

Arslan, A. M., Agatz, N., Kroon, L., Zuidwijk, R., 2018. Crowdsourced delivery–A dynamic pickup and delivery problem with ad hoc drivers. Transportation Science 53 (1), 222–235.

Baldacci, R., Bartolini, E., Mingozzi, A., 2011. An exact algorithm for the pickup and delivery problem with time windows. Operations Research 59 (2), 414–426.

Carbone, V., Rouquet, A., Roussat, C., 2017. The rise of crowd logistics: A new way to co-create logistics value. Journal of Business Logistics 38 (4), 238–252.

CCTV, 2019. Is it a little expensive to take a taxi during spring festival? didi has started to add the 'spring festival service fee', which will be paid in full to drivers. is that acceptable? (accessed on 3 October 2020).
URL `https://baijiahao.baidu.com/s?id=1624184727529237207&wfr=spider&for=pc`.

Cheah, S., Wang, S., 2017. Big data-driven business model innovation by traditional industries in the chinese economy. Journal of Chinese Economic and Foreign Trade Studies 10 (3), 229–251.

Cheng, G., Guo, D., Shi, J., Qin, Y., 2019. Smart city-wide package distribution using crowdsourced public transportation systems. IEEE Internet of Things Journal 6 (5), 7584–7594.

Chiang, W.-C., Russell, R., Xu, X., Zepeda, D., 2009. A simulation/metaheuristic approach to newspaper production and distribution supply chain problems. International Journal of Production Economics 121 (2), 752–767.

Coindreau, M.-A., Gallay, O., Zufferey, N., 2019. Vehicle routing with transportable resources: Using carpooling and walking for on-site services. European Journal of Operational Research 279 (3), 996–1010.

Cortés, C. E., Matamala, M., Contardo, C., 2010. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. European Journal of Operational Research 200 (3), 711–724.

Devari, A., Nikolaev, A. G., He, Q., 2017. Crowdsourcing the last mile delivery of online orders by exploiting the social networks of retail store customers. Transportation Research Part E: Logistics and Transportation Review 105, 105–122.

Didi, 2020a. Discussion about how does Didi Chuxing assign orders to drivers. (accessed on 15 June 2020).
URL http://mini.eastday.com/a/190915105925277.html.

Didi, 2020b. Homepage of Didi Chuxing. (accessed on 15 June 2020).
URL https://www.didiglobal.com/about-didi/about-us.

Didi, 2020c. How does Didi Chuxing assign orders to drivers ? (accessed on 15 June 2020).
URL http://www.didiabc.com/new/568.html.

Didi, 2020d. New passenger cancellation control scheme. (accessed on 15 June 2020).
URL http://www.didiabc.com/news/203.html.

DOL, 2019. State minimum wage laws. (accessed on 15 June 2019).
URL https://www.dol.gov/whd/minwage/america.htm.

Dupljanin, D., Mirkovic, M., Dumnic, S., Culibrk, D., Milisavljevic, S., Sarac, D., 2019. Urban crowdsourced last mile delivery: Mode of transport effects on fleet performance. International Journal of Simulation Modelling 18 (3), 441–452.

Frehe, V., Mehmann, J., Teuteberg, F., 2017. Understanding and assessing crowd logistics business models–using everyday people for last mile delivery. Journal of Business & Industrial Marketing 32 (1), 75–97.

Gschwind, T., Irnich, S., Rothenbächer, A.-K., Tilk, C., 2018. Bidirectional labeling in column-generation algorithms for pickup-and-delivery problems. European Journal of Operational Research 266 (2), 521–530.

Hosseinabadi, A. A. R., Vahidi, J., Balas, V. E., Mirkamali, S. S., 2018. Ovrp_gels: solving open vehicle routing problem using the gravitational emulation local search algorithm. Neural Computing and Applications 29 (10), 955–968.

Huang, K., Ardiansyah, M. N., 2019. A decision model for last-mile delivery planning with crowdsourcing integration. Computers & Industrial Engineering 135, 898–912.

Kafle, N., Zou, B., Lin, J., 2017. Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery. Transportation Research Part B: Methodological 99, 62–82.

Karaoglan, I., Altiparmak, F., Kara, I., Dengiz, B., 2012. The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach. Omega 40 (4), 465–477.

Kunze, O., 2016. Replicators, ground drones and crowd logistics a vision of urban logistics in the year 2030. Transportation Research Procedia 19, 286–299.

Li, H., Lim, A., 2003. A metaheuristic for the pickup and delivery problem with time windows. International Journal on Artificial Intelligence Tools 12 (02), 173–186.

Li, J., Zheng, Y., Dai, B., Yu, J., 2020. Implications of matching and pricing strategies for multiple-delivery-points service in a freight o2o platform. Transportation Research Part E: Logistics and Transportation Review 136, 101871.

Masson, R., Lehuédé, F., Péton, O., 2013. An adaptive large neighborhood search for the pickup and delivery problem with transfers. Transportation Science 47 (3), 344–355.

Masson, R., Trentini, A., Lehuédé, F., Malhéné, N., Péton, O., Tlahig, H., 2017. Optimization of a city logistics transportation system with mixed passengers and goods. EURO Journal on Transportation and Logistics 6 (1), 81–109.

Meituan-Dianping, 2020a. Frequently asked questions of Meituan-Dianping Takeout. (accessed on 15 June 2020).
URL `https://peisong.meituan.com/app/riderRecruitmentFusion/index?cityCode=100000&channelCode=bd60&recruitType=102#/info`.

Meituan-Dianping, 2020b. Media report of Meituan-Dianping Takeout. (accessed on

15 June 2020).
URL https://waimai.meituan.com/story?next_step=/newhome/news/list.

Morphy, E., 2013. About Walmart's idea to crowdsource its same-day delivery service. (accessed on 28 March 2019).
URL https://www.forbes.com/sites/erikamorphy/2013/03/28/about-walmarts-idea-to-crowdsource-its-same-day-delivery-service/#a7337675e5e7.

Pan, S., Chen, C., Zhong, R. Y., 2015. A crowdsourcing solution to collect e-commerce reverse flows in metropolitan areas. IFAC-PapersOnLine 48 (3), 1984–1989.

Polat, O., Kalayci, C. B., Kulak, O., Günther, H.-O., 2015. A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit. European Journal of Operational Research 242 (2), 369–382.

Rai, H. B., Verlinde, S., Merckx, J., Macharis, C., 2017. Crowd logistics: an opportunity for more sustainable urban freight transport? European Transport Research Review 9 (3), Article number 39.

Ren, S., Luo, F., Lin, L., Hsu, S.-C., Li, X. I., 2019. A novel dynamic pricing scheme for a large-scale electric vehicle sharing network considering vehicle relocation and vehicle-grid-integration. International Journal of Production Economics 218, 339–351.

Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation science 40 (4), 455–472.

Rougès, J.-F., Montreuil, B., 2014. Crowdsourcing delivery: New interconnected business models to reinvent delivery. Proceedings of the 1st International Physical Internet Conference, Québec, Canada, 1–19.

Russell, R., 2013. A constraint programming approach to designing a newspaper distribution system. International Journal of Production Economics 145 (1), 132–138.

Sadilek, A., Krumm, J., Horvitz, E., 2013. Crowdphysics: Planned and opportunistic crowdsourcing for physical tasks. Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media, Cambridge, Massachusetts, USA, 536–545.

Schrijver, A., 1986. Theory of Linear and Integer Programming. John Wiley & Sons, New York, USA.

Setzke, D. S., Pflügler, C., Schreieck, M., Fröhlich, S., Wiesche, M., Krcmar, H., 2017. Matching drivers and transportation requests in crowdsourced delivery systems. Proceedings of the Twenty-third Americas Conference on Information Systems, Boston, USA.

StatistaChina, 2019. China: retail e-commerce revenue forecast from 2017 to 2023. (accessed on 16 September 2019).

URL https://www.statista.com/statistics/246041/e-commerce-revenue-forecast-in-china/.

StatistaUS, 2019. E-commerce in the United States – statistics and facts. (accessed on 16 September 2019).
URL https://www.statista.com/topics/2443/us-ecommerce/.

Tang, W., Zhang, K., Ren, J., Zhang, Y., Shen, X. S., 2019. Privacy-preserving task recommendation with win-win incentives for mobile crowdsourcing. Information Sciences 527, 477–492.

Ting, C.-K., Liao, X.-L., 2013. The selective pickup and delivery problem: Formulation and a memetic algorithm. International Journal of Production Economics 141 (1), 199–211.

Uber, 2019. Payments and earnings. (accessed on 15 June 2019).
URL https://www.uber.com/en-GH/drive/resources/payments/.

Walmart, 2018. Walmart tests new last-mile grocery delivery service. (accessed on 17 September 2019).
URL https://corporate.walmart.com/newsroom/2018/09/05/walmart-tests-new-last-mile-grocery-delivery-service.

Wang, X., Agatz, N., Erera, A., 2018. Stable matching for dynamic ride-sharing systems. Transportation Science 52 (4), 850–867.

Wang, Y., Saksena, M., 1999. Scheduling fixed-priority tasks with preemption threshold. Proceedings of the Sixth International Conference on Real-Time Computing Systems and Applications, Hong Kong, 328–335.

Yu, V. F., Jewpanya, P., Redi, A. P., 2016. Open vehicle routing problem with cross-docking. Computers & Industrial Engineering 94, 6–17.

Yuan, Y., Chu, Z., Lai, F., Wu, H., 2020. The impact of transaction attributes on logistics outsourcing success: A moderated mediation model. International Journal of Production Economics 219, 54–65.