Joint Optimization of Incentive and Routing for Crowdsourced Last-mile Delivery

Dongze Li[1] and Fangni Zhang[1]

[1]Department of Data and Systems Engineering, The University of Hong Kong, Hong Kong, China

Short summary

This study presents a pioneering model, the Crowdsourced Incentive-Driven Vehicle Routing Problem with Time Windows (CS-IDVRPTW), designed to jointly optimize routing and incentive strategies for the crowdsourced last-mile delivery. This model uniquely considers the order acceptance probability of crowd-carriers, which is influenced by incentives and detour distances. The CS-IDVRPTW model is constructed as a Mixed Integer Nonlinear Programming (MINLP) problem with the objective of minimizing total expected delivery costs. The model is formulated into two versions: an arc-based formulation and a route-based formulation. Based on the route-based formulation, we propose a branch-and-price-and-cut (BPC) algorithm to obtain the exact solution. Extensive numerical studies validate the algorithm's ability to consistently yield optimal solutions with notable efficiency, outperforming commercial solvers such as Gurobi across various instances. Lastly, a sensitivity analysis is conducted, investigating the impact of crowd-carriers' order acceptance probability under different additional incentives and detour distance sensitivities.
Keywords: Branch-and-price-and-cut, Crowdshipping, Incentive design, Vehicle routing problem

1 Introduction

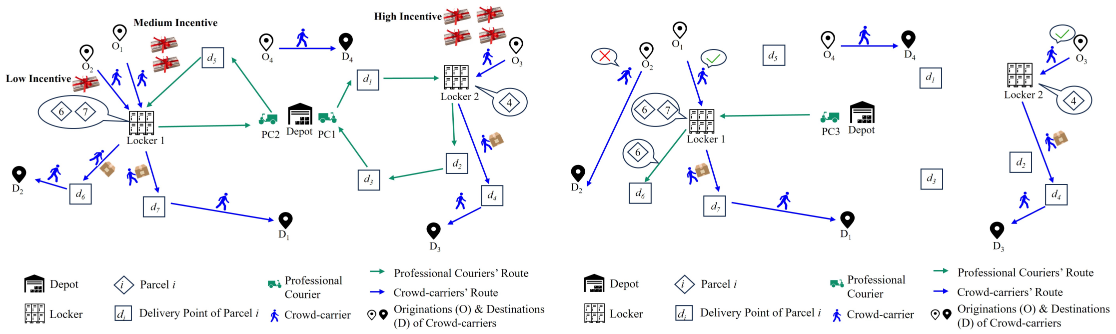
In the fast-paced era of e-commerce and on-demand economies, last-mile delivery has emerged as the pivotal yet most challenging aspect of logistics. Consumers demand ever-faster deliveries, while urban traffic congestion, escalating delivery costs, and mounting environmental concerns place immense pressure on traditional logistics systems. Amid this tension, crowdsourced delivery (crowdshipping) has emerged as a game-changing innovation. By leveraging the idle capacity of ordinary citizens' vehicles and their mobility, crowdshipping promises to revolutionize logistics by reducing costs, increasing flexibility, and enhancing delivery reach (Le et al., 2019; Pourrahmani & Jaller, 2021).

But this silver bullet is not without its caveats. While the crowdshipping model holds immense potential for utilizing underused societal resources, it faces significant hurdles: how can platforms attract and motivate casual participants to accept delivery tasks while maintaining efficiency and quality of service? Crowd-carriers often have other priorities, making their participation uncertain and variable (Hou et al., 2022; Mousavi et al., 2022). Without effective incentives and optimized delivery routes, platforms risk failed deliveries, inefficiencies, and spiraling costs.

This study addresses this critical gap by proposing a crowdsourced incentive-driven vehicle routing problem with time windows (CS-IDVRPTW). By integrating dynamic incentive mechanisms with sophisticated route planning, this study aims to balance cost efficiency, service reliability, and participant satisfaction, unlocking the true potential of crowdshipping in urban logistics.

Specifically, the CS-IDVRPTW model introduces a comprehensive framework that strategically allocates customer orders between professional couriers and crowd-carriers, enabling a collaborative approach to last-mile delivery. At the outset, all parcels and professional couriers are stationed at a central depot. The platform takes charge of assigning orders, predicting the acceptance probabilities of crowd-carriers based on their detour costs, and offering tailored incentives to motivate participation. Once decisions are made, professional couriers depart from the depot carrying assigned parcels. They not only deliver orders directly but also deposit parcels designated for crowd-carriers into strategically placed lockers. If crowd-carriers accept the platform's incentives and assigned tasks, they proceed from their starting locations to retrieve parcels from the lockers, transship them to the customers, and then continue to their intended destinations. Conversely, if crowd-carriers decline the orders, the platform must dispatch professional couriers from the depot to complete the delivery, incurring additional costs.

Fig.1 illustrates an example of the CS-IDVRPTW model within an urban network consisting of a depot, two lockers (Locker1, Locker2), and four crowd-carriers ($CS1, CS2, CS3, CS4$), with their respective starting and ending points labeled as $O1 - O4$ and $D1 - D4$. Fig.1(a) presents the order allocation, route planning, and incentive policies generated by the platform under the CS-IDVRPTW model, while Fig. 1(b) demonstrates the actual order acceptance by crowd-carriers after receiving the platform's assignments and incentives, as well as the subsequent parcel retrieval process in cases of order rejection. In Fig. 1(a), the delivery of seven parcels is depicted, with the platform assigning two professional couriers ($PC1, PC2$) and three crowd-carriers ($CS1, CS2, CS3$) to complete the tasks. Parcels 1-3 are directly delivered by $PC1$ to the customer delivery points, while parcel 4 is stored in Locker2 for transshipment by $CS3$. Similarly, parcel 5 is directly delivered by $PC2$, but parcels 6 and 7 are temporarily stored in Locker1 for subsequent transshipment by $CS2$ and $CS1$, respectively. To increase the likelihood of order acceptance, the platform offers additional incentives to each assigned crowd-carrier ($CS1, CS2, CS3$). The incentives are represented as monetary bundles, with $CS1$, $CS2$, and $CS3$ receiving medium, low, and high-level incentives, respectively. Fig. 1(b) shows the actual order acceptance by crowd-carriers following the platform's optimized order allocation and incentive decisions. It is observed that $CS2$ ultimately declines to transship parcel 6. As a result, the platform arranges for another professional courier ($PC3$) to retrieve parcel 6 from Locker1 and complete the delivery to the customer, incurring additional costs.



(a) Order allocation from the platform **(b)** Post-optimization stage

Fig. 1. An example of the CS-IDVRPTW model.

This study's contributions are threefold:

Firstly, we introduce a novel model CS-IDVRPTW. This model first jointly optimizes routing and incentive strategies in a crowdsourced last-mile delivery environment, addressing the critical gaps left by previous studies that focused solely on either routing or incentive mechanisms. The CS-IDVRPTW model also proposes a collaborative delivery structure that strategically allocates orders between professional couriers and crowd-carriers. This collaborative approach leverages the strengths of both types of carriers, minimizes operational costs, and optimizes service reliability, particularly in densely populated urban areas.

Secondly, the CS-IDVRPTW is formulated as a Mixed Integer Nonlinear Programming (MINLP) problem aimed at minimizing delivery costs. Given its NP-hard nature, a branch-and-price-and-cut (BPC) algorithm is proposed to obtain the exact solution of the model. Extensive numerical studies are conducted to demonstrate that the proposed algorithm consistently achieves optimal solutions with remarkable efficiency, surpassing the performance of commercial solvers such as Gurobi across a range of tested instances.

Thirdly, sensitivity analyses are conducted to explore the impact of varying crowd-carrier acceptance probabilities under different levels of sensitivity to detour distance and incentives. These analyses aim to investigate how total delivery costs are influenced by crowd-carrier acceptance probabilities in response to changes in these factors. The experimental results validate the effectiveness of the CS-IDVRPTW model across diverse scenarios, further extending its applicability and demonstrating its robustness in varying operational contexts.

2 Problem description

Problem settings

The CS-IDVRPTW is defined on a directed graph $G = (V, A)$, representing a network where n packages are transported from a depot to various delivery points. The vertex set V is partitioned as $V = V_C^O \cup V_C^D \cup V_T \cup V_R \cup \{0, n+1\}$. Here, V_C^O and V_C^D represent the origins and destinations of crowd-carriers, respectively. V_T denotes the set of lockers used for transshipment between professional couriers and crowd-carriers, and V_R represents all customer delivery requests, with n being the total number of requests. The nodes 0 and $n+1$ both represent the depot, where professional couriers depart from node 0 and return to node $n+1$. Furthermore, F represents the fleet of professional couriers at the depot responsible for package transportation. At the same time, C denotes the set of crowd-carriers who are distributed across the urban network and willing to participate in package delivery. The arc set $A = \{(i,j)|i,j \in V, i \neq j\}$ includes all directed arcs, each associated with a travel time t_{ij} and a travel cost c_{ij}.
All customer delivery requests are dispatched by the platform to both professional couriers and crowd-carriers, who jointly fulfill these requests. Professional couriers are responsible, according to the platform's dispatching decisions, for either delivering packages directly to the destination or transporting them to a locker, where crowd-carriers can complete the transshipment. Crowd-carriers receive delivery orders along with incentives from the platform and may choose to accept these orders based on their willingness. In the model, this acceptance decision is characterized by the acceptance probability $P(\mathcal{D}, \rho)$, which depends on the detour distance $\mathcal{D}$ for the crowd-carrier and the incentive ρ offered by the platform. The specific form of $P(\mathcal{D}, \rho)$ can be fitted based on historical big data from the logistics platform, and it typically satisfies $\frac{\partial P}{\partial \mathcal{D}} < 0$ and $\frac{\partial P}{\partial \rho} > 0$.
For any crowd-carrier $i \in C$, their transportation journey begins at the origin point $O_i \in V_C^O$ and ends at the destination point $D_i \in V_C^D$. If they accept the platform's order, they will detour to transfer packages from lockers to the requested destinations. It is important to note that, to enhance delivery efficiency, the platform does not know in advance whether crowd-carriers will accept their assigned orders once all professional couriers have departed from the depot for initial deliveries. This implies that orders stored in lockers but rejected by crowd carriers must be retrieved by professional couriers for subsequent delivery to customers, resulting in additional penalty costs.

Arc-based formulation

This section presents the arc-based formulation (ABF) of the CS-IDVRPTW model, detailed as follows:
The objective of the model is to minimize the total platform costs, which include the costs incurred by professional couriers (Z_{PC}), incentive costs for crowd-carriers (Z_{IC}), and the expected costs accounting for crowd-carriers' order acceptance probabilities ($\mathbb{E}[Z_{CS}]$). The objective function is defined as follows:

$$\min \quad Z_{PC} + Z_{IC} + \mathbb{E}[Z_{CS}] \tag{1a}$$

$$\text{where} \quad Z_{PC} = \sum_{k \in F} \sum_{j \in \{n+1\} \cup V_R \cup V_T} \sum_{i \in \{0\} \cup V_R \cup V_T} c_{ij} x_{ijk}, \tag{1b}$$

$$Z_{IC} = \sum_{k \in C} \sum_{p \in V_T} \sum_{i \in V_R} \rho_k \mathcal{D}_{pik} \mu_{pik}, \tag{1c}$$

$$\mathbb{E}[Z_{CS}] = \sum_{k \in C} \sum_{p \in V_T} \sum_{i \in V_R} \Big[P(\mathcal{D}_{pik}, \rho_k) \rho_0 \mathcal{D}_{pik} + \big(1 - P(\mathcal{D}_{pik}, \rho_k)\big)\big(c_{0p} + c_{pi} + c_{i,n+1} - \rho_k \mathcal{D}_{pik}\big) \Big] \mu_{pik}. \tag{1d}$$

$$\min \quad \sum_{k \in F} \sum_{j \in \{n+1\} \cup V_R \cup V_T} \sum_{i \in \{0\} \cup V_R \cup V_T} c_{ij} x_{ijk} + \sum_{k \in C} \sum_{p \in V_T} \sum_{i \in V_R} \Big[P(\mathcal{D}_{pik}, \rho_k)(\rho_0 + \rho_k)\mathcal{D}_{pik} + \big(1 - P(\mathcal{D}_{pik}, \rho_k)\big)\big(c_{0p} + c_{pi} + c_{i,n+1}\big) \Big] \mu_{pik}, \tag{2a}$$

$$\text{where} \quad \mathcal{D}_{pik}(\mu_{pik}) = \big(c_{O_k p} + c_{pi} + c_{iD_k} - c_{O_k D_k}\big)\mu_{pik}, \forall p \in V_T, \forall i \in V_R, \forall k \in C. \tag{2b}$$

$$\min \sum_{k\in F}\sum_{j\in\{n+1\}\cup V_R\cup V_T}\sum_{i\in\{0\}\cup V_R\cup V_T} c_{ij}x_{ijk} + \sum_{p\in V_T}\sum_{i\in V_R}\sum_{k\in C} A(\mu_{pik},\rho_k)\mu_{pik}, \tag{3a}$$

$$\text{where} \quad A(\mu_{pik},\rho_k) = \big[(\rho_0+\rho_k)\big(c_{O_kp}+c_{pi}+c_{iD_k}-c_{O_kD_k}\big) - \big(c_{0p}+c_{pi}+c_{i,n+1}\big)\big] \cdot P(\mathcal{D}_{pik}(\mu_{pik}),\rho_k) + \big(c_{0p}+c_{pi}+c_{i,n+1}\big). \tag{3b}$$

c_{ij} is the travel cost for a professional courier along arc $(i,j)\in A$. Let x_{ijk} be a binary variable that takes the value 1 if the professional courier k travels along arc $(i,j)\in A$, and 0 otherwise. Therefore, Eq. (1b) calculates the costs incurred by professional couriers. Also, let μ_{pik} be a binary variable that takes the value 1 if crowd-carrier k is assigned to a locker p to pick up a package of customer request i, and 0 otherwise. ρ_k, as another decision variable, represents the additional incentive rate per unit of detour distance for crowd-carrier k. $\mathcal{D}_{pik}$ is the detour distance for crowd-carrier k to tranship the package from locker p to customer request i. Then Eq. (1c) calculates the total additional incentives offered to all crowd-carriers. In Eq. (1d), $P(\mathcal{D}_{pik},\rho_k)$ is the order accept probability for crowd-carrier k under detour distance $\mathcal{D}_{pik}$ and incentive rate ρ_k. ρ_0 is the basic incentive rate per unit of detour distance for crowd-carriers. Therefore, $\rho_0\mathcal{D}_{pik}$ calculates the initial compensation offered to crowd-carrier k if he transships the package of customer i from locker p. $c_{0p}+c_{pi}+c_{i,n+1}$ is the extra cost to retrieve the package from the locker for subsequent delivery to the customer if the offer is rejected, and crowd-carriers will also return the additional incentives $\rho_k\mathcal{D}_{pik}$ to the platform in this case. By substituting Eq. (1b), Eq. (1c), and Eq. (1d) into Eq. (1a), we obtain the expanded objective function Eq. (2a) in the nonlinear form. In addition, Eq. (2b) indicates $\mathcal{D}_{pik}$ is a function of μ_{pik}. Therefore, Eq. (2a) can also be expressed as a combination of Eq. (3a) and Eq. (3b). It is important to note that, since μ_{pik} is a binary variable, we have $\mu_{pik}^2=\mu_{pik}$.
The objective function is subject to several constraints, including flow conservation constraints, capacity constraints, time window constraints, crowd-carrier and professional courier collaboration constraints, and some equations and inequalities to determine the task allocations.

Route-based formulation

To more efficiently solve the problem using the BPC algorithm, this section reformulates the CS-IDVRPTW as a route-based formulation (RBF), where route costs arise from the travel of professional couriers and the allocation of crowd carriers' orders along with associated incentives. As indicated in Eq. (3a), $\sum_{p\in V_T}\sum_{i\in V_R}\sum_{k\in C} A(\mu_{pik},\rho_k)$ can be interpreted as the cost incurred by crowd-carriers for delivery, encompassing both travel expenses and platform incentives.
Let R denote the set of feasible routes. Associate each route $r\in R$ with a binary decision variable λ_r, which indicates whether route r is selected. Let ϕ_{ijr} be a binary variable specifying whether route r travels arc $(i,j)\in A$, and let z_{pikr} be a binary variable indicating whether a crowd-carrier $k\in C$ transships the package in the locker $p\in V_T$ for the customer request $r\in V_R$ on route r. It is important to note that, $\sum_{r\in R}\mu_{pik}=z_{pikr}, \forall p\in V_T, i\in V_R, k\in C$. Therefore, the cost of a route r is calculated by

$$c_r = \sum_{i\in V_C\cup V_L\cup\{0\}}\sum_{j\in V_C\cup V_L\cup\{n+1\}} c_{ij}\phi_{ijr} + \sum_{p\in V_T}\sum_{i\in V_R}\sum_{k\in C} A(z_{pikr},\rho_k)z_{pikr} \tag{4}$$

In addition, let θ_{ir} be a binary parameter that takes value 1 if either a customer request or locker $i\in V_R\cup V_T$ is directly visited in route r. The RBF is presented as follows:

$$\min \sum_{r\in R} c_r\lambda_r \tag{5}$$

$$\text{s.t.} \sum_{r\in R}(\theta_{ir} + \sum_{p\in V_T}\sum_{k\in C} z_{pikr})\lambda_r = 1, \forall i\in V_R, \tag{6}$$

$$\sum_{r\in R}\sum_{p\in V_T}\sum_{i\in V_R} z_{pikr}\lambda_r \le 1, \forall k\in C, \tag{7}$$

$$\sum_{r\in R}\lambda_r \le \overline{F}, \tag{8}$$

$$\lambda_r \in \{0, 1\}, \forall r \in R \quad (9)$$

Objective (5) minimizes the total delivery cost incurred by both professional couriers and crowd-carriers for the selected routes. Constraints (6) ensure that each delivery request is visited exactly once. Constraints (7) restrict each crowd-carrier to be assigned at most one order. Constraints (8) limit the number of vehicles available for use. Constraints (9) define the domains of the decision variables.

3 Methodology

Fig. 2 illustrates the flowchart of the proposed BPC algorithm. The BPC algorithm addresses each node of the branch-and-bound tree by embedding a column generation procedure to compute lower bounds through solving linear relaxations. Subsequently, additional subset row cuts (SRCs) are incorporated to strengthen these lower bounds.

As the core of the algorithm, column generation is an iterative procedure that addresses the linear relaxation of the route-based model (5)-(9), referred to as the linear master problem. The process begins with an initial set of columns that ensures model feasibility by covering all customer requests. At each iteration, a restricted linear master problem (RLMP), consisting of a subset of route variables $R' \in R$ is solved using the simplex method. This produces dual solutions, which serve as inputs for the pricing problem. The reduced costs of edge travel and crowd-carrier assignments are then updated based on the dual solutions. A comprehensive bi-directional labeling algorithm is employed to solve the pricing problem, identifying new columns with negative reduced costs. To enhance computational efficiency, an upper limit, sol_{max}, is set on the number of columns generated per iteration. Optimality is confirmed when all labels are generated, and no additional column with a negative reduced cost is identified. Subsequently, SRCs are added to further improve the lower bounds. If the optimal solution is non-fractional, the algorithm updates the upper bound for the CS-IDVRPTW and checks the termination condition. Otherwise, branching strategies are applied to ensure the algorithm ultimately yields an optimal integer solution.

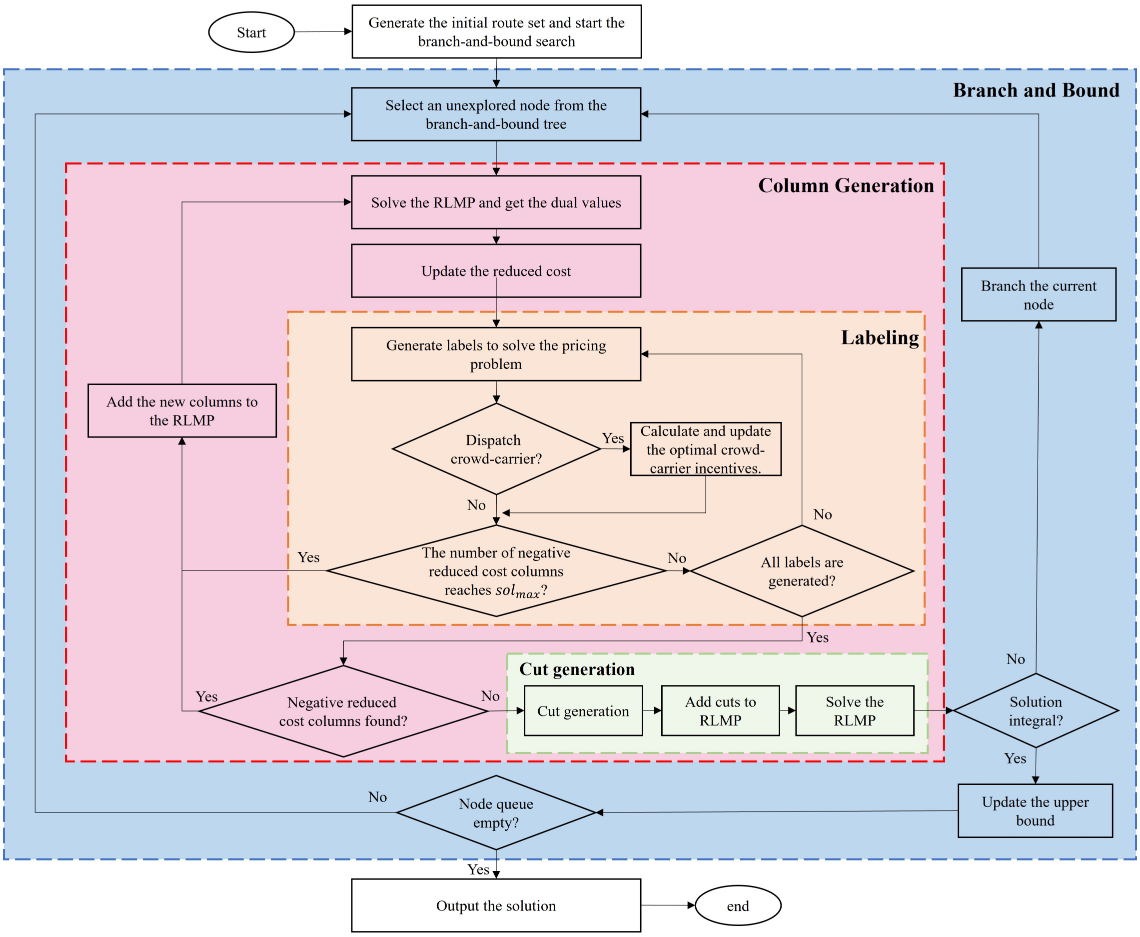


Fig. 2. The flowchart of the proposed branch-and-price-and-cut (BPC) algorithm

4 Results and discussion

In this section, we extensively test the proposed BPC to solve the CS-IDVRPTW problem, demonstrating its efficacy. All experiments were conducted on a computer outfitted with an AMD Ryzen 7 5800H CPU operating at 3.2 GHz and equipped with 16 GB of RAM.

Given that $P(\mathcal{D}, \rho)$ is estimated using historical data from a logistics platform, obtaining such comprehensive data directly from real-world operations presents significant challenges. To overcome this and to accurately capture the relationship between the acceptance probability P, the detour distance $\mathcal{D}$, and the platform incentive ρ, we formulated $P(\mathcal{D}, \rho)$ using a sigmoid function, as shown in Eq.(10).

Tables 1 present a comparative analysis of the computational outcomes produced by three algorithms: Gurobi, the Branch-and-Price (BP) algorithm, and the proposed BPC algorithm. These algorithms were applied to Solomon's dataset, specifically cases "r101" to "r112". The case names are appended with suffixes 'S' and 'M', which represent small- and medium-scale instances, respectively. For the small-scale instances, the scenarios incorporate 10 customers, 3 lockers, and 5 crowd-carriers. Meanwhile, the medium-scale instances consider 25 customers, 5 lockers, and 10 crowd-carriers. It should be noted that the positions of the lockers and the starting and ending points for crowd-carriers are generated randomly. In addition, the results displayed in Table 1 have been obtained by setting the parameter values of α and β at 0.02 and 4, respectively.

$$P(\mathcal{D}, \rho) = \frac{1}{1 + e^{\alpha \mathcal{D} - \beta \rho}} \tag{10}$$

Table 1 illustrates that both the BP and the BPC algorithm are capable of determining the exact optimal solution for all instances. More specifically, for small-scale instances, the proposed BPC algorithm is nearly 110 times faster on average than Gurobi in deriving the optimal solutions. For medium-scale instances, Gurobi fails to find the optimal solution within 10 hours for all instances and the average gap is 16.95%. In contrast, the proposed BPC algorithm can derive the optimal solution in an average time of 795.733 seconds, thereby significantly enhancing the computational

Table 1 Experiment results

Cases	Gurobi				BP			BPC		
	Time (s)	LB	UB	Gap	Time (s)	LB	UB	Time (s)	LB	UB
R101_S	365.33	238.39	238.39	0.00%	0.51	238.39	238.39	0.46	238.39	238.39
R102_S	857.57	217.39	217.39	0.00%	7.52	217.39	217.39	9.42	217.39	217.39
R103_S	904.94	217.39	217.39	0.00%	7.76	217.39	217.39	8.90	217.39	217.39
R104_S	227.33	198.21	198.21	0.00%	2.30	198.21	198.21	2.19	198.21	198.21
R105_S	264.83	237.78	237.78	0.00%	0.99	237.78	237.78	1.24	237.78	237.78
R106_S	246.63	208.81	208.81	0.00%	2.28	208.81	208.81	1.31	208.81	208.81
R107_S	243.09	208.81	208.81	0.00%	2.12	208.81	208.81	1.52	208.81	208.81
R108_S	543.84	198.21	198.21	0.00%	8.53	198.21	198.21	10.34	198.21	198.21
R109_S	181.76	219.98	219.98	0.00%	5.21	219.98	219.98	3.53	219.98	219.98
R110_S	249.48	212.91	212.91	0.00%	3.36	212.91	212.91	1.77	212.91	212.91
R111_S	242.94	201.03	201.03	0.00%	3.05	201.03	201.03	3.18	201.03	201.03
R112_S	744.00	198.21	198.21	0.00%	13.39	198.21	198.21	2.21	198.21	198.21
Average	422.65			0.00%	4.75			3.84		
R101_M	>36000	511.41	543.52	5.88%	2020.56	543.52	543.52	507.18	543.52	543.52
R102_M	>36000	385.16	471.49	18.31%	250.58	459.32	459.32	194.65	459.32	459.32
R103_M	>36000	334.22	410.21	18.52%	383.54	410.21	410.21	346.14	410.21	410.21
R104_M	>36000	316.37	388.14	18.49%	786.34	388.14	388.14	1221.56	388.14	388.14
R105_M	>36000	430.29	491.11	12.38%	564.04	486.33	486.33	271.86	486.33	486.33
R106_M	>36000	362.67	418.13	13.26%	367.31	418.13	418.13	235.50	418.13	418.13
R107_M	>36000	325.99	400.53	18.61%	4660.02	399.08	399.08	2781.04	399.08	399.08
R108_M	>36000	314.66	379.53	17.09%	4370.33	377.02	377.02	1879.90	377.02	377.02
R109_M	>36000	357.58	433.80	17.57%	1494.14	426.10	426.10	1294.47	426.10	426.10
R110_M	>36000	305.31	417.88	26.94%	347.14	401.50	401.50	252.26	401.50	401.50
R111_M	>36000	333.33	400.56	16.78%	1305.42	399.17	399.17	436.73	399.17	399.17
R112_M	>36000	302.19	375.16	19.50%	286.54	394.10	394.10	127.50	394.10	394.10
Average	>36000			16.95%	1403.00			795.73		

Note: **LB**: Lower Bound. **UB**: Upper Bound. **Gap**=(UB-LB) / UB · 100%

speed.

Furthermore, we observe that although the BP algorithm without cuts can also notably improve the computational speed compared to Gurobi, it is still slower than the BPC algorithm. This discrepancy becomes more pronounced as the size of the instances increases. The inclusion of cuts in the algorithm, despite making the resolution of the RLMP more complex, effectively elevates the lower bound of the solution, thus further reducing the search range.

In addition to the above, we have conducted experiments under different α and β values. Due to space limits, the results are not shown here. We found that the smaller the α/β value, the more orders will be assigned to the crowd-carriers, and the lower the overall expected cost of delivery.

5 Conclusions

The CS-IDVRPTW model proposed in this study is a pioneering attempt to jointly optimize routing and incentive strategies in a crowdsourced last-mile delivery setting. By reasonably estimating the influence of additional incentives and detour distances on the order acceptance probability of crowd-carriers, the platform can effectively allocate orders to professional couriers and crowd-carriers. Additionally, it can provide suitable incentives to crowd-carriers, thereby reducing the total expected delivery cost of the platform.

To solve the CS-IDVRPTW model, a self-designed BPC algorithm is introduced. According to the results of computational experiments, the BPC algorithm significantly accelerates the computational speed compared to the Gurobi solver.

Furthermore, we conducted a sensitivity analysis based on the order acceptance probability of crowd-carriers under different additional incentives and detour distance sensitivities. We found that when crowd-carriers are more sensitive to incentives and less sensitive to detour distances, the platform will assign more orders to them, and the total delivery cost will be lower.

Acknowledgements

The work described in this paper was partially supported by the National Natural Science Foundation of China (No. 72101222, 72371216) and the Research Grants Council of Hong Kong (No. 27202221, 17205124, T32-707-22).

References

Hou, S., Gao, J., & Wang, C. (2022). Optimization framework for crowd-sourced delivery services with the consideration of shippers' acceptance uncertainties. *IEEE Transactions on Intelligent Transportation Systems*, *24*(1), 684–693.

Le, T. V., Stathopoulos, A., Van Woensel, T., & Ukkusuri, S. V. (2019). Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence. *Transportation Research Part C: Emerging Technologies*, *103*, 83–103.

Mousavi, K., Bodur, M., & Roorda, M. J. (2022). Stochastic last-mile delivery with crowd-shipping and mobile depots. *Transportation Science*, *56*(3), 612–630.

Pourrahmani, E., & Jaller, M. (2021). Crowdshipping in last mile deliveries: Operational challenges and research opportunities. *Socio-Economic Planning Sciences*, *78*, 101063.