GRIFFITH UNIVERSITY

Multi-stage two-echelon crowdsourcing logistics assignment model with future committing drivers

Author

Li, MY, Zhang, WS, Wu, Y, Wang, L

Published

2023

Journal Title

Journal of Cleaner Production

Version

Version of Record (VoR)

DOI

10.1016/j.jclepro.2023.139397

Rights statement

© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Downloaded from

http://hdl.handle.net/10072/426909

Griffith Research Online

https://research-repository.griffith.edu.au

Journal of Cleaner Production 428 (2023) 139397

Contents lists available at ScienceDirect

Journal of Cleaner Production

journal homepage: www.elsevier.com/locate/jclepro

Multi-stage two-echelon crowdsourcing logistics assignment model with future committing drivers☆

Ming-Yang Li [a], Wen-Song Zhang [a], Yong Wu [b,*], Lei Wang [c]

[a] Business School, Faculty of Economics, Liaoning University, Shenyang, China
[b] Department of Business Strategy and Innovation, Griffith University, Gold Coast Campus, QLD 4222, Australia
[c] School of Public Administration, Dongbei University of Finance and Economics, Dalian, 116025, China

ARTICLE INFO

Handling Editor: Govindan Kannan

Keywords:
Logistics
Low-carbon
Crowdsourcing
Multi-stage
Dynamic task assignment
Hybrid algorithm

ABSTRACT

With the popularity of low-carbon economy, governments of all countries have introduced energy-saving and emission reduction policies. Crowdsourcing logistics, a logistics assignment model developed under the sharing economy, can effectively use transport resources from the public and thus meaningfully contribute to sustainability. It has important practical significance both for energy saving and emission reduction in last-mile logistics distributions, and hence plays an important role in tackling the challenges associated with last-mile and same-day deliveries. This paper investigated the crowdsourcing logistics task assignment problem where logistics platforms can perform multi-stage task assignment. The information of all tasks to be delivered in each stage is known *a priori*, however the information of crowdsourcing drivers in future stages is not known completely. A multi-stage two-echelon dynamic task assignment model (MS-2E-DAM) was developed for the problem and a heuristic which combines genetic algorithm and tabu search (GA-TS) was developed. Its performance was benchmarked with CPLEX 12.10 for small-size problems and the results demonstrated the effectiveness of the proposed heuristic approach. For large-size problems, the proposed approach can help reduce the overall cost by 1.94% over traditional assignment approaches. Sensitivity analyses on three key parameters helped identify the key factors that affect the system cost, and several management suggestions were proposed based on the results.

1. Introduction

With the intensification of the greenhouse effect, climate change has become one of the most serious challenges facing humankind today (Hashemi, 2021; Li et al., 2018). In the meantime, online shopping is getting more popular with the development of information technology and the change of consumption patterns, leading to an unprecedented requirement of services and increased carbon emissions (Liu et al., 2019). EMarketer (2019) estimated that the global online shopping expenditure would rise from 14.1% of the global total retail sales ($3.535 trillion) in 2019 to 22% by 2023. To meet the corresponding logistics needs, a large deal of road transport is used, leading to the rise of logistics companies, traffic congestion, air pollution, fuel and energy crises (Galkin et al., 2021; Strulak-Wójcikiewicz and Wagner, 2021).

Therefore, there is a need to find solutions that would promote not only economic development but also environmental and social sustainability (Strulak-Wójcikiewicz and Wagner, 2021). Crowdsourcing logistics, as a form of sharing economy, could offer a solution to these challenges (Suh et al., 2012; Mladenow et al., 2016; Modaresnezhad et al., 2020). Based on the crowdsourcing logistics concept, logistics companies can partially rely on resources from the general public for parcel delivery. For example, companies like Amazon, Walmart, DHL etc. have used crowdsourcing logistics for same day delivery and last-mile delivery (Arslan et al., 2019). Crowdsourcing logistics helps logistics service providers address delivery demand fluctuations, reduce number of self-owned delivery vehicles, cut operating costs, and improve last-mile delivery service.

Given the importance of using crowdsourcing logistics, this paper proposes a multi-stage two-echelon dynamic task assignment model

☆ Funding: This work was partly supported by the Humanities and Social Science Foundation of Ministry of Education of China (No. 19YJA630037), and the National Natural Science Foundation of China (No. 71974067).

* Corresponding author.
E-mail addresses: lmy_neu@163.com (M.-Y. Li), zhangwensong123@yeah.net (W.-S. Zhang), yong.wu@griffith.edu.au (Y. Wu), leonwang521@gmail.com (L. Wang).

https://doi.org/10.1016/j.jclepro.2023.139397
Received 3 January 2023; Received in revised form 11 October 2023; Accepted 17 October 2023
Available online 19 October 2023
0959-6526/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

(MS-2E-DAM) to minimize the overall system cost. In the proposed model, a rolling horizon assignment method is adopted for the crowdsourcing logistics task assignment. The model combines the information of crowdsourcing drivers and tasks in all stages and attempts to obtain the optimal task assignment results. At each stage, the task assignment first determines the currently known assignable drivers in all stages, then calculates the driver's delivery remuneration based on multiple factors such as the detour distance, the number and the volume of tasks (used interchangeably with 'parcels' in this paper). The problem is formulated as a cost minimization one and a hybrid heuristic algorithm using Genetic Algorithm and Tabu Search (GA-TS) is constructed. The model proposed in this paper can effectively reduce assignment costs, make better use of the capacity of crowdsourcing logistics drivers, and provide an effective solution to the problem of crowdsourcing logistics dynamic task assignment.

The proposed MS-2E-DAM approach in this paper has economic, environmental, and social contributions to logistics platforms, crowdsourcing drivers, as well as the society:

1. For logistics companies, crowdsourcing logistics can balance operating cost and equipment utilization with fluctuating demand for logistics service. In addition, crowdsourcing logistics provides important competitive advantages for logistics companies, such as reduction in costs, improvement in speed and responsiveness, reduction in lead times, and increase in flexibility and so on (Bathke and Münch, 2023).
2. For crowdsourcing drivers, they can earn some extra money (Arslan et al., 2019), which offsets their cost of driving.
3. For the society, crowdsourcing logistics can promote conservation of natural resources (especially oil), reduce traffic congestion and air pollution, and improve utilization of social resources (Paloheimo et al., 2016; Arslan et al., 2019; Gatta et al., 2019; Barbosa et al., 2023).

The rest of this paper is organized as follows. Section 2 reviews the related crowdsourcing logistics research. Section 3 provides a conceptual description of the proposed problem, followed by its mathematical formulation in Section 4. Section 5 introduces the GA-TS algorithm to solve the model. Section 6 conducts numerical experiments and sensitivity analyses to demonstrate the rationality and effectiveness of the proposed approach. Section 7 summarizes the paper and discusses future research directions.

2. Literature review

Mehmann et al. (2015) reviewed the status of crowdsourcing logistics and discussed its applications. Since then, scholars have conducted research on crowdsourcing logistics from multiple perspectives. These included research on the concept, the impact, and the significance of crowdsourcing logistics (Mladenow et al., 2016; Carbone et al., 2017; Frehe et al., 2017), on factors that affect the decision-making of services assignment (Buldeo Rai et al., 2017; Ermagun and Stathopoulos, 2018; Punel et al., 2018; Le and Ukkusuri, 2019; Bathke and Münch, 2023), on task assignment methods of crowdsourcing logistics (Archetti et al., 2016; Wang et al., 2016; Devari et al., 2017; Kafle et al., 2017; Akeb et al., 2018; Behrend and Meisel, 2018; Allahviranloo and Baghestani, 2019; Dahle et al., 2019; Dayarian and Savelsbergh, 2020; Macrina et al., 2020), and on pricing and compensation schemes for crowdsourcing drivers (Le et al., 2021).

This paper belongs to one of the key problems in crowdsourcing logistics research: the arrangement of pickups and deliveries. This has been mostly studied in two forms: single- and two-echelon systems (Kafle et al., 2017). In what follows, this section briefly reviews the relevant literature on the two forms.

2.1. Single-echelon systems

Single-echelon systems are suitable for urban areas of small sizes. It considers crowdsourcing drivers picking up tasks in city distribution center (or depot) and then delivering tasks directly to customers (Kafle et al., 2017).

Research on the single-echelon problem appeared more in the literature. For example, Archetti et al. (2016) developed a task assignment model and designed a multi-start heuristic to obtain the best feasible solution. Wang et al. (2016) transformed the problem of large-scale mobile crowd task assignment into a network min-cost flow problem, and proposed a pruning technique that can significantly reduce the network scale. Aimed at reducing assignment costs and total emissions, and ensuring the speed and reliability of parcel distribution, Devari et al. (2017) tackled the last-mile delivery problem using information from customers' social networks. Behrend and Meisel (2018) combined item sharing and crowdsourcing to construct a platform-based decision-making model which was solved by a heuristic algorithm.

Allahviranloo and Baghestani (2019) built a dynamic optimization model to study the pickup/delivery activities between the carrier and the requester in the peer-to-peer crowdsourcing model, and verified the model with data from Los Angeles and Orange Counties. Arslan et al. (2019) researched the dynamic assignment problem of tasks and drivers, proposed a dynamic rolling horizon framework to assign parcels to ad-hoc drivers, and developed an exact solution algorithm based on matching formulas. Dahle et al. (2019) considered the detour cost of crowdsourcing drivers for one or more tasks, designed three compensation schemes, and used an optimization model to solve the optimal assignment scheme. Dayarian and Savelsbergh (2020) considered a highly dynamic and stochastic same-day delivery environment and developed two rolling horizon dispatching approaches, one of which incorporates probabilistic information about future online orders and in-store customer arrivals. Seghezzi and Mangiaracina (2022) analyzed the impact of multi-parcel crowdsourcing logistics on delivery costs compared to the traditional approach and applied their analytical model to a case in Milan, Italy. Barbosa et al. (2023) assumed that not all the proposed delivery tasks will necessarily be accepted by crowdsourcing drivers and applied logistic regression to model their willingness to undertake a delivery as part of a new dynamic compensation scheme.

2.2. Two-echelon systems

Compared with single-echelon systems, two-echelon systems have satellite facilities (or sub-depots) to relay the parcels between city distribution center and customers (Kafle et al., 2017). There are a few scholars who have conducted research on the two-echelon systems. For example, Akeb et al. (2018) proposed neighbor relays as a collaborative logistics distribution solution in dense urban environments to deal with the cases when the customers are not at home. Kafle et al. (2017) considered trucks as a mobile parcel transfer point (sub-depot) while observing the delivery time constraints as well as the time matching between the trucks and crowdsourcing drivers. Macrina et al. (2020) built a mixed integer programming model which allows the crowdsourcing drivers to pick up tasks from their chosen depot and solved the problem through a metaheuristic. Feng et al. (2021) investigated an integrated production and transportation scheduling problem. They formulated the problem into a mixed-integer linear program and proved its strong NP-hardness. A Genetic Algorithm (GA) and a lower bound were then developed to solve it. Sampaio et al. (2020) analyzed the potential benefits of transfers in pickup and delivery operations in urban areas and considered a multi-depot pickup and delivery problem with time windows and transfers. They developed an adaptive large neighborhood search algorithm which effectively identifies transfer opportunities and synchronizes driver operations. Voigt and Kuhn (2022) assumed that the crowdsourcing logistics platform uses trust-generating mechanisms

Table 1
Feature comparison between this paper and key references.

Reference	Uncertain crowdsourcing drivers	Different task sizes	Two or moreechelons	Dynamic
Dayarian and Savelsbergh (2020)	Yes	No	No	Yes
Zhen et al. (2021)	No	No	No	No
Ahamed et al. (2021)	No	Yes	No	No
Janjevic et al. (2021)	No	Yes	Yes	No
Kong et al. (2021)	Yes	No	No	Yes
Kafle et al. (2017)	No	Yes	Yes	No
Archetti et al. (2016)	No	No	Yes	No
Lan et al. (2022)	No	Yes	Yes	No
This paper	Yes	Yes	Yes	Yes

to increase the reliability of crowdsourcing drivers and transshipment points to better integrate them. A solution approach based on the adaptive large neighborhood search was then developed to solve the problem.

Two-echelon systems are usually adopted for large cities as multiple depots can cover large areas more easily. In reality, a crowdsourcing logistics platform could have multiple task assignments at different times during a day, at which points sets of tasks are allocated to available crowdsourcing drivers. The duration between two assignment times is referred as a 'stage' in this paper. In multi-stage problems, it is obvious that the task assignment results of one stage could affect all subsequent stages. In existing two-echelon system approaches, however, each stage of task assignment is considered separately rather than combining the drivers and tasks information of all stages into a comprehensive assignment problem. Therefore, a holistic view across multiple stages would be necessary in order to offer the best possible solutions.

Although the research of crowdsourcing logistics has advanced in recent years, there are still different aspects that worth further investigation. For example, there is relative less research on two-echelon dynamic assignment problems. There also lacks research which considers uncertain commitment of crowdsourcing drivers. Furthermore, no research has combined the drivers and tasks information of all stages in the same day into a comprehensive assignment problem. To fill these gaps, the MS-2E-DAM approach is proposed in this paper. Table 1 highlights the innovative aspects offered by this paper and compares its key features with those related from the literature.

3. Problem definition and model setup

3.1. Problem definition

In the crowdsourcing logistics MS-2E-DAM problem, the number of tasks (parcels to be delivered, referred to as tasks hereafter) is known *a priori*, but the drivers participating in crowdsourcing are not *fully* known in advance. That is, the information of all the drivers who have announced to participate in the current stage assignment is known. Information about *some* drivers in future stages is also known. However, there might be additional drivers in future stages whose information is not known yet. The crowdsourcing logistics platform (the platform) will conduct multi-stage task assignments, depending on the tasks' target delivery times and priorities. Given this, tasks in the current stage need not to be completely assigned, as some tasks could be reserved for more suitable drivers in later stages.

In this problem, tasks are released stage-wise for crowdsourcing drivers to bid. Based on the bidding and assignment results, the tasks will be transported by the platform's own truck from depot to sub-depots for crowdsourcing drivers to pick up. These drivers need to arrive their destinations on time and therefore have time window restrictions. At each stage, a task can only be assigned to one driver; however, a driver can choose to deliver multiple tasks at the same time.

Fig. 1 illustrates the assignment process. It is assumed that there are three stages on the day, and the current stage is the first stage. Note that the current stage changes over time as the assignment is rolling. In the first stage, the information of drivers and tasks in all stages (i.e., all known driver and task information of the first, second and third stages) will be used for assignment. It is noted that only the assignment in the current stage (the first stage) is actually assigned, i.e., some tasks are assigned to drivers in the current stage and the rest tasks are left to the next stages. The assignment of tasks in later stages is only pre-assignment, and it may be changed due to the participation of new drivers. For example, in Fig. 1, there is one new driver in the second stage and two new drivers in the third stage. These new drivers might change the pre-assignment of tasks from previous stages when MS-2E-DAM rolls to its corresponding stages.

3.2. Sets, parameters, and variables

The MS-2E-DAM is specified on an undirected, complete graph $G = (V, E)$ where node set V is the combination of: (i) the set of delivery customers P; (ii) the set of depot and sub-depots B; (iii) the set of crowdsourcing drivers $N = N_A \cup N_F$, where N_A is the set of origin nodes of crowdsourcing drivers and N_F is the set of destination nodes of crowdsourcing drivers. The edge set E corresponds to links connecting the nodes in V. For the convenience of description, the following sets and parameters, decision variables are used:

Sets and parameters:

- D: the set of crowdsourcing drivers.
- T: Maximum number of assignment stages.
- Q^d: Maximum loading capacity of driver d.
- S^t: Assignment time of stage t.
- a^d: Announcement time of driver d participating in assignment bidding.
- e^d: Earliest departure time of driver d.
- l^d: Latest departure time of driver d.
- $\overline{l^d}$: Latest arrival time of driver d.
- γ_{uv}: The travel distance from node u to node v.
- s_b: The travel speed of the depot truck.
- s_d: The travel speed of crowdsourcing drivers.
- L^d: The original route distance when driver d does not participate in assignment.
- λ: Detour compensation per kilometer for a driver's additional travel distance.
- η: The cost per kilometer for the depot truck.
- α: Compensation per unit volume of task for a driver who carries freight.
- β: Compensation per task for a driver who carries freight.
- ∂: Compensation for uncompleted tasks to customers, that is, compensation provided for uncompleted tasks on the day.

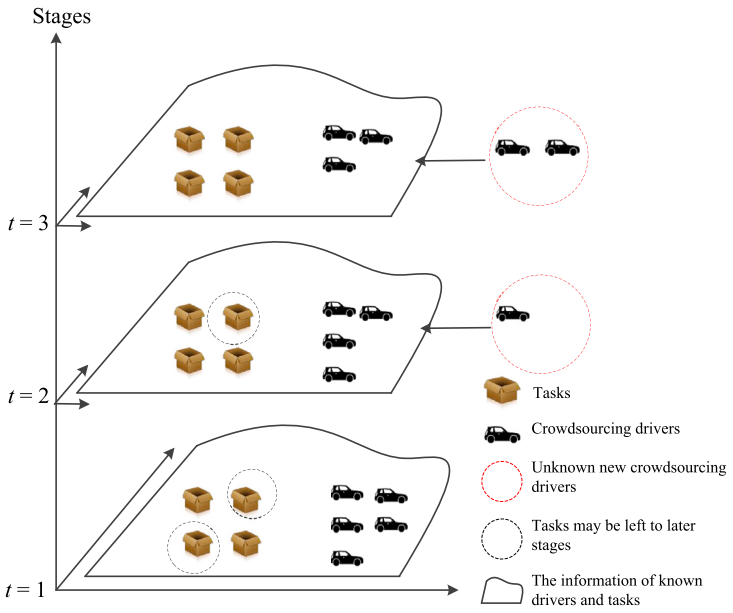

Fig. 1. An illustration of tasks assignment process.

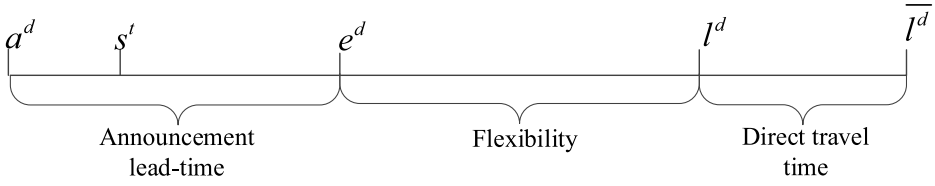

Fig. 2. Driver's time window for participating stage t assignment.

Decision variables:

- δ_b^t: Arrival time of the truck at depot b in stage t. This variable is initiated to the stage starting time.
- ω_b^{dt}: Arrival time of driver d at depot b in stage t.
- μ_v: The order in which the driver or truck visits node v.
- q_v^d: Task delivery volume of driver d at node v .
- Y^d: Binary variable taking value 1 if driver d is assigned and 0 otherwise.
- Z_{uv}^t: Binary variable taking value 1 if the depot truck travels from node u to node v in stage t and 0 otherwise.
- X_{uv}^d: Binary variable taking value 1 if driver d travels from node u to node v and 0 otherwise.

3.3. Model assumptions

The following observations and assumptions are provided for the mathematical model formulation:

1. Each crowdsourcing driver has an origin and a destination node (Arslan et al., 2019).
2. Each customer has a unique location and corresponds to a single delivery demand Kafle et al. (2017).
3. Each crowdsourcing driver can only participate in the assignment bidding within a time window. This time window starts from the driver announcement time a^d and ends at the driver's earliest departure time e^d. Fig. 2 shows the case that $a^d < S^t < e^d$ for driver d, so that driver d can participate in stage t assignment.
4. A driver's assignment bidding is accepted into a stage based on the earliest departure time of the driver. As shown in Fig. 3, S^{t+1} is immediately before the earliest departure time e^d of driver d, therefore driver d will participate in stage $t+1$ assignment.
5. As shown in Fig. 4, once a driver accepts a task (or tasks), the latest departure time of the driver will be advanced and the driver's flexibility will be reduced to accommodate the picking up and delivery of the task(s). The time for the driver to pick up the task(s) at sub-depot b must also be after the truck delivers at the sub-depot.
6. All tasks originate from the depot. Fig. 5 shows that tasks are first transported by a truck from the depot to the sub-depots, from where the tasks are handed over to the crowdsourcing drivers for delivery. The capacity for the truck is assumed to be able to accommodate all the tasks to be delivered in a certain stage.
7. Tasks that cannot be completed on the day will incur compensations for the affected customers.

4. Methodology and model

As discussed in Section 3, the MS-2E-DAM problem needs to determine which driver will deliver which task from which sub-depot, and in the case of multiple tasks being assigned to one driver, the delivery sequence to customers of the driver. In this sense, MS-2E-DAM involves both supply chain network design (at the operational level) under uncertainty (Govindan et al., 2017) and capacitated location-routing problem with time windows (Prodhon and Prins, 2014). In each stage of the MS-2E-DAM problem, it not only considers the known information of tasks and drivers in the current stage, but also the known information of tasks and drivers in later stages. Therefore, the tasks released at the current stage may be left to later stages for assignment to more suitable drivers to achieve an overall better delivery assignment. Fig. 6 presents a schematic view of the proposed MS-2E-DAM approach.

Let us denote the current stage as stage $\bar{t}$. Let $P_{\text{rem}}^{\bar{t}}$ be the set of tasks left over from the previous stages. Let χ be known percentage

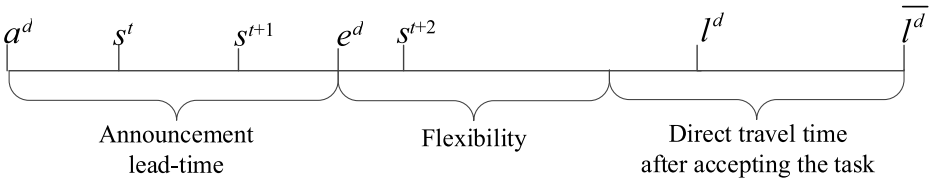

Fig. 3. Assignment stage determination using driver's time window.

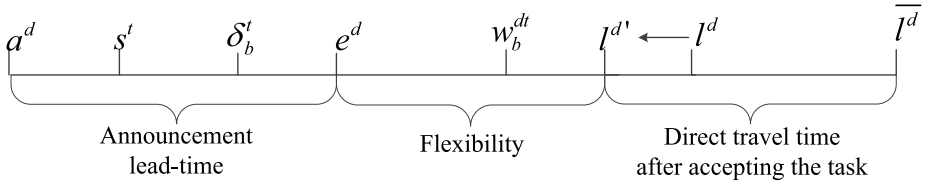

Fig. 4. Change of driver's latest departure time after task acceptance.

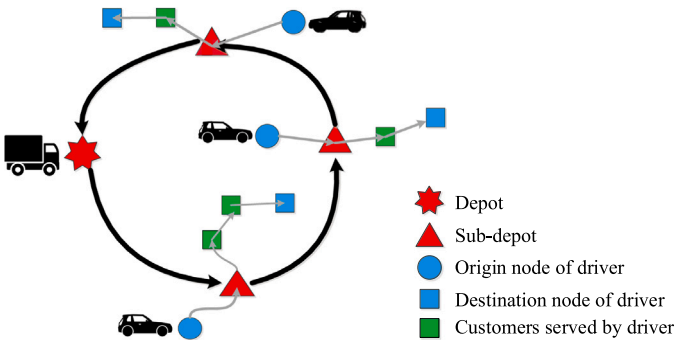

Fig. 5. An illustration of the truck and drivers delivering tasks.

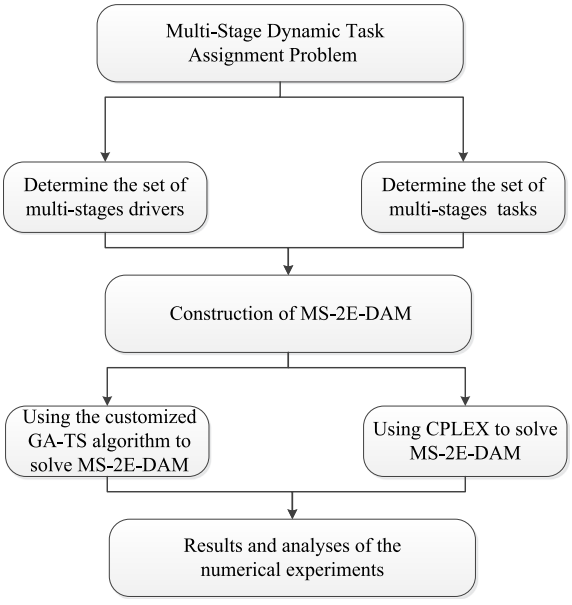

Fig. 6. A schematic view of the proposed MS-2E-DAM approach.

of drivers. Therefore, $\chi^{t-\bar{t}}$ indicates the proportion of known drivers who decide to participate in stage t assignment to all drivers who will eventually participate in stage t assignment at the time of stage $\bar{t}$.

4.1. Determining the set of multi-stage drivers and tasks

In order to minimize the overall cost on the day, stage $\bar{t}$ assignment must consider not only the tasks and participating drivers at stage $\bar{t}$, but also those at later stages. At each stage of the assignment, the information of the drivers in later stages cannot be completely known. For example, it is impossible to know the information of drivers who have not announced to participate in the assignment. The further away a stage is, the less information can be obtained. Therefore, before stage $\bar{t}$ assignment, it is necessary to determine the information of known drivers participating in assignment and tasks need to be assigned in each stage. The specific formulas are as follows:

$$D_{\text{new}}^t = \chi^{t-\bar{t}} D^t,$$

where D_{new}^t refers to the set of drivers who have declared to participate in stage t assignment at stage $\bar{t}$, $t = (\bar{t} \ldots T)$.

$$D_{\text{all}}^{\bar{t}} = \sum_{t=\bar{t}}^{T} D_{\text{new}}^t,$$

where $D_{\text{all}}^{\bar{t}}$ indicates the set of all the drivers who are known to participate in assignments of all the rest stages at stage $\bar{t}$. With D_{new}^t and $D_{\text{all}}^{\bar{t}}$, the corresponding sets of N_{new}^t and $N_{\text{all}}^{\bar{t}}$ can be determined. Apparently, if there are tasks left from previous stages, they need to be included for consideration in the current stage:

$$P_{\text{new}}^{\bar{t}} = P^{\bar{t}} \cup P_{\text{rem}}^{\bar{t}},$$

where $P_{\text{new}}^{\bar{t}}$ refers to the set of tasks that need to be assigned in stage $\bar{t}$, which is the union of the new task set $P^{\bar{t}}$ for stage $\bar{t}$, and the set $P_{\text{rem}}^{\bar{t}}$ of tasks left over from previous stages. Therefore, the set of all tasks that have not been assigned on the day at stage $\bar{t}$, i.e., $P_{\text{all}}^{\bar{t}}$ is:

$$P_{\text{all}}^{\bar{t}} = P_{\text{new}}^{\bar{t}} \cup \sum_{t=\bar{t}+1}^{T} P^t.$$

4.2. Model formulation

The following is the optimization model when the current stage is $\bar{t}$, in which, except for the $\bar{t}$ stage, the assignment of subsequent stages are all pre-assignment. When MS-2E-DAM rolls to the next stage, i.e., stage $\bar{t}+1$, all available tasks will be re-assigned due to the emergence of new drivers so as to ensure that each assignment is the optimal solution using the known information. This cycle will continue until the last stage.

4.2.1. Objective function

$$\begin{aligned}\min \Omega =& \eta \sum_{t=\bar{t}}^{T} \sum_{u\in B} \sum_{v\in B} \gamma_{uv} Z_{uv}^{t} + \\ & \lambda\Big(\sum_{d\in D_{all}^{\bar{t}}} \sum_{u,v\in B\cup N_{all}^{\bar{t}}\cup P_{all}^{\bar{t}}} \gamma_{uv} X_{uv}^{d} - \sum_{d\in D_{all}^{\bar{t}}} L^{d} Y^{d} \Big) + \\ & \sum_{d\in D_{all}^{\bar{t}}} \Big(\alpha \sum_{v\in P_{all}^{\bar{t}}} q_{v}^{d} + \beta \sum_{u,v\in B\cup P_{all}^{\bar{t}}} \sum X_{uv}^{d} \Big) + \\ & \partial\Big(|P_{all}^{\bar{t}}| - \sum_{d\in D_{all}^{\bar{t}}} \sum_{u,v\in B\cup P_{all}^{\bar{t}}} \sum X_{uv}^{d} \Big)\end{aligned} \tag{1}$$

Eq. (1) aims at minimizing the overall cost, which consists of the truck operating cost, the compensation paid to committed crowdsourcing drivers' additional travel across all stages, the compensation for the total quantity and volume of parcels transported by crowdsourcing drivers, and the penalty cost for tasks that are not completed on the day.

4.2.2. Constraints

$$\sum_{v\in P_{all}^{\bar{t}}} q_{v}^{d} \leqslant Q^{d}, d \in D_{all}^{\bar{t}} \tag{2}$$

Constraint (2) controls the actual load of driver d from node u to node v does not exceed the maximum load capacity of the driver.

$$\sum_{d\in D_{\text{all}}^{\bar{t}}} \sum_{u\in P_{\text{all}}^{\bar{t}}} X_{uv}^{d} \leqslant 1, \quad v \in P_{\text{all}}^{\bar{t}} \tag{3}$$

$$\sum_{d\in D_{\text{all}}^{\bar{t}}} \sum_{v\in P_{\text{all}}^{\bar{t}}} X_{uv}^{d} \leqslant 1, \quad u \in B \cup P_{\text{all}}^{\bar{t}} \tag{4}$$

Constraints (3) and (4) specify that the drivers cannot visit the same delivery node more than once. That is, a task can only be delivered by at most one crowdsourcing driver.

$$\mu_{u} - \mu_{v} - |P_{all}^{\bar{t}}| X_{uv}^{d} \leqslant |P_{all}^{\bar{t}}| - 1, \quad u, v \in B \cup N_{all}^{\bar{t}} \cup P_{all}^{\bar{t}}, \mu_{u}, \mu_{v} \in N^{*} \tag{5}$$

$$\mu_{u} - \mu_{v} - |B| Z_{uv}^{t} \leqslant |B| - 1, \quad u, v \in B, \mu_{u}, \mu_{v} \in N^{*} \tag{6}$$

Constraints (5) and (6) prevent sub-tours.

$$\sum_{d\in D_{\text{new}}^{t}} \sum_{u,v\in B\cup P^{t+1}\cup\cdots\cup P^{n}} \sum X_{uv}^{d} = 0, \quad t = (\bar{t} \ldots T) \tag{7}$$

Constraint (7) indicates that once a driver decides to participate in stage t assignment, the driver will not be able to accept tasks in subsequent stages.

$$a^{d} \leqslant S^{t} \leqslant e^{d}, \quad d \in D_{\text{new}}^{t}, t = (\bar{t} \ldots T) \tag{8}$$

Constraint (8) denotes the time window that a driver can participate in the assignment.

$$\delta_{b}^{t} \leqslant w_{b}^{dt}, \quad d \in D_{\text{new}}^{t}, b \in B, t = (\bar{t} \ldots T) \tag{9}$$

Constraint (9) means that the arrival time of driver d at sub-depot b in stage t must be after the arrival time of the truck at sub-depot b in stage t.

$$w_{b}^{dt} - \sum_{u\in N_{A}} \gamma_{ub} X_{ub}^{d} / s_{c} \geqslant e^{d}, \quad d \in D_{\text{new}}^{t}, b \in B, t = (\bar{t} \ldots T) \tag{10}$$

$$w_{b}^{dt} \leqslant l^{d}, \quad d \in D_{\text{new}}^{t}, b \in B, t = (\bar{t} \ldots T) \tag{11}$$

Constraints (10) and (11) guarantee that the arrival time of driver d at sub-depot b in stage t should be between the driver's earliest departure time and the driver's latest departure time, while ensuring enough time for traveling among nodes.

$$w_{b}^{dt} + \sum_{u,v\in B\cup P\cup N_{F}} \sum \gamma_{uv} X_{uv}^{d} / s_{c} \leqslant \overline{l^{d}}, \quad d \in D_{\text{new}}^{t}, b \in B, t = (\bar{t} \ldots T) \tag{12}$$

Constraint (12) ensures that drivers will have enough time to reach their destination on time after their task deliveries.

$$\delta_{b}^{t} + \gamma_{bb'} Z_{bb'}^{t} / s_{t} - M(1 - Z_{bb'}^{t}) \leqslant \delta_{b'}^{t}, \quad b, b' \in B, t = (\bar{t} \ldots T) \tag{13}$$

Constraint (13) allocates enough time for the depot truck to travel among the depot and the sub-depots (where M is a sufficiently large number).

$$X_{uv}^{d} \leqslant Y^{d}, \quad d \in D_{\text{all}}^{\bar{t}}, u, v \in B \cup N_{all}^{\bar{t}} \cup P_{all}^{\bar{t}} \tag{14}$$

Constraint (14) enforces the assignment logic, i.e., a driver can only execute a task after the task is assigned to the driver.

$$X_{uv}^{d} = (0, 1), \quad d \in D_{\text{all}}^{\bar{t}}, u, v \in B \cup N_{all}^{\bar{t}} \cup P_{\text{all}}^{\bar{t}} \tag{15}$$

$$Y^{d} = (0, 1), \quad d \in D_{\text{all}}^{\bar{t}} \tag{16}$$

$$Z_{uv}^{t} = (0, 1), \quad u, v \in B, t = (\bar{t} \ldots T) \tag{17}$$

Constraints (15) to (17) define X_{uv}^{d}, Y^{d} and Z_{uv}^{t} as binary variables.

5. Solution method

The MS-2E-DAM problem is a variant of vehicle routing problems with time windows which belongs to NP-hard problem. Given the inherent intractability of MS-2E-DAM, a hybrid heuristic algorithm is proposed which combines Genetic Algorithm (GA) and Tabu Search (TS). Genetic Algorithm (GA) was first developed by Goldberg and Holland (1988), which is suitable for optimization of large-scale complex problems (Zhou et al., 2019; Derbel et al., 2012; Ardjmand et al., 2015). Tabu search (TS) was provided by the University of Colorado system scientist Professor Glover in 1986, which has been widely used for solving vehicle routing problems (Gmira et al., 2021). It has strong local solving ability and fast convergence speed. Specifically, this paper uses GA to allocate crowdsourcing drivers to stages and tasks and applies TS to optimize the sequence of visits for each driver. Therefore, the hybrid heuristic algorithm, denoted as 'GA-TS' in this paper, utilizes the advantages of both optimization techniques.

Algorithm 1 provides a schematic description of the GA-TS. After all relevant parameters are set, the GA population is initialized using four methods (Section 5.1). The chromosomes are then optimized by TS (Section 5.2) to determine the task delivery sequences, followed by their fitness function evaluations (in Section 5.3). Based on the fitness function values, the chromosomes for the next generation are selected as per Section 5.4, and are applied with crossover (Section 5.5) and mutation (Section 5.6) operations to produce the chromosomes for the next generation.

5.1. Initialization

In this paper, natural numbers are used to encode solutions for MS-2E-DAM. Each chromosome codes the solution in the stages to be scheduled. Within each stage, there are two parts. The first part shows the sub-depots visited by the crowdsourcing drivers, which uses $|D^{t}|_{\max}$ numbers with each number representing the sub-depot that the corresponding crowdsourcing driver visits. The second part encodes the task assignment to the crowdsourcing drivers, which consists of $|P|$ numbers with the ith number indicating the delivering crowdsourcing driver for the ith task. When a task is not assigned to any driver in a stage, the corresponding number will be 0. Fig. 7 illustrates how the encoding works. There are three stages, six tasks (two in each stage), six drivers (two in each stage), and three depots from which

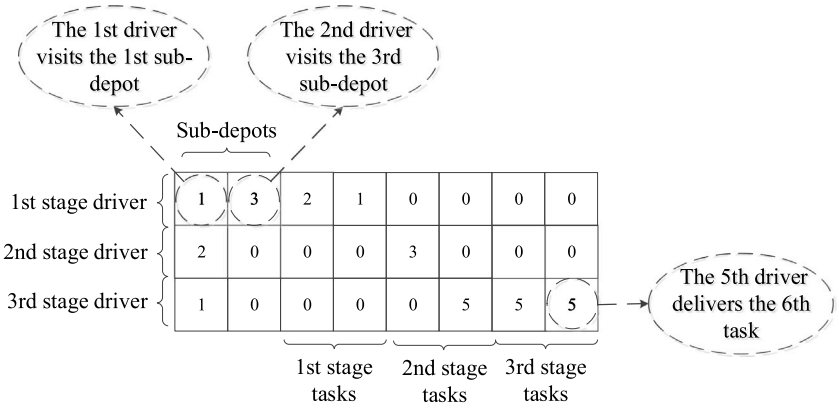

Fig. 7. An illustrative example of encoding.

Algorithm 1: GA-TS

```
Parameter initialization (e.g., population size pop_size, max number of generations Max_Gen, crossover and mutation rates P_c and P_m);
Population initialization;
Gen = 1;
while Gen ⩽ Max_Gen do
    Using TS to determine the task delivery sequences for crowdsourcing drivers, and the corresponding depot visit sequences for the depot truck;
    Calculate the fitness function of all chromosomes;
    if Gen ≠ Max_Gen then
        Select pop_size chromosomes for the next generation;
        Apply crossover and mutation operations;
    Gen = Gen +1;
Report the best solution found;
```

the drivers can pick up tasks. The first two genes in each stage of the chromosome indicate from which sub-depots the drivers accept their tasks; the following genes specify the assignment of tasks to drivers.

In the initialization phase, codes for the sub-depots from which the drivers pick up their tasks are randomly generated. It is noted that tasks accepted by one driver should not be more than the driver's maximum loading capacity. The following four methods are used for task assignment in order to construct a better and diversified initial population. The first one uses pure randomization and the rest three are based on Roulette wheels:

1. Randomly assign tasks to drivers;
2. Based on the average distance between the origin of a driver and all the sub-depots, assign each driver a weight. The smaller the average distance, the greater the weight for the Roulette wheel;
3. Based on the destination distance between a driver and a task, assigning each driver a weight. The closer the distance, the greater the weight for the driver to get the task for the Roulette wheel;
4. Based on the original route distance of drivers, assigning each driver a weight. The longer the original route distance, the greater the weight for the Roulette wheel.

5.2. Tabu search

Each chromosome is decoded by TS to determine the sequence of tasks delivered by drivers and the sequence of sub-depots visited by the depot truck. Fig. 8 provides an example in which the 5th driver arrives at the 2nd sub-depot to pick up tasks 1, 2 and 4; the 6th driver arrives at the 1st sub-depot to pick up tasks 3, 5 and 6. With the assignment of tasks, the delivery sequences of drivers 5 and 6 are determined based on cross exchanges. For instance, the 1st task of the 5th driver is exchanged with the 2nd task, and the 1st task of the 6th driver is exchanged with the 3rd task.

The sequence of sub-depots to be visited by the depot truck is determined once the sequences of tasks delivered by drivers are finalized. Specifically, the truck visit sequence is determined in three steps: (1) determine all the sub-depots visited by the truck in each stage; (2) construct an initial truck visit sequence according to drivers' time windows, and; (3) use TS to improve the initial sequence, noting that the truck needs to depart from and finally return to the depot. It should be noted that the fitness value of a chromosome will be penalized (see Section 5.3) if there are drivers who cannot meet the time window determined by the depot truck. Fig. 9 illustrates how the sub-depot visiting sequence is determined by TS with two-point cross exchanges.

When determining the routes for both the crowdsourcing drivers and the depot truck, the initial solutions of the TS are randomly generated and neighborhood movements are to randomly exchange node visit sequences as per Fig. 8. The total travel distance of each truck or driver is set as the evaluation function, a penalty value is applied when the route does not meet the drivers' time windows. The common aspiration criterion, which accepts better solutions than the current best known solution, is adopted in this paper.

5.3. Fitness function

The fitness value of a chromosome is calculated using Eq. (18):

$$f = \Omega + \text{Penalty} \times N_{\text{infeasible}} \tag{18}$$

Here, f denotes the fitness function value of the chromosome; Ω is the objective function value of Eq. (1); *Penalty* is a sufficient large number; $N_{\text{infeasible}}$ refers to the number of infeasible routes in the chromosome.

5.4. Selection

The chromosomes for the next generation are chosen using tournament selection. An elite retention strategy is also applied to retain the τ ($\tau > 0$) best chromosomes. The tournament selection method works as follows: (1) randomly select φ ($\varphi > 2$) chromosomes from the population (each individual has the same probability of being selected); (2) take the chromosome with the best fitness function value into the next generation; (3) repeat the previous step until *pop_size* chromosomes are selected.

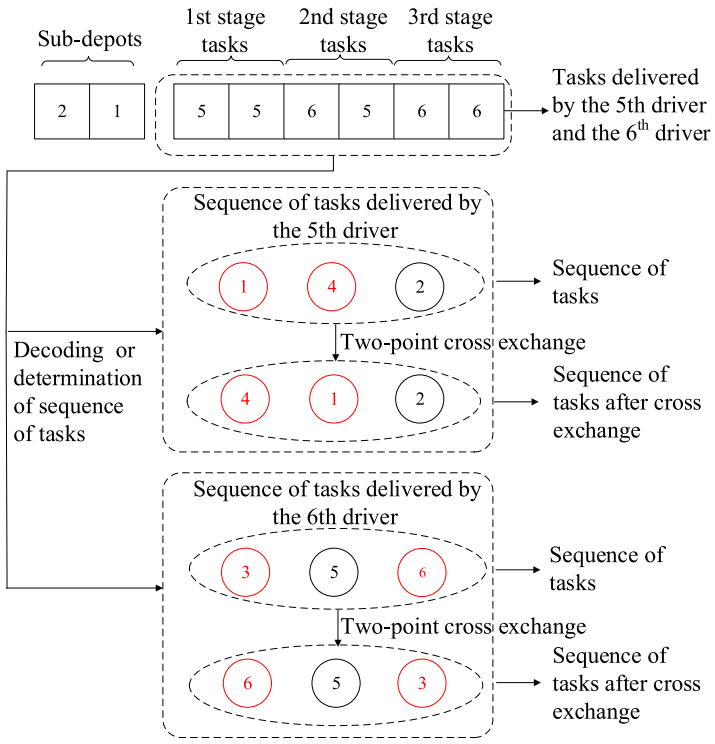

Fig. 8. An illustrative example of task delivery sequence determined by tabu search.

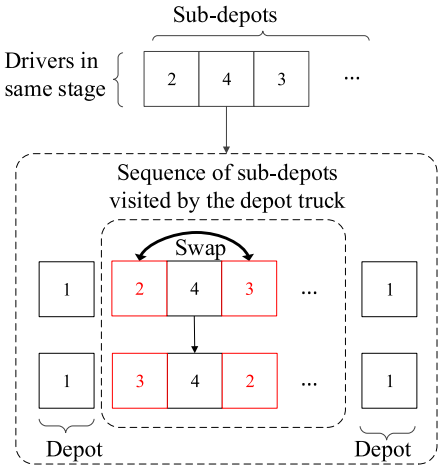

Fig. 9. An illustrative example of the sequence of sub-depots determined by tabu search.

5.5. Crossover

The crossover operator is a modified two-point crossover from the classical two-point crossover. The chromosomes in the current population are randomly mated to form *pop_size*/2 parent pairs. In order to increase diversity and improve the quality of offspring, the crossover rate P_c is determined using the adaptive function based on Srinivas and Patnaik (1994):

$$P_c = \begin{cases} \rho_1 \times \frac{f_{\text{avg}} - f_m}{f_{\text{avg}} - f_{\text{min}}}, & f_{\text{avg}} \geqslant f_{\text{m}} \\ \rho_2, & f_{\text{avg}} < f_{\text{m}} \end{cases}$$

Here, f_{avg} is the average fitness function values of all chromosomes, f_{min} is the minimum fitness function value of all chromosomes, f_m is the smaller fitness function value of the selected parent pair, $0.0 < \rho_1 < \rho_2 < 1.0$. Fig. 10 shows the modified two-point crossover operation, where the two crossover points are randomly selected and parts of the chromosomes are exchanged to form two new offspring.

5.6. Mutation

There are two mutation operators proposed in this paper which are modified based on classical point mutations. In order to help the algorithm escape from a local optimal solution, the mutation rate P_m is determined using the adaptive function in Srinivas and Patnaik (1994):

$$P_m = \begin{cases} \rho_3 \times \frac{f_{\text{avg}} - f}{f_{\text{avg}} - f_{\text{min}}}, & f_{\text{avg}} \geqslant f \\ \rho_4, & f_{\text{avg}} < f \end{cases}$$

Here, f_{avg} is the average fitness function values of all chromosomes, f_{min} is the minimum fitness function value of all chromosomes, f is the fitness function value of the selected chromosome to be mutated, $0.0 < \rho_3 < \rho_4 < 1.0$.

The first mutation operator consists of three steps: (1) randomly select multiple tasks in the selected chromosome; (2) randomly assign selected tasks to any driver who can accept the tasks; (3) re-determine the sub-depots visited by the drivers. Fig. 11 illustrates how the first mutation operator works. Here, numbers in the dashed circles denote the drivers who deliver the selected tasks before the mutation operation, red numbers denote the drivers who deliver the selected tasks after the mutation operation.

The second mutation operator consists of two steps: (1) randomly select a driver who have been assigned; (2) all tasks of the selected driver are assigned to other drivers who can accept the tasks, as shown in Fig. 12.

6. Numerical experiments

This section presents the numerical experiments and analyzes the results. All experiments were conducted using MATLAB 2018b on an Intel Core i7 7700HQ 2.8 GHz laptop with 8 GB RAM.

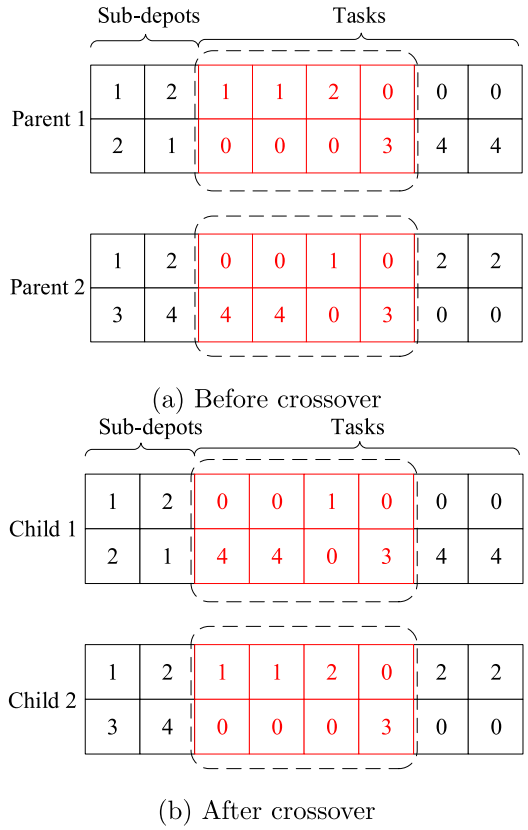

Fig. 10. The crossover operator.

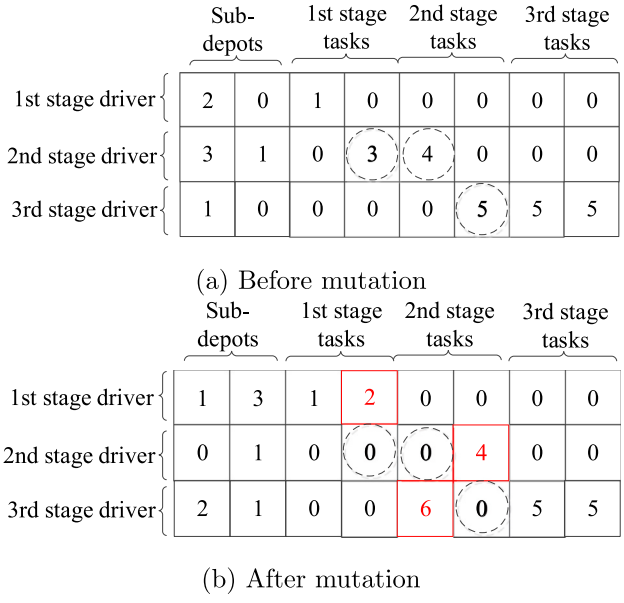

Fig. 11. The first mutation operator. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.1. Parameter values

In the experiments, the origin and destination nodes for crowdsourcing drivers were randomly generated in a 50 km × 50 km rectangular plane. So were the delivery nodes for tasks, and the location of the depot and sub-depots. The capacity of crowdsourcing drivers was set to

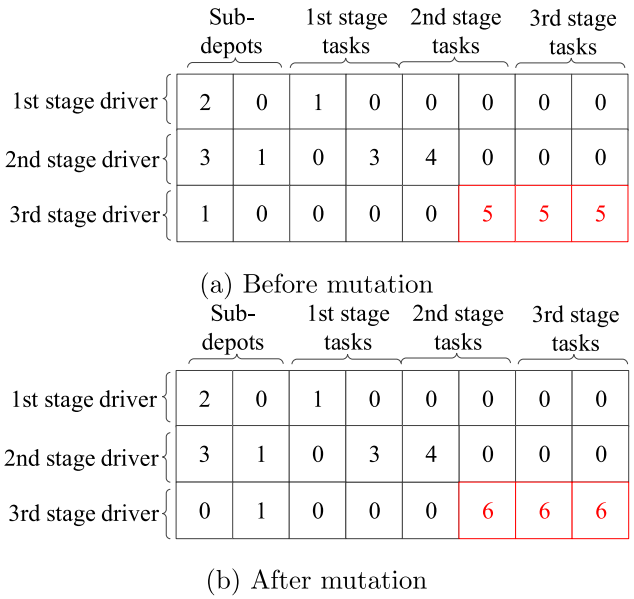

Fig. 12. The second mutation operator.

$Q^d = 75$. The volume of a task was a random number generated from a uniform distribution $U(1, 50)$. The cost per kilometer of the depot truck $\eta = 1$; the compensation per kilometer for a driver's additional travel due to detour $\lambda = 0.5$; the compensation per unit task volume $\alpha = 0.5$, and the compensation per task $\beta = 1$. The penalty for uncompleted tasks was set as $\partial = 1000$. The crowdsourcing drivers' travel speed s_c was set to 60 km/h and the depot truck's travel speed s_t was set to 40 km/h.

Preliminary numerical tests were conducted to set the parameters for the GA-TS. As a result, the number of generations and the population size in the GA were set to 100. The adaptive parameters to control the crossover probability were set as $\rho_1 = 0.5$, $\rho_2 = 0.9$. For mutation probability, $\rho_3 = 0.5$, $\rho_4 = 0.8$. The maximum number of iterations in the TS was set to 30, length of the tabu list was set to $\left\lceil \sqrt{\frac{N^d \times (N^d - 1)}{2}} \right\rceil$ (N^d is number of tasks delivered by driver d. $\lceil \cdot \rceil$ denotes the rounding up operator).

6.2. Small-size problems

The GA-TS was first tested and its performance compared with CPLEX 12.10 for a batch of 15 small-size problems. In this batch, the drivers were assumed to only accept tasks from the depot. There were three stages a day, namely at 9am, 1pm and 5pm. The total number of tasks ($|P|$), the number of tasks in each stage ($|P^t|$), the total number of drivers ($|D|$), and the number of drivers in each stage ($|D^t|$) are shown in Table 2. The driver confirmation rate $\chi = 0.8$.

Each problem was solved five times using the GA-TS algorithm to account for the randomness in the solution approach and the average of the five runs was reported. Table 2 presents the total costs using CPLEX (c_M) and GA-TS (c_H), and their corresponding CPU time (in seconds). The results show that CPLEX needs longer solving time than the GA-TS algorithm does for all the small-size problems. It should also be noted that as the size of the problem increases, the solving time gap between CPLEX and GA-TS quickly increases. This indicates that CPLEX will not be suitable for large-size problems. Fig. 13 visually depicts the cost differences between CPLEX and the GA-TS algorithm. It can be observed that the gaps are rather minimal. Overall, as shown in Table 2, most of the cost gaps between CPLEX and GA-TS (i.e., $(c_H - c_M)/c_M$) is less than 3%, with the biggest one less than 7%; CPLEX achieves better results at the cost of significantly longer computational time when the number of tasks is 15 or more; the coefficient of variation (CoV) for GA-TS is small (the largest one is 2.93%) which indicates that GA-TS is very robust.

Table 2
The results of small-size problems.

Prob.	$\|P\|$	$\|P^t\|$	$\|D\|$	$\|D^t\|$	CPLEX		GA-TS			$\frac{c_H - c_M}{c_M}$
					c_M	CPU (s)	c_H	CoV (%)	CPU (s)	Gap (%)
S1	6	2	6	2	346.06	8.7	346.06	0.00	8.0	0.00
S2	6	2	12	4	282.12	10.8	282.12	0.00	12.6	0.00
S3	6	2	18	6	294.81	12.7	294.81	0.00	16.8	0.00
S4	9	3	12	4	326.02	14.9	326.91	0.39	17.7	0.27
S5	9	3	18	6	417.58	15.3	428.09	2.93	17.7	2.51
S6	9	3	24	8	340.93	19.2	345.79	2.92	19.1	1.42
S7	12	4	12	4	466.64	22.1	475.71	0.62	22.0	1.92
S8	12	4	18	6	446.89	35.5	461.47	2.05	19.7	3.26
S9	12	4	24	8	394.59	45.5	397.14	0.23	29.5	0.60
S10	15	5	12	4	529.47	628.4	542.55	2.88	29.0	2.47
S11	15	5	18	6	528.91	323.1	564.48	1.74	31.0	6.72
S12	15	5	24	8	428.67	420.1	447.70	1.62	27.2	4.44
S13	18	6	12	4	692.40	42.9	713.03	2.01	40.7	2.98
S14	18	6	18	6	582.70	5,400.0	604.29	2.08	56.2	3.70
S15	18	6	24	8	714.53	11,160.0	762.25	1.16	47.5	6.68

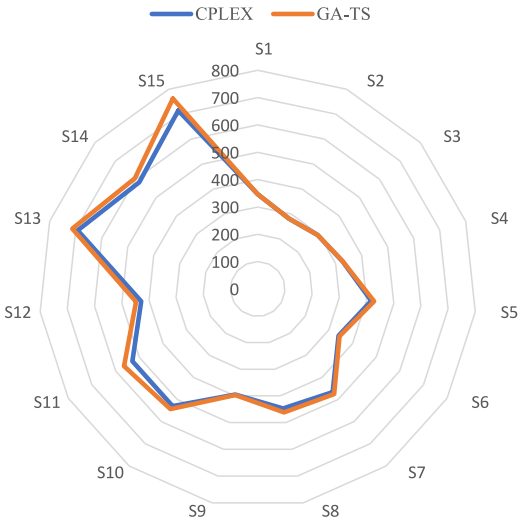

Fig. 13. Cost comparison between CPLEX and the GA-TS algorithm for small-size problems.

6.3. Large-size problems

For large-size problems, it was assumed that there were six sub-depots and there were three stages a day. The total number of tasks was 60; the number of tasks in each stage was 20. The total number of drivers was 90; and the number of drivers in each stage was 30. The driver confirmation rate $\chi = 0.8$.

In total, 10 problem instances were generated and solved by the GA-TS algorithm. For comparison, the GA-TS algorithm was also used to solve each problem instance without considering the future driver and task information, which is referred to as the single-stage assignment model (SSAM). Each problem was solved five times using MS-2E-DAM and SSAM, and the average of the five runs was reported in Table 3.

It can be observed from Table 3 that SSAM has higher cost than the MS-2E-DAM algorithm across all problem instances. MS-2E-DAM outperforms SSAM by 1.94% on average, with the best instance reaching 3.93%. This indicates that there is a cost competitive edge by considering future committing drivers and task information. Indeed, such information enables MS-2E-DAM to make full use of the capacity of crowdsourcing drivers and to find the more suitable drivers for tasks, thereby reducing the cost for logistics platforms.

Cost, in this case, is a proxy of the vehicle distance traveled. Therefore, the cost reductions showing in the numerical experiments clearly indicate the potential of the proposed MS-2E-DAM approach's contribution to sustainable last-mile logistics operations, especially in terms of reduced carbon emissions. Note that the cost reductions in Table 3 are based on crowdsourcing logistics conducted in single stages and hence highlighting the benefits of incorporating future committing drivers.

Table 3
The results of large-size problems.

Prob.	SSAM Cost (c_S)	MS-2E-DAM Cost (c_H)	$\frac{c_S - c_H}{c_H}$ (%)
L1	4,897.12	4,740.32	3.20
L2	4,206.88	4,176.14	0.73
L3	4,971.62	4,807.10	3.30
L4	3,581.82	3,545.99	1.00
L5	4,579.10	4,563.88	0.33
L6	4,078.58	3,918.26	3.93
L7	4,240.86	4,222.80	0.42
L8	4,211.92	4,064.25	3.50
L9	3,973.74	3,960.31	0.34
L10	4,386.12	4,289.31	2.20
Mean	4,312.78	4,228.84	1.94

6.4. Sensitivity analyses

This section analyzes the influence of three parameters on the cost of task assignment: the compensation per kilometer for a driver (λ), the cost per kilometer for the depot truck (η), and the compensation per unit volume of a task for a driver (α). The influence of the three parameters on the cost of task assignment was tested on one instance, where there were six sub-depots and three stages a day. The total number of tasks was 90; the number of tasks in each stage was 30; the total number of drivers was 60; and the number of drivers in each stage was 20. For this test instance, 1.0 was used for the initial values of each parameter and the results were used as the cost benchmark. The parameters were then increased or decreased with a step of 0.5 to test how the total cost responded to the parameter changes.

Table 4 presents the total system cost, the distance traveled by the depot truck, the total detour distance traveled by all crowdsourcing drivers, and the cost change from the cost benchmark in percentages. It can be observed that when λ, η, and α change, the total system costs change correspondingly. However, the impacts on the total system cost of these parameters are different. Changes in λ and α have a greater impact on the cost, while η has a smaller impact on the cost.

Figs. 14 to 16 further graphically present how the system responds to the change of parameters. Based on the trends, the following observations can be drawn:

1. Due to the highly cost-sensitive nature of λ, the crowdsourcing platform should pay more attention to exploring more payment options and strengthen cost control for crowdsourcing drivers;

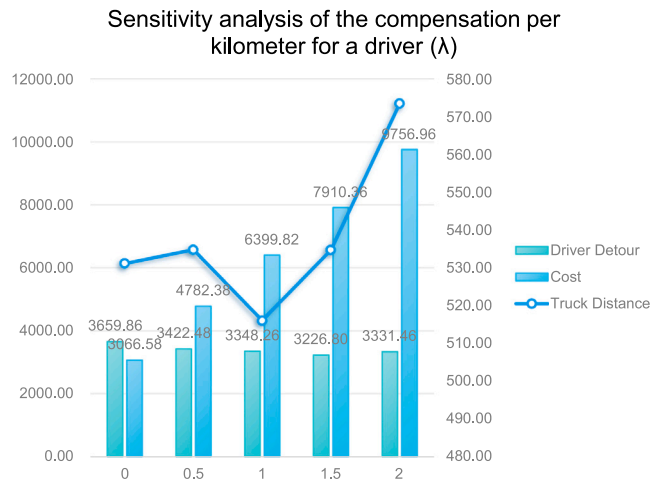

Fig. 14. Sensitivity analysis of parameter λ.

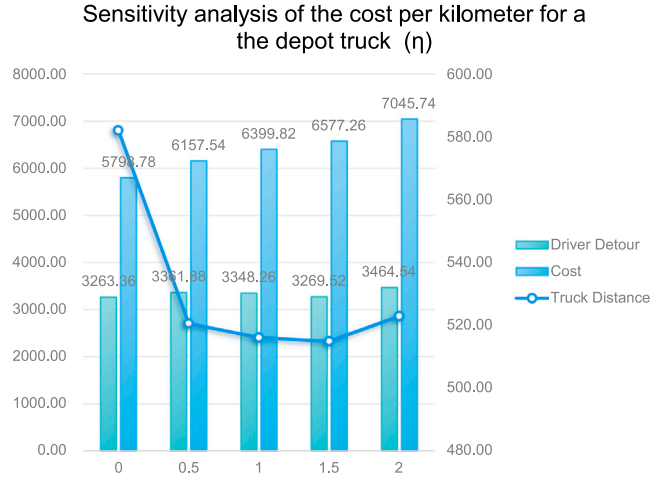

Fig. 15. Sensitivity analysis of parameter η.

Table 4
Sensitivity analysis of three parameters.

Parameter	Value	Cost	Truck distance	Driver detour	Cost change (%)
	0.0	3,066.58	531.12	3,659.86	−52.08
	0.5	4,782.38	534.77	3,422.48	−25.27
λ	1.0	6,399.82	516.07	3,348.26	–
	1.5	7,910.36	534.70	3,226.80	23.60
	2.0	9,756.96	573.45	3,331.46	52.46
	0.0	5,798.78	582.03	3,263.36	−9.39
	0.5	6,157.54	520.56	3,361.88	−3.79
η	1.0	6,399.82	516.07	3,348.26	–
	1.5	6,577.26	514.88	3,269.52	2.77
	2.0	7,045.74	522.88	3,464.54	10.09
	0.0	4,058.52	556.68	3,425.64	−36.58
	0.5	5,209.04	556.38	3,339.90	−18.61
α	1.0	6,399.82	516.07	3,348.26	–
	1.5	7,756.34	423.06	3,445.74	21.20
	2.0	8,823.38	569.52	3,273.00	37.87

2. The two task-related parameters, i.e., the compensation per kilometer for a driver (λ) and the compensation per unit volume of a task for a driver (α) are key to influencing the overall system cost, which is not surprising. This highlights the need for the crowdsourcing platform to optimize the task assignment and encourage multiple stage task assignment to minimize the drivers' detour distance, and maximizing the use of capacity for crowdsourcing drivers.
3. The change of η has less impact on overall system cost, total truck distance traveled, and overall driver detour distance. This indicates that the depot truck operations are less affected by its operating cost due to its indispensability.

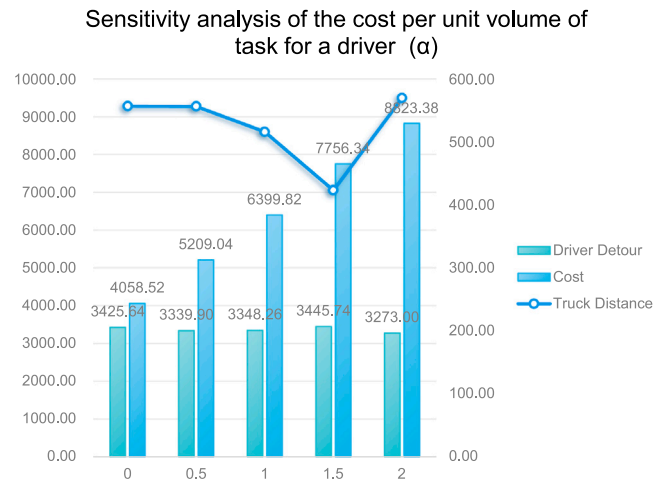

Fig. 16. Sensitivity analysis of parameter α.

7. Conclusion and future research

Vehicle fuel consumption and carbon emissions in the process of logistics operations have great environmental impacts, which render them as one of the major challenges for sustainable development. Crowdsourcing logistics can effectively reduce the use of vehicles and vehicle distance traveled to contribute to sustainable logistics. This paper proposed a multi-stage two-echelon dynamic task assignment approach for crowdsourcing logistics, which realistically captured the uncertainty of driver commitment over the task assignment horizon. The problem was formulated into a mathematical model and solved using CPLEX 12.10 for some small-size problems. Given the inherent intractability of the problem, a heuristic which combines genetic algorithm and tabu search was designed. The benchmark results against CPLEX confirmed that the heuristic can provide near optimal solutions with reasonable computational time for daily operational use. This paper also demonstrated the advantage of using the multi-stage approach over single stage approach and identified key parameters which affect the total system cost.

Compared with existing research, the contributions of this study are as follows. First, this paper fully considers the driver and task information in future stages, and comprehensively takes these into consideration for the assignment and dispatch of all delivery tasks. This is more in line with the reality and can provide a realistic reference for the actual operations of logistics platforms. Second, this paper explores a new way for crowdsourcing logistics task assignment. The account for the uncertain commitment of crowdsourcing drivers makes the proposed method suitable and practical for daily decision-making for crowdsourcing logistics platforms, which contributes to the research in the field of crowdsourcing logistics with uncertain information. Finally, a hybrid heuristic combining genetic algorithm and tabu search is proposed to exploit the characteristics of the research problem in this paper. The feasibility and effectiveness of the algorithm are verified via numerical experiments. The numerical results further provide insights into key areas where crowdsourcing logistics platforms should focus on.

This paper conducted preliminary research on the crowdsourcing logistics dynamic task assignment problem. However, there are still factors that worth further incorporation, such as uncertain traffic environment, uncertain starting points of the driver and so on. This paper could be explored further in a few ways to enable it better reflect real-world dispatching requirements. First, it is useful to consider the crowdsourcing logistics task assignment problem with uncertain traffic environment. In urban settings, traffic accidents and traffic congestions are common which will lead to difficulties in estimating the driving time. The general assumption of driver's unit travel time is constant will be challenged in this condition. Second, while a driver might commit to a certain stage of task assignment, the starting point of the driver might be uncertain. Therefore, it is worthwhile to model this into the decision-making process. Lastly, it is worthwhile to incorporate customer delivery time windows. Crowdsourcing logistics plays an important (and maybe the most important) role in customer's overall delivery experience, and getting parcels delivered as per customer instructions would be highly meaningful.

CRediT authorship contribution statement

Ming-Yang Li: Conceptualization, Methodology, Writing – original draft, Supervision, Funding acquisition. **Wen-Song Zhang:** Methodology, Software, Investigation, Validation, Visualization, Writing – original draft, Writing – revision. **Yong Wu:** Conceptualization, Methodology, Formal analysis, Writing – review & editing, Writing – revision, Supervision. **Lei Wang:** Conceptualization, Methodology, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partly supported by the Humanities and Social Science Foundation of Ministry of Education of China (No. 19YJA630037), and the National Natural Science Foundation of China (No. 71974067). The authors would like thank the two anonymous reviewers and the editor, for their constructive and pertinent comments and feedback. This paper would have not been in its current form without their valuable inputs.

References

Ahamed, T., Zou, B., Farazi, N.P., Tulabandhula, T., 2021. Deep reinforcement learning for crowdsourced urban delivery. Transp. Res. Part B Methodol. 152 (August), 227–257.

Akeb, H., Moncef, B., Durand, B., 2018. Building a collaborative solution in dense urban city settings to enhance parcel delivery: An effective crowd model in Paris. Transp. Res. Part E Logist. Transp. Rev 119, 223–233.

Allahviranloo, M., Baghestani, A., 2019. A dynamic crowdshipping model and daily travel behavior. Transp. Res. Part E Logist. Transp. Rev 128 (2018), 175–190.

Archetti, C., Savelsbergh, M., Speranza, M.G., 2016. The vehicle routing problem with occasional drivers. Eur. J. Oper. Res 254 (2), 472–480.

Ardjmand, E., Weckman, G., Park, N., Taherkhani, P., Singh, M., 2015. Applying genetic algorithm to a new location and routing model of hazardous materials. Int. J. Prod. Res. 53 (3), 916–928.

Arslan, A.M., Agatz, N., Kroon, L., Zuidwijk, R., 2019. Crowdsourced delivery – A dynamic pickup and delivery problem with ad hoc drivers. Transp. Sci. 53 (1), 222–235.

Barbosa, M., Pedroso, J.P., Viana, A., 2023. A data-driven compensation scheme for last-mile delivery with crowdsourcing. Comput. Oper. Res 150 (2022).

Bathke, H., Münch, C., 2023. From occasional to active crowdshippers: The significance of couriers' characteristics. IEEE Trans. Eng. Manag. 1–16.

Behrend, M., Meisel, F., 2018. The integration of item-sharing and crowdshipping: Can collaborative consumption be pushed by delivering through the crowd? Transp. Res. Part B Methodol 111, 227–243.

Buldeo Rai, H., Verlinde, S., Merckx, J., Macharis, C., 2017. Crowd logistics: an opportunity for more sustainable urban freight transport? eur. Transp. Res. Rev. 9 (3), 1–13.

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

Dahle, L., Andersson, H., Christiansen, M., Speranza, M.G., 2019. The pickup and delivery problem with time windows and occasional drivers. Comput. Oper. Res. 109, 122–133.

Dayarian, I., Savelsbergh, M., 2020. Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. Prod. Oper. Manag. 29 (9), 2153–2174.

Derbel, H., Jarboui, B., Hanafi, S., Chabchoub, H., 2012. Genetic algorithm with iterated local search for solving a location-routing problem. Expert Syst. Appl. 39 (3), 2865–2871.

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. Transp. Res. Part E Logist. Transp. Rev. 105, 105–122.

EMarketer, 2019. Global ecommerce 2019. https://www.emarketer.com/content/global-ecommerce-2019.

Ermagun, A., Stathopoulos, A., 2018. To bid or not to bid: An empirical study of the supply determinants of crowd-shipping. Transp. Res. Part A Policy Pract. 116 (March), 468–483.

Feng, X., Chu, F., Chu, C., Huang, Y., 2021. Crowdsource-enabled integrated production and transportation scheduling for smart city logistics. Int. J. Prod. Res. 59 (7), 2157–2176.

Frehe, V., Mehmann, J., Teuteberg, F., 2017. Understanding and assessing crowd logistics business models – using everyday people for last mile delivery. J. Bus. Ind. Mark 32 (1), 75–97.

Galkin, A., Schlosser, T., Cardenas, I., Hodakova, D., Capayova, S., 2021. Freight demand and supply assessment for implementation of crowdsourcing technology: A case study in Bratislava. Slovakia. Sustain 13 (7), 3865.

Gatta, V., Marcucci, E., Nigro, M., Patella, S.M., Serafini, S., 2019. Public transport-based crowdshipping for sustainable city logistics: Assessing economic and environmental impacts. Sustain 11 (1).

Gmira, M., Gendreau, M., Lodi, A., Potvin, J.Y., 2021. Tabu search for the time-dependent vehicle routing problem with time windows on a road network. Eur. J. Oper. Res 288 (1), 129–140.

Goldberg, D.E., Holland, J.H., 1988. Genetic algorithms and machine learning. Mach. Learn 3 (2), 95–99.

Govindan, K., Fattahi, M., Keyvanshokooh, E., 2017. Supply chain network design under uncertainty: A comprehensive review and future research directions. European J. Oper. Res. 263 (1), 108–141.

Hashemi, S.E., 2021. A fuzzy multi-objective optimization model for a sustainable reverse logistics network design of municipal waste-collecting considering the reduction of emissions. J. Clean. Prod 318 (2020).

Janjevic, M., Merchán, D., Winkenbach, M., 2021. Designing multi-tier, multi-service-level, and multi-modal last-mile distribution networks for omni-channel operations. Eur. J. Oper. Res. 294 (3), 1059–1077.

Kafle, N., Zou, B., Lin, J., 2017. Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery. Transp. Res. Part B Methodol. 99 (December), 62–82.

Kong, X.T., Kang, K., Zhong, R.Y., Luo, H., Xu, S.X., 2021. Cyber physical system-enabled on-demand logistics trading. Int. J. Prod. Econ. 233, 108005.

Lan, Y.L., Liu, F., Ng, W.W., Gui, M., Lai, C., 2022. Multi-objective two-echelon city dispatching problem with mobile satellites and crowd-shipping. IEEE Trans. Intell. Transp. Syst. 1–14.

Le, T.V., Ukkusuri, S.V., 2019. Influencing factors that determine the usage of the crowd-shipping services. Transp. Res. Rec. 2673 (7), 550–566.

Le, T.V., Ukkusuri, S.V., Xue, J., Van Woensel, T., 2021. Designing pricing and compensation schemes by integrating matching and routing models for crowd-shipping systems. Transp. Res. Part E Logist. Transp. Rev 149, 102209.

Li, G., Zheng, H., Ji, X., Li, H., 2018. Game theoretical analysis of firms' operational low-carbon strategy under various cap-and-trade mechanisms. J. Clean. Prod. 197, 124–133.

Liu, S., Zhang, Y., Liu, Y., Wang, L., Wang, X.V., 2019. An 'Internet of Things' enabled dynamic optimization method for smart vehicles and logistics tasks. J. Clean. Prod. 215, 806–820.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., Laporte, G., 2020. Crowd-shipping with time windows and transshipment nodes. Comput. Oper. Res 113.

Mehmann, J., Frehe, V., Teuteberg, F., 2015. Crowd logistics – A literature review and maturity model. Innov. Strateg. Logist. Supply Chain. 117–145.

Mladenow, A., Bauer, C., Strauss, C., 2016. crowd logistics: The contribution of social crowds in logistics activities. Int. J. Web Inf. Syst 12 (3), 379–396.

Modaresnezhad, M., Iyer, L., Palvia, P., Taras, V., 2020. Information Technology (IT) enabled crowdsourcing: A conceptual framework. Inf. Process. Manag. 57 (2).

Paloheimo, H., Lettenmeier, M., Waris, H., 2016. Transport reduction by crowdsourced deliveries – A library case in Finland. J. Clean. Prod. 132, 240–251.

Prodhon, C., Prins, C., 2014. A survey of recent research on location-routing problems. European J. Oper. Res. 238 (1), 1–17.
Punel, A., Ermagun, A., Stathopoulos, A., 2018. Studying determinants of crowd-shipping use. Travel Behav. Soc 12, 30–40.
Sampaio, A., Savelsbergh, M., Veelenturf, L.P., Van Woensel, T., 2020. Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers. Networks 76 (2), 232–255.
Seghezzi, A., Mangiaracina, R., 2022. Investigating multi-parcel crowdsourcing logistics for B2C e-commerce last-mile deliveries. Int. J. Logist. Res. Appl. 25 (3), 260–277.
Srinivas, M., Patnaik, L.M., 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans. Syst. Man Cybern. 24 (4), 656–667.
Strulak-Wójcikiewicz, R., Wagner, N., 2021. Exploring opportunities of using the sharing economy in sustainable urban freight transport. Sustain. Cities Soc. 68, 102778.
Suh, K., Smith, T., Linhoff, M., 2012. Leveraging socially networked mobile ICT platforms for the last-mile delivery problem. Environ. Sci. Technol. 46 (17), 9481–9490.
Voigt, S., Kuhn, H., 2022. Crowdsourced logistics: The pickup and delivery problem with transshipments and occasional drivers. Networks 79 (3), 403–426.
Wang, Y., Zhang, D., Liu, Q., Shen, F., Lee, L.H., 2016. Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions. Transp. Res. Part E Logist. Transp. Rev 93, 279–293.
Zhen, L., Wu, Y., Wang, S., Yi, W., 2021. Crowdsourcing mode evaluation for parcel delivery service platforms. Int. J. Prod. Econ. 235 (February).
Zhou, L., Lin, Y., Wang, X., Zhou, F., 2019. Model and algorithm for bilevel multisized terminal location-routing problem for the last mile delivery. Int. Trans. Oper. Res 26 (1), 131–156.