Computers & Industrial Engineering 196 (2024) 110476

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Dynamic pickup and delivery problem for autonomous delivery robots in an airport terminal☆

Joonhwa Jeong [a], Ilkyeong Moon [a,b,*]

[a] Department of Industrial Engineering, Seoul National University, Seoul 08826, Republic of Korea
[b] Institute of Engineering Research, Seoul National University, Seoul 08826, Republic of Korea

ARTICLE INFO

Keywords:
Dynamic pickup and delivery problem
Autonomous delivery robot
Battery recharging strategy
Peak period management

ABSTRACT

Autonomous delivery robot (ADR) operation for short-range delivery purposes is becoming an increasingly popular mode of service. Assigning orders to robots is critical in ensuring high quality in these automated delivery operations. This paper considers a dynamic pickup and delivery problem using autonomous robots (DPDP-AR), where a fleet of ADRs picks up desired items from stores and delivers them to customers. The arrival time of orders is uncertain, with a hard delivery deadline for each order. This study considered the battery consumption of ADRs, which has been less extensively covered in previous research, and devised a battery recharging strategy for the operation. To handle this stochasticity and dynamism in the problem, we developed a reassignment algorithm that reschedules previously assigned orders at the arrival of a new order. Additionally, as periods of high demand can be estimated, peak time management of the battery is proposed to enhance ADR utilization at peak periods. Computational experiments were performed on real-world ADR food delivery service instances in an airport terminal. Test instances on various demand scenarios demonstrated improvement in service quality when devised policies were used compared to current practice. We substantiated that the proposed algorithms found efficient solutions within short computation times, validating their applicability to real-world operations.

1. Introduction

The contactless trend of commerce triggered by the COVID-19 pandemic has dramatically changed delivery process. One significant change was the introduction of alternative delivery services in various business sectors. Online platforms had an influx of new customers unfamiliar with online purchasing, and food delivery services had to deal with explosive demand (Suguna, Shah, Raj, & Suresh, 2021). As a result, last-mile service managers had to prioritize and organize huge quantities of products of various types to meet constantly increasing orders. Speed, efficiency, transparency, personalized experience, and other factors were needed for a stable last-mile operation (Mangiaracina, Perego, Seghezzi, & Tumino, 2019). As it was challenging for many service providers to possess these qualities while satisfying demand, there has been a constant need for innovative methods of operation.

Implementing autonomous robots for last-mile delivery is suggested as a way for efficient operation. The development of self-driving technology and artificial intelligence has paved the way for commercializing autonomous vehicles and robots that can navigate without human control. Autonomous robots were mainly used for material handling at warehouses and distribution centers. With the help of relaxed regulations related to the operation of pedestrian roads, the application of autonomous robots has significantly expanded in the past few years. Models specifically customized for delivery were developed, and these so-called autonomous delivery robots (ADRs) are increasingly taking up larger proportion of last-mile delivery tasks.

As ADRs are expected to enhance delivery productivity and reduce delivery times, the global ADR market, estimated at 211.5 million USD in value, is expected to grow 34.9 percent annually for the next ten years. Various companies that can fulfill technological advancements for ADR development are entering the business and releasing products. *Starship Technologies*, founded in 2014, developed an autonomous delivery robot that can travel at pedestrian speed and is providing service mainly on college campuses in the USA (Chen, Demir, Huang, & Qiu, 2021). Other companies such as *Kiwi*, *Nuro*, and *Ottonomy* are also providing delivery services using autonomous robots. One current limitation of the industry is that the service is provided in small,

☆ The authors are grateful for the valuable comments from the editor-in-chief, the associate editor, and three anonymous reviewers. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00218913 and NRF-2021R1A4A1029124).
* Corresponding author at: Department of Industrial Engineering, Seoul National University, Seoul 08826, Republic of Korea.
E-mail addresses: juna529@snu.ac.kr (J. Jeong), ikmoon@snu.ac.kr (I. Moon).

https://doi.org/10.1016/j.cie.2024.110476

Available online 14 August 2024

restricted areas, as there are regulations related to the travel of ADRs on roads. There is a move toward weakening guidelines in various countries to keep pace with technological advancement and market expansion.

There are several competitive advantages of utilizing ADRs instead of human drivers. Several issues arise from human traits and characteristics. While using delivery personnel has played a valuable role as the conventional delivery method, there are innate limitations such as hour-of-service constraint and fatigue issues. Similar concerns need not arise when considering ADRs as long as the robots are adequately charged and supervised. In addition, there is ample possibility that human drivers can adjust to the operational decisions made by the central system. An example can be drivers choosing delivery routes based on empirical knowledge rather than on the optimal route suggested by the system. Even though the assignment of orders is based on the best routing decision, the best result cannot be guaranteed in this situation. In a centralized control system of ADRs, on the other hand, the existing plan can be maintained as there is no discretion by individual couriers.

The most prominent advantage of ADR operation is its flexibility. Unforeseen events and uncertainties can disrupt predetermined delivery plans, which emphasizes the importance of the ability to adapt readily to new changes. As communication between the dispatcher and couriers is more nimble and straightforward using ADRs, new opportunities for system improvements arise. Notably, there exist cases where reassigning pre-assigned orders enables better solutions. Reassignment of orders is considered impractical when utilizing human drivers because human drivers behave more like individual entities. Canceling previous assignments and assigning orders to new drivers could lead to systemic confusion. However, changing the delivery schedule is more convenient for ADR-based delivery, where robot delivery schedules are managed centrally. Consequently, the modification of the previous assignment becomes more realistic.

Even with the competitive advantages, several challenges are present that delay the adoption of ADR in dynamic pickup and delivery operations. ADR lacks the ability as a courier to respond to emergencies through independent decision-making. While there are certain situations where the judgment of a human courier leads to better consequences, currently developed ADRs struggle to perform such roles. These limitations make businesses reluctant to replace human couriers with ADRs. Essential environmental conditions, such as the need for environments without stairs, also limit the scope of use of ADR operations.

Lack of practical experience and related studies are unavoidable and critical obstacles in ADR adoption. As ADR is in the early implementation phase of technology, usage cases are inevitably scarce. The lack of comprehensive studies indicates a significant research gap in this area. Among current literature related to ADR utilization in delivery, not many studies incorporated unique characteristics of ADR. The distinctive benefits of ADR, such as flexible operation, reduced cost, and longer service time, are not harnessed sufficiently. Many studies inadequately reflected several factors in utilizing ADR. A notable example is battery management, where simplified management schemes based on unrealistic assumptions were introduced in many articles.

In this study, we leveraged the opportunity obtained through ADR use to make more efficient dispatching decisions in every period of delivery. We propose a reassignment policy that reassigns previously assigned orders with the newly arriving order. We also present a battery management strategy applicable in practical operation and a peak time management methodology concerning batteries. Such methodologies are devised and implemented to increase service quality by minimizing the number of rejected orders and reducing customer waiting times. Through extensive experimentation results, we confirm that the two proposed solution methodologies yield improved outcomes for the problem.

Contributions of this study can be outlined as follows:

1. We propose a mathematical formulation of pickup and delivery problems with battery constraints and rejecting options included in the formulation.
2. We incorporate relevant factors of the emerging field of ADR operation into the problem definition. An easily applicable battery management methodology for ADR operation is introduced.
3. We develop a novel dispatching policy and demonstrate improved results compared to the current policy in use.
4. We constructed test instances to simulate a real operation environment and tested the effectiveness of the proposed algorithm.

The remainder of this paper is organized as follows. Section 2 presents a literature review of related studies. In Section 3, we describe the aspects of the problem and formulate a mathematical model for the problem. We propose methodologies developed to achieve improved solutions in Section 4. In Section 5, we present test results of the proposed methodologies compared to benchmarks. In Section 6, the paper concludes with a summary and suggests future research opportunities.

2. Literature review

In this section, we provide an overview of relevant research. It is well known that optimization problems with stochastic and dynamic characteristics are challenging. In addition, algorithmic strategies suggested in the transportation field cannot usually scale up to real-world problems (Powell, Simao, & Bouzaiene-Ayari, 2012). Various studies emerged that sought effective solutions to practical concerns. We review several categories regarding delivery problems in dynamic situations and fleet management using autonomous vehicles.

2.1. Dynamic pickup and delivery

The dynamic pickup and delivery problem (DPDP) is the field most closely related to this study. DPDP is a variant of pickup and delivery problem, where items or people should be transported from the pickup location to the delivery location to satisfy demand. A distinction is made in DPDP in that demand arrives dynamically during the time horizon. Extensive reviews of DPDP can be found in Cai et al. (2023), Psaraftis, Wen, and Kontovas (2016) and Soeffker, Ulmer, and Mattfeld (2022). In particular, Cai et al. (2023) presents a recent review of the literature and several applications on this topic.

The choice of objective function varied, depending on the focus of the study. Several studies that emphasized minimizing operational costs aimed to reduce costs incurred while using the service. Sun, Yang, Shi, and Zheng (2019) considered a real-time distribution strategy in urban areas in the DPDP context to minimize the total distribution cost of vehicles. The dynamic insertion method was adapted from the heuristic algorithm developed in the study to expand the range of solution searches. Arslan, Agatz, Kroon, and Zuidwijk (2019) provided a variant of DPDP in which ad-hoc drivers participate as crowdsourcers for the transportation service. Based on the routing solution from the subproblem, optimal matching of jobs and drivers is obtained in every iteration of the rolling horizon approach. Crowdsourcing improved the system's flexibility, with more tasks being matched to the drivers, and decreased the total cost. Additionally, studies such as Ma, Hao et al. (2021), Tirado and Hvattum (2017) and Zhu and Sheu (2018) proposed frameworks for cost minimization in DPDP.

Other studies that put more emphasis on the customer side developed solutions to maximize the service level, such as the number of accepted requests, or to minimize customer dissatisfaction, such as customers' waiting times. Cortés, Sáez, Núñez, and Muñoz-Carpintero (2009) focused on minimizing the user cost, composed of users' total travel time and waiting time. The stochastic effect of DPDP, which has not been considered a factor of importance, was considered with high significance. Predicting future states was included in dynamic state formulation to prepare for rerouting in the future. Solution methodology based on particle swarm optimization provided improved results compared to myopic models. Tao, Zhuo, and Lai (2023) aimed to satisfy passenger welfare by minimizing travel time while minimizing the energy cost of electric vehicles (EVs). Constraints related to the charging station, the battery status, and road conditions were considered, and routing decisions were made to find the best routes for each EV. Other publications, such as those by Cheng, Liao, and Hua (2017), Ghiani, Manni, Quaranta, and Triki (2009), and Sheridan et al. (2013), also intended to search for methodologies that could enhance the service level provided to customers.

There have been extensive studies on dynamic pickup and delivery in recent years to address problems related to the expanding delivery industry. Additional conditions such as time windows, multiple depot locations, and the use of green vehicles were dealt with in various studies. Ghiani, Manni, and Manni (2022) introduced an anticipatory algorithm that can be applied to real-time fleet management using parametric policy function approximation. The estimation of instance features was updated dynamically and applied to the base policy with a supervised learning model. Farazi, Zou, and Tulabandhula (2022) introduced a heuristic-embedded deep reinforcement learning (DRL) algorithm to solve the problem in a crowdshipping circumstance. The availability and capacity of each participant were also uncertain in the problem, and the trained DRL model was shown to decrease the shipping cost significantly. Xu and Wei (2023) integrated transshipment and a last in first out constraint to the DPDP context. Several heuristics and Q-learning algorithms were applied in solution methodology to modify decisions according to dynamic demand. The results of the bi-objective optimization problem posed the trade-off relationship between customer satisfaction and vehicle traveling distance.

Other recent publications also addressed DPDP in various application scenarios. Similar to Du, Zhang, Wang, and Lau (2023) and Xu and Wei (2023) also addressed the problem with the last in first out constraint to better represent the cases for single access point vehicles. Several intuitive strategies were implemented in the hierarchical optimization framework to minimize delay after deadline. Cai, Zhu, Lin, Ming, and Tan (2024) proposed a decomposition-based solution method that decomposes a multi-objective DPDP into subproblems. A tabu search algorithm was effectively applied to excel benchmark algorithms concerning total cost. Gao, Zhang, Zhang, and Zhao (2024) discussed the setting with electric vehicles to serve passengers and parcel orders that arrive with uncertainty. A reinforcement learning-based algorithm was developed along with two rolling horizon heuristics.

Table 1 compares relevant literature and this study. There have been publications that modeled operations with ADR adoption. As it is more realistic to consider deliveries to have specific deadlines in practical delivery contexts, various studies have included constraints related to time windows. Most of the studies assumed an environment with sufficient resources to satisfy all demands, and the main focus was on how to make the best use of the given resources to complete all tasks. Given that managing the current fuel/battery level and deciding the timing of recharge impose additional complexity on the problem, constraints and decisions related to the battery were excluded in many studies. In comparison, this study considered the possibility that not enough couriers are present to meet every customer's deadline, which is a frequently encountered situation in newly launched services. In addition, battery recharge constraints and a battery managing scheme were introduced to enhance the real-world applicability of the model.

The research most similar to our study was conducted by Ulmer et al. (2021). This research proposes a restaurant meal delivery problem, a modified version of the dynamic pickup and delivery problem. Demand arrival is stochastic, and a dispatching decision is made at the arrival of each order. The unknown orders that arrive in the future makes decision-making more challenging, as the best decision in the present may not be the best decision with newly arrived orders. As making changes to the previous assignment is impossible in this problem, a policy that includes the time buffer and postponement of orders is presented to overcome this uncertainty.

In comparison to Ulmer et al. (2021), using ADRs in this study provides an opportunity for modification in the previous assignment, as the entire fleet of ADRs is controlled via a centralized system. Therefore, we harnessed the advantage of utilizing ADRs and devised a reassignment policy that provides the potential for more flexible operation. In order to cope with the battery constraints arising from the use of ADR, a method for efficiently charging the battery was proposed to minimize the impact on overall utilization. In addition, unlike Ulmer et al. (2021), we exploited demand information from the external demand factors and proposed methodologies to better handle peak period demand.

2.2. Dynamic dispatching

Problems in numerous fields of study share similarities with DPDP, namely dial-a-ride and same-day delivery problems. In same-day delivery problems, orders must be delivered by the day's end and are dispatched to a fleet of vehicles. Klapp, Erera, and Toriello (2018) proposed the formulation of a deterministic model for the same-day delivery system. Based on the deterministic model, an a priori solution approach was developed, along with the creation of several dynamic policies. An analysis explored the trade-off relationship between minimizing the total cost and maximizing order coverage. Van Heeswijk, Mes, and Schutten (2019) used linear value function approximation, and an approximate dynamic programming algorithm was proposed to solve a delivery dispatching problem in urban consolidation centers. Voccia, Campbell, and Thomas (2019) proposed a routing strategy considering information about future requests. Online purchases were considered in the study, and analysis related to the distribution of time windows was conducted.

2.3. Autonomous vehicle operation

With the adoption of autonomous vehicles rapidly influencing various fields, a growing number of studies have considered fleet management of autonomous vehicles (AVs). Srinivas, Ramachandiran, and Rajendran (2022) provided a literature review on autonomous robot-driven delivery. Jun, Lee, and Yih (2021) solved the pickup and delivery problem with autonomous mobile robots. A novel mathematical model incorporating a recharging design was introduced, and two heuristic algorithms were presented. Bongiovanni et al. (2019) proposed a mixed-integer linear programming formulation representing an electric autonomous dial-a-ride problem. While the study looked at finding the most cost-effective route to serve previously arrived orders, the study also looked at the timing of charging station visits and the amount of necessary battery recharging.

Among the research that covered AV operations, literature on shared-use electric vehicle (EV) operations holds significant relevance to this study. Although EVs possess a clear advantage in environmental aspects, facile adoption has proved challenging due to several shortcomings, such as more extended battery charging time, shorter travel distance, and higher cost. Loeb and Kockelman (2019) concentrated on this trade-off and analyzed the effectiveness of adopting EVs to shared use vehicle services. As associated costs of EVs are gradually decreasing, the adoption efficiency was tested on various cost scenarios. The result disclosed that enhanced performance was found with reduced charging times and improved travel distances, although performance fell slightly short of that exhibited by gasoline-powered vehicles. Hyland and Mahmassani (2018) addressed the on-demand shared-use AV mobility service (SAMS) problem in dynamic scenarios, treating it as a sequential stochastic control problem. The study emphasized dispatching decisions over routing decisions and proposed six AV-traveler assignment strategies. Al-Kanj, Nascimento, and Powell (2020) proposed an approximate dynamic programming method for assigning travel requests to vehicles and deciding the timing and amount of recharging. With the given information expressed as the

Table 1
Comparison of related studies and this study.

Author (Year)	Problem type	ADR	Time windows	Rejection	Fuel/Battery recharge
Hyland and Mahmassani (2018)	Dynamic vehicle routing	✓			
Bongiovanni, Kaspi, and Geroliminis (2019)	Dial-a-ride	✓	✓		✓
Steever, Karwan, and Murray (2019)	Dynamic pickup and delivery		✓		
Ulmer and Streng (2019)	Same-day delivery	✓			
Ulmer, Thomas, Campbell, and Woyak (2021)	Dynamic pickup and delivery		✓	✓	
Xu and Wei (2023)	Dynamic pickup and delivery		✓		
Gao et al. (2024)	Share-a-ride		✓	✓	✓
This study	Dynamic pickup and delivery	✓	✓	✓	✓

state of a Markov decision process, the impact of decisions made in the present on the future was examined. Compared to the myopic policy, the approximate dynamic programming approach with hierarchical aggregation generated more revenue, with reduced recharging events during peak hours.

Most of the literature is based on a simulation-based approach, suggesting heuristics applicable in real-world instances. Compared to other fields of research concerning vehicles, research that looks at improving and expanding the use of ADRs is limited. Mainly, research is needed that adequately incorporates real-world decisions.

2.4. Battery recharge

Prior research on recharging has mainly been conducted on vehicle routing problem (VRP) and pickup and delivery problem (PDP) with electric vehicles and drones (Ma, Hu, Chen, Wang and Wu, 2021; Messaoud, 2022). These studies can be classified into three categories according to the recharging strategy: full recharging, partial recharging, and battery swapping. Full recharging indicates the battery being charged to a complete level every recharge. This strategy can reduce the probability of incurring an empty battery during operation and is relatively more straightforward to implement, so many studies adopted the strategy as the battery management scheme (Chen, Zhang, Pourbabak, Kavousi-Fard, & Su, 2016; Erdoğan & Miller-Hooks, 2012; Hiermann, Puchinger, Ropke, & Hartl, 2016; Lin, Zhou, & Wolfson, 2016; Schneider, Stenger, & Goeke, 2014; Zhang, Zhang, Gajpal, & Appadoo, 2019).

The partial recharging strategy recharges only a portion of the battery at every stop. Choosing the amount of recharge varies depending on the study. Vehicles can be recharged by a fixed amount of battery every time, or the recharging amount can be considered as a decision variable and determined considering future requests (Bongiovanni et al., 2019; Cortés-Murcia, Prodhon, & Afsar, 2019; Keskin & Çatay, 2016; Macrina, Pugliese, Guerriero, & Laporte, 2019; Sassi & Oulamara, 2017). Because the recharging amount per recharge is smaller than full recharging, the partial recharging strategy enables more flexible management and improved system performance. Several studies combined full and partial recharging to aid in making decisions when using a more adaptable method (Felipe, Ortuño, Righini, & Tirado, 2014; Jun et al., 2021). Lastly, the battery swapping strategy replaces the battery with a fully charged battery at every recharge stop. It is similar to the full recharging strategy with a shortened recharging time. Battery swapping is being widely adopted in multiple studies (Adler & Mirchandani, 2014; Hof, Schneider, & Goeke, 2017; Jie, Yang, Zhang, & Huang, 2019; Masmoudi, Hosny, Demir, Genikomsakis, & Cheikhrouhou, 2018; Sayarshad, Mahmoodian, & Gao, 2020; Soysal, Cimen, & Belbağ, 2020; Verma, 2018). Still, other aspects, such as vehicle structure, battery configuration, and replacing personnel should be suited to enable battery swapping.

To summarize the literature review, recent studies on dynamic pickup and delivery have been actively conducted due to the increase in related services. Studies with similar structures, such as dial-a-ride, same-day delivery, and dynamic dispatching, are also being actively pursued. Meanwhile, the advancement of related technologies has led to the widespread adoption of autonomous vehicles and robots in various fields. Consequently, research has been conducted on incorporating autonomous vehicles into existing systems and applying their unique characteristics. Finally, concerning battery charging, recharging strategies can be categorized into three groups, and the advantages and disadvantages of each strategy have been analyzed.

3. Problem description and mathematical model

In this section, we state the problem and propose a mathematical formulation. A detailed explanation of the problem is presented in Section 3.1. Section 3.2 presents the battery management methodology applied in this study. Section 3.3 introduces the mathematical formulation of the static version of the problem.

3.1. Problem statement

This study was initiated based on the practical implementation of food delivery services using ADRs at Incheon International Airport. The service is operated by *Woowa Brothers Corp.*, a company with more than 60 percent of market share in South Korea's food delivery app industry. Since 2022, the company has conducted a pilot program of ADR-driven food delivery service at the airport terminal. The method by which service is performed is as follows. People who arrive at the terminal and wait for their flights use the mobile app to place an order. These customers can choose between predefined restaurant lists. Once the order is submitted, it is dispatched to an ADR. The dispatched ADR departs from the station, picks up the food from the restaurant, and arrives at the gate to wait for the customer pickup. After the delivery terminates, the ADR returns to the station.

Based on this service, we introduce a dynamic pickup and delivery problem using autonomous robot (DPDP-AR). This problem is a variant of the well-known dynamic pickup and delivery problem (DPDP). Even though DPDP and related research fields have gained considerable attention in recent years, only a few studies address online order and delivery issues (Cai et al., 2023). Furthermore, the need for problem statements with realistic constraints and more implementable strategies in the ADR-driven delivery literature has come to the forefront (Srinivas et al., 2022). Even though battery management is essential to real-world ADR fleet management, it is not considered in many studies. Most previous studies considering battery constraints adopted battery swapping, which has an implementation advantage but lacks operational efficiency. We seek inspiration from the current research gap and aim to incorporate more realistic constraints, such as battery time and management strategies in centralized vehicle management.

The DPDP-AR is composed of a set of orders $\mathcal{J} = \{J_1, J_2, \dots, J_l\}$ that need to be served by a fleet of ADRs $\mathcal{M} = \{M_1, M_2, \dots, M_m\}$. The time horizon is finite $\mathcal{T} = [0, t_{max}]$. A set of nodes $\mathcal{N} = \{N_1, N_2, \dots, N_n\}$ represents locations that are visited during delivery. An order, s, comprises four characteristics: order time (t^s), deadline (d^s), pickup restaurant (R^s), and delivery location (D^s). The order time and deadline exist within the time horizon $t^s, d^s \in \mathcal{T}$. The pickup restaurant and delivery location are elements of the set of restaurants $\mathcal{R} = \{R_1, R_2, \dots, R_o\}$ and the set of delivery locations $\mathcal{D} = \{D_1, D_2, \dots, D_p\}$ respectively.

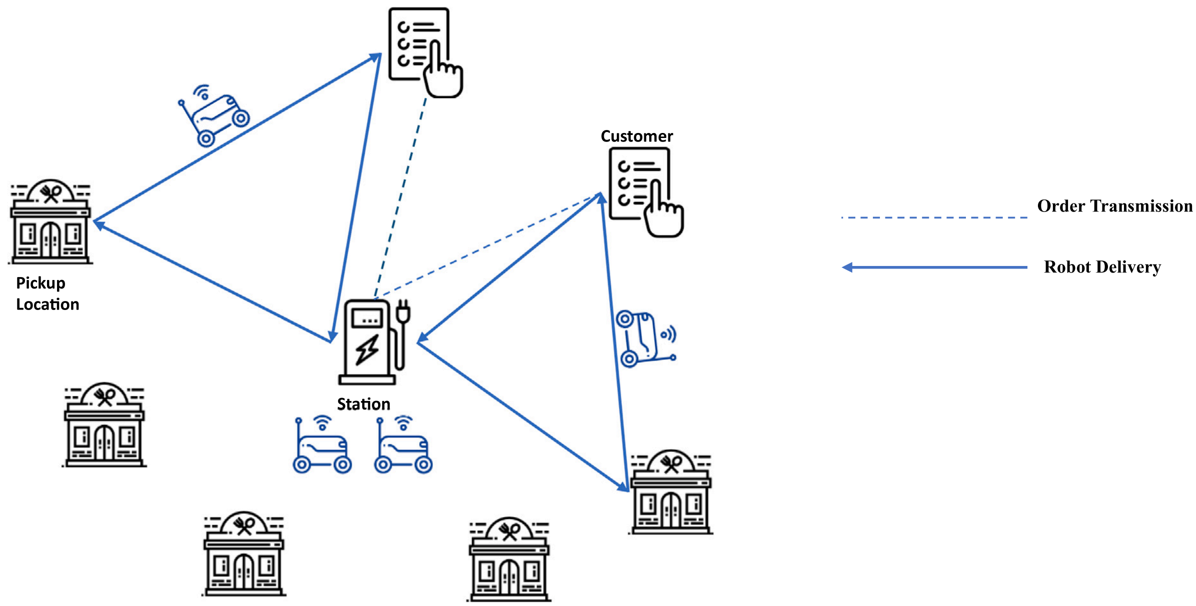


Fig. 1. Illustrative example of DPDP-AR operation.

Every restaurant and delivery location has a corresponding node, $\mathcal{R}$, $\mathcal{D} \subset \mathcal{N}$, and the distance between nodes is determined as the physical distance between two locations. ADRs are located at the ADR station before and after each delivery. The decision epoch occurs each time an order is submitted. At every decision epoch, a decision is made based on the decision state at the time. The elements that constitute the decision state are the earliest available time and battery level of each ADR and the list of orders dispatched earlier but not yet started on deliveries. The earliest available time of an ADR indicates the fastest time the ADR can begin delivering a new order after finishing all the pre-assigned deliveries. With the peak time management scheme described in Section 4.2, consideration of whether it is peak time is also included as an element of the decision state.

The DPDP-AR operation is illustrated in Fig. 1. If an order S is dispatched to an ADR, the order is added to the schedule of the ADR, and delivery starts upon reaching the delivery start time in the schedule. The ADR launches from the station to visit the restaurant R^S to pick up the order. After pickup, the ADR heads toward the delivery location D^S to deliver the food to the customer. The ADR returns to the station once the customer order has been completed. The ADR battery is recharged under certain charging conditions after completion of each order. As the delivery robots follow the predetermined path to serve orders with a fixed starting and ending location, any additional routing decisions are not considered.

We implement the DPDP-AR in the airport food delivery service. The elements of the problem are described below.

1. *Customer orders.* Every order is created from the flight schedule, which indicates that every customer is a passenger on a certain flight. Order arrival is stochastic and becomes known to the system when the order is submitted.
2. *Order arrival interval and deadline.* Every flight has a possible order arrival interval, which denotes the time interval in which the passenger can submit an order. The length of a possible order arrival interval is identical in every flight. There is also a deadline for each order, which is a certain time before the departure time of the corresponding flight. This deadline indicates the boarding time, so it acts as a hard deadline that makes it impossible for the customer to receive late deliveries. The elements are depicted in Fig. 2.
3. *Autonomous robot delivery.* The ADRs are fully charged at the start of the operation. The battery is discharged during delivery, and the discharged amount is proportional to the distance traveled. One order is served per trip.
4. *Penalty of rejection.* Each rejection of an order incurs a penalty cost of α.

The dispatcher aims to maximize the level of customer satisfaction provided by the service. In other words, the objective is to minimize dissatisfaction incurred by the service. Such dissatisfaction can occur in two ways. If an order is served, the customer's waiting time is calculated as the time between the order and the delivery arrival time. As the customer waits longer for the delivery, the customer's dissatisfaction increases. If an order is rejected, a penalty cost of α is imposed per rejection. The objective is to minimize the sum of two elements of dissatisfaction. The objective function can be written as follows:

$$min \sum_{i \in \mathcal{O} \setminus \mathcal{O}^{\mathcal{R}}} (z^i - r^i) + \alpha \left| \mathcal{O}^{\mathcal{R}} \right| \tag{1}$$

$\mathcal{O}$ indicates a set of all orders, $\mathcal{O}^{\mathcal{R}}$ indicates a set of rejected orders, z^i indicates the delivery arrival time of order i, r^i indicates the order release time of order i.

One possible distortion in the objective function value is due to the bi-objective characteristic. The delayed time for served orders has specific time units, while the unit penalty α for rejected orders does not. If α is set relatively low, the resulting solution would reject orders even when it is possible to serve it, which is inconsistent with the goal of this study. If α is set relatively high, ensuring fast delivery for served orders would not be considered as significant as needed in the objective function value. In order to handle this issue with the parameter value, we set α as marginally higher than the maximum possible delay while maintaining proximity in value. This method of parameter value choice would balance the impact of rejection while also inducing the desired manner of serving as many customers as possible.

In this study, the following assumptions are made.

1. Once an order is submitted, the customer cannot withdraw the order.
2. Customers make orders only if they have enough time left to receive delivery, considering the travel time and deadline.
3. Customers have a homogeneous preference on the choice of restaurants to deliver to.

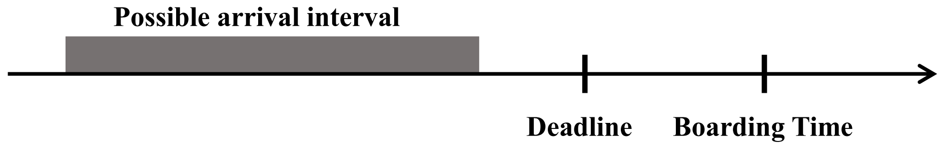


Fig. 2. Illustrative example of order arrival interval and deadline.

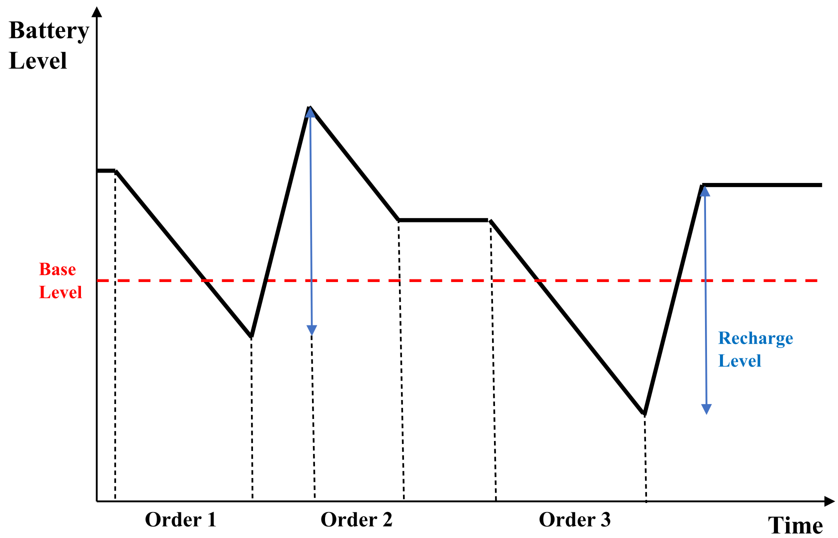


Fig. 3. Overview of the (BL, RL) policy.

4. The ready time of restaurants is zero. In other words, the ADR can pick up the demanded item once it arrives at the restaurant without any delay.
5. The travel time of ADRs for the delivery is deterministic, and is proportional to the distance traveled. Assuming a constant speed of the ADR and no congestion in the environment, the travel time is the distance traveled divided by the speed of the ADR.

3.2. Battery management

As the movement toward zero carbon emissions is gaining importance worldwide, electrically-powered vehicles such as autonomous vehicles and drones are utilized in various applications. Some preliminary work was carried out to design infrastructure for these vehicles, including charging station locations (Hwang & Lim, 2022; Kchaou-Boujelben & Gicquel, 2020). One common deficiency of electrically powered vehicles is the limited capacity of batteries. The complete battery capacity is more limited, and recharging takes longer. Therefore, deciding on battery usage and recharging emerged as a complex combinatorial optimization problem.

In this study, we applied a newly devised partial recharging strategy to perform battery management. The proposed strategy is similar to the well-known (r, Q) policy used in inventory management. As in (r, Q) policy where a quantity of Q is ordered every time the inventory level reaches or falls below r, a predefined quantity RL is charged if the battery level drops below a certain level BL after completion of an order. This paper refers to this parameterized policy as (BL, RL) policy, where BL indicates base level and RL indicates recharge level. Fig. 3 illustrates how the policy operates. Battery level is reviewed after completion of each order. In Fig. 3, after delivery of Order 1 and Order 3, the battery level is shown as being under the base level. Therefore, a fixed amount of charge (recharge level) is recharged. In the case of Order 2, because the battery level after completion is not below the base level, no recharge is performed.

The (BL, RL) policy has competitive advantages over alternative management policies. As this policy shares commonalities with partial recharging, the smaller size of the individual recharge units in the (BL, RL) policy allows for greater operational flexibility than does the full recharging strategy. The partial recharging strategy is known to increase system efficiency but comes at the expense of greater computational complexity. Especially in previous studies where recharging times at each stop were considered as decision variables, most situations had information about the upcoming order available, making it relatively straightforward to determine the appropriate recharge level. However, as the circumstances of this study are dynamic, and the arrival of future orders is uncertain, applying previous methodologies may not guarantee efficiency. Charging the predefined amount of RL in every recharge in (BL, RL) policy helps use partial recharging while mitigating the drawbacks associated with partial recharging. To deal with uncertainties in the problem and make prompt recharging decisions simultaneously, we developed and implemented a (BL, RL) policy with an adaptability advantage and reduced complexity.

3.3. The static model

We formulate the static model of the problem. While the system acquires information about an order at the ordering point in the actual case, this model considers a virtual case where all the information about demand is known before the start of the operation. As the feasible region of the static case includes the feasible region of the dynamic case, obtaining more efficient solutions is expected. This model is a variation of a static pickup and delivery problem. As is the case for the dynamic situation in this study, the routing decision is not considered. Instead, robots follow predefined routes between locations.

This problem shares a similar structure with the job-shop scheduling problem in that an optimal sequence of individual jobs needs to be sought. We adopt the graph-based formulation of the identical machine scheduling problem introduced in Yalaoui and Nguyen (2021). The mixed-integer linear programming (MILP) formulation comprises nodes representing jobs and arcs indicating predecessor–successor relationships between orders. An initial node exists, and nodes that represent the first job of each machine have connecting arcs with the initial node. We modified the formulation presented by Yalaoui and Nguyen (2021) to suit our problem. First, a rejection node connected to the initial node is introduced. Second, elements of the problem, such as order placement time and a hard deadline, are added as constraints. Lastly, the battery management scheme introduced in Section 3.2 is also integrated into the model. Such management implementations are battery discharge proportional to the processing time and charging under a certain battery level. The decision about when to process orders in which dispatch sequence is made in the model.

The indices, sets, parameters, and decision variables utilized in the mathematical formulation are defined as follows:

Indices and sets

$\mathcal{J}$	Set of orders, $j \in \mathcal{J} = \{1, 2, \ldots, n\}$
$\mathcal{J}'$	Set of orders including job t, $j \in \mathcal{J}' = \{1, 2, \ldots, n, t\}$
$\mathcal{J}''$	st of orders including job 0, $j \in \mathcal{J}'' = \{s, 1, \ldots, n\}$
$\mathcal{J}'''$	Set of orders including job $0, n+1$, $j \in \mathcal{J}''' = \{s, 1, 2, \ldots, n, t\}$
$\mathcal{M}$	Set of robots, $i \in \mathcal{I} = \{1, 2, \ldots, m\}$

Parameters

r_j	Order release time of order j
p_j	Processing time of order j
d_j	Deadline of arrival of order j
α	Penalty of rejection
$\overline{C}$	Upper bound of completion time
β	Full battery capacity of each robot
γ	Recharge rate
δ	Discharge rate
π	Battery recharge level point
ρ	Battery recharge quantity

Decision variables

b_j	Battery level of the robot after completion of order j
x_{ij}	1 if order j is processed right after order i on the same robot, 0 otherwise
y_j	1 if battery is recharged after completion of order j, 0 otherwise
z_j	Completion time of order j
w_j	Completion time of order j including recharge

The mathematical model is as follows:

$$\min \quad \sum_{j\in\mathcal{J}} (z_j - r_j) + \alpha \sum_{j\in\mathcal{J}} x_{tj} \tag{2}$$

$$\text{s.t.} \quad z_j \geq r_j + (1 - x_{tj})p_j, \quad \forall j \in \mathcal{J} \tag{3}$$

$$w_j = z_j + \frac{\rho}{\gamma} y_j, \quad \forall j \in \mathcal{J} \tag{4}$$

$$z_j \geq w_i + p_j - \overline{C}(1 - x_{ij}) \quad \forall i \in \mathcal{J}'' \forall j \in \mathcal{J} \tag{5}$$

$$z_j \leq d_j + \overline{C} x_{tj} \quad \forall j \in \mathcal{J} \tag{6}$$

$$\sum_{i\in\mathcal{J}'''} x_{ij} = 1 \quad \forall j \in \mathcal{J}' \tag{7}$$

$$\sum_{i\in\mathcal{J}} x_{ji} \leq 1 - x_{tj} \quad \forall j \in \mathcal{J} \tag{8}$$

$$\pi - b_j \leq \beta y_j \quad \forall j \in \mathcal{J}'' \tag{9}$$

$$\pi - b_j \geq \beta(y_j - 1) \quad \forall j \in \mathcal{J}'' \tag{10}$$

$$b_j \geq b_i + \rho y_i - \delta p_j - \beta(1 - x_{ij}) \quad \forall i \in \mathcal{J}'', \forall j \in \mathcal{J}' \tag{11}$$

$$\sum_{j\in\mathcal{J}'} x_{sj} \leq m + 1 \tag{12}$$

$$z_s = 0 \tag{13}$$

$$z_t = 0 \tag{14}$$

$$b_s = \beta \tag{15}$$

$$x_{st} = 1 \tag{16}$$

$$x_{jj} = 0 \quad \forall j \in \mathcal{J}''' \tag{17}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in \mathcal{J}''' \tag{18}$$

$$y_j \in \{0, 1\} \quad \forall j \in \mathcal{J} \tag{19}$$

$$0 \leq b_j \leq \beta \quad \forall j \in \mathcal{J} \tag{20}$$

$$z_j \geq 0 \quad \forall j \in \mathcal{J} \tag{21}$$

$$w_j \geq 0 \quad \forall j \in \mathcal{J} \tag{22}$$

In the formulation, we added artificial nodes s and t in the graph formulation of the problem. Node s is the start node used to ensure the constraint related to the number of robots. Node t is the rejection node, and all rejected orders are succeeding nodes of this node. By introducing two artificial nodes, we formulated this problem as a tree where each node is connected to the preceding and succeeding node of the corresponding robot. In addition to set J, which includes all nodes that correspond to proper orders, sets J', J'', and J''' were introduced to include either node s or node t to implement constraints for the graph model formulation.

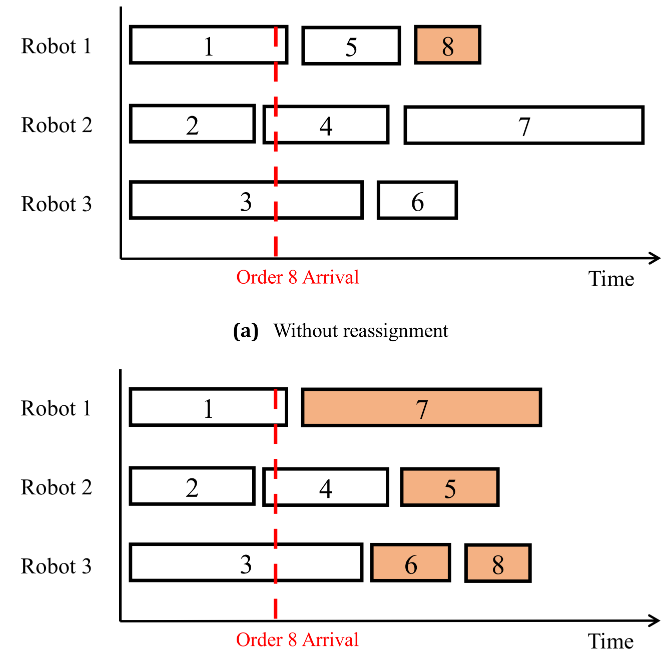
The objective function of the problem (2) minimizes the weighted sum of the waiting time for accepted orders and the penalty cost for rejected orders. The waiting time indicates the time between order placement and delivery completion. Constraint (3) ensures the time relation within orders. Constraint (4) expresses the balance equation in the case of battery recharging. Constraint (5) ensures the time relation between consecutive orders. Constraint (6) requires order completion before the deadline for accepted orders. Constraint (7) requires every order to have a preceding order. Constraint (8) ensures every order except for the rejected orders has at most one subsequent order. Constraints (9) and (10) is used to apply rules related to battery recharging. If battery falls below the predetermined level π, battery is recharged, and otherwise it is not recharged. Constraint (11) provides a balance equation for the battery in case of recharging. Constraint (12) guarantees the number of robots. Constraints (13) and (14) initialize completion time and battery level before the start of operation. Constraint (15) ensures that battery is fully charged at the start of operation. Constraints (16) and (17) provide conditions for the graph model. Constraints (18)–(22) enforce the domain of binary variables and real variables.

4. Solution methodology

In this section, we introduce the solution methodology for the proposed DPDP-AR. We devised an event-based rolling horizon method to handle the demand that arrives dynamically during the time horizon. In this method, a dispatching decision of orders is made every time an order arrives. Section 4.1 presents the reassignment policy utilized in the rolling horizon framework inspired by the beneficial characteristics of using ADRs. Section 4.2 proposes a methodology that includes adaptive battery management to handle peak time demand.

4.1. Reassignment policy

One advantage of utilizing an ADR setup is the flexibility it brings in processing orders. Rescheduling orders can significantly burden the system in human courier-based delivery, as adjustment and notification of new schedules confuse individual drivers. Therefore, studies that utilized human drivers in dynamic pickup and delivery problems and dial-a-ride problems attempted to find effective dispatch methods under given circumstances without the possibility of rescheduling. The issues



(a) Without reassignment

(b) With reassignment

Fig. 4. Comparison of schedule with and without reassignment.

were handled in either of two ways. One approach is the reactive policy, which decides based on the requested orders. Reyes, Erera, Savelsbergh, Sahasrabudhe, and O'Neil (2018) introduced a rolling horizon algorithm in which orders were assigned in every period with given information. The other approach is an anticipatory policy, where information about future requests influences decisions. Ghiani et al. (2022) designed an anticipatory policy applicable in large instances. A new parametric policy function approximation approach that exploits real-time instance information was developed. Li et al. (2021) also suggested a machine learning-based anticipatory policy that estimates future demand distribution with arrived requests.

Considering the interval between orders, the solution time at each period should be fast enough to maintain ongoing service in the real-world application. In this study, we leverage the traits of ADR and introduce the reassignment policy. In brief, the reassignment policy dispatches multiple orders that satisfy a particular criterion, including the new order at every decision epoch.

A comparison of assignments with and without reassignment is illustrated in Fig. 4. Previously assigned orders of each robot are presented as a schedule in the timeline, and two cases of order assignment at the arrival of order 8 are shown in Figs. 4(a) and 4(b). We introduce the term *waiting order* to indicate orders that arrived and were assigned previously but have not yet commenced delivery at the assigned ADR. In Fig. 4(a), orders 5, 6, and 7 are *waiting orders* as their delivery start time is after the red line that indicates the arrival of order 8. If reassignment is not considered, Order 8 is assigned to robot 1, as in Fig. 4(a). If reassignment is considered, *waiting orders* and the new order are all considered assignable, as in orders 5, 6, 7, and 8. In this case, order 8 is assigned to robot 3, and Orders 5, 6, and 7 are transferred to another robot compared to the original schedule. A more balanced schedule of orders is established in Fig. 4(b).

As a result of reassignment, future operations can benefit from increased flexibility. Considering that more orders are the subject of assignment in the reassignment policy, more efficient solutions can be found in every decision epoch. Moreover, a reassignment policy can overcome the systemic congestion issue when orders with long travel times arrive, leading to decreased utilization. These lengthy and burdensome orders can be reassigned to the most unoccupied robot in every period, which will, in the long term, lead to an increase in the whole system's capacity.

Algorithm 1 illustrates the reassignment procedure that operates every time a new order arrives. The input is the result of the algorithm operation at the arrival of the immediately preceding order, with the information about the newly arrived order. First, the *waiting orders* are classified from the assigned order set and are added to the unassigned order set, U. Each ADR's battery level and earliest available time are updated to incorporate the exclusion of *waiting orders* from the assigned order set. The new order is added to set U, and RP, formulated by mixed-integer linear programming (MILP), is solved to assign the orders in U to ADRs. The assigned order set, rejected order set, and the completion time of orders are updated by using the solutions of RP. Subsequently, the ADRs' battery levels and earliest available times are updated according to the reassignment solution. Lastly, the (BL, RL) policy introduced in Section 3.2 is implemented. After the completion of assigned orders, the ADRs with battery levels lower than BL are recharged in the amount of RL. The earliest available time and battery level of the recharged robot are updated.

Algorithm 1 Reassignment algorithm

Input: set of assigned orders A, set of rejected orders B, completion time of assigned orders C_i, status of each ADR(battery level b_r, charge time c_r, earliest available time e_r), information about the new order n(order time o_n, processing time p_n, deadline d_n)
Output: updated A, B, C_i, e_r, b_r
Initialize unassigned order set U;
$t \leftarrow o_n$;
for $r \in \mathcal{R}$ **do**
  if $c_r < t$ **then**
    Add $waiting\ orders$ of r to U;
    Remove $waiting\ orders$ of r from A;
    Update b_r, e_r;
  end
end
Add the new order n to $\mathcal{U}$;
Solve RP on the elements of U and get the reassigned solution;
Update A, B, C using the RP solutions;
Update b_r, e_r according to the RP solutions;
for $r \in \mathcal{R}$ **do**
  if $b_r \leq BL$ **then**
    $e_r \leftarrow e_r + \frac{RL}{rc}$;
    $b_r \leftarrow b_r + RL$;
    $c_r \leftarrow e_r$;
  end
end

In the MILP formulation of RP, a new parameter, e_i, is added to indicate the earliest available time of robot i. This indicates the earliest time robot i can embark on the new order after completing all the pre-assigned orders, and the value is transmitted from the iteration immediately before. As there is a difference in the earliest available time of robots during operation, the identical robot scheduling of the static model had to be modified. Therefore, binary decision variable f_{ij} was introduced to signify if order j is the first order of robot i.

RP

$$\min \quad \sum_{j\in\mathcal{J}}(z_j - r_j) + \alpha \sum_{j\in\mathcal{J}} x_{tj} \tag{23}$$

$$\text{s.t.} \quad z_j \geq r_j + (1 - x_{tj})p_j, \qquad \forall j \in \mathcal{J} \tag{24}$$

$$z_j \geq z_i + p_j - \overline{C}(1 - x_{ij}) \qquad \forall i \in \mathcal{J}'', \forall j \in \mathcal{J} \tag{25}$$

$$z_j \leq d_j + \overline{C} x_{tj} \qquad \forall j \in \mathcal{J} \tag{26}$$

$$\sum_{i\in\mathcal{J}'''} x_{ij} = 1 \qquad \forall j \in \mathcal{J}' \tag{27}$$

$$\sum_{i\in\mathcal{J}} x_{ji} \leq 1 - x_{tj} \qquad \forall j \in \mathcal{J} \tag{28}$$

$$\sum_{i\in\mathcal{M}} f_{ij} - 1 \leq 1 - x_{sj} \qquad \forall j \in \mathcal{J} \tag{29}$$

$$\sum_{i\in\mathcal{M}} f_{ij} - 1 \geq x_{sj} - 1 \qquad \forall j \in \mathcal{J} \tag{30}$$

$$\sum_{j\in\mathcal{J}} f_{ij} \leq 1 \qquad \forall i \in \mathcal{M} \tag{31}$$

$$z_j \geq e_i + p_j - \overline{C}(1 - f_{ij}) \qquad \forall i \in \mathcal{M}, \forall j \in \mathcal{J} \tag{32}$$

$$\sum_{j\in\mathcal{J}'} x_{sj} \leq m + 1 \tag{33}$$

$$z_s = 0 \tag{34}$$

$$z_t = 0 \tag{35}$$

$$x_{st} = 1 \tag{36}$$

$$x_{jj} = 0 \qquad \forall j \in \mathcal{J}''' \tag{37}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in \mathcal{J}''' \tag{38}$$

$$z_j \geq 0 \qquad \forall j \in \mathcal{J} \tag{39}$$

$$f_{ij} \in \{0, 1\} \qquad \forall i \in \mathcal{M}, \forall j \in \mathcal{J} \tag{40}$$

The objective function (23) and Constraints (24)–(28), (33)–(39) are identical to the static model, with constraints related to battery management excluded. Constraints (29) and (30) ensure that the first orders of each robot have the corresponding f_{ij} value. Constraint (31) expresses the necessary conditions for variable f_{ij}. Constraint (32) guarantees new delivery starts after the earliest available time of each robot. Constraint (40) enforces the domain of the binary variable.

4.2. Peak time management

One distinctive characteristic of this problem is that an external source allows the estimation of demand distribution. As all the customers are passengers waiting for their flights, information about demand can be extracted from the flight schedule. The gained information becomes the source of developing adaptive strategies for the varying demand. Estimating peak time is especially important from an operational perspective, as marginal cost or benefit rapidly increases near the peak hours (Wu, Yücel, & Zhou, 2022).

There are numerous actual operations where the timing of peak demand can be estimated. One type of operation predicts peak demand by exploiting information from an external source. The schedule of the external source serves as a guide for demand estimation. This presumable demand can be found in shops near public transportation terminals, restaurants near megaplex theaters, and many other locales. These circumstances have nearby gathering hubs with scheduled events that influence demand, and the peak periods can be estimated from the schedule of the influencing entity. For instance, restaurants near megaplex theaters can be speculated to have peak demand an hour before the peak time of the theater, as people would have a meal before entering the theater.

Another type of operation has a structure that inherently exhibits a significant difference in demand between peak and non-peak periods. Examples of such operations include restaurants, coffee shops, gyms, and e-commerce ordering platforms. Peak periods of these business processes can be estimated from historical data and used for further improvement in operation.

As demonstrated in the preceding examples, there are various real-life scenarios in which the operator can predict peak periods. The operations vary from offline shops such as restaurants and coffee shops to online ordering systems. Management strategies specifically adaptable for peak periods are essential, as demand in these periods takes up a significant portion. With a differentiated strategy for peak periods, the operator is expected to achieve more efficient results with the same resources.

In this paper, we adopt peak time management to enhance vehicle operation efficiency. Fig. 5 explains the process of deriving the peak time. First, the order arrival interval is obtained from the flight schedule. The length of the order arrival interval is identical at every flight, ending sometime before the deadline, as shown in Fig. 2. Second, the time horizon is partitioned into time slots of identical length. The number of overlapping order arrival intervals is counted for each time slot. Lastly, based on the counting results of each time slot, we derive the peak time by selecting a certain number of time slots with the highest counts.

In developing a flexible strategy for peak time management, we propose a parameterized management methodology for battery recharging. The method is based on the (BL, RL) policy proposed in Section 3.2. Instead of applying a unique (BL, RL) pair throughout the time horizon, we introduce another pair named (BLP, RLP) employed during peak times. Fig. 6 illustrates the difference in parameter values implemented in peak time management. BLP is set lower than BL to utilize more battery capacity and reduce the number of battery recharges during peak times. RLP is set lower than RL to allow for decreased recharging time for individual recharges. The variations made in these parameters are applied to increase the utilization of ADRs during peak times. Any reduction in the objective function is expected to minimize the possibility of ADRs being inoperative due to recharging.

5. Numerical experiments

In this section, we present the results of numerical experiments. In Section 5.1, we explain the environment and the elements of the test instances, including demand generation. In Section 5.2, the reassignment policy proposed in Section 4.1 is compared with other benchmark policies. In Section 5.3, the effectiveness of peak time management methodology introduced in Section 4.2 is measured.

5.1. Description of the test instances

The experiments are performed to represent the ADR-based food delivery service at Incheon International Airport. Components of the service are pickup restaurants, several service regions that indicate the gates where customers can order, and a station where ADRs return after each delivery. The distance between locations is calculated based on the Incheon International Airport Corporation (IIAC) airport map, depicted in Fig. 7. Experiments were conducted on a PC with an AMD Ryzen 5 7600X 6-Core CP, a 4.70 GHz processor, and 32 GB of RAM with a Windows 10 64-bit system. Test instances were generated by using Python 3.9. Solution approaches, policies, and benchmarks were developed and tested with FICO Xpress 8.12 and Xpress-Optimizer version 38.01.04.

We tested our proposed algorithm on randomly generated instances based on the actual airport flight schedule. From the passenger flight operation records of one month, from July 1, 2023, to July 31, 2023, provided by IIAC, we generated random demand based on the flight schedule. Demand was generated individually for each flight, and every demand arrived between the corresponding flight's order arrival interval. The deadline for an order is set as 30 min before departure to indicate boarding time. The order arrival interval starts 120 min before departure and ends 45 min before departure, as customers are assumed not to order delivery if the boarding time is imminent. The orders' arrival is assumed to follow a Poisson distribution with mean λ, and therefore, the time between orders follows an Exponential distribution with mean $\frac{1}{\lambda}$.

Parameter values were set for the experiment to simulate real-world problems more accurately. The time horizon is 480 min, as the actual operation time is 8 h from 9 a.m. to 5 p.m. The number of ADRs

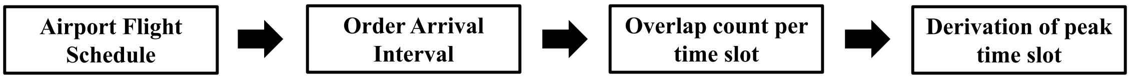


Fig. 5. Process of peak time derivation.

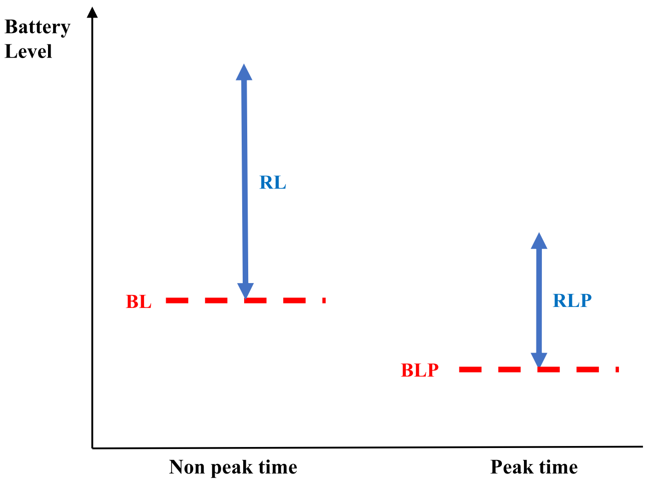


Fig. 6. Illustration of peak time management.

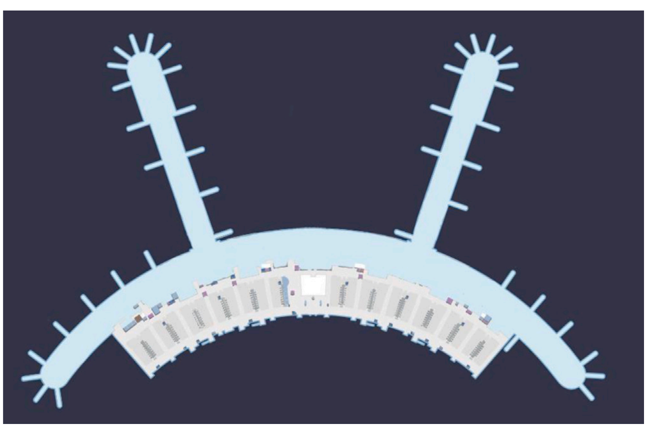


Fig. 7. Map of service delivery area.

is set as 5. In the real-world service, 4 ADRs are operating, but the number is expected to increase in the near future. In implementing the (BL, RL) policy, the base level is 50 percent, and the recharge level is 30 percent. Therefore, a battery amount of 30 percent is recharged every time the battery level falls below 50 percent after completion of an order. The total battery capacity is 100 percent. The discharge rate is 0.0167 percent/s, and the recharge rate is 0.0333 percent/s, which is consistent with the rates of ADRs in operation. In calculating each order's travel time, the ADR's speed and the distance between each node are considered. The speed of the ADR is assigned as a fixed value of 1.5 m/s as the highest speed of the currently utilized model. The distance between nodes was calculated based on the building outline of Incheon International Airport. The travel time is calculated as the distance divided by the speed of the ADR. In calculating the objective function, the penalty of cost rejection α is set as 100. To minimize rejection situations, the value of α is slightly higher than the maximum delay for an order, which is 90 min.

5.2. Performance analysis of reassignment policy

We conducted experiments to assess the effectiveness of the reassignment policy on the generated problem instances. As ADR-driven delivery is a recently introduced service, it is hard to speculate on variations in demand. We aim to demonstrate that our proposed policy can perform well regardless of variations in demand scenarios. Therefore, we assigned different values for the parameter λ of Poisson distribution and experimented on three situations: low, moderate, and high demand. Considering the range of service regions and ADR speed limitations, we set the parameter value while avoiding a steep rise in rejected orders. Information about these situations is listed in Table 2. We generated ten random instances for each demand case for each of the 20 flight schedules, totaling 200 instances.

Table 2
Description of demand cases.

	Low	Moderate	High
λ	0.5	1	1.25
Avg. number of orders	54.6	113.8	147.2

Benchmark algorithms are required to show the performance of the proposed algorithm. We selected two benchmarks: $Myopic_A$ and $Myopic_B$. $Myopic_A$ represents the policy that is commonly used in practice. In the current practice, an order is assigned rather myopically; in other words, an order is assigned to the driver that is considered the fastest to initiate and finish the delivery (Al-Kanj et al., 2020; Ulmer et al., 2021). Therefore, no other information except for the current schedule and availability affects the dispatching decision in this policy. As a variation of $Myopic_A$, we devised $Myopic_B$, where a battery-related scheme is also examined. Instead of selecting the ADR with the best availability, as in $Myopic_A$, two ADRs with the best availability are initially selected as candidates, and the ADR with a higher battery level is dispatched. This policy is built upon the assumption that using batteries evenly between ADRs may increase system efficiency.

In addition to two benchmarks, the static model of the problem is also compared. The signifier "*Static*" is chosen as the result of the model's ability to solve the static pickup and delivery problem introduced in Section 3.3. Assuming that all the information about demand is known before the start of the operation, the result of *Static* provides optimal dispatch and reject decisions for all orders with complete information. As our problem is a dynamic version of the static problem, we employ the *Static* result as a lower bound for the result of other policies.

The performance of the reassignment policy is assessed based on two indicators. First, we compute the objective function value and make a comparison with benchmark algorithms. The objective function comprises the sum of waiting time for accepted orders and the sum of penalty cost for rejected orders, as shown in Eq. (1). Primarily, we compute the improvement of the proposed policy over two benchmarks, $Myopic_A$ and $Myopic_B$. The improvement is defined as the difference in the objective function value of two policies divided by the objective function value of the benchmark:

$$\text{Improvement} = \frac{\text{Benchmark solution} - \text{Policy solution}}{\text{Benchmark solution}} \times 100(\%) \tag{41}$$

Second, the number of rejected orders is compared between policies. We measure the rate at which orders are accepted and rejected. The acceptance and rejection rates are calculated to examine the quality of the dispatching method in terms of service level.

We checked the computational time of each policy. Every order was dispatched in under 20 s of computational time for the reassignment policy and the two dynamic benchmarks. This result makes the policies applicable for the operation's order dispatching/rejection decision-making in dynamic situations. We obtained optimal solutions

Table 3
Experiment results of low demand case.

No.	Objective function				Number of rejections		
	$Static$	RP	$Myopic_A$	$Myopic_B$	RP	$Myopic_A$	$Myopic_B$
1	689.52	704.51	954.87	745.60	0.30	0.30	0.30
2	667.66	684.92	931.02	730.69	0.20	0.20	0.30
3	633.86	659.83	876.95	699.84	0.20	0.10	0.10
4	686.14	694.22	934.21	734.93	0.30	0.30	0.30
5	733.11	752.27	1028.34	804.52	0.20	0.20	0.10
6	672.67	686.76	953.91	752.89	0.00	0.20	0.10
7	634.91	648.95	877.94	671.40	0.10	0.00	0.00
8	666.35	703.19	942.07	713.42	0.50	0.50	0.30
9	674.20	704.80	947.06	747.57	0.30	0.30	0.40
10	688.97	711.13	983.90	762.48	0.00	0.10	0.00
11	685.07	734.18	969.77	756.10	0.20	0.10	0.10
12	607.82	618.84	819.99	633.87	0.50	0.50	0.50
13	636.59	643.55	878.24	670.93	0.00	0.00	0.00
14	697.00	737.20	988.36	767.90	0.30	0.40	0.20
15	681.54	703.55	943.70	733.45	0.40	0.40	0.30
16	776.68	820.87	1108.25	880.93	0.50	0.50	0.30
17	672.93	679.34	929.22	715.17	0.10	0.10	0.10
18	627.93	630.94	864.23	657.21	0.00	0.00	0.00
19	644.16	659.58	900.48	686.61	0.10	0.20	0.10
20	674.61	697.80	945.25	778.46	0.10	0.10	0.10
21	733.22	763.89	1031.83	810.83	0.20	0.10	0.20
22	770.77	804.52	1080.92	876.31	0.60	0.50	0.40
23	732.54	757.54	1032.16	817.26	0.10	0.10	0.20
24	606.67	611.86	817.26	645.30	0.10	0.10	0.10
25	683.43	692.11	955.25	734.11	0.00	0.00	0.00
26	703.43	773.27	1049.55	864.87	0.10	0.30	0.30
27	739.51	756.56	1052.03	798.36	0.20	0.40	0.20
28	690.75	710.59	944.55	724.91	0.40	0.40	0.30
29	664.18	688.51	905.72	692.85	0.50	0.40	0.30
30	685.01	689.99	940.11	742.91	0.20	0.20	0.20
31	657.84	663.66	906.88	707.11	0.10	0.20	0.10
Avg.	681.29	702.87	951.42	743.83	0.22	0.23	0.19

Table 4
Experiment results of moderate demand case.

No.	Objective function				Number of rejections		
	$Static$	RP	$Myopic_A$	$Myopic_B$	RP	$Myopic_A$	$Myopic_B$
1	1351.36	2416.04	3124.92	3130.12	5.10	7.30	6.70
2	1385.58	2564.09	3367.51	3214.15	6.10	7.80	7.50
3	1360.61	2278.00	3030.64	3107.19	4.50	5.60	7.00
4	1091.47	1375.97	1899.14	1933.38	0.00	0.40	0.70
5	1296.99	2028.23	2647.69	2659.58	3.50	4.60	5.00
6	1416.44	2575.33	3364.84	3376.96	6.10	8.00	8.80
7	1301.79	1911.21	2635.86	2578.68	2.70	4.30	5.00
8	1408.02	2680.76	3344.23	3303.53	6.00	7.40	7.60
9	1238.78	1908.98	2419.09	2348.10	3.40	4.82	3.56
10	1451.32	2884.59	3777.96	3829.81	7.20	10.10	11.00
11	1441.57	2358.13	3132.66	3138.94	4.70	6.30	6.70
12	1240.74	1855.19	2485.42	2368.95	3.00	3.80	3.30
13	1371.91	2345.61	3221.76	3191.57	5.20	7.60	8.50
14	1422.26	2857.54	3697.72	3703.84	6.80	9.20	8.70
15	1324.62	2153.84	2850.05	2676.99	2.70	3.90	3.80
16	1440.36	2630.10	3645.38	3602.74	5.70	9.00	8.40
17	1358.72	2165.19	2852.73	2691.46	4.40	5.70	5.60
18	1372.66	2627.06	3433.93	3503.08	5.50	7.40	8.10
19	1334.15	2129.64	2840.35	2844.21	3.60	4.60	5.30
20	1381.31	2516.47	3211.46	3295.94	5.80	6.60	7.60
21	1483.24	2855.55	3618.50	3683.09	5.80	7.00	8.10
22	1357.95	2040.01	2855.92	2843.86	3.00	4.90	5.90
23	1555.42	3360.13	4480.73	4388.93	8.40	12.60	13.50
24	1346.70	2174.01	2844.63	2684.28	4.50	5.70	5.40
25	1279.55	1941.02	2588.08	2391.13	2.80	3.60	3.30
26	1379.54	2415.52	3068.20	3038.56	5.50	6.40	6.80
27	1491.61	2847.87	3570.83	3567.42	7.40	8.30	9.00
28	1414.56	2578.90	3435.91	3348.81	5.20	7.50	7.30
29	1365.65	2373.92	3010.86	2873.47	4.80	5.60	5.70
30	1421.92	2817.23	3577.15	3554.66	6.80	8.80	9.70
31	1355.83	1938.45	2580.86	2454.11	3.50	4.40	4.40
Avg.	1369.12	2374.34	3116.61	3075.08	4.83	6.43	6.71

for $Static$ in low demand cases, but it is well known that a $Static$ problem cannot scale to a large size. Therefore, in the moderate demand case, we replaced it with the best lower bound value of the $Static$ problem under a computational time limit of 600 s. In the high demand case, as the gap between the best lower bound and the best feasible solution was extremely high, we excluded the $Static$ problem in the analysis.

We now analyze the experiment results of the low demand case. Table 3 shows the results of experiments. The result of the reassignment policy is marked as RP. RP outperformed the result of $Myopic_A$ and $Myopic_B$. On average, RP resulted in 26.1 percent of improvement over $Myopic_A$, and 5.51 percent of improvement over $Myopic_B$. The RP objective function was significantly lower than the objective function of $Myopic_A$ and $Myopic_B$ in all instances. $Myopic_B$ obtained a lower objective function than $Myopic_A$. As the demand rate is relatively low, less than one order was rejected on average for all policies. The number of rejections was slightly lower in $Myopic_B$ compared to the other two policies.

With a low number of rejected orders, it becomes evident that in the case of low demand, a substantial majority of orders can be handled through available ADRs without rejection. Because the utilization of ADRs is relatively lower than higher demand cases, the number of recharges during operation is comparatively indifferent between policies. Under these circumstances, in low demand cases, the difference in objective function entirely relies on the ability to dispatch orders to ADRs more proficiently. It can be inferred that RP can develop a more efficient decision than can benchmark policies regarding assigning orders to ADRs.

The result of the moderate demand case is shown in Table 4. As more demand arrives at the same time horizon, the number of rejected orders increases greatly compared to the number in low demand cases. The average number of rejected orders exhibited significant fluctuations depending on the date, ranging from less than one rejection to as high as ten or more rejections. In this high variation circumstance, RP exhibited fewer rejections than $Myopic_A$ and $Myopic_B$, with about two fewer rejections per instance. The gap of the objective function between $Static$ and the three dynamic policies, including RP, increased greatly, as the objective function of two myopic policies more than doubled the objective function of the $Static$ model. This result indicates the increased difficulty of the dynamic problem compared to the static problem as the scale and complexity increase. Similarly to the low demand case, RP recorded a reduced objective function compared to $Myopic_A$ and $Myopic_B$. RP showed objective function improvement of 23.8 percent and 22.8 percent over $Myopic_A$ and $Myopic_B$. Even though the problem became more complex with increased orders, RP showed a clear competitive advantage over two myopic policies. As more orders arrived with shorter time between orders, more orders were classified as $waitingorders$. Therefore, congestion or concentration of orders was avoided using RP, and more efficient solutions were derived.

Table 5 shows the experiment result of the high demand case. The number of rejections increased substantially compared to low and moderate demand cases. Therefore, rejecting an appropriate number of orders gained more importance. When the average number of arriving orders increased by 29.3 percent compared to moderate demand, the average number of rejected orders increased by 231.7 percent for RP. The objective function more than doubled for RP with the increase in rejected orders, but still, the enhanced result was obtained compared to benchmark policies. The improvement of the objective functions compared to $Myopic_A$ and $Myopic_B$ were computed as 21.0 percent and 20.9 percent, respectively. The improvements decreased compared to the moderate demand case. The difference between the number of rejected orders increased, where RP rejected 16 orders while myopic policies rejected more than 20 orders.

To summarize the result, RP proved to show substantial performance improvement compared to the two benchmarks in all demand cases. Fig. 8 outlines the improvement shown when using RP compared

Table 5
Experiment results of high demand case.

No.	Objective function			Number of rejections		
	RP	$Myopic_A$	$Myopic_B$	RP	$Myopic_A$	$Myopic_B$
1	5953.64	7179.04	7041.88	22.70	27.60	27.80
2	4658.23	5988.73	6023.39	16.00	21.60	21.80
3	4535.67	5938.61	5947.60	15.70	21.20	21.20
4	3583.47	4651.59	4886.88	9.60	12.80	14.20
5	4643.11	5736.01	5709.80	16.40	19.30	19.20
6	4673.36	5973.95	6036.58	15.90	21.10	22.20
7	3963.18	5206.58	5173.37	10.70	15.30	15.10
8	4468.54	5703.65	5837.14	15.30	20.20	20.50
9	4464.09	5645.13	5780.34	14.80	19.90	19.80
10	4268.51	5652.25	5497.69	14.10	20.00	20.00
11	4079.64	5195.83	5160.36	12.10	15.30	15.50
12	2994.95	3931.06	3936.96	6.60	9.80	9.90
13	5209.34	6474.03	6342.53	20.40	25.30	25.40
14	4744.76	6157.69	6032.91	16.00	21.70	21.90
15	4028.28	5308.64	5298.09	12.20	16.40	16.70
16	5278.54	6855.85	6807.45	19.20	25.50	25.40
17	4559.97	5931.85	6016.98	15.20	21.10	21.90
18	4857.61	6020.62	5983.67	16.70	21.20	21.60
19	4456.05	5447.36	5559.13	15.70	19.20	19.90
20	5869.76	6963.40	6833.51	24.70	29.10	28.70
21	4998.87	6453.34	6339.65	17.40	23.10	23.20
22	4185.25	5563.24	5612.60	12.60	17.40	18.10
23	6371.89	7553.24	7406.18	26.00	30.40	30.00
24	4368.08	5356.86	5291.72	15.10	18.30	17.10
25	4146.03	5143.14	5100.65	13.70	16.60	15.80
26	4319.12	5510.61	5596.50	14.00	18.70	19.20
27	5692.11	7261.75	7405.35	21.80	27.80	28.10
28	5057.46	6480.93	6515.80	18.80	25.30	26.10
29	3346.69	4334.47	4303.61	9.30	11.90	12.00
30	6014.34	7085.96	7039.21	23.10	26.60	27.50
31	4436.01	5793.12	5803.75	14.90	20.20	20.60
Avg.	4652.47	5887.05	5881.33	16.02	20.64	20.85

Table 6
Average waiting time of served orders.

Demand	RP	$Myopic_A$	$Myopic_B$
Low	12.52	17.08	13.32
Moderate	17.36	23.04	22.45
High	23.25	30.21	30.05

to when using $Myopic_A$. The objective function showed an improved result of more than a 20 percent decrease in the objective function in all demand cases. The number of rejections was lower in RP, as seen in Fig. 8(b), which lowered the penalty cost of rejected orders. However, the reduction of the objective function of RP was higher than the reduced objective function due to rejection. Therefore, it can be said that there is a significant improvement in the service level for served orders. This improvement can be attributed to efficient dispatching decisions made in the reassignment policy. This is especially true of orders that arrived earlier but have not yet started delivery yet, *waiting orders*, which are being assigned together. More flexible adjustment to any imbalance or hindrance in the system could lead to faster deliveries in the long run. Table 6 illustrates the average waiting time of served orders. The value was compared to assess the performance for only served orders rather than rejected orders. It was observed that RP resulted in reduced waiting times across all demand cases compared to $Myopic_A$ and $Myopic_B$. This result indicates that the number of rejected orders decreased, and better service quality was ensured for served orders.

One notable experiment result is that more rejection occurred in the low demand case when using RP compared to when using $Myopic_B$. This result is in contrast to a decrease in objective function. One possible explanation for this outcome is that each order received multiple chances to be rejected. In myopic policies, whether an order is dispatched to an ADR or is rejected is decided only one time when

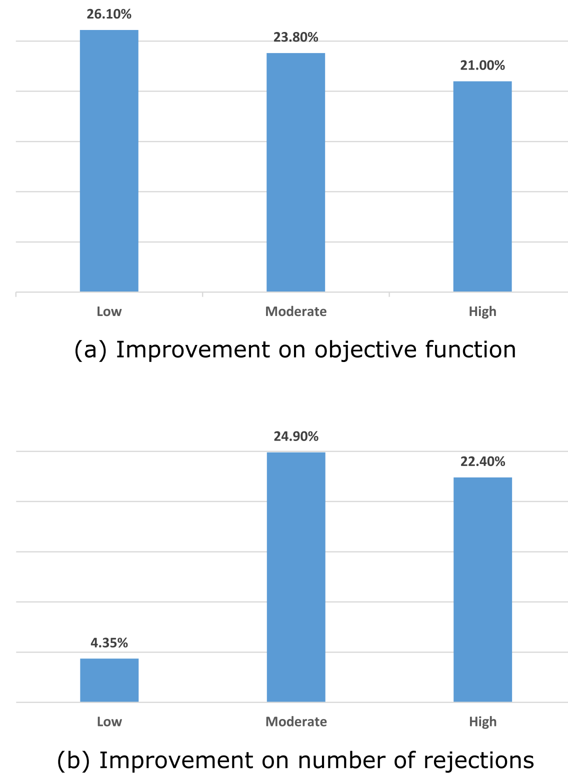


Fig. 8. Improvement on objective function and number of rejections.

the order arrives. However, as every arrived order is an input of reassignment before delivery starts, the chance of rejection remains even after an order is dispatched first. Orders not previously rejected could become the subject of rejection during the following process as a means to increase system performance. In other words, when utilizing RP, order rejection could be used to resolve situations with insufficient capacity. Still, the slight increase in rejection contributed to significantly increased delivery efficiency for the remaining orders.

5.3. Performance analysis of peak time management

This section evaluates the performance of the peak time management (PTM) strategy. As the strategy is expected to be effective in situations with densely arriving orders, we perform experiments on moderate demand and high demand cases. PTM strategy is applied to the reassignment policy and compared with the basic model. The base level was set as 50 percent, and the recharge level was set as 30 percent before PTM strategy was applied, as explained in 5.1. With the adoption of PTM strategy, the base level changed to 30 percent and the recharge level changed to 15 percent. These levels have a lower value than the base level and recharge level of non-peak times, as illustrated in Fig. 6. The parameter values were applied in the periods that were classified as peak periods by the demand information. This adjustment is intended to temporarily reduce battery charging time during peak periods to increase utilization. As in Section 5.2, we evaluate the effectiveness of PTM based on the experiment results on data of 31 days, with ten instances per day, by comparing objective function and number of rejections.

In the experiment, the number of orders that arrived during peak and non-peak periods are compared to verify the classification of peak periods. For high demand cases, 20.86 orders arrived per hour during peak periods, whereas 16.67 orders arrived per hour during non-peak periods. Therefore, any strategy change for peak periods impacts more orders compared to orders in non-peak periods. The peak time

Table 7
Experiment results of peak time management.

No.	Moderate demand				High demand			
	Objective function		Number of rejections		Objective function		Number of rejections	
	RP	$RP + PTM$	RP	$RP + PTM$	RP	$RP + PTM$	RP	$RP + PTM$
1	2416.04	2190.43	5.10	4.40	5953.64	4684.67	22.70	14.20
2	2564.09	2497.55	6.10	5.40	4658.23	4006.69	16.00	12.40
3	2278.00	2113.67	4.50	3.90	4535.67	3629.37	15.70	9.40
4	1375.97	1306.47	0.00	0.00	3583.47	3372.81	9.60	9.00
5	2028.23	1933.64	3.50	3.70	4643.11	4050.12	16.40	11.30
6	2575.33	2404.24	6.10	5.50	4673.36	3839.82	15.90	10.00
7	1911.21	2216.38	2.70	4.60	3963.18	3979.46	10.70	12.00
8	2680.76	2770.95	6.00	6.70	4468.54	4772.86	15.30	17.60
9	1908.98	1861.93	3.40	3.20	4464.09	4434.66	14.80	14.20
10	2884.59	2760.47	7.20	6.80	4268.51	4156.69	14.10	13.10
11	2358.13	2412.61	4.70	4.70	4079.64	4038.19	12.10	11.20
12	1855.19	1892.03	3.00	2.80	2994.95	2957.82	6.60	7.00
13	2345.61	2376.37	5.20	5.50	5209.34	5345.01	20.40	20.80
14	2857.54	2742.68	6.80	6.30	4744.76	4446.55	16.00	14.70
15	2153.84	2086.13	2.70	2.70	4028.28	3907.42	12.20	11.40
16	2630.10	2690.74	5.70	6.00	5278.54	5091.82	19.20	18.40
17	2165.19	2089.16	4.40	4.10	4559.97	4122.92	15.20	12.60
18	2627.06	2313.51	5.50	3.70	4857.61	4060.17	16.70	12.30
19	2129.64	2033.36	3.60	3.10	4456.05	3794.13	15.70	12.10
20	2516.47	2295.06	5.80	5.20	5869.76	4665.70	24.70	15.70
21	2855.55	2546.11	5.80	4.30	4998.87	4310.91	17.40	12.00
22	2040.01	1972.92	3.00	2.50	4185.25	3861.20	12.60	11.00
23	3360.13	3329.78	8.40	9.40	6371.89	5917.75	26.00	23.50
24	2174.01	2269.68	4.50	5.20	4368.08	4534.06	15.10	15.20
25	1941.02	2062.61	2.80	3.20	4146.03	4147.12	13.70	13.80
26	2415.52	2193.33	5.50	3.90	4319.12	3628.88	14.00	9.20
27	2847.87	2580.72	7.40	5.70	5692.11	4777.09	21.80	14.50
28	2578.90	2217.36	5.20	3.30	5057.46	4204.79	18.80	11.30
29	2373.92	2334.64	4.80	4.60	3346.69	3336.69	9.30	9.30
30	2817.23	2869.94	6.80	7.30	6014.34	5147.79	23.10	16.90
31	1938.45	2028.51	3.50	3.70	4436.01	4218.86	14.90	12.70
Avg.	2374.34	2303.00	4.83	4.56	4652.47	4240.07	16.02	13.19

management strategy is expected to increase the utilization of ADRs in the short term. However, the efficiency of the strategy during the whole time horizon is an aspect that needs to be revealed through research, as more recharging occasions are required to recharge ADRs during non-peak periods. We experimented with test instances to verify if peak time management is compelling enough to overcome the trade-off situation.

Table 7 presents the result of the experiment. By implementing PTM, the objective function and the number of rejections are reduced by 3.01 percent and 5.59 percent in the moderate demand case. In the high demand case, the objective function and the number of rejections were reduced by 8.86 percent and 17.67 percent. Compared to RP, $RP + PTM$ showed a decrease in average objective function and number of rejections, especially when the demand rate was higher. The experiment results correspond to our intuition in implementing the PTM method. By applying PTM, we increased ADR utilization during peak times when demand tends to be concentrated. Recharging frequency and the time required for recharging decreased, making more time available for delivery of orders. A decrease in the number of rejections in both moderate and high demand can result from this factor.

We analyzed the results of the experiment to examine the effects of peak time management. Fig. 9 depicts two statistics obtained from the experiment. First, the waiting time for orders that arrive during peak periods is calculated. When the peak time management strategy was implemented, customers had to wait three fewer minutes on average to get the orders delivered. This fast dispatching and delivery result indicates that ADRs can handle more orders with a speedier process capacity if PTM is implemented. The average number of recharges during the whole process is also calculated. If PTM is used, about two more recharges for each ADR, and nine more for the entire system occur. This outcome is due to applying two (BL, RL) pairs in PTM. In a non-peak period after a peak period, ADRs suddenly face a higher base level when deciding to recharge. More ADRs are recharged under the higher base and recharge levels in these situations. Even though more time is spent on recharging, PTM still yielded lower objective function values in most instances. This result can give insight into the fact that by using PTM, ADRs are utilized more during periods of many orders and recharged more during periods of fewer orders, indicating an efficient operation.

In a small number of instances, the objective function of $RP+PTM$ increased compared to the result of RP. The reason for this outcome is twofold. First, as mentioned above, the number of recharges increases in times after peak time as the base level that triggers recharges increases. In this situation, an ADR's total capacity decreases, leading to an inevitable loss in processing ability. Second, PTM can have side effects when an extended sequence of peak time slots exists. As PTM aims to benefit from a temporary increase in capacity with a minor sacrifice in future capacity, prolonged peak times might lead to inferior performance. Focusing on the rise in the objective function occurring in several instances from the same date, it can be seen that the distribution of peak time impacts the performance of PTM.

6. Conclusions

With the increasing utilization of ADRs in diverse domains, the significance of enhancing ADR driven delivery's effectiveness is also on the rise. In the process, taking account of the unique characteristics of ADRs is an essential factor in taking full advantage of this emerging technology. However, as the dynamic pickup and delivery problem is known initially as a complex problem, finding efficient solutions with additional constraints related to ADR becomes challenging. Furthermore, in dynamic situations, the increasing need for timely solutions raises the complexity of finding practical policies in real-world scenarios.

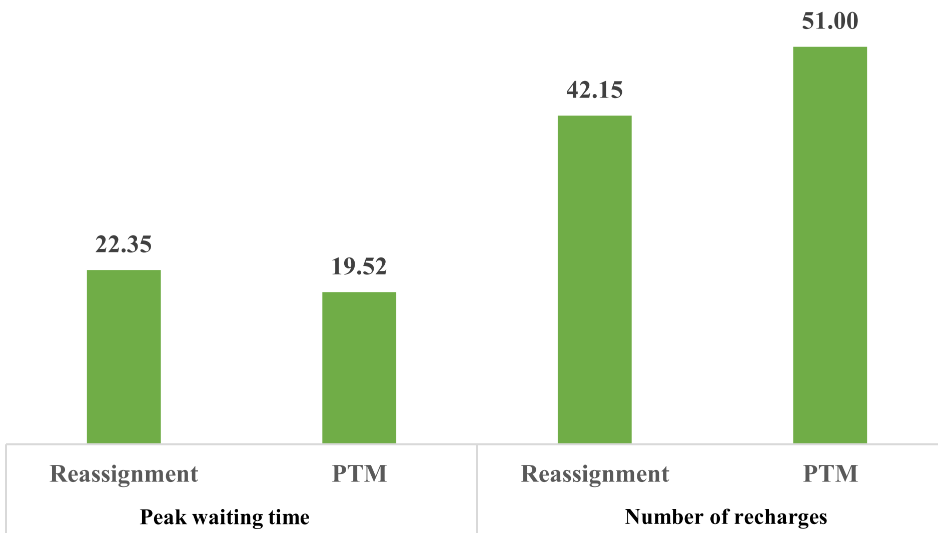


Fig. 9. Comparison of waiting time during peak periods and number of recharges.

In this study, we proposed a solution methodology to address such challenging situations. We defined a novel DPDP-AR problem with the ADR features included in the formulation. The reassignment policy is presented, and the mathematical formulation is suggested based on the formulation of the static model. A practical battery management strategy and peak time management methodology are proposed concerning batteries. Experiment results verified improved solution quality of the reassignment policy compared to the policy in practical use, and implementation of peak time management was shown to enhance the solution quality even further.

This paper presents a guideline for novice operators to operate ADR and to process orders efficiently. Novice operators can utilize the reassignment algorithm suggested in this study to effectively dispatch dynamic orders to ADR within a short computation time. With the intuitive (BL, RL) policy, the operator can also easily handle battery recharging, which is one of the few inconveniences of adopting this new technology. Based on the operating environment, battery-related parameters can be adjusted to enhance efficiency. In specific environments where peak demand periods can be estimated, the peak time management strategy introduced in this study can also be applied to increase utilization during peak periods. Consequently, this paper can serve as an example of harnessing the advantages of utilizing ADR while mitigating the inconvenience caused by it to increase the operational productivity of delivery services.

Several limitations exist in this study. The travel time of ADRs is calculated as a deterministic value of distance divided by speed, and any factor that could affect travel time, such as congestion, is not considered. Furthermore, the speed of ADRs is also a fixed value, in contrast to the varying speed of ADRs during the actual operation. In addition, the ready time of restaurants is assumed to be zero in the experiment. For further research, we aim to expand the problem with more practical applications. In addition to stochastic order arrivals, order reservations could be added to the model. Policies that can manage both preannounced orders and stochastic orders could be devised to solve the problem. ADR speed could be relaxed to be an adjustable variable and could be added as a decision variable considering the varying battery discharge rates. Anticipatory policies based on forecasts of future demand are options in terms of policy-making. Approximate dynamic programming and reinforcement learning methodologies could be applied to make decisions based on future demand information.

CRediT authorship contribution statement

Joonhwa Jeong: Writing – original draft, Formal analysis, Conceptualization. **Ilkyeong Moon:** Writing – review & editing, Validation, Supervision, Project administration.

Data availability

The authors are unable or have chosen not to specify which data has been used.

Acknowledgments

The authors are grateful for the valuable comments from the editor-in-chief, the associate editor, and three anonymous reviewers. This work was supported by the National Research Foundation of Korea (NRF), Republic of Korea grant funded by the Korea government (MSIT), Republic of Korea(No. RS-2023-00218913 and NRF-2021R1A4A1029124).

References

Adler, J. D., & Mirchandani, P. B. (2014). Online routing and battery reservations for electric vehicles with swappable batteries. *Transportation Research, Part B (Methodological)*, *70*, 285–302.

Al-Kanj, L., Nascimento, J., & Powell, W. B. (2020). Approximate dynamic programming for planning a ride-hailing system using autonomous fleets of electric vehicles. *European Journal of Operational Research*, *284*(3), 1088–1106.

Arslan, A. M., Agatz, N., Kroon, L., & Zuidwijk, R. (2019). Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, *53*(1), 222–235.

Bongiovanni, C., Kaspi, M., & Geroliminis, N. (2019). The electric autonomous dial-a-ride problem. *Transportation Research, Part B (Methodological)*, *122*, 436–456.

Cai, J., Zhu, Q., Lin, Q., Ma, L., Li, J., & Ming, Z. (2023). A survey of dynamic pickup and delivery problems. *Neurocomputing*, Article 126631.

Cai, J., Zhu, Q., Lin, Q., Ming, Z., & Tan, K. C. (2024). Decomposition-based multiobjective evolutionary optimization with tabu search for dynamic pickup and delivery problems. *IEEE Transactions on Intelligent Transportation Systems*.

Chen, C., Demir, E., Huang, Y., & Qiu, R. (2021). The adoption of self-driving delivery robots in last mile logistics. *Transportation Research Part E: Logistics and Transportation Review*, *146*, Article 102214.

Chen, T., Zhang, B., Pourbabak, H., Kavousi-Fard, A., & Su, W. (2016). Optimal routing and charging of an electric vehicle fleet for high-efficiency dynamic transit systems. *IEEE Transactions on Smart Grid*, *9*(4), 3563–3572.

Cheng, X., Liao, S., & Hua, Z. (2017). A policy of picking up parcels for express courier service in dynamic environments. *International Journal of Production Research*, *55*(9), 2470–2488.

Cortés, C. E., Sáez, D., Núñez, A., & Muñoz-Carpintero, D. (2009). Hybrid adaptive predictive control for a dynamic pickup and delivery problem. *Transportation Science*, *43*(1), 27–42.

Cortés-Murcia, D. L., Prodhon, C., & Afsar, H. M. (2019). The electric vehicle routing problem with time windows, partial recharges and satellite customers. *Transportation Research Part E: Logistics and Transportation Review*, *130*, 184–206.

Du, J., Zhang, Z., Wang, X., & Lau, H. C. (2023). A hierarchical optimization approach for dynamic pickup and delivery problem with LIFO constraints. *Transportation Research Part E: Logistics and Transportation Review*, *175*, Article 103131.

Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, *48*(1), 100–114.

Farazi, N. P., Zou, B., & Tulabandhula, T. (2022). Dynamic on-demand crowdshipping using constrained and heuristics-embedded double dueling deep Q-network. *Transportation Research Part E: Logistics and Transportation Review*, *166*, Article 102890.

Felipe, Á., Ortuño, M. T., Righini, G., & Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, *71*, 111–128.

Gao, Y., Zhang, S., Zhang, Z., & Zhao, Q. (2024). The stochastic share-a-ride problem with electric vehicles and customer priorities. *Computers & Operations Research*, *164*, Article 106550.

Ghiani, G., Manni, A., & Manni, E. (2022). A scalable anticipatory policy for the dynamic pickup and delivery problem. *Computers & Operations Research*, *147*, Article 105943.

Ghiani, G., Manni, E., Quaranta, A., & Triki, C. (2009). Anticipatory algorithms for same-day courier dispatching. *Transportation Research Part E: Logistics and Transportation Review*, *45*(1), 96–106.

Hiermann, G., Puchinger, J., Ropke, S., & Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, *252*(3), 995–1018.

Hof, J., Schneider, M., & Goeke, D. (2017). Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops. *Transportation Research, Part B (Methodological)*, *97*, 102–112.

Hwang, S. W., & Lim, S. (2022). The charging infrastructure design problem with electric taxi demand prediction using convolutional LSTM. *European Journal of Industrial Engineering*, *16*(6), 651–678.

Hyland, M., & Mahmassani, H. S. (2018). Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. *Transportation Research Part C (Emerging Technologies)*, *92*, 278–297.

Jie, W., Yang, J., Zhang, M., & Huang, Y. (2019). The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. *European Journal of Operational Research*, *272*(3), 879–904.

Jun, S., Lee, S., & Yih, Y. (2021). Pickup and delivery problem with recharging for material handling systems utilising autonomous mobile robots. *European Journal of Operational Research*, *289*(3), 1153–1168.

Kchaou-Boujelben, M., & Gicquel, C. (2020). Locating electric vehicle charging stations under uncertain battery energy status and power consumption. *Computers & Industrial Engineering*, *149*, Article 106752.

Keskin, M., & Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C (Emerging Technologies)*, *65*, 111–127.

Klapp, M. A., Erera, A. L., & Toriello, A. (2018). The dynamic dispatch waves problem for same-day delivery. *European Journal of Operational Research*, *271*(2), 519–534.

Li, X., Luo, W., Yuan, M., Wang, J., Lu, J., Wang, J., et al. (2021). Learning to optimize industry-scale dynamic pickup and delivery problems. In *2021 IEEE 37th international conference on data engineering* (pp. 2511–2522). IEEE.

Lin, J., Zhou, W., & Wolfson, O. (2016). Electric vehicle routing problem. *Transportation Research Procedia*, *12*, 508–521.

Loeb, B., & Kockelman, K. M. (2019). Fleet performance and cost evaluation of a shared autonomous electric vehicle (SAEV) fleet: A case study for austin, texas. *Transportation Research Part A: Policy and Practice*, *121*, 374–385.

Ma, Y., Hao, X., Hao, J., Lu, J., Liu, X., Xialiang, T., et al. (2021). A hierarchical reinforcement learning based optimization framework for large-scale dynamic pickup and delivery problems. *Advances in Neural Information Processing Systems*, *34*, 23609–23620.

Ma, B., Hu, D., Chen, X., Wang, Y., & Wu, X. (2021). The vehicle routing problem with speed optimization for shared autonomous electric vehicles service. *Computers & Industrial Engineering*, *161*, Article 107614.

Macrina, G., Pugliese, L. D. P., Guerriero, F., & Laporte, G. (2019). The green mixed fleet vehicle routing problem with partial battery recharging and time windows. *Computers & Operations Research*, *101*, 183–199.

Mangiaracina, R., Perego, A., Seghezzi, A., & Tumino, A. (2019). Innovative solutions to increase last-mile delivery efficiency in B2C e-commerce: a literature review. *International Journal of Physical Distribution & Logistics Management*, *49*(9), 901–920.

Masmoudi, M. A., Hosny, M., Demir, E., Genikomsakis, K. N., & Cheikhrouhou, N. (2018). The dial-a-ride problem with electric vehicles and battery swapping stations. *Transportation Research Part E: Logistics and Transportation Review*, *118*, 392–420.

Messaoud, E. (2022). Solving a stochastic programming with recourse model for the stochastic electric capacitated vehicle routing problem using a hybrid genetic algorithm. *European Journal of Industrial Engineering*, *16*(1), 71–90.

Powell, W. B., Simao, H. P., & Bouzaiene-Ayari, B. (2012). Approximate dynamic programming in transportation and logistics: a unified framework. *EURO Journal on Transportation and Logistics*, *1*(3), 237–284.

Psaraftis, H. N., Wen, M., & Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks*, *67*(1), 3–31.

Reyes, D., Erera, A., Savelsbergh, M., Sahasrabudhe, S., & O'Neil, R. (2018). The meal delivery routing problem. *Optimization Online*, *6571*, 2018.

Sassi, O., & Oulamara, A. (2017). Electric vehicle scheduling and optimal charging problem: complexity, exact and heuristic approaches. *International Journal of Production Research*, *55*(2), 519–535.

Sayarshad, H. R., Mahmoodian, V., & Gao, H. O. (2020). Non-myopic dynamic routing of electric taxis with battery swapping stations. *Sustainable Cities and Society*, *57*, Article 102113.

Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, *48*(4), 500–520.

Sheridan, P. K., Gluck, E., Guan, Q., Pickles, T., Balcıog, B., Benhabib, B., et al. (2013). The dynamic nearest neighbor policy for the multi-vehicle pick-up and delivery problem. *Transportation Research Part A: Policy and Practice*, *49*, 178–194.

Soeffker, N., Ulmer, M. W., & Mattfeld, D. C. (2022). Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research*, *298*(3), 801–820.

Soysal, M., Cimen, M., & Belbağ, S. (2020). Pickup and delivery with electric vehicles under stochastic battery depletion. *Computers & Industrial Engineering*, *146*, Article 106512.

Srinivas, S., Ramachandiran, S., & Rajendran, S. (2022). Autonomous robot-driven deliveries: A review of recent developments and future directions. *Transportation Research Part E: Logistics and Transportation Review*, *165*, Article 102834.

Steever, Z., Karwan, M., & Murray, C. (2019). Dynamic courier routing for a food delivery service. *Computers & Operations Research*, *107*, 173–188.

Suguna, M., Shah, B., Raj, S. K., & Suresh, M. (2021). A study on the influential factors of the last mile delivery projects during Covid-19 era. *Operations Management Research*, 1–14.

Sun, B., Yang, Y., Shi, J., & Zheng, L. (2019). Dynamic pick-up and delivery optimization with multiple dynamic events in real-world environment. *IEEE Access*, *7*, 146209–146220.

Tao, Y., Zhuo, H., & Lai, X. (2023). The pickup and delivery problem with multiple depots and dynamic occasional drivers in crowdshipping delivery. *Computers & Industrial Engineering*, *182*, Article 109440.

Tirado, G., & Hvattum, L. M. (2017). Improved solutions to dynamic and stochastic maritime pick-up and delivery problems using local search. *Annals of Operations Research*, *253*, 825–843.

Ulmer, M. W., & Streng, S. (2019). Same-day delivery with pickup stations and autonomous vehicles. *Computers & Operations Research*, *108*, 1–19.

Ulmer, M. W., Thomas, B. W., Campbell, A. M., & Woyak, N. (2021). The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transportation Science*, *55*(1), 75–100.

Van Heeswijk, W. J., Mes, M. R., & Schutten, J. M. (2019). The delivery dispatching problem with time windows for urban consolidation centers. *Transportation Science*, *53*(1), 203–221.

Verma, A. (2018). Electric vehicle routing problem with time windows, recharging stations and battery swapping stations. *EURO Journal on Transportation and Logistics*, *7*(4), 415–451.

Voccia, S. A., Campbell, A. M., & Thomas, B. W. (2019). The same-day delivery problem for online purchases. *Transportation Science*, *53*(1), 167–184.

Wu, O. Q., Yücel, Ş., & Zhou, Y. (2022). Smart charging of electric vehicles: An innovative business model for utility firms. *Manufacturing & Service Operations Management*, *24*(5), 2481–2499.

Xu, X., & Wei, Z. (2023). Dynamic pickup and delivery problem with transshipments and LIFO constraints. *Computers & Industrial Engineering*, *175*, Article 108835.

Yalaoui, F., & Nguyen, N. Q. (2021). Identical machine scheduling problem with sequence-dependent setup times: MILP formulations computational study. *American Journal of Operations Research*, *11*(1), 15–34.

Zhang, S., Zhang, W., Gajpal, Y., & Appadoo, S. (2019). Ant colony algorithm for routing alternate fuel vehicles in multi-depot vehicle routing problem. In *Decision Science in Action: Theory and Applications of Modern Decision Analytic Optimisation* (pp. 251–260). Springer.

Zhu, L., & Sheu, J.-B. (2018). Failure-specific cooperative recourse strategy for simultaneous pickup and delivery problem with stochastic demands. *European Journal of Operational Research*, *271*(3), 896–912.