INTELLIGENT VEHICLE SCHEDULING FOR GREEN AIRPORT BAGGAGE TRANSPORT SERVICE

XINYUE WANG

MPhil

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University

Department of Aeronautical and Aviation Engineering

Intelligent Vehicle Scheduling for Green Airport Baggage Transport Service

Xinyue Wang

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Philosophy

August 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

________________________ (Signed)

____Xinyue WANG_______ (Name of student)

ABSTRACT

Efficient airport baggage transport is critical for improving airport operation efficiency and quality. In practice, the baggage transport is usually achieved by the cooperation of tractors and trailers under the drop-and-pull mode. Recently, new electric autonomous vehicles have been introduced to promote the intelligent and sustainable development of airports. However, scheduling baggage transport vehicles presents significant challenges due to the complex relationships among tractors, trailers, and flights, which are further addressed by considering the recharging decision-making problem of electric autonomous vehicles. Besides, the airport ground handling is a highly dynamic and uncertain scenario, particularly at busy hub airports.

To address these challenges, this thesis reviewed the literature related to vehicle scheduling for airport baggage transport services. Based on the previous studies and the intelligent development process of airports, this research focuses on vehicle scheduling under two operating modes: multi-trailer drop-and-pull baggage transport and electric auto-dolly-based baggage transport.

For the multi-trailer drop-and-pull baggage transport, this study develops a two-stage scheduling model for tractors and trailers under the drop-and-pull mode, as well as designing an efficient hybrid intelligence-based solution algorithm. Specifically, the Adaptive Large Neighborhood Search is taken as the foundation of the algorithm, with carefully designed operators. Besides, two key methods are introduced to enhance the efficiency of the algorithm, including a K-means clustering-based initialization method and a topological sort-based solution evaluation method.

For the electric auto-dolly-based baggage transport, a simplified scheduling model is established based on the model of Vehicle Routing Problem, which is then modeled into the Markov Decision Process of improvement heuristic. Then, a scheduling algorithm that integrates reinforcement learning and the Transformers variant-based deep learning model is improved, with specifically designed problem embeddings to effectively present the constraints on service time and electricity consumption, thus improving the algorithm convergence speed.

Finally, supported by the flight and map data collected from real-world airports, a SUMO-based integrated airport service vehicle scheduling simulation platform is established. Simulation experimental results are analyzed to improve the algorithm and provide references for airport service vehicle scheduling in practice.
Keywords: airport baggage transport, airport service vehicle scheduling, adaptive large neighborhood search, reinforcement learning

PUBLICATIONS ARISING FROM THIS THESIS

JOURNAL PAPERS

[1] X. Zhang, **Wang, X.**, W. Dong, and G. Xu*, "Sequential feature-augmented deep multilabel learning for compound fault diagnosis of rotating machinery with few labeled and imbalanced data," *Computers and Operations Research*, 2025.

[2] **Wang, X.**, G. Xu*, Z. Zhou, and Y. Zou, "Sequential feature-augmented deep multilabel learning for compound fault diagnosis of rotating machinery with few labeled and imbalanced data," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 12, pp. 13 947–13 955, 2024.

[3] **Wang, X.**, G. Xu*, and X. Zhang, "Hybrid intelligence-based vehicle scheduling for multi-trailer drop-and-pull airport baggage transport," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Under review, 2024.

CONFERENCE PAPERS AND PRESENTATIONS

[1] **Wang, X.** and G. Xu*, "Adaptive large neighborhood search-based vehicle scheduling for multi-trailer drop-and-pull airport baggage transport service," in *2024 International Forum on Shipping, Ports and Airports (IFSPA)*, 2024.

[2] **Wang, X.**, Y. Long, G. Xu*, and Y. Liu, "Integrated tractor and trailer scheduling for airport baggage transport service," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2023, pp. 3894–3899.

ACKNOWLEDGMENTS

I am deeply indebted to my esteemed supervisor, Dr. Xu Gangyan, an Assistant Professor in the Department of Aeronautical and Aviation Engineering at The Hong Kong Polytechnic University. Dr. Xu has been an unwavering source of intellectual mentorship, generously devoting his expertise to nurturing my academic growth. His profound insights have been instrumental in shaping my understanding, while his willingness to share his extensive research experiences and academic journey has been a true inspiration.

My sincere gratitude extends to the entire community of the Intelligent and Networked Systems Group within our department. Their collective encouragement and unwavering support have formed an indispensible backbone throughout my research endeavor. The valuable suggestions and stimulating discussions with my colleagues have significantly enriched the quality of my work.

I remain immensely grateful to my parents and friends for their unconditional love, boundless support, and unwavering patience, which have been the constants in my journey. Their belief in me has been a beacon guiding me through challenges.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Efficient airport ground handling services are essential for the smooth functioning of airports and the operation of airlines while ensuring the level of aviation safety and security [1]–[3]. It was reported that approximately 5-10% of flight delays can result from inadequate airport ground handling services [4], and accidents during ground operations are frequently reported, resulting in significant flight delays and even leading to the loss of ground operator lives [5]. Insufficient and inefficient ground services have become major obstructions to further decreasing airport capacity. For instance, Singapore aviation firms are ramping up hiring ground service operators to cope with the recent surge in air travel.

Among various airport ground handling services, baggage transport plays an essential role. In 2017, the daily baggage handling volume of the Hong Kong International Airport (HKIA) reached 80000 pieces. In 2022, airports worldwide successfully managed an impressive 4.5 billion bags, and it is predicted that the size of the airport baggage handling system market worldwide will reach 16.1 billion U.S. dollars. Efficient airport baggage transport service is also recognized as a key indicator for airport operation performance. Any inefficient operations or mistakes would delay the flight directly and even lead to a cascade effect for all upcoming flights. In the practices of most existing airports, baggage transport service is usually operated by motorized tractors and non-motorized trailers under the drop-and-pull mode, as shown in Fig. 1.2 (a). Trailers are non-motorized and only used to hold passenger baggage, while motorized tractors are utilized to tow trailers between the Baggage Handling Area (BHA) and aircraft stands.

In last few years, the vigorous development of the aviation industry has heightened the demand for improved operational efficiency and quality of airport ground handling [6]. Nowadays, many airports are transitioning towards higher automation and sustainability [7], [8]. Governments are actively supporting the development of intelligent airports. For example, China has announced many civil aviation policies, actively promoting automation and electrification [9], which is shown in Fig. 1.1. Global airports are also introducing new equipment for airport baggage transport service, like self-driving tractors in Hong Kong, electric auto-dollies in Singapore, baggage loading robots in the United Kingdom, etc [10], [11]. The adoption of clean fuels and electric ground equipment effectively reduces the airport's carbon emissions [12]. Fig. 1.2 (b) shows the electric auto-dolly developed by *AURRIGO*, which is under test at the Singapore Changi International Airport.

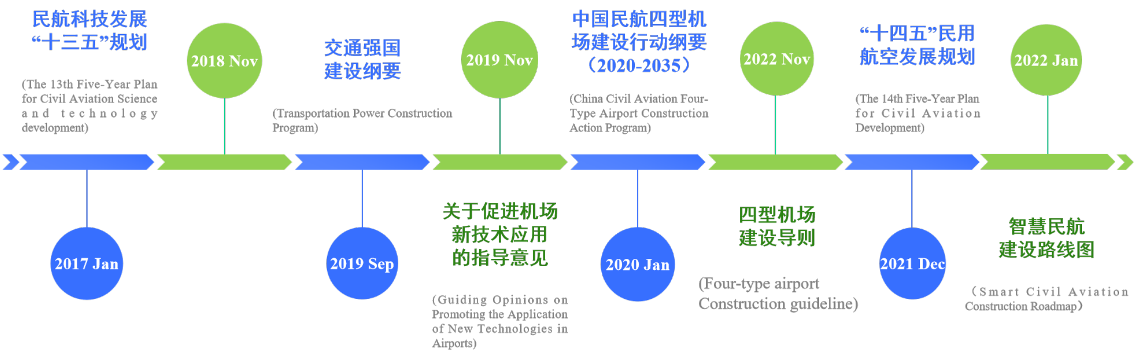


Figure 1.1: The civil aviation policies announced by the China Aviation Administration of China from 2017 to 2022.

Nowadays, airports primarily use two modes: the existing and widely used tractor-trailer mode, and the emerging electric auto-dolly mode. Fig. 1.3 shows their difference, which are mainly demonstrated from the following perspectives:

1. All electric auto-dollies are self-motorized, while the movement of trailers completely relies on tractors under the drop-and-pull mode;

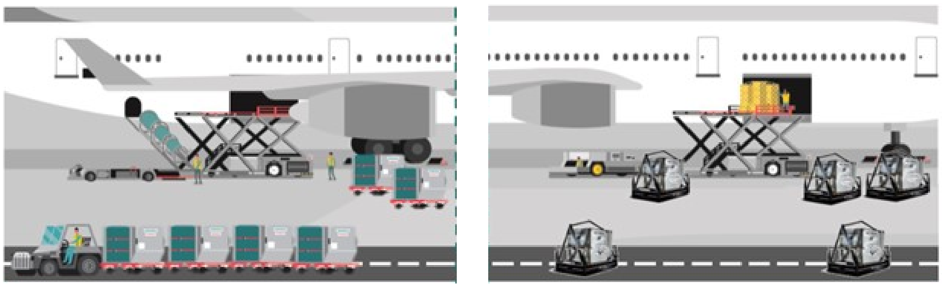
2. The equipped robot arms on auto-dollies can make the process of baggage load-

ing and unloading unmanned, while loading and unloading baggage for trailers are implemented manually;

3. Auto-dolly is generally more expensive than the tractor and trailer because of its equipped smart devices and sensors.

(a) (b)

Figure 1.2: Airport baggage transport vehicles: (a). tractor and trailers; (b). electric auto-dolly.



(a) (b)

Figure 1.3: The two different airport baggage transport modes: (a). tractor-trailer mode; (b). electric auto-dolly mode.

Following the existing multi-trailer drop-and-pull baggage transport enables flexible coupling and decoupling of tractors and trailers as required [13], potentially lowering operational costs. However, it greatly increases the complexity of operation scheduling. Although adopting new-type vehicles, i.e., electric auto-dollies, can simplify the scheduling decision-making process, their high cost may necessitate scheduling under limited dolly resources. According to our investigations, the scheduling of tractors and trailers is usually done manually based on predefined rules or expert experiences. For example, schedulers always dispatch the closest tractor to the flight, and the tractor driver usually looks for empty trailers while moving. Such manual decisions are inefficient and cannot well cope with the dense and dynamic demands, especially in large airports during busy hours [14]. Thus, automatic tools are urgently needed to support vehicle scheduling, which is essential in improving the performance of airport ground handling while reducing operating costs and flight delays [15].

Many works have been conducted to optimize the scheduling efficiency of airport ground handling services [16]–[18] under dynamic and uncertain environments [19], [20] and with electric vehicles [21]. Generally, the scheduling problem of airport service vehicles is regarded as the Vehicle Routing Problem (VRP) [14]. It is usually modeled as Integer Programming (IP) and Mixed Integer Programming (MIP) [22], [23], and is solved by mature solvers such as CPLEX and heuristic algorithms. However, considering the operational features of airport baggage transport, there are still several limitations to be addressed.

On the one hand, the specialized roles of tractors and trailers are not classified in most existing works but treated as fixed units, which is rarely aligned with the practical application and would hinder the effective utilization of the multi-trailer capability of tractors [24]. Prior studies demonstrate that decoupled trailer operation systems, where tractors and trailers are independently scheduled, enable dynamic resource allocation and operational flexibility [25]. This approach proves particularly advantageous in scenarios involving prolonged cargo transfer processes [26], such as airport baggage handling systems, where manual loading/unloading of luggage containers onto trailers creates significant time bottlenecks. Empirical evidence from Cui *et al.* [27] further suggests that coordinated scheduling optimization of transport units under such decoupled mechanisms could enhance baggage logistics throughput by 18-22% in typical hub operations. On the other hand, current research related to airport ground electric vehicle scheduling primarily focuses on the scheduling of aircraft tractors [21], [28], [29]. However, these findings can not directly guide electric auto-dolly scheduling, as the operation modes and contents of different ground services vary significantly from each other.

Moreover, many challenges still exist in vehicle scheduling for airport baggage transport. The first difficulty arises from the strict requirements of the airport baggage transport service. A single flight usually requires multiple trailers (or dollies) to serve, and the number of trailers (or dollies) needed differs from the number of flights. Such divisible demand settings will greatly increase the complexity of dolly scheduling, which is a typical variant of the split delivery vehicle routing problem [30]. Besides, baggage transport is usually conducted under tight time constraints, making it challenging to solve real-world large-scale cases in a short time. Another challenge comes from the operation mode of baggage transport vehicles. When adopting tractors and trailers, the complex route dependencies between a tractor and multiple trailers must be considered. Similarly, when employing electric auto-dollies, it is necessary to make recharging decisions without compromising operational efficiency. Therefore, how to efficiently obtain effective schedules in a short time is always a great challenge.

In summary, to promote the development of airport ground handling, this thesis aims to achieve the optimization of vehicle scheduling for baggage transport service based on the existing related research.

1.2 RESEARCH SCOPE AND OBJECTIVES

Based on the above analysis, this study aims to achieve intelligent vehicle scheduling for reducing airport operating costs, and further promoting the green airport baggage transport service. The research objectives are as follows:

1. **Achieve efficient vehicle scheduling of multi-trailer drop-and-pull airport baggage transport.** Firstly, based on the operating characteristics of multi-trailer drop-and-pull baggage transport service, a tractor and trailer scheduling model is established. Then, a heuristic scheduling algorithm is developed to solve this problem, followed by specialized initial solution generation algorithm, local operators, and an acceleration algorithm for large-scale instances.

2. **Design an effective electric auto-dolly scheduling method to improve sustainable airport baggage transport.** Firstly, the model construction for the electric auto-dolly scheduling problem is investigated. Then, the development of the reinforcement learning-based scheduling algorithm is researched, including the design of the Markov Decision Process, problem embedding, and training algorithm.

3. **Conduct empirical analysis on the constructed airport vehicle scheduling simulation platform.** An integrated vehicle scheduling simulation platform for airport ground handling is constructed. Then, guidelines for the real-world airport baggage transport service are developed based on empirical analysis.

1.3 THESIS ORGANIZATION

The research framework of intelligent vehicle scheduling for green airport baggage transport service is shown in 1.4. According to the research framework, this paper is organized as follows:

First, a brief introduction of the research background, research scopes, and research objectives is developed in Chapter 1.

Chapter 2 reviews the research studies related to airport baggage transport optimization and the approaches for vehicle routing problems. Finally, the research potential in the domain is summarized in this section.

Chapter 3 illustrates how to achieve efficient vehicle scheduling of multi-trailer drop-and-pull airport baggage transport. We first defined the tractor and trailer scheduling problem in the airport baggage transport scenario. And we established the mathmatical formulations of the integrated tractor and trailer scheduling problem. Then we decomposed it into a two-stage scheduling model to decrease the solving complexity. Finally, a hybrid intelligence-based algorithm that integrates K-means clustering and Adaptive Large Neighborhood Search is developed to efficiently solve the two-stage scheduling problem. The algorithm performances are validated by comparison experiments.

Chapter 4 designs an effective electric auto-dolly scheduling method for improving green airport baggage transport. So we first constructed a simplified electric auto-dolly scheduling model to decrease the model-solving complexity. Then we define the process of solving this problem as a Markov Decision Process, including designing scenario-specific state embeddings, actions, and the reward function. To solve this problem, a Proximal Policy Optimization (PPO) policy with Transformers structure-based encoder and decoder and Curium Learning (CL) strategy is developed. Extensive experiments are conducted to verify the model and algorithm effectiveness.

Chapter 5 firstly introduces the simulation platform construction, which is achieved by processing flight information and airport ground map data through the Open Street Map (OSM) and the Simulation of Urban Mobility (SUMO). Then, an energy consumption model is established to measure the energy consumption and emissions of different types of vehicles. Based on it, the two modes that adopt fuel tractors and electric auto-dollies are compared. A statistical analysis of energy consumption for airport baggage transport service is also conducted in Chapter 5 to provide guidance to real-world airport operations.

Chapter 6 concludes this thesis and discusses future work.

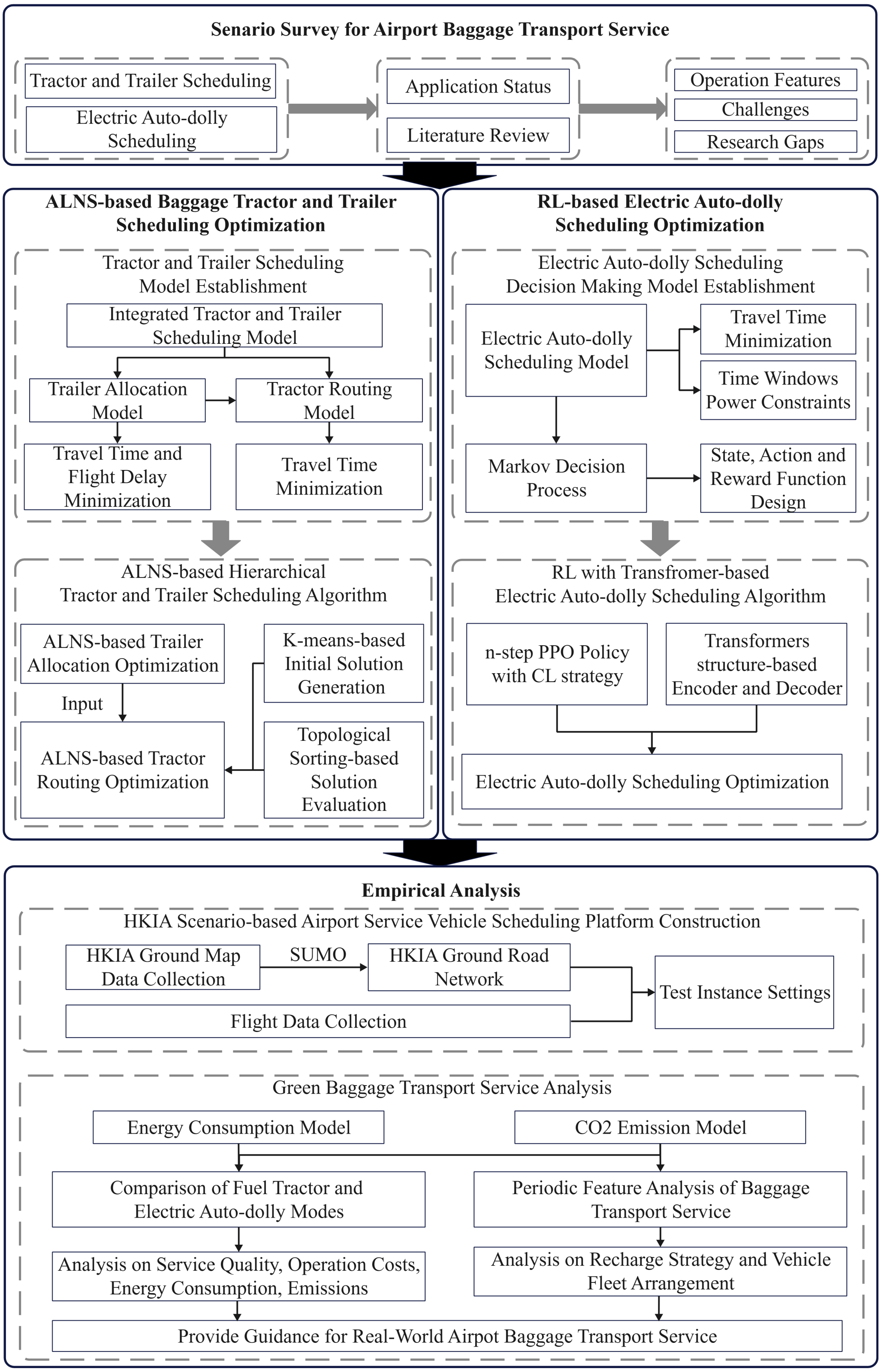


Figure 1.4: The research framework of intelligent vehicle scheduling for green airport baggage transport service.

2 LITERATURE REVIEW

In this chapter, the literature on relevant studies of operations research in airport ground handling and optimization methods is reviewed. First, the state-of-the-art vehicle scheduling research for airport ground handling services is summarized to investigate compensatory and potential research directions of baggage transport vehicle scheduling problems. Regarding the different challenges presented by the multi-trailer drop-and-pull mode and the electric auto-dolly for baggage transport operations, the drop-and-pull routing problem and the electric vehicle routing problem are reviewed, respectively. In the aspect of optimization methods, the reinforcement learning-based methods for routing problems are investigated. Finally, the research limitations and gaps are summarized.

2.1 VEHICLE SCHEDULING FOR AIRPORT GROUND HANDLING

Focusing on improving the operational efficiency of various airport ground services, extensive works have been conducted for the scheduling of different types of service vehicles, such as de-icing vehicles [31], [32], ferry buses [33], fuelling vehicles [34], and aircraft towing vehicles [35]. Meanwhile, some works considered the interactions among different vehicles to improve the overall efficiency of airport ground handling, e.g., the scheduling of multi-type airport service vehicles with service priority [36], [37] and collaborative optimization of different ground activities [38].

With the popularity of electric vehicles, some scholars have also begun to investigate the application of electric vehicles in airport ground service. In recent studies, Bao *et al.*

developed an integrated operational framework for hybrid fuel-electric aircraft tractors. This model simultaneously optimizes temporal efficiency, energy expenditure, and carbon footprint by incorporating auxiliary power unit (APU) energy substitution effects. Then, an enhanced adaptive large-scale neighborhood search heuristic is designed to obtain near-optimal solutions for this complex optimization challenge [28]. Zoutendijk *et al.* also investigated the electric aircraft tractor routing and recharging problem, while considering a limit for the supply of energy.

Table 2.1: Summary of literature related to vehicle scheduling for airport baggage transport service.

Literature	Objective (Minimum)	Time window	Vehicle type classification[1]	Capacity =1	Capacity >1	Two-way service[2]	Drop-and-pull	Method
[39]	The total cost of fixed and fuel consumption of tractors	✓	✓		✓	✓		ALNS
[40]	The total cost of fixed consumption of all vehicles and fuel consumption of tractors	✓	✓		✓		✓	ALNS
[41]	The number of used vehicles and the total travel distance of vehicles	✓			✓			GA
[18]	The number of used vehicles and the total extra time cost of vehicles	✓		✓				NSGA-II
[42]	The vehicle travel time	✓			✓			LNS
[43]	The number of undelivered baggage and the vehicle travel time	✓			✓			Greedy
This paper	The total travel time of tractors	✓	✓		✓	✓	✓	ALNS

[1] Distinguishing between the tractor and trailer in the baggage transport vehicle.
[2] Allowing vehicles to serve both departure and arrival flights in one trip.

In recent years, as an essential part of airport ground handling, vehicle scheduling in airport baggage transport has been extensively studied, and related works are summarized in Table 2.1. Most works focused on minimizing operation costs, including the number of vehicles adopted, travel time, fuel consumption, etc. Some considered reducing flight delays, e.g., Padrón *et al.* studied the bi-objective collaborative scheduling of multiple service vehicles to minimize flight waiting time [44]. Considering the constraints of vehicle capacity and service time window, Clausen *et al.* and Guo *et al.* modeled the vehicle scheduling problem as a Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) [41], [43]. Meanwhile, various heuristic algorithms have been developed, including Variable Neighborhood Search (VNS) [44], Large Neighborhood Search (LNS)

[42], and ALNS [39]. Besides, some attempted to integrate learning-based methods with heuristic methods. For example, Zhou *et al.* proposed a learning-assisted LNS method that integrates imitation learning and the graph convolutional network [42]. They further improved the heuristic policy with an attention-based neural network trained with reinforcement learning [14].

2.2 DROP-AND-PULL ROUTING PROBLEM

Table 2.2: Summary of literature related to tractor and trailer drop-and-pull problem.

Literature	Application	Vehicle type	Number of Trailers			Mixed Transport[1]	Routes Interdependence[2]	Method
			=1	=2	>2			
[45]	Intercity Freight	Truck+trailer	✓					Branch-and-price-and-cut
[46]			✓					Hybrid metaheuristic
[24]	Container drayage at ports	Tractor+trailer	✓				✓	Branch-and-price-and-cut
[47]				✓		✓		GA
[48]				✓		✓	✓	ALNS
[49]			✓					Branch-and-price-and-cut
This paper	Airport baggage transport	Tractor+trailer			✓	✓	✓	ALNS

[1] Allowing tractors to transport both vacant and full trailers in one trip
[2] A trailer can be transported alternately by multiple tractors.

The drop-and-pull mode has been widely applied in freight transportation. In intercity freight transportation, a vehicle is typically composed of a truck and a trailer [50], where both truck and trailer can load cargo, allowing for the shifting of cargo between them. In such problems, the transport demand usually involves distributing cargo from the depot to customers or collecting cargo from customers, and scheduling the truck to transport both vacant and full trailers in one trip is rarely considered. The common scenario is that some customers could only be served by trailers, while others can only be served by trucks. Thus, the truck sometimes needs to drop the trailer off at a transit station before implementing tasks [45], [46].

Another typical application is the container drayage problem at ports. Different from trucks, the tractor cannot carry cargo and is usually only used to pull trailers. One tractor normally can pull at most two trailers [47], [48]. In particular, A foldable container drayage problem is researched in [26], demonstrating tractors' capability to transport 4-

6 folded units versus single loaded containers. For the solution algorithms, Song *et al.* formulated a branch-and-price-and-cut framework addressing synchronized drayage routing with rigid synchronization constraints [49]. In their work, one drayage task finished by the same tractor is forbidden, so there are no route dependencies between different tractors [49]. Moghaddam et al. established a multi-modal container routing paradigm incorporating heterogeneous container dimensions and decoupled tractor-trailer operations, significantly expanding solution space dimensionality [51]. Similarly, tractor change is not allowed during one task, and the tractor can only engage in single-trip services.

2.3 ELECTRIC VEHICLE ROUTING PROBLEM

Over the past decade, electric vehicles (EVs) have garnered significant attention and have witnessed a substantial surge in market share, which is prompted by the mounting global consciousness towards environmental sustainability. Recently, EVs have been widely used in public transportation systems [52], freight [53], yard [54], and last-mile transportation [55].

Recent advancements in sustainable logistics have driven significant research efforts toward optimizing electric vehicle (EV) fleet management systems [56]. A critical development in this domain is the Electric Vehicle Routing Problem (EVRP), a specialized adaptation of the classical VRP. Unlike conventional VRP formulations, EVRP explicitly incorporates battery capacity limitations, charging station location constraints, and time-energy consumption coupling effects [57]. With the consideration of the routing constraints, charging operations, etc, many works have emerged over the years focusing on EVRP variants, which can be classified as following attributes:

Charing and discharging attributes. Charging and discharging attributes are popularly discussed in many works, as they are very important for a realistic EVRP. Conventional charging modeling approaches typically posit a direct linear relationship between state-of-charge (SOC) and charging duration [58]. However, emerging research incorporates state-dependent charging profiles that better reflect electrochemical dynamics. Montoya *et al.* extended the EVRP by considering a non-linear charging function coupled with a hybrid metaheuristic combining iterated local search with heuristic concentration techniques to handle solution space discontinuities. [59]. Besides, a linear model between the energy consumption and travel time is commonly used for the vehicle discharging process. But it actually may be influenced by many other realistic factors, like speed, weight [60], temperature [61], etc. Lastly, battery swapping is also investigated in some research to introduce an alternative strategy for replenishing the energy of EVs [62], [63].

Recharge station attributes. In most of the literature, the recharging station information is assumed to be known in advance. However, this approach doesn't always align with real-world complexities, such as the rapidly evolving urban landscapes. This interdependence necessitates co-optimization of geo-spatial allocation of charging stations and operational scheduling of electric vehicles [64], [65]. Advanced EVRP variants further integrate power grid operational constraints, like capacity thresholds of substations, peak load balancing mechanisms at charging hubs, and so on [56]. Moreover, different recharge stations may provide different recharge speeds with different prices, which can also affect the selection of recharge stations in the EVRP [66].

Vehicle fleet attributes. The operational constraints of EVs, particularly regarding payload-range trade-offs, drive logistics research studies toward heterogeneous fleet compositions integrating EVs with conventional fuel vehicles (FVs) [67]. Compared to traditional fuel vehicles, the energy cost of EVs are relatively lower, followed by a higher purchase cost. Hybrid electric vehicles (HEVs) further complicate the advantages of EVs and FVs through dual-propulsion architectures, enabling dynamic energy source switching between battery packs and diesel generators. Hiermann *et al.* addressed the mixed fleet scheduling problem that contains FVs, EVs, and HEVs, which was solved by a hybrid meta-heuristic framework [68].

2.4 REINFORCEMENT LEARNING FOR ROUTING PROBLEMS

The temporal state transition modeling capabilities of reinforcement learning (RL) have positioned it as a dominant paradigm for combinatorial routing optimization. RL-driven frameworks demonstrate particular effectiveness in addressing NP-hard challenges, such as traveling salesman problems, Capacity-constrained routing, and demand-coupled logistics [69]–[72]. Current methodological innovations on RL-based algorithms primarily involve the following perspectives,

RL algorithms. RL algorithms mainly include two principal optimization paradigms according to the differences in learning objectives [73], [74]. The first value-centric approach approximates state-action value functions, which enables solution construction by temporal difference estimation. Representative implementations include Deep Q-Networks (DQN) [75], [76] and its variants, double DQN and Dueling DQN [77]. The second category is policy-gradient paradigms, which directly parameterize the policy to optimize action selection probability distribution through gradient ascent on expected returns. The existing popular policy-based RL algorithms include REINFORCE [70], [78] and Proximal Policy Optimization (PPO) [79], [80]. In existing research, policy-based RL algorithms are the mainstream of RL algorithms designed for routing problems, due to their advantages in handling high-dimensional action spaces and naturally supporting stochastic policies, which can be beneficial in uncertain environments.

In the domain of combinatorial routing optimization, both value-centric and policy-gradient RL algorithms have been studied [73]. On the one hand, value functions guide the decision-making process, enabling the system to learn optimal routes through iterative evaluations. On the other hand, policy-gradient methods have advantages in exploring the action space and can adapt more swiftly to complex and high-dimensional decision-making scenarios in routing. Existing popular policy-based RL methods include REINFORCE [70], [78], Proximal Policy Optimization (PPO) [79], [80], etc.

Heuristic types of RL policy. Generally, the heuristic types of RL policy can be categorized as constructive [71], [81], and improvement heuristics [82], [83]. The construction methods learn a policy to build solutions by iteratively selecting nodes from the problem graph. Although this approach can save more inference time cost, it often lacks the ability to obtain (near-)optimal solutions [84]. On the contrary, with the initialized solutions, improvement strategies utilize local search mechanisms to iteratively optimize the initial solution. Typical examples include the 2-opt operator proposed by [82], the ruin-and-repair operation from [85], and the node-swapping technique developed in [86]. Improvement model-based RL is advantageous for its scalability in complex environments but may suffer from sample inefficiency.

Deep learning models of RL policy. Various deep learning architectures such as Recurrent Neural Networks (RNN) [86]–[88], Graph neural networks (GNN) [89], [90], and Transformers [71], [91] have been employed. Different from RNN and GNN, Transformers allow for parallel processing of input sequences, which significantly speeds up training and inference. The multi-head self-attention mechanism enhances the ability of Transformers to effectively capture long-range dependencies in the data. These advantages make Transformer particularly well-suited for complex RL tasks.

2.5 SUMMARY

In the above literature review, many scholars have made some achievements in the research of airport baggage transport vehicle scheduling. However, there are still some limitations need to be further investigated.

Firstly, previous works provide extensive knowledge on scheduling baggage transport vehicles. However, they seldom considered the drop-and-pull mode or considered both departure and arrival flights together. Works are still needed on the integrated scheduling of tractors and trailers with the consideration of various practical requirements so as to support real-life operations. Besides, it is important to highlight that simulations and experiments conducted in existing related research often lack the utilization of actual flight and map information obtained from real-world airports.

Secondly, to the best of our knowledge, vehicle scheduling under drop-and-pull mode in airport baggage transport is still not well studied. Different from previous works, a tractor can pull more than three trailers simultaneously in airport baggage transport. Meanwhile, considering both arrival and departure flights, the tractor should be allowed to transport both vacant and fully loaded trailers in one trip.

Thirdly, although many real-world EVRP variants have been well studied, research on electric auto-dolly scheduling is still very limited. Unlike the scenarios of cargo and public transport systems, where electric vehicle power consumption can be more accurately estimated, the working status and energy consumption of electric auto-dollies are highly correlated with flight plans, which increases the difficulty in making charging decisions. Besides, due to capacity limitations, electric auto-dollies need to make multiple trips between the BHA and aircraft stands. Incorporating charging-related decisions into such multi-trip scenarios would greatly increase the complexity of the problem, which in turn would require a more efficient solution method.

Finally, although reinforcement learning has been proven to be effective in solving large-scale combinatorial optimization problems recently However, to our knowledge, most works still mainly focus on some classical problems with few constraints (e.g. TSP, VRP, and the Jop Shop Problem), and they are rarely tested on datasets from real scenarios. Moreover, the electric auto-dolly scheduling problem in this study needs to take into account the uncertain charging time, which is also a challenge for the training and convergence of reinforcement learning algorithms.

Based on the above analysis, this study aims to fill the gaps in the current research on airport baggage transport vehicle scheduling. A vehicle scheduling problem for multi-trailer drop-and-pull baggage transport is first studied, which enables the flexible coupling and decoupling of tractors and trailers. Then, a hybrid intelligence-based algorithm integrating K-means clustering and ALNS is designed to solve this problem. Besides, the electric auto-dolly scheduling problem is solved by a reinforcement learning and Transformers variant-based algorithm, which shows its advantages in large-scale real-world instances. Finally, supported by real-world airport information, an integrated airport service vehicle scheduling simulation platform is established to contribute to algorithm improvement and provide references for future airport baggage transport.

3 VEHICLE SCHEDULING FOR MULTI-TRAILER DROP-AND-PULL AIRPORT BAGGAGE TRANSPORT

This chapter aims to propose an efficient scheduling method for multi-trailer drop-and-pull airport baggage transport. Firstly, a two-stage model for the multi-trailer drop-and-pull problem in airport baggage transport is developed, which can effectively decrease the computation complexity without affecting the scheduling performance. Secondly, a hybrid intelligence-based method that integrates K-means clustering and Adaptive Large Neighborhood Search (ALNS) is developed to efficiently solve the two-stage scheduling problem. To further improve the algorithm efficiency, several effective operators and a topological sort-based solution evaluation method are proposed that could accelerate the computing processes and effectively cope with large-scale problems. Finally, extensive experimental case studies are conducted to verify the effectiveness of the proposed method and provide benchmarks for future works.

3.1 PROBLEM DESCRIPTION

In a typical airport baggage transport scenario, the drop-and-pull mode with motorized tractors and non-motorized trailers is popularly adopted. Generally, there are two main transport tasks. One is for departing flights. The baggage will be collected at the BHA and loaded onto the trailers waiting there. After the baggage check-in process (a cut-off time at the BHA before each flight), these trailers will be towed by tractors to the corresponding aircraft stand, and then the baggage will be loaded onto the aircraft by ground handling operators. The process should be conducted within strict time windows according to the flight plan to ensure the aircraft can take off on time. The other is for arrival flights. After an aircraft arrives at the aircraft stand, the baggage will be unloaded from the aircraft and loaded onto the trailers waiting there. These trailers will then be towed by tractors to the BHA. The process should also be conducted efficiently to minimize the waiting time of passengers at the baggage claim area. The process is illustrated in Fig. 3.1.

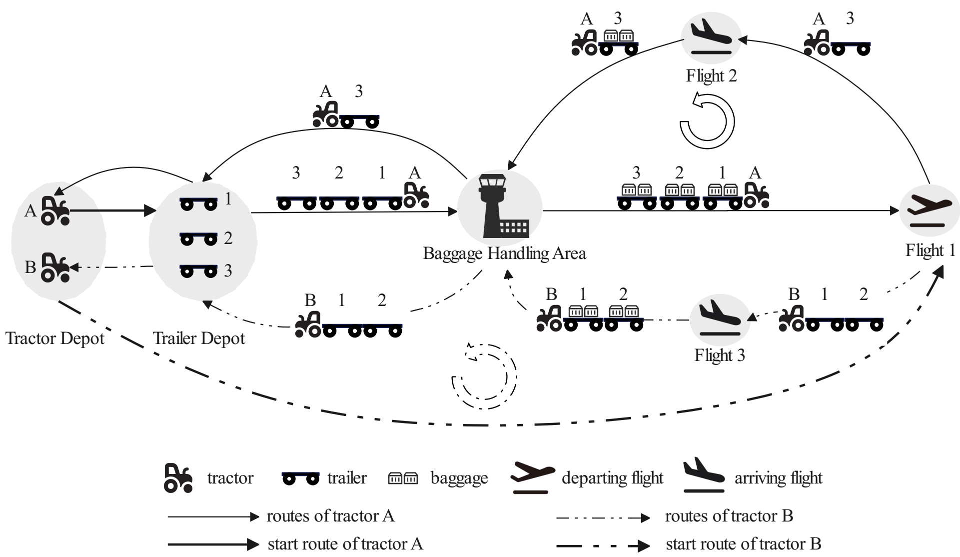


Figure 3.1: The example of 2 tractors and 3 trailers transporting baggage for 3 flights.

Considering a specific planning period, the schedules of tractors and trailers are decided based on the information of arrival and departure flights simultaneously, including the arrival/departure time and the number of trailers required for each flight. Specifically, for each flight, the information required is listed as follows:

1. *Number of Trailers Required*: According to the volume of baggage the flight carries, the number of trailers needed will be known in advance before the scheduling starts.

2. *O-D Information*: The origination and destination information of each baggage transport demand will be generated based on the nature of the arrival/departure flight and the stand that the corresponding aircraft parks.

3. *Earliest Baggage Loading Start Time*: It refers to the earliest time that the baggage can be loaded onto the trailers.

4. *Latest Baggage Unloading End Time*: It refers to the latest time for completing unloading baggage from trailers.

With the above information, the schedules of tractors and trailers in a given period can be made. For example, considering the scenario in Fig. 3.1, the number of trailers needed for Flight 1, 2, and 3 are 3, 1, and 2, respectively. A feasible solution is that Flight 1 is served by all three trailers, Flight 2 is served by Trailer 3, and Flight 3 is served by Trailer 1 and 2. Tractor A departs from the tractor depot, picks up all three trailers at the trailer depot, and tows them to the aircraft stand of Flight 1. After disconnecting Trailer 1 and 2, Tractor A tows Trailer 3 to serve Flight 2, while Trailer 1 and 2 are left at the aircraft stand of Flight 1. After the baggage is unloaded, they will be towed by Tractor B to serve Flight 3 before returning to the depot. In this process, one tractor can pull multiple trailers simultaneously, and the tractors and trailers can flexibly connect and disconnect when needed.

3.2 INTEGRATED MODEL

Considering an airport baggage transport scenario in a given period, the sets of tractors and trailers are M and K, respectively. The notations adopted are given in Table 3.1. According to the above analysis, the typical locations involved in the problem include the depot of tractors/trailers, BHA, and aircraft stands. Since the tractors and trailers can load and unload baggage at any aircraft stand and BHA, to make the problem clear, this paper defines these locations as pickup and delivery nodes of baggage in the model. Thus, the transport network can be denoted as $G_M = (N_M, A_M)$. Here, $N_M = I_S \cup I_E \cup \{s_M\}$, which includes baggage loading and unloading nodes of all flights in set F and the trailer depot s_M.

Table 3.1: Notations for Chapter 3.

	Indicates
f	Index for flights.
i, j	Index for baggage loading and unloading nodes, $i, j \in N_M$.
u, v	Index for trailer pickup and delivery nodes, $u, v \in N_K$.
s_M	The depot of trailers, $s_M \in N_M$.
s_K	The depot of tractors, $s_K \in N_K$.
f_S	The baggage loading node of flight f, $f_S \in N_M$.
f_E	The baggage unloading node of flight f, $f_E \in I_E$.
p_i^m	The pickup request from trailer m at node i, $p_i^m \in N_K$.
d_i^m	The delivery request from trailer m at node i, $d_i^m \in N_K$.
p_l^m	The lth pickup request of trailer m.
d_l^m	The lth delivery request of trailer m.
$m, \tilde{m}$	Indicating trailers, $m, \tilde{m} \in M$.
k	Indicating tractors, $k \in K$.
	Sets
M	Set of trailers.
M^*	Set of used trailers in the solution of the trailer allocation problem.
K	Set of tractors.
F	Set of flights.
I_S	Set of the baggage loading nodes of all flights, $I_S \subset N_M$.
I_E	Set of the baggage unloading nodes of all flights, $I_E \subset N_M$.
N_M	Set of service nodes for trailers and s_M.
N_K	Set of all possible pickup and delivery nodes.
N_K^*	Set of pickup and delivery request from set M^* and s_K.
P_m	Set of pickup requests from trailer m.
D_m	Set of delivery requests from trailer m.
P	Set of pickup requests from M^*, $P = \bigcup_{m \in M} P_m \subset N_K^*$.
D	Set of delivery requests from M^*, $D = \bigcup_{m \in M} D_m \subset N_K^*$.
	Parameters
t_{ij}	Travel time from node i to j.
$\tilde{t}_{uv}$	Travel time from node u to v.
el_f	Earliest baggage loading start time for flight f.
e_u	Earliest pickup or delivery start time for node u.
t_F	Allowed duration time of baggage transport for each flight.
t_M	Average time for baggage loading or unloading.
r_u	Load of node u (r_s denotes the load of depot).
Q	Capacity of a tractor.
W	A sufficiently large positive constant.
	Decision Variables
x_{ij}^m	$= 1$, if trailer m successively serves at node i and node j, or 0 otherwise.
y_{uv}^k	$= 1$, if tractor k travels from node u to node v, or 0 otherwise.
z_u	The order of tractors visiting node u.
BL_i	Baggage loading or unloading start time at node i.
B_u	Pickup or delivery start time at node u.
R_u^k	Load of tractor k after serving node u.

There are two key decisions. One is deciding the route of trailers, denoted as x_{ij}^m. It equals 1 if trailer m successively serves node j after severing node i or 0 otherwise. The other is deciding the route of tractors, denoted as y_{uv}^k. The routes of tractors are highly dependent on the routes of trailers. For example, if $x_{ij}^m = 1$, there must be a tractor visiting node i to pick up trailer m and then delivering it to node j. Thus, the routes of tractors are defined on the network $G_K = (N_K, A_K)$, where $N_K = \{p_i^m, d_i^m | i \in N_M, m \in M\} \cup \{s_K\}$, where p_i^m and d_i^m indicate the request to pickup trailer m at node i and deliver trailer m to node i, respectively. It is noted that N_K contains all possible pickup and delivery requests from all trailers at all nodes in N_M. Therefore, $y_{uv}^k = 1$ if trailer k implement request at node u and v successively, or 0 otherwise.

The objective of airport baggage transport vehicle scheduling is to minimize the operation cost of all the vehicles while ensuring service quality under strict time requirements. Thus, the model can be developed as follows:

$$\text{Minimize} \sum_{u,v \in N_K} \sum_{k \in K} \tilde{t}_{uv} y_{uv}^k \tag{3.1}$$

s.t. :

$$\sum_{i \in N_M} x_{s_M i}^m = \sum_{i \in N_M} x_{i s_M}^m = 1, \forall m \in M \tag{3.2}$$

$$\sum_{u \in N_K} x_{s_K u}^k = \sum_{u \in N_K} x_{u s_K}^k = 1, \forall k \in K \tag{3.3}$$

$$\sum_{j \in N_M, j \neq i} x_{ij}^m = \sum_{j \in N_M, j \neq i} x_{ji}^m, \forall i \in N_M, m \in M \tag{3.4}$$

$$\sum_{v \in N_K, v \neq u} y_{uv}^k = \sum_{v \in N_K, v \neq u} y_{vu}^k, \forall v \in N_K, k \in K \tag{3.5}$$

$$\sum_{m \in M} x_{f_S f_E}^m = 1, \forall f \in F \tag{3.6}$$

$$\sum_{m \in M} \sum_{j \in N_M, j \neq i} x_{ij}^m = 1, \forall i \in N_M \backslash \{s_M\} \tag{3.7}$$

$$\sum_{m \in M} \sum_{j \in I_E, i \neq j} x_{ij}^m = 0, \forall i \in I_E \tag{3.8}$$

$$\sum_{k\in K}\sum_{v\in N_K, v\neq p_i^m} y_{p_i^m v}^k \leq 1, \forall i \in N_M, m \in M \tag{3.9}$$

$$\sum_{k\in K}\sum_{i\in N_M}\Big(\sum_{v\in N_K, v\neq p_i^m} y_{p_i^m v}^k + \sum_{v\in N_K, v\neq d_i^m} y_{d_i^m v}^k\Big) \leq \tag{3.10}$$

$$W * \sum_{j\in N_M\setminus\{s_M\}} x_{s_M j}^m, \forall m \in M$$

$$\sum_{k\in K}\sum_{v\in N_K, v\neq p_i^m} y_{p_i^m v}^k = \sum_{j\in N_M, j\neq i} x_{ij}^m, \forall i \in N_M, m \in M \tag{3.11}$$

$$\sum_{k\in K}\sum_{v\in N_K, v\neq d_j^m} y_{v d_j^m}^k = \sum_{i\in N_M, i\neq j} x_{ij}^m, \forall j \in N_M, m \in M \tag{3.12}$$

$$\sum_{k\in K}\sum_{\tilde{m}\in M, \tilde{m}\neq m}\sum_{v\in N_K, v\neq p_i^{\tilde{m}}} y_{p_i^{\tilde{m}} v}^k \leq W(1 - x_{ij}^m), \tag{3.13}$$

$$\forall i \in N_M\setminus\{s_M\}, j \in N_M, i \neq j, m \in M$$

$$\sum_{k\in K}\sum_{\tilde{m}\in M, \tilde{m}\neq m}\sum_{v\in N_K, v\neq d_j^{\tilde{m}}} y_{v d_j^{\tilde{m}}}^k \leq W(1 - x_{ij}^m), \tag{3.14}$$

$$\forall i \in N_M, j \in N_M\setminus\{s_M\}, i \neq j, m \in M$$

$$\sum_{v\in N_K, v\neq p_i^m} y_{p_i^m v}^k - \sum_{v\in N_K, v\neq d_j^m} y_{v d_j^m}^k \leq W(1 - x_{ij}^m), \tag{3.15}$$

$$\forall i, j \in N_M, i \neq j, m \in M, k \in K$$

$$\sum_{v\in N_K, v\neq p_i^m} y_{p_i^m v}^k - \sum_{v\in N_K, v\neq d_j^m} y_{v d_j^m}^k \geq W(x_{ij}^m - 1), \tag{3.16}$$

$$\forall i, j \in N_M, i \neq j, m \in M, k \in K$$

$$\max\{0, r_u\} \leq R_u^k \leq \min\{Q, Q + r_u\}, \forall u \in N_K, k \in K \tag{3.17}$$

$$R_u^k + r_v \leq R_v^k + W\left(1 - y_{uv}^k\right), \forall u, v \in N_K, u \neq v, k \in K \tag{3.18}$$

$$B_{p_i^m} \geq \max\{e_{p_i^m}, B_{d_i^m} + t_M\}, \forall m \in M, i \in N_M\setminus\{s_M\} \tag{3.19}$$

$$e_{p_{s_M}^m} \leq B_{p_{s_M}^m} \leq B_{d_{s_M}^m}, \forall m \in M \tag{3.20}$$

$$B_{d_i^m} \geq e_{d_i^m}, \forall m \in M, i \in N_M \tag{3.21}$$

$$B_{d_i^m} \leq e_{d_i^m} + t_F, \forall m \in M, i \in I_E \tag{3.22}$$

$$B_u + \tilde{t}_{uv} \leq B_v + W\left(1 - y_{uv}^k\right), \forall u, v \in N_K, u \neq v, k \in K \tag{3.23}$$

$$B_{p_i^m} \leq B_{d_j^m} + W(1 - x_{ij}^m), \forall i, j \in N_M, i \neq j, m \in M \tag{3.24}$$

$$z_u - z_v + W * y_{uv}^k \leq W - 1, \forall u \in N_K, \tag{3.25}$$

$$v \in N_K \backslash \{s_K\}, u \neq v, k \in K$$

$$x_{ij}^m \in \{0, 1\}, \forall i, j \in N_M, m \in M \tag{3.26}$$

$$y_{uv}^k \in \{0, 1\}, z_u \geq 0, \forall u, v \in N_K, k \in K \tag{3.27}$$

The objective (3.1) is to minimize the total travel time of all tractors. Constraint (3.2) requires all trailers to be parked at s_M at the very beginning and returned there after serving all flights. Similarly, constraint (3.3) requires all tractors to start from the depot and return to s_K after completing all pickup and delivery requests. Constraints (3.4) and (3.5) are the flow balance constraints of trailers and tractors, respectively.

Constraints (3.6)-(3.16) are the baggage transport service demand constraints. In this work, the concept of *virtual flight* is introduced, based on which a flight requiring multiple trailers can be represented as multiple virtual flights requiring only one trailer. In this way, the modeling process can be greatly simplified. Subsequent discussions are all based on virtual flights.

Constraints (3.6)-(3.8) ensure that the baggage loading demands must be satisfied, including: 1) The baggage loading and unloading node of each flight must be visited by trailers once and only once; 2) If one trailer leaves the baggage loading node of any flight, it should move to its baggage unloading node directly; 3) If one trailer leaves the baggage unloading node of any flight, it cannot move to another unloading node directly.

Constraints (3.9)-(3.16) restrict the relationship between the routes of trailers and tractors. That is, the trailers can only be moved by tractors: 1) Each pickup or delivery request can be implemented at most once; 2) If trailer m is not used by any flight, it will not be picked up or delivered; 3) If trailer m did not access node i, it will not be picked up at node i; on the contrary, if trailer m accesses node i, there must be one and only one tractor picking up trailer m at node i. The same goes for the trailer delivery requests; 4) If trailer m successively visits node i and j, i.e. $x_{ij}^m = 1$, it will be picked up at node i and delivered to node j once by the same tractor, respectively, and other trailers will not be picked up at node i or delivered to node j.

Constraint (3.17) ensures that the capacity of any tractor should not exceed its capacity after visiting any node. Constraint (3.18) ensures the load consistency of tractors.

Constraints (3.19)-(3.24) are related to the time and order of tractors picking up and delivering trailers. Constraints (3.19)-(3.22) are the time window requirement for picking up and delivering trailers, including: 1) Every pickup and delivery request must be implemented after the allowed earliest pickup and delivery time; 2) If trailer m is needed to be delivered to a baggage unloading node, the arrival time of trailer m must be no later than $e_{d_i^m} + t_F$; 3) If trailer m is delivered to a node i outside the depot s_M, trailer m will be occupied for t_M to load or unload baggage. Thus, the time of trailer m being re-picked up $B_{p_i^m}$ must be no earlier than the end time of baggage loading or unloading; 4) All scheduled trailers must be picked up first and delivered to the depot s_M.

Constraint (3.23) ensures the time consistency of tractors. Constraint (3.24) requires that if trailer m needs to successively visit node i and node j, the time of delivering trailer m to node j must be no earlier than the time of picking it up at node i.

Constraint (3.25) is the Miller-Tucker-Zemlin sub-tour constraint to eliminate subtours in the routes of tractors [92]. Constraints (3.26)-(3.27) define the range of variables.

3.3 TWO-STAGE MODEL

The above integrated scheduling model is a special and complex variant of the classical Vehicle Routing Problem (VRP), which has already been proven to be NP-hard in the literature. Since the model involves cooperative routing of tractors and trailers, incorporating time windows and pickup and delivery requirements, it should also be NP-hard, demonstrating greater complexity. Moreover, the scale of network G_K expands exponentially with the increase of flights and trailers numbers, where $|N_K| = 2 * (2 * |F| + 1) * |M| + 1$. Numerous constraints are also introduced to delineate the route dependencies between tractors and trailers, leading to high model complexity. Therefore, to efficiently solve the problem, this part will re-formulate the integrated scheduling model as a two-stage scheduling model. The first stage optimizes the routes of trailers. Then, in the second stage, the routes of tractors will be optimized to fulfill the pickup and delivery requests from the generated trailer routes, rather than all potential requests N_K. The order of tractors transporting trailers is also determined by trailer routes, simplifying the route dependencies between trailers and tractors.

3.3.1 Trailer allocation model

Based on the integrated scheduling model, the trailer allocation model is defined on the transport network G_M with decision variables x_{ij}^m and BL_i. The mathematical formulation is presented below.

$$\text{Minimize} \sum_{i,j \in N_M} \sum_{m \in M} t_{ij} x_{ij}^m + \sum_{f \in F, f_E \in I_E} (BL_{f_E} - el_f) \tag{3.28}$$

s.t. :

$$\sum_{i \in N_M} x_{s_M i}^m = \sum_{i \in N_M} x_{i s_M}^m = 1, \forall m \in M \tag{3.29}$$

$$\sum_{j \in N_M, j \neq i} x_{ij}^m = \sum_{j \in N_M, j \neq i} x_{ji}^m, \forall i \in N_M, m \in M \tag{3.30}$$

$$\sum_{m \in M} x_{f_S f_E}^m = 1, \forall f \in F \tag{3.31}$$

$$\sum_{m \in M} \sum_{j \in N_M, j \neq i} x_{ij}^m = 1, \forall i \in N_M \backslash \{s_M\} \tag{3.32}$$

$$\sum_{m \in M} \sum_{j \in I_E, i \neq j} x_{ij}^m = 0, \forall i \in I_E \tag{3.33}$$

$$BL_{f_S} \geq el_f, \forall f_S \in I_S, f \in F \tag{3.34}$$

$$BL_{f_E} + t_M \leq el_f + t_F, \forall f_E \in I_E, f \in F \tag{3.35}$$

$$BL_i + t_{ij} + t_M \leq BL_j + W\left(1 - x_{ij}^m\right), \tag{3.36}$$

$$\forall i, j \in N_M, i \neq j, m \in M$$

$$x_{ij}^m \in \{0, 1\}, \forall i, j \in N_M, m \in M \tag{3.37}$$

The objective function (3.28) minimizes the total travel time of all trailers and the flight waiting time. The former objective aims to minimize operating costs, while the latter seeks to create opportunities for tractors to pick up and deliver multiple trailers in subsequent stages.

Constraints (3.29)-(3.33) ensure the flow balance and service demand are satisfied, which are consistent with constraints (3.2), (3.4), (3.6)-(3.8) in the integrated scheduling model. Constraints (3.34)-(3.36) describe the service time window and time consistency requirements. Constraint (3.37) defines the value range of x_{ij}^m, which is consistent with constraint (3.26).

3.3.2 Tractor routing model

Each movement of trailer m requires one pickup and one delivery operation. Thus, the demand arising from predetermined trailer routes can be denoted by sequential lists involving paired pickup and delivery requests $\{p_1^m, d_1^m, p_2^m, d_2^m, ..., p_{n_m}^m, d_{n_m}^m\}$, $\forall m \in M^*$, where M^* is the set of used trailers, n_m is the number of request pairs from trailer m, p_l^m and d_l^m denote the lth pair of pickup and delivery requests on the route of trailer m, respectively. Fig. 3.2 shows an example of transferring the route of trailer m into pickup and delivery requests for subsequent tractor routing. As shown in Fig. 3.2, trailer m is allocated to two flights, so the route of trailer m contains five nodes, including the depot s_M and baggage loading and unloading locations of two flights. Thus, the demand generated from trailer m can be denoted by a sequential list containing four pairs of pickup and delivery requests $\{p_1^m, d_1^m, p_2^m, d_2^m, p_3^m, d_3^m, p_4^m, d_4^m\}$.

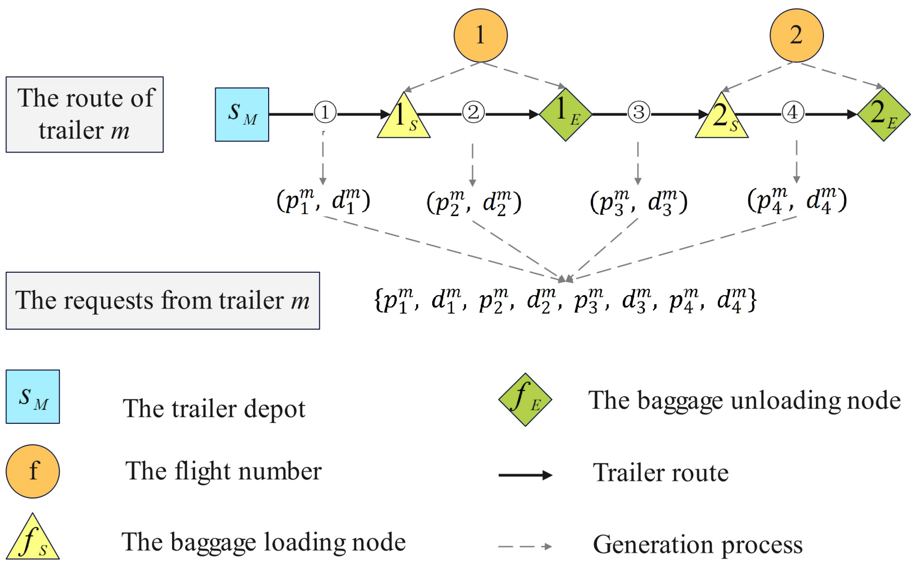


Figure 3.2: The pickup and delivery requests generation process of the route of trailer m

Tractors are scheduled to pick up and deliver trailers following the specified sequence within the required time window. The tractor routing problem is defined on a sub-network of G_K: $G_K^* = (N_K^*, A_K^*)$, where $N_K^* = P \cup D \cup \{s_K\}$, P and D are the set of pickup and delivery request nodes from trailer set M^*, respectively. The decision variables involve y_{uv}^k, z_u, B_u, and R_u^k. The mathematical formulation is presented below.

$$\text{Minimize} \sum_{u,v \in N_K^*} \sum_{k \in K} \tilde{t}_{uv} y_{uv}^k \tag{3.38}$$

s.t. :

$$\sum_{k \in K} \sum_{v \in N_K^*, v \neq u} y_{uv}^k = 1, \forall u \in P \tag{3.39}$$

$$\sum_{v \in N_K^*, v \neq p_l^m} y_{p_l^m v}^k = \sum_{v \in N_K^*, v \neq d_l^m} y_{v d_l^m}^k, \tag{3.40}$$

$$\forall p_l^m \in P_m, d_l^m \in D_m, m \in M^*, k \in K$$

$$\sum_{v \in N_K^*, v \neq u} y_{uv}^k = \sum_{v \in N_K^*, v \neq u} y_{vu}^k, \forall u \in P \cup D, k \in K \tag{3.41}$$

$$\sum_{v \in N_K^*} y_{s_K v}^k = \sum_{u \in N_K^*} y_{u s_K}^k = 1, \forall k \in K \tag{3.42}$$

$$B_u + \tilde{t}_{uv} \leq B_v + W\left(1 - y_{uv}^k\right), \forall u, v \in N_K^*, u \neq v, k \in K \tag{3.43}$$

$$R_u^k + r_v \leq R_v^k + W\left(1 - y_{uv}^k\right), \forall u, v \in N_K^*, u \neq v, k \in K \tag{3.44}$$

$$e_u \leq B_u, \forall u \in P \cup D \tag{3.45}$$

$$B_{d_l^m} \leq e_{d_l^m} + t_F, \forall d_l^m \in D_m, m \in M^* \tag{3.46}$$

$$B_{p_l^m} \leq B_{d_l^m}, \forall p_l^m \in P_m, d_l^m \in D_m, m \in M^* \tag{3.47}$$

$$B_{d_l^m} + t_M \leq B_{p_{l+1}^m}, \forall p_{l+1}^m \in P_m, d_l^m \in D_m, m \in M^* \tag{3.48}$$

$$\max\{0, r_u\} \leq R_u^k \leq \min\{Q, Q + r_u\}, \forall u \in N_K^*, k \in K \tag{3.49}$$

$$z_u - z_v + W * y_{uv}^k \leq W - 1, \forall u \in N_K^*, \tag{3.50}$$

$$v \in N_K^* \backslash \{s_K\}, u \neq v, k \in K$$

$$y_{uv}^k \in \{0, 1\}, z_u \geq 0, \forall u, v \in N_K^*, k \in K \tag{3.51}$$

The objective function (3.38) minimizes the total travel time of tractors, aligning with the optimization objective in (3.1).

Constraints (3.39)-(3.41) ensure that each request node in N_K^* is visited exactly once, and the lth pickup and delivery nodes p_l^m and d_l^m are visited by the same tractor. Constraint (3.42) is consistent with constraint (3.3). Consistency of the time and capacity are ensured by constraints (3.43) and (3.44). Constraints (3.45) and (3.46) restrict the time window for picking up and delivering trailers. Constraints (3.47) and (3.48) require that the transportation sequence of trailer m adheres to the order $\{p_1^m, d_1^m, p_2^m, d_2^m, \ldots, p_{n_m}^m, d_{n_m}^m\}$, ensuring that pickups and deliveries for trailer m occur in the prescribed manner. Constraints (3.49)-(3.51) are consistent with constraints (3.17), (3.25), and (3.27) in the integrated scheduling model.

3.4 METHODOLOGY

Although the complexity of the scheduling model has been reduced by dividing it into two stages, it is still difficult to solve efficiently by exact optimal solution algorithms, especially when the number of tractors and trailers is large. Besides, in practice, the solution algorithms should be efficient enough to dynamically cope with the uncertainties of flight departures and arrivals. Therefore, this part develops an efficient hybrid-intelligence based solution algorithm that integrates the diversity of ALNS, Simulated Annealing (SA), and K-means clustering [93]. The overall framework of the algorithm is shown in Fig. 3.3. It starts by obtaining the information of the flight, tractor, and trailer within a planning horizon, usually a couple of hours. Then, the routes of trailers and tractors will be generated sequentially. After that, the final results will be decoded by integrating the routes of both trailers and tractors.

Specifically, considering the advantages of reducing the chance of being trapped in local optima [94], ALNS [95] is adopted as the foundation of the solution algorithm. Meanwhile, to further improve the efficiency and performance of the solution algorithm, several key methods are developed. Firstly, effective destruction and repair operators were designed based on the specific features of the tractor and trailer scheduling problems, which could further enhance the search capability. Secondly, since the generation of initial solution for ALNS is vital for the overall performance. However, existing methods, e.g., greedy, CW-saving, often take a long time to produce a relatively good initial solution. Thus, an efficient K-means clustering-based initial solution generation algorithm is developed to speed up convergence and save search time. Thirdly, it is found that checking whether the complex constraints in the second stage are satisfied is also time-consuming. Thus, a topological sort-based solution evaluation algorithm is proposed to accelerate the process of ALNS. In the following, the detailed design of the hybrid-intelligence based scheduling algorithm will be further discussed.

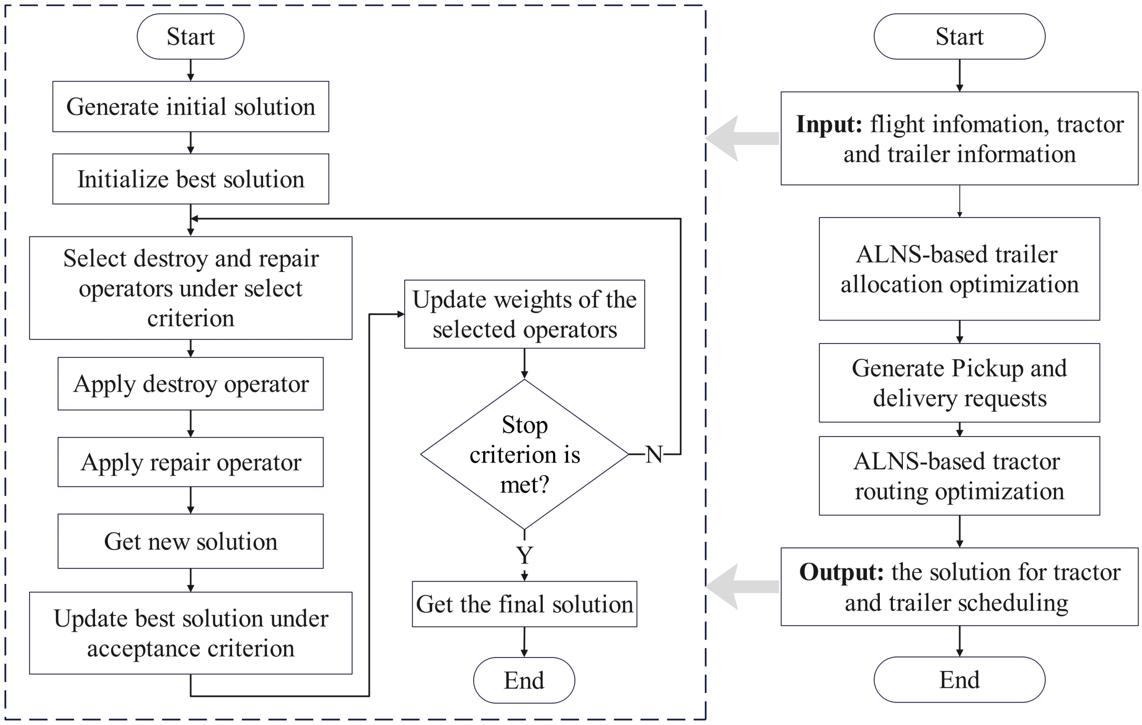


Figure 3.3: The framework of the hybrid intelligence-based two-stage scheduling algorithm.

3.4.1 ALNS-based trailer allocation algorithm

The ALNS-based trailer allocation algorithm is shown in Fig. 3.3, and the detailed processes are introduced below.

Initial solution generation. The natural number encoding scheme is adopted for trailer route encoding, s_M is denoted by 0, and each node in N_M responds to a natural number. We adopt the regret insertion heuristic [96] to obtain an initial solution:

- Step 1: Sort the nodes in N_M according to the arrival or departure time of flights.
- Step 2: Try to add node i to the end of each trailer's route and calculate the increased cost of each route while checking the solution feasibility for each insertion. Recording all feasible insertions and their increased costs.
- Step 3: If multiple feasible insertions exist, insert node i into the incumbent solution at the location with the second lowest increased cost. If there is only one feasible

location, add node i there.

- Step 4: Delete n_0 nodes with the maximum cost while there are still nodes not inserted, but no feasible insertion. The cost of demand j between i and k is calculated by: $c(i) = c(i, j) + c(j, k) - c(i, k)$. n_0 is a random integer with a range 0 to $\mu_0 * |N_M|$.

- Step 5: Stop and return the initial solution if all nodes are inserted; otherwise, return to *Step 1*.

Destroy and repair operators design. Five destroy operators are designed for the trailer allocation problem:

- Trailer-Random Removal (TA-RR): It randomly removes n_r nodes from the current solution, which could help diversify the search space.

- Trailer-Maximum Travel Time Removal (TA-MTTR): It removes n_r nodes with the maximum travel time from the current solution. The travel time of node i is calculated by $tt(j) = t(i, j)$, where node j is the next node of node i in the corresponding trailer route.

- Trailer-Maximum Service Time Removal (TA-MSTR): It removes n_r baggage unloading nodes with the maximum service end time BL_{f_E} from the current solution.

- Trailer-Maximum Cost Removal (TA-MCR): It removes n_r nodes with the maximum saved objective value from the current solution. It is evaluated by the objective value $z_M(i)$ after removing node i.

- Trailer-Maximum Cost Route Removal (TA-MCRR): It removes the trailer route with the maximum objective value from the current solution.

Note that the number of demand nodes to be removed n_r is counted by: $n_r = \mu_0 * |N_M| + \mu_1 * n_{iter}$, where n_{iter} is the current number of iterations, μ_0 and μ_1 are the parameters.

The proposed three repair operators are also listed below:

- Trailer-Greedy Insertion (TA-GI): It inserts demand i to the place with the minimum increased objective value.

- Trailer-Regret Insertion (TA-RGI): It inserts demand i to the place with the second minimum increased objective value.

- Trailer-Random Insertion (TA-RDI): It inserts demand i to a random feasible place to diversify the search space.

Acceptance and selection criterion. The SA criterion is adopted here for accepting solutions, which uses an updated temperature to determine the possibility of accepting the incumbent solution. SA requires three parameters: the initial temperature T_{start}, the final temperature θT_{start}, and the temperature updating step γ. The current temperature is updated by $T = \max\{T - \gamma, \theta T_{start}\}$. The operator selection criterion used here is the roulette wheel selection scheme, which iteratively updates operator scores according to the performance of operators.

3.4.2 ALNS and K-means clustering-based tractor routing algorithm

The tractor routing algorithm also follows the procedure shown in Fig. 3.3. It integrates ALNS with the K-means clustering-based initial solution generation method and topological sort-based solution evaluation method.

Initial solution generation. The framework of the proposed K-means clustering-based initial solution generation algorithm is shown in Algorithm 1. The basic idea of this algorithm is to assign the pickup and delivery requests with close distance and similar time

Algorithm 1 K-means Clustering-based Initial Solution Generation Framework

Input: Routes of all trailers $\{R^m\}$, service time lists on routes of all trailers $\{BL^m\}$, tractor set K, used trailer set M^*

Output: Initial routes of tractors $\{R^k\}$

Similarity matrix $M^s \leftarrow None$
for $R^a, R^b \in \{R^m\}, BL^a, BL^b \in \{BL^m\}, a \neq b$ **do**
    $M^s_{ab} \leftarrow \sum\limits_{u \in R_a} \underset{|BL_u - BL_v|}{\arg\min}\ tt^{'}(u, v)$
end for
$n_k \leftarrow \max\{1, \beta * \min\{|K|, |M^*|\}\}$
Clusters of trailer routes $C \leftarrow$K-means based Clustering(M^s, n_k)
Route of each tractor $R^k \leftarrow None$
for cluster $c \in C$ **do**
    for $R^m \in c$ **do**
        Generating pickup and delivery request sequence RS^m from R^m
        Randomly choose an empty tractor route R^k
        Last inserted request $r^m_l \leftarrow None$
        The feasible insertion set of the last inserted request $U^*_l \leftarrow None$
        Flag $\leftarrow 0$
        for request $r^m \in RS^m$ **do**
            while Flag = 0 **do**
                Set U^* as the set of all feasible insertions of r_m
                if $U^* \neq \emptyset$ **then**
                    Set u^* as the insertion in U^* with the minimum cost
                    Insert r^m to R^k at the location of u^*
                    $U^* \leftarrow U^* - \{u^*\}$
                    $r^m_l \leftarrow r^m, U^*_l \leftarrow U^*$
                    Flag $\leftarrow 1$
                else if $U^*_l \neq \emptyset$ **then**
                    Remove $R^k \leftarrow R^k - \{r^m_l\}$
                    Set u^*_las the insertion in U^*_l with the minimum cost
                    Insert r^m_l to R^k at u^*_l
                    $U^*_l \leftarrow U^*_l - \{u^*_l\}$
                else
                    Insert r^m to any feasible location on the route of other tractors
                end if
            end while
        end for
    end for
end for
return $\{R^k\}$

windows to the same tractor can better take advantage of multi-trailer capacity. Therefore, we developed a K-means-based algorithm to cluster the trailer routes according to the similarity of distance and service time, and then allocate the requests on the trailer routes belonging to the same cluster to the same tractor. The metric defined to measure the similarity between two trailer routes R_a and R_b is defined by $M^s_{ab} = \sum_{u \in R_a} \arg\min_{|BL_u - BL_v|} tt^{'}(u, v)$, where R_a is assumed to be the shorter route, BL_u and BL_v are the service time of node u and v, respectively, and $tt^{'}(u, v)$ is the distance from node u to v.

To obtain clusters of trailer routes, the similarity matrix M_s is fed into the K-means clustering algorithm to categorize similarity into n_k levels, where n_k is an adaptable parameter that varies in accordance with the problem's scale. Subsequently, routes that share the highest similarity ranking are aggregated into the same cluster. Moreover, to enhance tractor utilization, scenarios where a cluster only contains a single trailer route or where the quantity of clusters exceeds the number of tractors are precluded by amalgamating these clusters into larger ones. Finally, the requests belonging to the same cluster are inserted preferentially into the feasible location with the lowest cost of the same tractor route, as described in Algorithm 1.

Destroy and repair operators design. Three destroy operators are designed as follows:

- Tractor-Random Removal (TR-RR): It randomly removes $n^{'}_r$ pairs of pickup and delivery requests from the current solution, which helps to diversify the search space.
- Tractor-Maximum Travel Time Removal (TR-MTTR): It removes $n^{'}_r$ pairs of pickup and delivery requests with the maximum tractor travel time from the current solution. The travel time of request pair (u, v) is counted by: $tt^{'}(u, v) = t(u_0, u) + t(v_0, v)$, while u_0 and v_0 are the former request of u and v, respectively.
- Tractor-Maximum Service Time Removal (TR-MSTR): It removes $n^{'}_r$ pairs of pickup and delivery requests with the maximum service end time from the current solution.

Here, $n_r^{'}$ is counted by: $n_r^{'} = \mu_0 * |N_M| + \mu_1 * n_{iter}^{'}$, where $n_{iter}^{'}$ is the iteration number. The repair operators are listed as follows.

- Tractor-Greedy Insertion (TR-GI): It inserts the pair of pickup and delivery request (u, v) to the place with the minimum increased objective value.

- Tractor-Regret Insertion (TR-RI): It inserts the pair of pickup and delivery request (u, v) to the place with the second minimum increased objective value.

- Tractor-Local Insertion (TR-LI): It inserts the pair of pickup and delivery request (u, v) to the place with the minimum increased objective value in one tractor route.

Acceptance and selection criterion. The SA criterion and roulette wheel selection scheme are also adopted here as the solution acceptance criterion and the operator selection criterion, respectively.

Topological sort-based solution evaluation. Constraints (3.47) and (3.48) require that the order of tractors transporting trailers must follow the determined order of trailers serving flights, as well as the generated pickup and delivery sequences. However, enumerating whether these constraints are satisfied by checking B_u is time-consuming. Thus, a topological sort-based solution evaluation method is introduced here to enhance the efficiency of the scheduling algorithm. We first construct a network G_K^+ containing both the routes of tractors and trailers to trace the sequence of tractors transporting trailers. If the required sequences are broken, there must be at least one directed cycle existing in the G_K^+. Then, topological sort is applied to check the existence of directed cycles. The key steps of this algorithm are listed here:

- Constructing the directed graph $G_K^+ = (N_K^*, A_K^+)$, where A_K^+ is composed of the routes of tractors and specific pickup and delivery sequences from trailers.

- Using the topological ordering to check whether G_K^+ contains any directed cycle. If the directed cycle exists, then this solution is not feasible; otherwise, it is feasible.

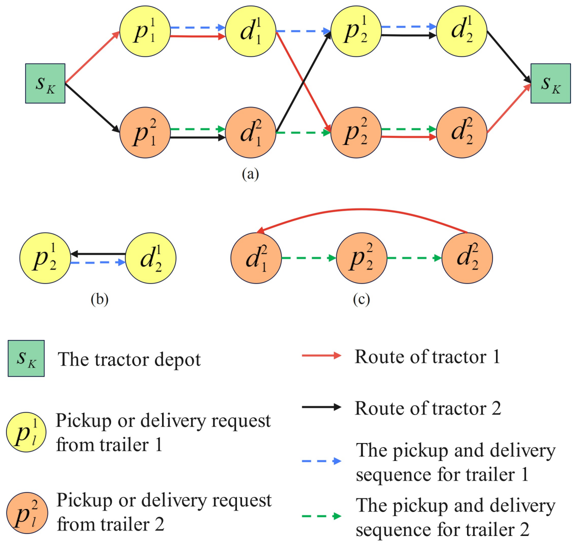


Figure 3.4: The network that includes the tractor routes and order Constraints from trailers.

A simple example is given in Fig. 3.4(a) about the feasible solutions for tractor routing, in which the pickup and delivery requests from 2 trailers are served by 2 tractors. The hard arrows represent the routes of 2 tractors, while the dashed arrows are the specific pickup and delivery sequences, also representing constraints (3.47) and (3.48). If any directed cycle exists in G_K^+, it must be composed of the hard and dashed arrows, as constraints (3.50) have ensured that there is no directed cycle in tractor routes (hard arrows). Therefore, the directed cycle in G_K^+ only exists when the solution breaks the determined sequences, as shown in Fig. 3.4 (b) and (c).

3.5 CASE STUDY

This section presents computational experiments to validate the proposed models and solution algorithm.

3.5.1 Test instances

We take HKIA as an example to conduct the experiments, which one of the busiest international airports around the world that connects about 220 destinations around the world and is operated by about 120 airlines, with a passenger throughput of 71.5 million passengers [97] and a cargo throughput of 4.8 million tons in 2019[1]. We constructed the road network of airport ground vehicles in HKIA, a total of 71 aircraft stands are included here, mainly located around Terminal 1. The detailed road network construction process will be introduced in Chapter 6. Without loss of generality, we consider each flight needs 2 or 3 trailers [39]. The allowed duration of baggage transport is set as $t_F = 60$ minutes, the average baggage loading or unloading time $t_M = 10$ minutes, the capacity of the tractor $Q = 6$, and the speed of the tractor is to be 20 km/h.

With the above analysis, three types of instances are generated to simulate different scenarios:

- Random instances: It is designed to simulate the off period. The gates of flights are randomly selected, and flight arrival or departure times are randomly generated.
- Cluster instances: It is designed to simulate the busy period. In each instance, departing and arriving flights have a higher chance of being allocated at adjacent stands with very close time slots.
- Practical instances: It is designed to simulate real-world scenarios. Each instance contains real flight information collected from HKIA, and the numbers of tractors and trailers are set to be 20 and 30, respectively.

3.5.2 Performance evaluation

To evaluate the performance of the proposed two-stage scheduling model and hybrid-intelligence-based solution algorithm, we compared their results with those obtained by Gurobi (version 10.0.03, 64 bits) with the integrated scheduling model on generated random instances, since Gurobi can only solve the small-scale instances of this complex problem.

[1] https://www.hongkongairport.com/tc/the-airport/hkia-at-a-glance/fact-figures.page

Table 3.2: Comparison of Gurobi and the proposed algorithm based on the integrated and two-stage scheduling on small instances.

Instances	Integrated Model+Gurobi						Two-stage Model+Gurobi						Integrated Model+Proposed				Two-stage Model+Proposed			
	Feas.	LB	BestObj	#K	#M	T_run (s)	Feas.	LB	BestObj	#K	#M	**T_run (s)**	Cost	#K	#M	T_run (s)	Cost	#K	#M	T_run (s)
1	Opt.	4.99	4.99	1	3	205.50	Opt.	4.99	4.99	1	3	7.87	**4.99**	1	3	6.29	**4.99**	1	3	**4.70**
2	Opt.	7.23	7.23	1	2	2.90	Opt.	7.23	7.23	1	2	0.24	**7.23**	1	2	1.80	**7.23**	1	2	**1.22**
3	Opt.	12.05	12.05	1	2	3.70	Opt.	12.05	12.05	1	2	0.41	**12.05**	1	2	1.77	**12.05**	1	2	**1.20**
4	Opt.	5.09	5.09	1	2	2.70	Opt.	5.09	5.09	1	2	0.14	**5.09**	1	2	1.58	**5.09**	1	2	**1.20**
5	Opt.	8.04	8.04	1	3	44.30	Opt.	8.04	8.04	1	3	8.63	**8.04**	1	3	6.36	**8.04**	1	3	**4.80**
6	Opt.	5.32	5.32	1	3	52.50	Opt.	5.32	5.32	1	3	6.11	**5.32**	1	3	5.92	**5.32**	1	3	**5.47**
7	Opt.	5.94	5.94	1	2	3.50	Opt.	5.94	5.94	1	2	0.19	**5.94**	1	2	1.85	**5.94**	1	2	**1.86**
8	Opt.	5.24	5.24	1	2	2.90	Opt.	5.24	5.24	1	2	0.30	**5.24**	1	2	3.20	**5.24**	1	2	**2.40**
9	Opt.	4.19	4.19	1	2	2.70	Opt.	4.19	4.19	1	2	0.28	**4.19**	1	2	1.50	**4.19**	1	2	**1.53**
10	Opt.	21.63	21.63	1	2	849.00	Opt.	21.63	21.63	1	2	6.95	**21.63**	1	3	8.39	**21.63**	1	2	**2.99**
11	Opt.	18.23	18.23	1	2	2313.90	Opt.	18.23	18.23	1	2	13.28	**18.23**	1	3	9.12	**18.23**	1	2	**3.61**
12	Opt.	10.85	10.85	1	2	63.50	Opt.	10.85	10.85	1	2	3.77	**10.85**	1	2	8.65	**10.85**	1	2	**2.87**
13	Opt.	26.56	26.56	1	2	6066.50	Opt.	26.56	26.56	1	2	11.54	28.86	1	2	5.26	28.86	1	2	**2.61**
14	Opt.	15.65	15.65	1	2	640.90	Opt.	15.65	15.65	1	2	11.94	17.96	1	2	4.96	**15.65**	1	2	**4.35**
15	Feas.	7.43	33.88	1	2	7200.00	Opt.	33.88	33.88	1	2	473.27	39.99	1	2	6.38	36.18	1	2	**4.90**
16	Feas.	9.17	14.13	1	2	7200.00	Opt.	14.13	14.13	1	2	53.70	14.32	1	2	4.80	14.32	1	2	**3.15**
17	Opt.	13.05	13.05	1	2	165.10	Opt.	13.05	13.05	1	2	10.72	**13.05**	1	2	15.42	**13.05**	1	2	**4.17**
18	Opt.	9.83	9.83	1	2	51.70	Opt.	9.83	9.83	1	2	3.47	**9.83**	1	2	13.33	**9.83**	1	2	**3.80**
19	Opt.	19.90	19.90	1	2	470.20	Opt.	19.90	19.90	1	2	8.81	**19.90**	1	2	16.95	26.13	1	2	**4.82**
20	Opt.	8.83	8.83	1	2	66.20	Opt.	8.83	8.83	1	2	1.55	**8.83**	1	2	14.56	**8.83**	1	2	**3.06**
21	Opt.	18.22	18.22	1	2	52.80	Opt.	18.22	18.22	1	2	4.67	**18.22**	1	2	8.00	**18.22**	1	2	**4.20**
22	Opt.	6.78	6.78	1	2	27.90	Opt.	6.78	6.78	1	2	1.58	**6.78**	1	2	7.88	**6.78**	1	2	**3.14**
23	Opt.	6.11	6.11	1	2	1499.60	Opt.	6.11	6.11	1	2	23.60	**6.11**	1	2	5.77	**6.11**	1	2	**4.57**
24	Feas.	0.13	17.02	1	3	14400.00	Opt.	17.02	17.02	1	3	2448.08	17.02	1	3	50.00	17.02	1	3	**11.75**
25	Feas.	0.07	31.20	1	3	14400.00	Feas.	24.63	31.20	1	3	14400.00	31.20	1	3	27.73	33.51	1	3	**23.48**
26	-	0.00	-	-	-	14400.00	Feas.	5.07	16.96	1	3	14400.00	27.91	1	3	56.12	16.96	1	3	**15.75**
27	-	0.00	-	-	-	14400.00	-	0.00	-	-	-	14400.00	**30.73**	1	3	66.02	**30.73**	1	3	**11.62**
28	Feas.	5.33	18.87	1	3	14400.00	Opt.	18.87	18.87	1	3	5104.68	29.87	1	3	137.15	32.23	1	3	**31.24**
29	Feas.	0.00	20.72	1	3	14400.00	Opt.	20.80	20.80	1	3	10978.05	26.12	1	3	21.37	26.12	1	3	**11.20**
30	-	2.82	-	-	-	14400.00	Feas.	13.78	24.12	1	3	14400.00	24.48	1	3	62.59	24.48	1	3	**8.87**
31	Feas.	8.29	14.52	1	3	14400.00	Opt.	15.47	15.47	1	3	3652.36	15.47	1	3	27.11	26.20	1	3	**11.63**
32	Feas.	14.22	22.23	1	3	14400.00	Opt.	22.32	22.32	1	3	3857.11	38.20	1	3	14.26	39.99	1	3	**19.47**
33	Feas.	2.07	6.24	1	3	14400.00	Opt.	6.24	6.24	1	3	339.33	7.99	1	3	35.31	7.99	1	3	**22.61**
34	Feas.	0.02	15.78	1	3	14400.00	Feas.	13.54	15.78	1	3	14400.00	25.45	1	3	110.81	25.45	1	3	**15.79**
35	-	0	-	-	-	14400.00	Opt.	11.07	11.07	1	3	342.95	**11.07**	1	3	119.28	**11.07**	1	3	**28.76**
Avg.		9.63	**13.82**	**1.00**	**2.35**	5708.23		12.87	14.17	1.00	2.41	2839.30	16.52	1.00	2.49	25.36	16.70	1.00	2.43	**8.25**

* Opt. : solving the instance to optimality within time limits;
* Feas.: only obtaining a feasible solution within time limits.

The results are reported in Table 3.2, where column "LB" is the lower bound obtained by the Gurobi solver, "Cost" is the total cost, "#K" and "#M" are the number of used tractors and trailers, and "T_run" is the running time (in seconds). On the one hand, the results show that compared with the integrated model, the two-stage scheduling model can obtain more feasible solutions in a much shorter running time while having little impact on the solution quality. While both using Gurobi and changing the model to the two-stage scheduling model, infeasible instances are reduced, and the average running time is decreased by 2265.92 seconds, which proves the effectiveness of the proposed two-stage model. On the other hand, our proposed scheduling algorithm can significantly enhance solving speed with the premise of approaching the optimal solution in most instances, both on the integrated and two-stage scheduling models. Thus, integrating the two-stage scheduling model and the proposed scheduling algorithm has a significant advantage on the balance of solution quality and running time, it can be seen that all instances were solved within 32 seconds.

3.5.3 Algorithm performance analysis

In this part, experiments are conducted to evaluate the performance of different parts of the proposed hybrid-intelligence based scheduling algorithm.

Parameter setting. The parameter setting for the hybrid-intelligence based solution algorithm is shown in Table 3.3.

Table 3.3: Parameters for ALNS.

Parameter	Value
n_{iter}	300
$n_{iter}^{'}$	200
T_{start}	1.0
θ	0.025
γ	0.95
μ_0	0.12
μ_1	0.005
β	0.7

Initial solution generation algorithm comparison. As shown in Table 3.4, the proposed K-means clustering-based initial solution generation algorithm for the tractor routing problem is compared with two other existing popular algorithms: the Cheapest Feasible Insertion (CFI) algorithm [27] and the Clark Wright (CW) Algorithm [98]. The experiment is implemented on random instances, and the number of flights ranges from 4 to 16. The average result of 20 cases with the same number of flights is regarded as the final result. Besides, the running time "T_run" is restricted to 1800 s. Table 3.4 shows that the running time of the three algorithms all increases with the increase of the instance scale. The CFI algorithm follows the greedy principle to insert the demand node into the location with the cheapest cost. Although the CFI algorithm can generate the solution with the cheapest cost, its running time is much longer than CW and our algorithm. The CFI algorithm cannot obtain a solution within the required time limit when $|F| > 10$. Compared with CFI, the running time of CW is reduced a lot on small instances, but its running time increases rapidly as the scale of the instances increases. Although the cost obtained by our proposed algorithm has a certain increase compared with CFI, its running time is much less than that of CFI and CW algorithms.

Table 3.4: The comparison for initial solution generation algorithms.

$\lvert F\rvert$	CFI		CW		Proposed	
	T_run (s)	Cost	T_run (s)	Cost	T_run (s)	Cost
4	178.44	**32.13**	20.61	64.64	**5.08**	45.21
6	609.17	**43.83**	71.79	80.74	**11.19**	57.69
8	1393.88	**60.35**	212.52	113.66	**18.40**	93.36
10	1775.09	**64.90**	453.71	127.01	**45.14**	98.43
12	1800.00	-	481.90	118.65	**30.36**	138.82
14	1800.00	-	833.93	151.04	**64.85**	169.84
16	1800.00	-	1257.62	143.25	**119.79**	175.37

* -: The initial solution generation is not finished within 1800 s.

Performance of topological sort-based solution evaluation algorithm. We compared the running time of the whole scheduling algorithm with and without the topological sort-based solution evaluation algorithm. The saved time per iteration when applying our algorithm is shown in Fig. 3.5. The experiment is also implemented on random instances with different problem scales. The average result of 20 instances on each problem scale is regarded as the final result. It could be seen that our algorithm effectively accelerated the process of entire scheduling algorithm. Besides, as the scale of the problem increases, our solution evaluation algorithm can effectively save much more running time.

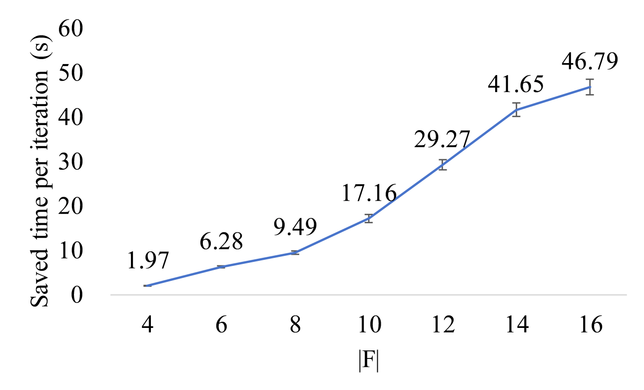
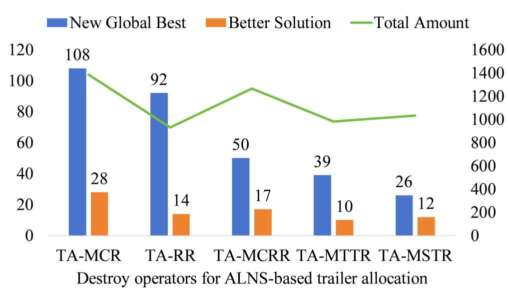


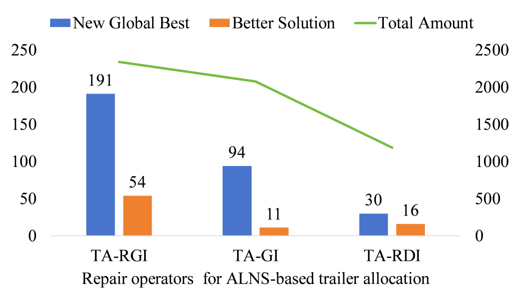
Figure 3.5: The time cost saved per iteration by the proposed algorithm on different instance scales.

Performance of destroy and repair operators. The number of times that each operator leads to a new global best or better solution, and the used times of each operator are presented in Fig. 3.6. The experiments are implemented on both random and cluster instances, in which the number of flights ranges from 4 to 16. According to the results, the operator that produced the most New Global Best and Better Solution among all destroy operators in the trailer allocation stage is TA-MCR, which is also the most frequently used. Besides, TA-RCI performs best among repair operators. As for the subsequent tractor routing stage, TR-WTTR and TR-GI perform best in removal and repair operators, respectively. The influence of different destroy and repair operators on solution search is evaluated by the following two metrics:

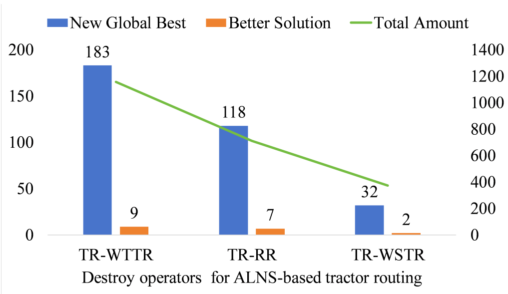
- New Global Best: The generated candidate solution is a new global best solution.
- Better Solution: The generated candidate solution is better than the incumbent solution.



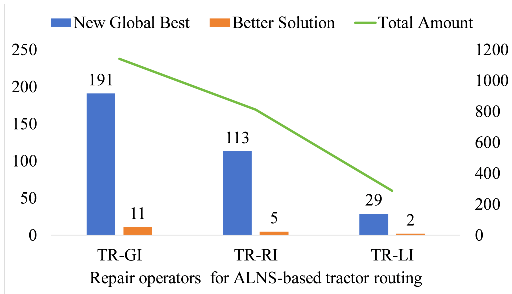
(a)



(b)



(c)



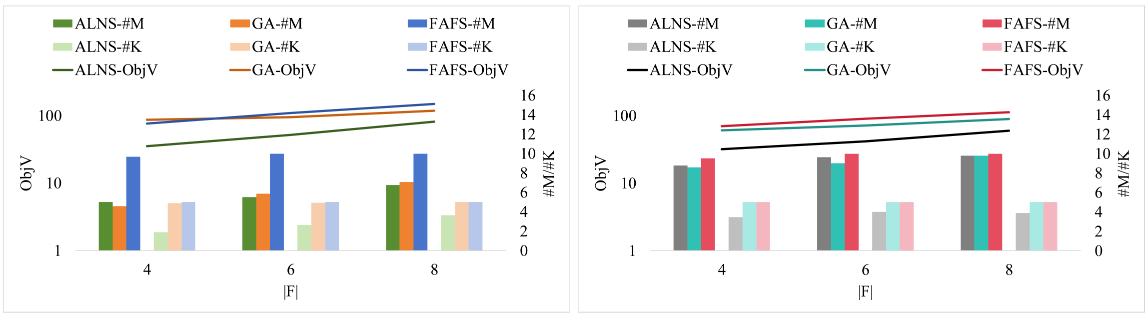
(d)

Figure 3.6: The performance of destroy and repair operators.

3.5.4 Result comparison on three types of instances

We implemented the simulation experiments based on two other heuristic algorithms for comparison on the above three types of instances. The "First Arrival First Serve" (FAFS) scheme simulates the practical scheduling strategy at HKIA. It always assigns the trailer closest to its baggage loading location. For each trailer requiring pickup, the tractor that can reach the trailer's location the fastest is assigned. Another algorithm is the Genetic Algorithm (GA)-based two-stage trailer and tractor scheduling scheme [41], and the number of iterations follows the parameter setting in our algorithm.

The scheduling results obtained by our algorithm, GA, and FAFS are shown in Fig. 3.7 and Table 3.5. For random and cluster instances, the number of flights ranges from 4 to 8, and the final result is the average of 20 cases. The results show that the solution of the proposed algorithm could save much more costs and tractor resources than FAFS and GA on all types of instances, especially in large-scale practical instances. It could be seen that sometimes it is difficult for GA to obtain feasible solutions, and the solutions produced by GA require more vehicle resources and operating costs.



(a) (b)

Figure 3.7: Result Comparison on (a). random and (b). cluster instances.

Table 3.5: Result comparison on practical instances.

Instances	$\|F\|$	Proposed			GA			FAFS		
		Cost	#K	#M	Cost	#K	#M	Cost	#K	#M
1	17	**247.51**	**10**	**22**	270.01	20	26	302.88	20	30
2	21	**283.64**	**11**	**28**	390.73	20	30	398.49	20	30
3	31	**389.42**	**18**	30	-	-	-	548.05	20	30
4	24	**243.92**	**14**	30	392.35	20	28	408.55	20	30
5	26	**373.27**	**16**	30	-	-	-	556.18	20	30
6	34	**373.51**	20	30	-	-	-	548.77	20	30
7	13	**134.23**	**10**	**25**	281.57	20	26	273.14	20	30
8	24	**218.84**	**12**	**28**	371.20	20	28	407.22	20	30

[*] -: The feasible solution is not obtained within the required iterations.

3.6 CONCLUDING MARKS

In this chapter, a practical tractor and trailer scheduling problem for airport baggage transport service is investigated. Different from previous research on drop-and-pull scheduling, this work allows each tractor to tow more than 3 trailers, which significantly increases the challenges of scheduling. In the scenario addressed, a tractor can tow multiple trailers, operating under the drop-and-pull mode that allows for the flexible detachment and reattachment of trailers as required. This operational flexibility introduces complex route dependencies between tractors and trailers, presenting a significant challenge in vehicle scheduling.

To tackle these challenges, a two-stage scheduling model is developed to reduce the complexity of the scheduling model and streamline the problem-solving process. Besides, a hybrid intelligence-based two-stage scheduling algorithm is introduced. It leverages a K-means clustering initial solution generation approach and employs a topological sort-based solution evaluation method to enhance algorithm efficiency. Experimental results show that the proposed algorithm significantly outperforms other scheduling methods in enhancing vehicle utilization and reducing operational costs.

4 ELECTRIC AUTO-DOLLY SCHEDULING FOR SUSTAINABLE AIRPORT BAGGAGE TRANSPORT

This chapter aims to propose an efficient electric auto-dolly scheduling method to reduce the operating cost of airport baggage transport service. Firstly, the practical problem of adopting a new type of electric auto-dolly in airport baggage transport service is investigated. Secondly, a simplified electric auto-dolly scheduling model is proposed, which effectively decreases the model-solving complexity. Then we define the process of solving this problem as a Markov Decision Process, including designing scenario-specific state embeddings. Thirdly, a scheduling algorithm based on RL and Transformers variant is improved, and the problem embedding is designed specifically, which can effectively represent the problem characteristics, thus improving the algorithm's convergence speed. Finally, extensive experimental case studies are conducted to verify the effectiveness of the proposed method and provide benchmarks for future works.

4.1 PROBLEM DESCRIPTION

Considering the scenario where a fleet of electric auto-dollies is adopted to implement the airport baggage transport tasks. Generally, there are two main transport tasks. One is for departing flights, the baggage needs to be collected at the BHA and then delivered to the aircraft stand. The other is for arriving flights, in which the baggage transport direction is reversed. The process should be conducted within strict time windows to ensure the aircraft can take off on time and reduce the time passengers wait for baggage. In a specific planning period, the schedules of electric auto-dollies are decided based on the information about flights, including 1) the number of dollies required; 2) the origination and destination location of baggage for each flight, specified by the corresponding aircraft stand and flight type (departing/arriving), and specified time windows for baggage transport.

Due to the capacity limitation, a dolly can usually only load part of the baggage for one flight simultaneously. Meanwhile, the electric auto-dollies may need to recharge their batteries to maintain their operations. A dolly is allowed to recharge multiple times during its route, and recharge stations are set near the BHA for dollies' easy access. Besides, the static charging speed model and the entire charging strategy are adopted in our problem, which means that the dollies are always fully charged. With the above information, the schedules of electric auto-dollies in a given period can be made, an example is shown in the left part of Fig. 4.1.

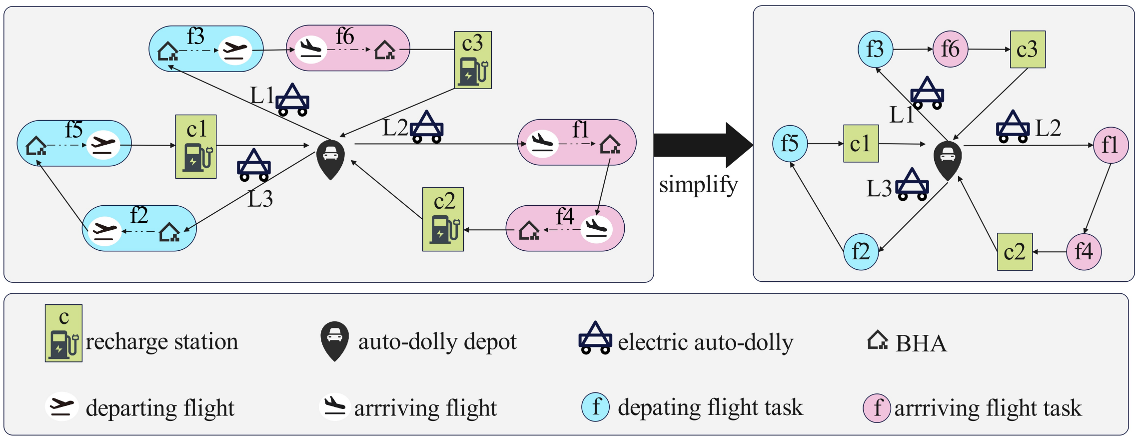


Figure 4.1: An instance of the electric auto-dolly scheduling problem with 3 dollies, 3 arriving flights, and 3 departing flights

4.2 ELECTRIC AUTO-DOLLY SCHEDULING MODEL

Considering an airport baggage transport scenario in a given time period, A fleet of electric auto-dollies L starting from the depot s_E is scheduled to serve flight tasks F. The notations adopted are given in Table 4.1.

Table 4.1: Notations for Chapter 4.

	Indicates
a, b	Index for flight task nodes, $u, v \in N_L$.
s_E	The depot of electric auto-dollies, $s_E \in N_L$.
l	Indicating electric auto-dollies, $m \in L$.
	Sets
L	Set of electric auto-dollies.
F	Set of flight tasks.
N_C	Set of dummy recharge stations.
N_L	Set of all nodes that dollies may access, $N_L = F \cup N_C \cup \{s_E\}$
	Parameters
t^{OD}_{ab}, t^{DO}_{ab}	Travel time from the baggage origination(destination) of task a to the destination(origination) of task b, note that $t^{OD}_{s_E s_E} = t^{OD}_{aa} = 0, \forall a \in N_C$
el_a	Earliest baggage loading start time for flight task a.
P_m	Maximum electricity capacity of the electric auto-dolly.
P_l	Lowest electricity threshold of the electric auto-dolly.
r	Recharge speed.
	Decision Variables
x^e_{abl}	$= 1$, if electric auto-dolly l travels from node a and node b, or 0 otherwise.
BE_a	Task start time at node a.
p_{al}	Electricity of electric auto-dolly l when leaving node a.

According to the above analysis, once a dolly is loaded with baggage at its origination, it has to first reach the baggage destination for unloading before continuing to transport other baggage. Based on this, we define a flight task $a \in F$ that includes the process of transporting baggage from the origin to the destination to simplify the modeling process. Besides, one flight requiring multiple dollies is modeled by multiple *virtual flights* requiring only one trailer. The set of *dummy recharge station* N_C is also introduced to enable the case that a dolly can be recharged multiple times. Thus, the transport network can be denoted as $G_L = (N_L, A_L)$, where $N_L = F \cup N_C \cup \{s_M\}$. The objective of electric auto-dolly scheduling is to minimize the operation cost of all dollies while ensuring service quality under strict time requirements, and the battery of all dollies is kept within the allowable range. Thus, the model can be developed as follows:

$$\text{Minimize} \sum_{a,b \in N_L} \sum_{l \in L} (t_{aa}^{OD} + t_{ab}^{DO}) x_{abl}^e \tag{4.1}$$

s.t. :

$$\sum_{a \in N_L} x_{s_E bl}^e = \sum_{a \in N_L} x_{a s_E l}^e = 1, \forall l \in L \tag{4.2}$$

$$\sum_{b \in N_L, b \neq a} x_{abl}^e = \sum_{b \in N_L, b \neq a} x_{bal}^e, \forall a \in N_L, l \in L \tag{4.3}$$

$$\sum_{l \in L} \sum_{b \in N_L} x_{abl}^e = 1, \forall a \in F \tag{4.4}$$

$$\sum_{l \in L} \sum_{b \in N_L} x_{abl}^e \leq 1, \forall a \in N_C \tag{4.5}$$

$$el_a + 2t_M + t_{aa}^{OD} \leq BE_a + 2t_M + t_{aa}^{OD} \leq el_a + t_F, \forall a \in F \tag{4.6}$$

$$BE_a + 2t_M + t_{aa}^{OD} + t_{ab}^{DO} - W(1 - x_{abl}^e) \leq BE_b, \tag{4.7}$$

$$\forall a \in N_L/N_C, b \in N_L, a \neq b, l \in L$$

$$BE_a + t_{aa}^{OD} + t_{ab}^{DO} + (P_m - p_{al})/r - W(1 - x_{abl}^e) \leq BE_b, \tag{4.8}$$

$$\forall a \in N_C, b \in N_L, a \neq b, l \in L$$

$$P_l \leq p_{al} \leq P_m, \forall a \in N_L, l \in L \tag{4.9}$$

$$p_{al} + W(1 - \sum_{b \in N_L} x_{abl}^e) \geq P_m, \forall l \in L, a \in N_C \cup \{s_E\} \tag{4.10}$$

$$p_{al} - (t_{aa}^{OD} + t_{ab}^{DO}) + W(1 - x_{abl}^e) \geq p_{bl}, \forall a, b \in N_L, a \neq b, l \in L \tag{4.11}$$

$$x_{abl}^e \in \{0, 1\}, \forall a, b \in N_L, l \in L \tag{4.12}$$

Constraints (4.2) ensure the routes of all dollies must start and end at the depot s_E. Constraints (4.3)-(4.4) ensure the flow balance and baggage transport demand are satisfied. Constraints (4.5) limit that each dummy recharge station can be accessed at most once. Constraints (4.6)-(4.8) describe the service time window and time consistency requirements. It is noted that the recharging time for a dolly with left electricity p_{al} is $(P_m - p_{al})/r$.

Constraints (4.9)-(4.11) describe the electricity range and consistency requirements. An allowed electricity threshold is set to extend the battery life of dollies. In particular, constraints (4.10) ensure that dollies are fully recharged when leaving the recharge station. Constraint (4.12) defines the value range of x^e_{abl}.

4.3 MARKOV DECISION PROCESS

The process of solving the electric auto-dolly scheduling problem by the reinforcement learning algorithm can be defined as a Markov Decision Process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$. The definitions of state $\mathcal{S}$, action $\mathcal{A}$, state transaction rule $\mathcal{T}$, and reward function $\mathcal{R}$ are introduced as follows.

State $\mathcal{S}$. The state at time t for the electric auto-dolly scheduling problem is defined to include the following contents: 1) features of the current solution δ_t, 2) the record of previous actions, and 3) the minimum objective function value so far, i.e.,

$$s_t = \{\{n(a)\}_{a\in|N_L|}, \{p_t(a)\}_{a\in|N_L|}, \mathcal{H}(t, K), f(\delta_{t^*})\} \tag{4.13}$$

where δ_t can be divided into two parts: $n(a)$ contains features of node a and $p_t(a)$ indicates the positional features of node a in δ_t (i.e., node positional feature); The most recent K previous actions at time t is restored in $\mathcal{H}(t, K)$; $f(\cdot)$ is the objective function to minimize, and $\delta^*_t = \arg\min_{\delta_{t'}\in\{\delta_0,\ldots,\delta_t\}} f(\delta_{t'})$. Specifically, $n(a)$ arranges depending on the node type a as shown in Table 4.2. Among them, the 2-dim coordinates, power restored at the recharge station, time windows, and task duration are fixed embedding decided by the specific problem instance settings. ”Power left” and ”Service time left” are two variables that change as the solution δ_t is updated, and they are used to describe the electric power left at node a and the time left after completing the flight task at node a, respectively.

Action $\mathcal{A}$. At each time step, the RL policy is required to remove one node from the current solution, and then reinsert it back to a specific place. So, action at time t is defined as $a_t = \{a, b\}$, which means the agent removes node a and then reinserts it to the location after node b.

State transaction rule $\mathcal{T}$. A deterministic transition rule is adopted here to perform a_t, which means that it always accepts the current action to generate a new solution, it is noted that infeasible actions will be masked, regardless of their impact on the objective value.

Reward $\mathcal{R}$. For each time index t, the reward mechanism assigns a value r_t calculated by subtracting the minimum of $f(\delta_{t+1})$ and $f(\delta^*)$ from $f(\delta^*)$, i.e.:

$$r_t = f(\delta^*) - \min\left[f(\delta_{t+1}), f(\delta_t^*)\right] \tag{4.14}$$

In this way, the total reward throughout the entire training or inference process equates to the cumulative reduction in cost compared to the objective value of the initial solutions.

Table 4.2: The embedding contents for different types of nodes.

Node type	Embedding contents
Dummy depot	Coordinates Power left
Dummy recharge station	Coordinates Power stored
Flight task	Coordinates of start Coordinates of end Time window Task duration Power left Service time left

4.4 METHODOLOGY

This section presents the details of our Deep Reinforcement Learning algorithm (DRL), which is an improvement on the Neural Neighborhood Search algorithm (N2S) proposed in [84]. The concrete implementation process of DRL on the electric auto-dolly scheduling problem is presented in Figure 4.2, which illustrates an example with 3 dollies, 3 recharge stations, and 6 flights. Our DRL policy follows a heuristic improvement policy to learn to improve the solution quality. First, the DRL encoder integrates the node features and positional features to embed an electric auto-dolly scheduling solution, in which the self-attention correlations of node and positional features are computed individually, respectively. Following the network in N2S, a Synthesis Attention (Synth-Att) mechanism is adopted to generate enhanced problem embeddings by synthesizing the two attention scores. After that, taking the encoder output as input, two DRL decoders are proposed to generate the node removal and reinsertion actions, respectively. Besides, the RL training algorithm also follows the settings in [84], which is the proximal policy optimization that integrates n-step return estimation and the curriculum learning strategy. The main components of DRL are introduced in the following sections.

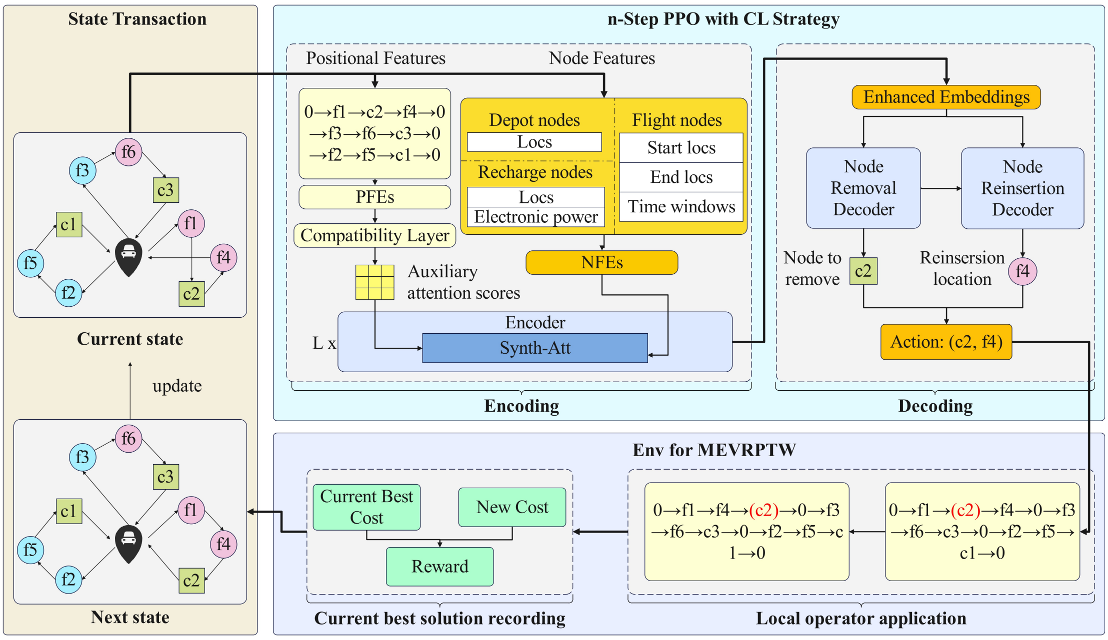


Figure 4.2: The inference framework of the RL-based electric auto-dolly scheduling algorithm.

4.4.1 Synth-Att mechanism enhanced encoder

We follow the network structures of the encoder to generate the problem embeddings for our electric auto-dolly scheduling problem.

Taking the state $s_t = \{\{n(a)\}_{a\in|N_L|}, \{p_t(a)\}_{a\in|N_L|}, \mathcal{H}(t, K)\}$ as input, the encoder is designed to learn problem embeddings and current solution representations from input $\{n(a)\}_{a\in|N_L|}$ and $\{p_t(a)\}_{a\in|N_L|}$. Specifically, the inputs are first projected into two kinds of embeddings, i.e., node feature embeddings (NFEs) $\{h_a\}_{a=0}^{|N_L|}$ and positional feature embeddings (PFEs) $\{g_a\}_{a=0}^{|N_L|}$. For each node a, NFE h_a with dimension $d_h = 128$ is set to be the linear projection of its node features $n(a)$. PFE g_a with dimension $d_g = 128$ is generated by the Cyclic Positional Encoding (CPE) scheme [84], which is an improvement on the positional encoding scheme in Transformers and can encode cyclic sequences more accurately. Besides, NFEs are set to serve as the primary set of embeddings, while PFEs function as auxiliary ones. Thus the multi-head auxiliary attention scores learned from PFEs can be obtained as follows,

$$\alpha_{a,b,m}^{\text{aux}} = \frac{1}{\sqrt{d_k}}(g_a W_m^{Q_{\text{aux}}})(g_b W_m^{K_{\text{aux}}})^T \tag{4.15}$$

where $W_m^{Q_{\text{aux}}} \in \mathbb{R}^{d_g\times d_q}, W_m^{K_{\text{aux}}} \in \mathbb{R}^{d_g\times d_k}$ are network parameters. m is the number of heads in the multi-head attention mechanism. Here m is set to 4, d_q, d_k, and d_g are all set to 32.

According to [72], directly integrating the two sets of embeddings can introduce undesired noise to self-attention. To solve this problem, a straightforward and versatile Synth-Att mechanism is introduced to the DRL, which functions by incorporating a MultiLayer Perceptron (MLP) ($2m * 2m * m$). The original self-attention scores from NFEs and the above auxiliary attention scores from PFEs are combined and fed into an element-wise MLP, allowing Synth-Att to synthesize them into comprehensive ones. The details are presented below, First, the self-attention scores $\alpha_{a,b,m}^{\text{self}}$ for NFEs is counted according to the parameters $W_m^Q \in \mathbb{R}^{d_h \times d_q}$ and $W_m^K \in \mathbb{R}^{d_h \times d_k}$ for m heads as follows,

$$\alpha_{a,b,m}^{\text{self}} = \frac{1}{\sqrt{d_k}}(h_a W_m^Q)(h_b W_m^K)^T \tag{4.16}$$

Thereafter, the attention scores α^{aux} and α^{self} are together fed into an element-wise three-layer MLP for computing the synthesized attention scores as follows,

$$\alpha_{a,b,1}^{\text{Synth}}, ..., \alpha_{a,b,m}^{\text{Synth}} = \text{MLP}(\alpha_{a,b,1}^{\text{self}}, ..., \alpha_{a,b,m}^{\text{self}}, \alpha_{a,b,1}^{\text{aux}}, ..., \alpha_{a,b,m}^{\text{aux}}) \tag{4.17}$$

Then, a Softmax layer is adopted to further normalize the obtained synthesized attention scores for each head m, and the output is $\tilde{\alpha}_{a,b,m}$. Finally, the outputs of the Synth-Att mechanism are given by Formula (4.19) with parameter $W^O \in \mathbb{R}^{md_v \times d_h (d_v = d_h/m)}$.

$$\text{head}_{a,m} = \sum_{b=1}^{N_L} \tilde{\alpha}_{a,b,m}(h_b W_m^{N_L}) \tag{4.18}$$

$$\tilde{h}_a = \text{Concat}[\text{head}_{a,1}, ..., \text{head}_{a,m}] W^O \tag{4.19}$$

To enhance the problem representation ability of the DRL, L encoders are stacked to construct the final DRL encoder $(L = 3)$. It is noted that the proposed encoder retains the structural integrity of the Transformers encoder, while replacing the original multi-head self-attention mechanism with the multi-head Synth-Att module.

4.4.2 DRL Decoder

The decoder designed in N2S is specifically designed for pickup and delivery problems. Here we make further improvements to extend it to VRP applications. The DRL decoder is designed to output two actions: node removal action and node reinsertion action. Firstly, the max-pooling layer is adopted to transfer the global representation of all embeddings into each individual one for each node a as follows,

$$\hat{h}_a = \tilde{h}_a W_h^{\text{local}} + \max[\{\tilde{h}_a\}_{a=1}^{|N_L|}] W_h^{\text{global}} \tag{4.20}$$

Then the enhanced embedding $\tilde{h}_a$ is input into the node removal decoder to choose the node to remove. After that, the node insert decoder integrates $\tilde{h}_a$ and the node removal decoder output to choose the node reinsertion location. The detailed structure of the node removal decoder and the node reinsertion decoder is introduced as follows:

Node removal decoder. Given the enhanced embeddings $\{\hat{h}_a\}_{a=1}^{|N_L|}$ and the action history $\mathcal{H}(t, K)$, the node removal decoder outputs a probability distribution over $|N_L|$ nodes to decide the removal node. Specifically, it first computes an evaluation score λ_a for each $a \in N_L$ to measure the cross attentions between node a and its neighbor nodes as follows:

$$\lambda_a = (\hat{h}_{\text{pred}(a)} W_\lambda^Q)(\hat{h}_a W_\lambda^K)^T + (\hat{h}_a W_\lambda^Q)(\hat{h}_{\text{succ}(a)} W_\lambda^K)^T \tag{4.21}$$

$$- (\hat{h}_{\text{pred}(a)} W_\lambda^Q)(\hat{h}_{\text{succ}(a)} W_\lambda^K)^T \tag{4.22}$$

where pred(a) and succ(a) are defined as the former and the successor nodes of a, respectively, and $W_\lambda^Q \in \mathbb{R}^{d_h \times d_h}, W_\lambda^K \in \mathbb{R}^{d_h \times d_h}$. $\lambda_{a,1}$ to $\lambda_{a,m}$ are obtained by using the multi-head technique, which are then fed into a three-layer MLP$_\lambda$ $(m + 4, 32, 32, 1)$ for each node as follows,

$$\tilde{\Lambda}_a = \text{MLP}_\lambda(\lambda_{a,1}, ..., \lambda_{a,m}, c(a), \mathbb{1}_{\text{last}(1)=a}, \mathbb{1}_{\text{last}(2)=a}, \mathbb{1}_{\text{last}(3)=a}) \tag{4.23}$$

where $c(a)$ represents the frequency of node a being selected as a removal node in the last K steps, and $\mathbb{1}_{\text{last}(1)=a}$ equals to 1 if node a was selected in the last step at time t; 0 otherwise. The final probability distribution for choosing the removal node is obtained by the activation of a Tanh function and the normalization by a Softmax function.

Node reinsertion decoder. Input a designated removal node b and the current state s_t, the reinsertion decoder is designed to generate the probability distributions for repositioning the removed node. Specifically, it computes insertion likelihoods for placing node a after each candidate position in the existing solution sequence. We designed two evaluation metrics $\mu^p[a,b]$ and $\mu^s[a,b]$ to measure the likelihoods of node a accepting node b as its new predecessor and successor nodes, respectively,

$$\mu^p[a,b] = (\hat{h}_a W_\mu^{Q_p})(\hat{h}_b W_\mu^{K_p})^T \tag{4.24}$$

$$\mu^s[a,b] = (\hat{h}_a W_\mu^{Q_s})(\hat{h}_b W_\mu^{K_s})^T \tag{4.25}$$

where $W_\mu^{Q_p}, W_\mu^{Q_s} \in \mathbb{R}^{d_h \times d_h}, W_\mu^{K_p}, W_\mu^{K_s} \in \mathbb{R}^{d_h \times d_h}$. Taking the scores as input, the decoder predicts the distribution of reinserting node a after node b using MLP_μ $(2m * 31x * 32 * 1)$,

$$\tilde{\mu}_b = \text{MLP}_\mu(\mu_1^p[\text{succ}(b), i], ..., \mu_m^p[\text{succ}(b), a], \mu_1^s[b, a], ..., \mu_m^s[b, a]) \tag{4.26}$$

where pred$(\cdot)$ and succ$(\cdot)$ is considered in the new solution without the node i. Similar to the node removal decoder, a Tanh function is also applied for activation, and infeasible nodes are masked before implementing the normalization by Softmax. Finally, a node b is sampled from all nodes in N_L according to the obtained distribution, which indicates that the location of reinserting the node a is right behind node b.

4.4.3 RL training algorithm

We follow the Proximal Policy Optimization [99] with n-step return estimation and a Curriculum Learning strategy (n-step PPO with CL strategy) used in [84] for the DRL policy training. PPO is a type of policy gradient reinforcement learning algorithm that strikes a balance between simplicity and performance. It is widely used due to its simplicity and strong performance across various tasks. Besides, a critic network is introduced here to enable the implementation of the actor-critic variant of PPO, which can facilitate the convergence of the PPO policy. Besides, the CL strategy is used here to avoid the agent lacking the opportunity to explore high-quality solutions during training by gradually improving the quality of initial states. Moreover, the n-step return estimation strategy extends the standard PPO by using n-step returns for updating the policy. Thus, the advantage estimates are calculated using returns over n steps rather than just single-step transitions, which can facilitate the model convergence and decrease the variance.

4.5 CASE STUDY

4.5.1 Experiment settings

The server infrastructure for this study comprised 4 RTX 4090 Ti GPU accelerators paired with an Intel Xeon Platinum 8370C multi-core CPU (2.80 GHz base frequency), ensuring parallel processing capabilities. Instances with three sizes, $20, 50$,and 100 are designed, where the nodes of each instance are uniformly located in the unit square $[0, 1]\times[0, 1]$. Besides, the initial solution δ_0 is generated randomly. Since the case of the depot and recharge stations being visited multiple times by multiple auto-dollies may exist, the solution length for each instance might be longer than $|F| + 1$. There is also a possibility that the length of multiple solutions varies on the same instance, this is because the number of sub-routes and vehicles may be different in different solutions. To solve this problem, multiple dummy depots and recharge stations are added to the end of the initial solutions. Each dummy depot can be regarded as one available dolly, and the number of dummy recharge stations indicates the allowed times of recharging for all dollies. In our experiments, we set $10, 20$, and 30 dummy depots, and $5, 10$, and 20 dummy recharge stations for three problem sizes, respectively. The time windows of flights randomly generated from a planning horizon $[0, 480]$min, the auto-dolly speed is set to 20 km/h, the auto-dolly endurance is set to $P_m = 120\ min$, the time for loading or unloading baggage is set to $t_M = 10$min, and the recharge speed is set to $r = 4$. To facilitate the model training, the planning horizon is scaled into $[0, 1]$min. Follow this, P_m is scaled to $120/480$min, t_M is set to $10/480$min.

4.5.2 Benchmark methods

Three algorithms are used in this experiment to compare and verify the proposed DRL method's performance as follows.

1. **Gurobi:** Gurobi is a state-of-the-art optimization solver designed to solve a wide range of mathematical programming problems. It functions here by providing a reference to the solution upper bound and giving a rough estimate of the problem complexity.

2. **FAFS:** First Arrive First Service algorithm (FAFS) assigns dollies to flight tasks in the order in which the service time window begins and dispatches the dolly that reaches the task node earliest. It is widely adopted with practical reference significance [74].

3. **ALNS:** Adaptive large neighborhood search integrates the advantages of multiple human-designed local operators, which works well in vehicle routing problems [39].

It is noted that all iterative algorithms share the same reference time (number of inference interactions) T_{infer} as our proposed DRL method.

4.5.3 Comparison experiments

The comparison experiments are developed on the instances of size 20, 50, and 100 with the uniform distribution. For ALNS and our DRL algorithm, we run each instance 20 times with randomly generated seeds, and the average result is taken for comparison as shown in Table 4.3. Besides, ”ObjV” indicates the objective value of the obtained solution, and ”T_run” indicates the running time of the algorithm.

It can be seen that the solver Gurobi can only get the optimal solution when the problem size is smaller than 50. Besides, ALNS can reach the near-optimal solution for small and medium-sized problems, but requires a long solving time. Compared to them, our DRL algorithm can reach a relatively satisfactory performance with a short running time, and the change in problem size has little effect on the running time.

Table 4.3: Results comparison on instances of small, medium, and large scales.

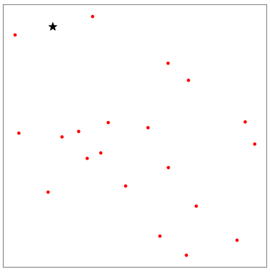
|F|	Gurobi		ALNS		FAFS		Ours	
	ObjV	T_run (s)	ObjV	T_run (s)	ObjV	T_run (s)	ObjV	T_run (s)
20	**0.008709***	0.538	0.009951	40.659	0.012760	**0.073**	0.010993	0.118
50	**0.020538***	33.071	0.023371	263.415	0.031472	0.306	0.030841	**0.219**
100	-	7200	**0.051855**	1145.631	0.061057	0.877	0.059311	**0.351**

*: The optimal solution is obtained.
-: The feasible solution is not obtained within the required time.

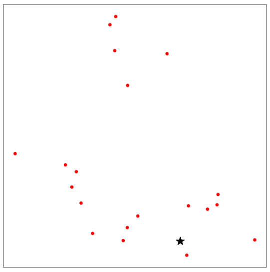
4.5.4 Generalization experiments

The generalization ability of our DRL algorithm is verified on instances with different problem sizes and distributions. For comparison, the node coordinates of test instances of size 20, 50, and 100 are generated on the unit square $[0,1]\times[0,1]$ with the normal, cluster, and center distribution, respectively. The example instances are shown in Fig. 4.3. In this experiment, the models are trained with instances in the uniform distribution with sizes of 20 and 50, respectively. The generalization performance and gaps to optima on instances of small and middle sizes with different distributions are shown in Table 4.4 and Table 4.5, respectively.

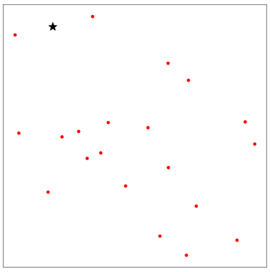
It can be seen that our DRL algorithm also shows good performance when being tested with unknown instances in the cluster and center distribution. Specifically, for the instances in three distributions with sizes of 20 and 50, the difference between the results of our algorithm and the optimal solutions obtained by Gurobi is in the range of [21%,



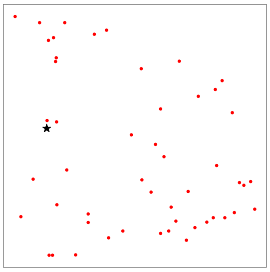
(a) size 20, uniform



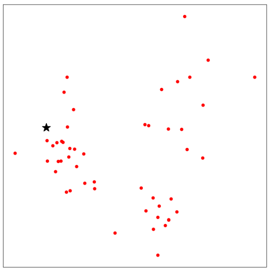
(b) size 20, cluster



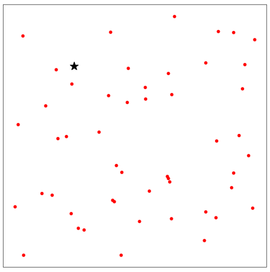
(c) size 20, center



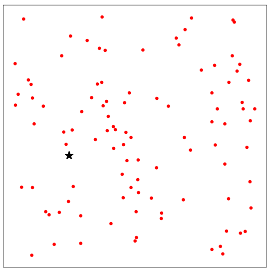
(d) size 50, uniform



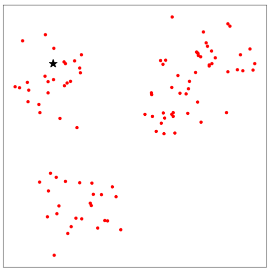
(e) size 50, cluster



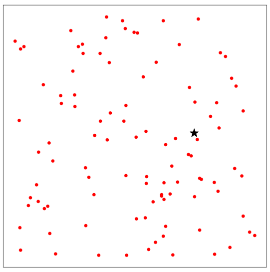
(f) size 50, center



(g) size 100, uniform



(h) size 100, cluster



(i) size 100, center

Figure 4.3: Example instances of size 20, 50, and 100 with the uniform, cluster, and center distribution.

Table 4.4: Generalization performance on instances with different instance scales and distributions.

Train		Uniform			Cluster			Center		
		20	50	100	20	50	100	20	50	100
Uniform	20	**0.010993**	0.030846	0.059313	**0.005818**	0.016527	**0.032014**	0.003668	0.009902	**0.019938**
	50	0.012604	**0.030841**	**0.059311**	0.005848	**0.016525**	0.032022	**0.003666**	**0.009900**	0.019945

Table 4.5: Gaps to Optima on instances of small and middle size with different distributions.

Train		Uniform		Cluster		Center	
		20	50	20	50	20	50
Uniform	20	**26.23%**	50.19%	**21.24%**	49.81%	24.57%	50.37%
	50	26.35%	**50.17%**	21.88%	**49.79%**	**24.52%**	**50.33%**

27%] and [49%, 51%], respectively. Besides, the models trained with instances of problem sizes 20 and 50 have little difference in generalization performance, which indicates that the reward space is effectively sampled.

4.6 CONCLUDING REMARKS

In this chapter, an electric auto-dolly scheduling problem for sustainable airport baggage transport service is investigated. In practice, one flight usually requires multiple dollies, and such divisible demands setting will greatly increase the complexity of dolly scheduling. It is even more complex as dollies are required to complete baggage transport within the required time window while recharging at the right time without compromising operational efficiency.

To tackle these challenges, a simplified electric auto-dolly scheduling model is proposed, which effectively decreases the model-solving complexity. Then, a scheduling algorithm that integrates deep reinforcement learning and the Transformers variant is developed. To effectively represent the problem characteristics and improve the algorithm convergence speed, the service time of flight tasks and the auto-dolly battery status are specifically embedded into the state space. Besides, a pair of destroy and reinsertion decoders are designed based on the model of Transformers to facilitate the solution quality improvement under the improvement heuristic policy. Finally, extensive experimental case studies are conducted to verify the effectiveness and generalization performance of the proposed method and provide benchmarks for future works.

5 EMPIRICAL ANALYSIS

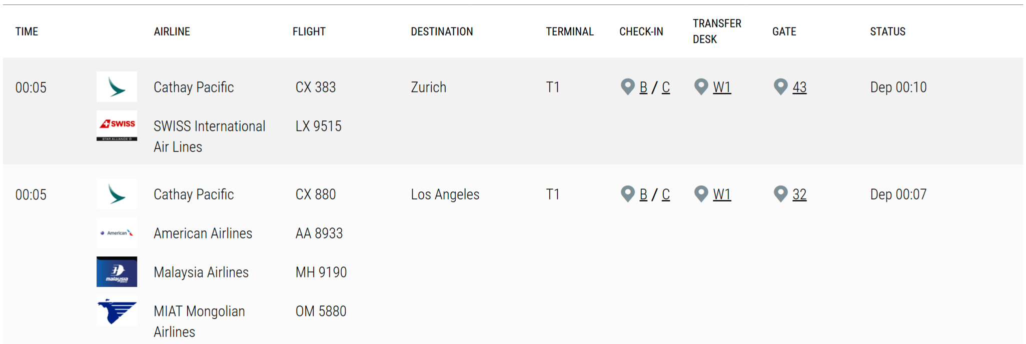
This chapter introduces the empirical analysis of baggage transport service based on the scenario of the Hong Kong International Airport. First, we built an integrated airport ground vehicle simulation platform based on the collected flight data and the ground vehicle road network obtained by SUMO of the Hong Kong International Airport. Secondly, to evaluate the operating costs and carbon emissions of baggage transport service under tractor-trailer and electric auto-dolly modes, the energy consumption models of fuel tractors and electric auto-dollies are formulated. Finally, based on the simulation platform and the above model, the practical application of baggage transport service is analyzed, including the comparison between the two operating modes in operating costs and carbon emissions, and the prediction of the sustainable development of baggage transport service in the future.

5.1 SIMULATION PLATFORM CONSTRUCTION

The construction of the simulation platform mainly includes the process of flight data collection and road network construction. The detailed implementation is described as follows.

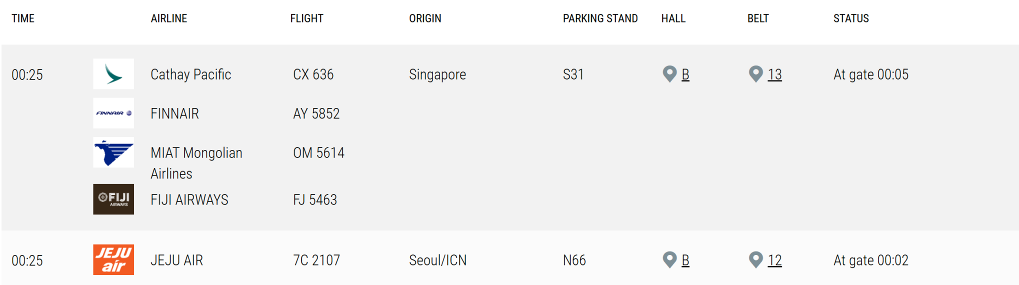
5.1.1 Flight data collection

The flight data of HKIA is obtained from the HKIA official website via Python's open source library *request* [100]. Fig. 5.1 shows some samples of flight information provided by the HKIA official website. A piece of departing flight information includes the airline, flight number, scheduled departure time, actual arrival time, boarding gate, etc. A piece of arriving flight information includes the airline, flight number, scheduled departure time, actual arrival time, aircraft parking stand, etc. We mainly collected flight data for eight months from March to May 2023, September to November 2023, and June to July 2024.



TIME	AIRLINE	FLIGHT	DESTINATION	TERMINAL	CHECK-IN	TRANSFER DESK	GATE	STATUS
00:05	Cathay Pacific	CX 383	Zurich	T1	B / C	W1	43	Dep 00:10
	SWISS International Air Lines	LX 9515						
00:05	Cathay Pacific	CX 880	Los Angeles	T1	B / C	W1	32	Dep 00:07
	American Airlines	AA 8933						
	Malaysia Airlines	MH 9190						
	MIAT Mongolian Airlines	OM 5880						

(a) Departing flights



TIME	AIRLINE	FLIGHT	ORIGIN	PARKING STAND	HALL	BELT	STATUS
00:25	Cathay Pacific	CX 636	Singapore	S31	B	13	At gate 00:05
	FINNAIR	AY 5852					
	MIAT Mongolian Airlines	OM 5614					
	FIJI AIRWAYS	FJ 5463					
00:25	JEJU AIR	7C 2107	Seoul/ICN	N66	B	12	At gate 00:02

(b) Arriving flights

Figure 5.1: Flight information samples of HKIA.

5.1.2 Road network construction

The construction of the map for the simulation platform involves three steps. Firstly, the airport ground area to be modeled is determined on the satellite map of HKIA, as shown in Fig. 5.2 (a). Here we mainly model the area around Terminal 1, including 61 aircraft stands. This is because the aircraft stands near Terminal 1 are typically *near aircraft stands* (connected to the corridor bridge), so the stand number corresponds to the gate number one by one, which can facilitate us to query the aircraft stand according to the boarding gate for departing flight as Table 5.1. Next, the road information file for airport ground service vehicles in the selected area is downloaded from the OpenStreetMap (OSM), as shown in Fig. 5.2 (b). OSM is a collaborative project to create a free, editable map of the world, which provides a wealth of geographic data. Finally, the above file is imported into the SUMO. SUMO is an open-source, highly portable, microscopic, and continuous road traffic simulation package designed to handle large road networks. We further edit the road network, marking the locations of the parking stands and BHA to generate the final road network as shown in Fig. 5.2 (c).

Furthermore, we extract the length, start node, and end node of each road in this network to generate the adjacency matrix of the road network with stand and BHA as nodes using the Dijkstra algorithm. This allows us to quickly get the distance between any two stands, or between one stand and the BHA, facilitating the operation of vehicle scheduling algorithms.

By constructing these detailed scenario maps using satellite imagery and road network data, the simulation platform can accurately simulate the HKIA scenario, enabling the simulation and evaluation of various airport service vehicle scheduling optimization.

Table 5.1: The correspondence between boarding gates and aircraft stands.

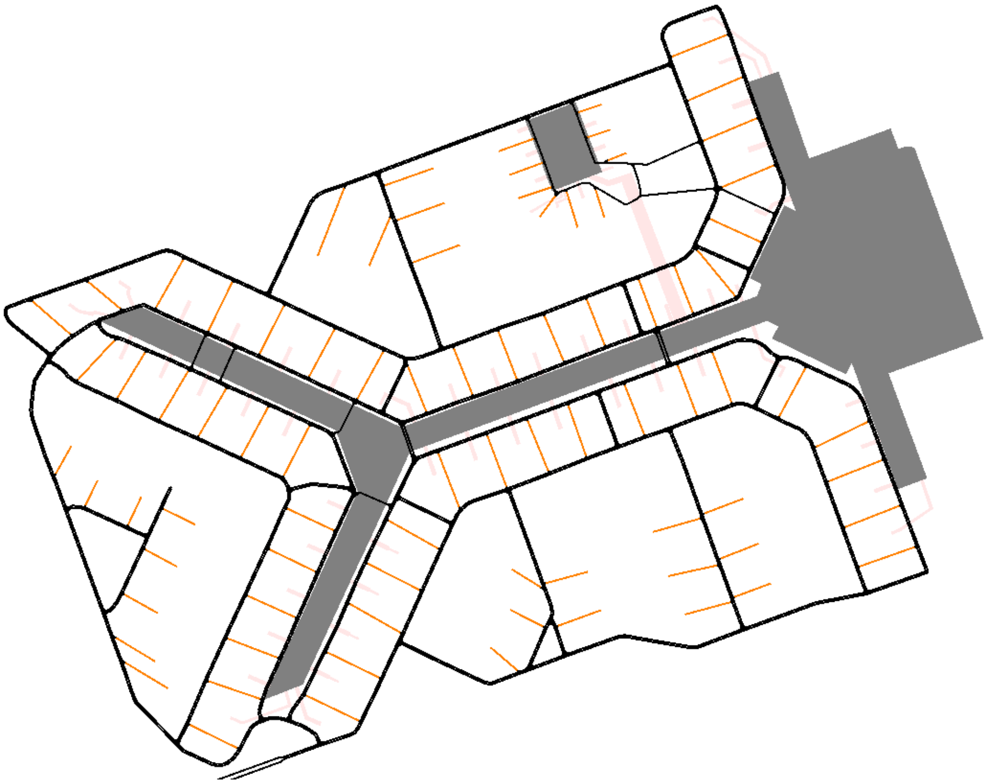
Boarding Gate	1	2	3	4	5	6	7	8	9	10	11	12
AIrcraft Stand	S1	S2	S3	S4	N5	N6	N7	N8	N9	N10	S11	N12

Boarding Gate	13	14	15	16	17	18	19	20	21	23	24	25
AIrcraft Stand	R13	R14	R15	R16	R17	R18	R19	R20	R21	S23	N24	S25

Boarding Gate	26	27	28	29	30	31	32	33	34	35	36	40
AIrcraft Stand	N26	S27	N28	S29	N30	S31	N32	S33	N34	S35	N36	W40

Boarding Gate	41	42	43	44	45	46	47	48	49	50	60	61
AIrcraft Stand	S41	W42	S43	W44	S45	W46	S47	W48	S49	W50	N60	W61

(a) Satellite map of chosen area in HKIA (b) Road network in OSM



(c) Road network in SUMO

Figure 5.2: HKIA ground map construction process.

5.2 SUSTAINABLE DEVELOPMENT ANALYSIS OF AIRPORT BAGGAGE TRANSPORT SERVICE

5.2.1 Vehicle energy consumption and carbon emission models

This section introduces the energy consumption and carbon emission models for airport baggage transport vehicles. Because trailers are non-motorized, we focus on the traditional fuel tractors and electric auto-dollies here. The notations used in this chapter are introduced in Table 5.2, and the parameter value settings follow the settings in [28] and [101]. Moreover, the emission factor of diesel oil k_f is set referring to the report of the Intergovernmental Panel on Climate Change. The carbon emissions of the power grid can be regarded as the carbon emissions of electric vehicles since electric vehicles produce little carbon emissions during driving. Thus k_e is set to $0.6\ kg/kWh$, which is the carbon emission factor of China Southern Power Grid in 2023.

Table 5.2: Notations for Chapter 5.

Parameters	Meaning	Value
ρ	Air density	$1.2041\ kg/m^3$
A_f	The frontal area of the vehicle	$3.912\ m^2$
C_D	Coefficient of Aerodynamic drag	0.6
f_{rl}	Rolling resistance constant	0.03
g	Gravity acceleration	$9.81\ m/s^2$
η	Diesel engine efficiency	0.4
η_{tf}	Travel train efficiency	0.9
η_d	Driveline efficiency	92%
η_m	Electric motor efficiency	91%
ξ	Fuel-to-air mass ratio	1
κ	Heating value of typical diesel fuel	$44kJ/g$
Ψ	Unit conversion factor	$737\ L/g$
f_c	Engine friction factor	$0.2\ kJ/r/L$
N	Engine speed	$33\ r/s$
D	Engine displacement	$5\ L$
k_f	Emission factor of diesel oil	$0.074\ kg/MJ$
k_e	Emission factor of electricity power grid	$0.6\ kg/kWh$

Mechanical energy calculation. First of all, according to basic physics, the required tractive effort for a vehicle driving is determined by three major resistances described as follows [102]:

$$F = m_k a + F_a + F_{rl} + F_g \tag{5.1}$$

where F is the tractive effort; m_k is vehicle mass; a is vehicle acceleration, and F_a, F_{rl}, and F_g are aerodynamic, rolling, and grade resistances, respectively, which can be calculated by:

$$\begin{cases} F_a = kv^2 = \frac{\rho}{2} C_D A_f v^2 \\ F_{rl} = f_{rl} m_k g \\ F_g = m_k g \sin\theta \end{cases} \tag{5.2}$$

Combining Formulas. (5.1), (5.2), then the required power p can be expressed as:

$$F = m_k a + \frac{\rho}{2} C_D A_f v^2 + f_{rl} m_k g + m_k g \sin\theta \tag{5.3}$$

The above equation can be applied to both fuel and electric vehicles. Assuming that airport ground roads are flat and tractors are traveling at a constant speed, thus $\theta = 0, a = 0\ m/s^2$. Thus, the required power p (in W) for a vehicle traveling at v to generate the above tractive force can be estimated using the following formula:

$$p = F \cdot v = (\frac{\rho}{2} C_D A_f v^2 + f_{rl} m_k g) v \tag{5.4}$$

Fuel and electricity consumption model. The fuel consumption f_{uv} (in L) of a fuel vehicle traveling from node u to v can be obtained by:

$$f_{uv} = \frac{\xi}{\kappa\Psi}(f_c \cdot N \cdot D + \frac{p}{\eta\eta_{tf}}) \cdot t_{uv} \tag{5.5}$$

Given the required mechanical power, the energy consumption g_{uv} of an electric vehicle traveling from node u to v can be defined by:

$$g_{uv} = \frac{pt_{uv}}{\eta_d\eta_m} \tag{5.6}$$

where t_{uv} (in s) is the time required for traveling from node u to v.

Carbon emission model. The mass of CO_2 emission produced by a fuel and electric vehicle traveling from node u to v can be obtained by Formula (5.7) and (5.8), respectively.

$$em^f_{uv} = k_f \cdot f_{uv} \tag{5.7}$$

$$em^e_{uv} = k_e \cdot g_{uv} \tag{5.8}$$

5.2.2 Fleet settings

To analyze the operating costs, energy consumption, and carbon emission when operating airport baggage transport service under two different modes practically, we investigated some energy consumption-related parameters of fuel tractors and electric auto-dollies, as shown in Table 5.3, referring to [28]. It is noted that the energy capacity and replenishment process of fuel tractors can be ignored because of the high range and short refueling time of fuel vehicles [28]. Besides, the unit electricity energy cost is set according to the announcement of CLP Power Limited in November 2023; the diesel oil price refers to the announcement of the Hong Kong Consumer Council in August 2024. The operating costs of the tractor-trailer mode and the electric auto-dolly mode are defined by the fuel and electricity costs, respectively, which are calculated by multiplying the unit energy price by the energy consumption.

Along with the settings in Chapter 3, the number of trailers/dollies required by each flight is randomly set to 2 or 3. The allowed duration of baggage transport is set as $t_F = 60$ minutes, the average baggage loading or unloading time $t_M = 10$ minutes, the capacity of the tractor $Q = 6$, and the speed of the tractor/dolly to be 20 km/h. The number of tractors, trailers, and dollies is set to 30, 50, and 50, respectively.

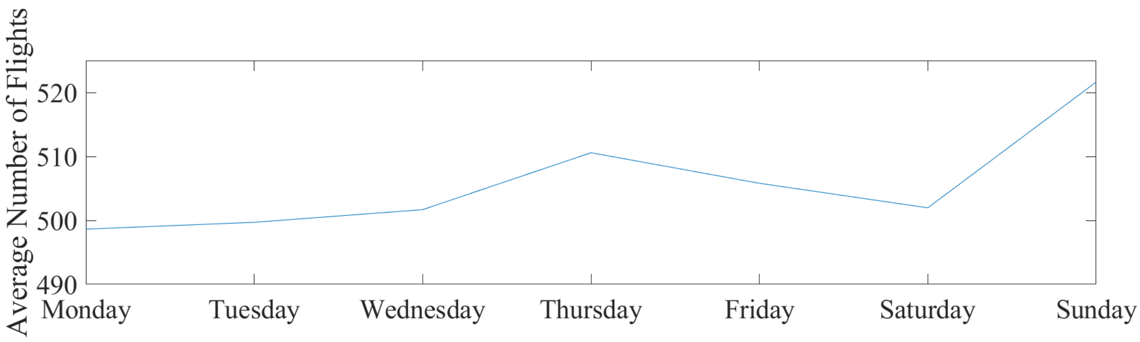
Table 5.3: Parameters of fuel tractors and electric auto-dollies.

Parameters	Fuel tractor	Electric auto-dolly
Energy capacity	-	100 kWh
Available energy capacity	-	100 kWh*70%
Vehicle mass	4000kg	1200kg
Energy dissipation rate	-	4
Average Speed	20 km/h	20 km/h
Unit Energy Price	22.45 HKD/L	1.429 HKD/kWh
Energy replenishment rate	-	90 kW

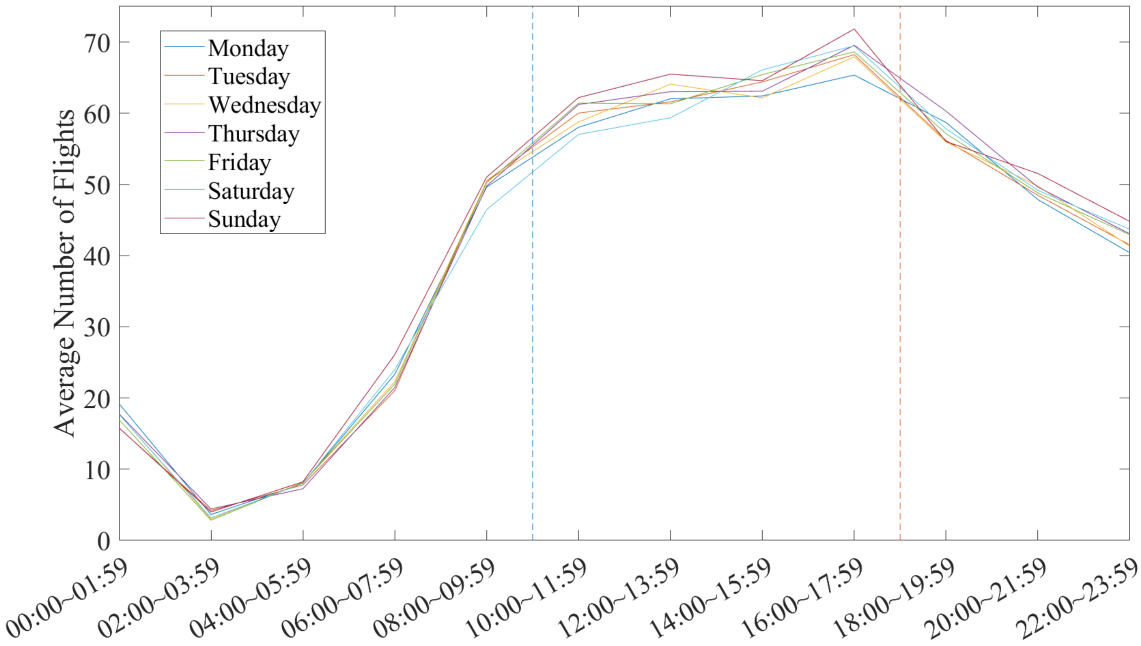
5.2.3 Comparison of two airport baggage transport service modes

At present, the mainstream airport baggage transport service is generally operated by tractors and trailers, and with the electrification and intelligence of airports, the future baggage transport service may be operated by electric auto-dollies. To analyze the differences between the two modes in terms of operating costs, sustainable development, etc., the baggage transport vehicle scheduling experiments are conducted on practical cases in various periods.

We first counted the average number of flights per day in a week and the average number of flights per two hours in a day, the statistical results are shown in Fig. 5.3. It can be seen that the days in a week with the fewest and most flights are Monday and Sunday, respectively. This is also in line with practical experience, as passengers tend to travel more on weekends. Besides, there are significant peaks and lows of flight numbers at different times in a day, which can be defined as 10:00-20:00 and 0:00-10:00, respectively. Based on the above statistical results, we chose all Mondays and Sundays in March 2023 for the experiment. The reason for choosing March 2023 is that there are no additional public holidays in 2023 March in Hong Kong, so the statistical regularity of flight data will not be affected by holidays.



(a) The average number of flights per day for a week



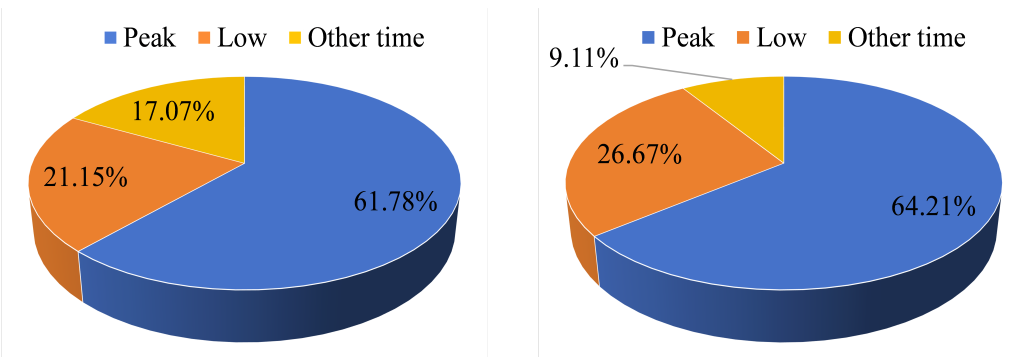
(b) The average number of flights per two hours in a day

Figure 5.3: The average number of flight statistics.

The performance of the tractor-trailer mode and the electric auto-dolly mode is compared here. Table 5.4 displays the comparison results on the cases of Monday and Sunday, while Table 5.5 shows the comparison results on the cases of peak and low periods. The experimental results are evaluated based on the travel time, number, operating costs, and carbon emission mass of all used vehicles. Firstly, the baggage transport service under the electric auto-dolly operation mode consumes less operating costs and generates less carbon emissions in the case of different time periods. The 30 tractors consume 6300-7100 HKD per day and emit 700-780 kg of CO_2; the average daily cost of 50 dollies is 210-220 HKD and the carbon emissions are 80-100 kg. Therefore, the daily operating costs and carbon emissions of the electric auto-dolly mode are about 3% and 12% of the tractor-trailer mode, respectively. The main energy consumption during the day is obviously during the peak hours. Fig. 5.4 shows that under both operation modes, the energy consumption during the 10-hour peak period reaches are at least 60%.

Table 5.4: The comparison of two modes on the cases of Monday and Sunday.

		Tractor-Trailer Mode					Electric Auto-dolly Mode			
		Travel Time (h)	Trailer Number	Tractor Number	Cost (HKD)	CO_2 (kg)	Travel Time (h)	Dolly Number	Cost (HKD)	CO_2 (kg)
Mon.	23/3/6	45.78	35	17	6406	702	62.47	50	**211**	**89**
	23/3/13	45.94	36	18	6430	705	62.36	50	**210**	**89**
	23/3/20	42.80	35	17	5989	656	60.12	50	**203**	**85**
	23/3/27	46.72	36	19	6538	716	64.42	50	**217**	**91**
	Avg.	45.31	35.5	17.75	6341	695	62.35	50	**210**	**88**
Sun.	23/3/5	48.21	38	20	6747	739	64.86	50	**219**	**92**
	23/3/12	45.91	39	18	6425	704	63.42	50	**214**	**90**
	23/3/19	47.82	40	19	6692	733	65.54	50	**221**	**93**
	23/3/26	50.68	38	19	7093	777	65.62	50	**221**	**93**
	Avg.	48.16	38.75	19	6739	739	64.86	50	**219**	**92**



(a) Tractor-trailer mode

(b) Electric auto-dolly mode

Figure 5.4: The percentage of energy consumed at peak, low, and other times of the day in two operation modes.

Table 5.5: The comparison of two modes on the cases of peak and low in one day.

			Tractor-Trailer Mode					Electric Auto-dolly Mode			
			Travel Time (h)	Trailer Number	Tractor Number	Cost (HKD)	CO_2 (kg)	Travel Time (h)	Dolly Number	Cost (HKD)	CO_2 (kg)
Peak	Monday	23/3/6	28.88	49	25	4042	443	34.24	50	**116**	**49**
		23/3/13	27.33	47	26	3825	419	34.38	50	**116**	**49**
		23/3/20	28.21	48	24	3948	433	35.03	50	**118**	**50**
		23/3/27	29.60	50	29	4142	454	37.55	50	**127**	**53**
	Sunday	23/3/5	28.25	47	27	3953	433	34.57	50	**117**	**49**
		23/3/12	28.33	50	27	3964	434	35.75	50	**121**	**51**
		23/3/19	28.34	50	26	3966	435	36.50	50	**123**	**52**
		23/3/26	32.05	48	27	4485	491	38.70	50	**131**	**55**
	Avg.		28.87	48.63	26.38	4041	443	35.84	50	**121**	**51**
Low	Monday	23/3/6	8.99	21	9	1258	138	17.11	50	**58**	**24**
		23/3/13	11.11	25	11	1555	170	17.84	50	**60**	**25**
		23/3/20	8.35	20	9	1169	128	17.01	50	**57**	**24**
		23/3/27	9.16	18	8	1282	141	10.93	50	**37**	**16**
	Sunday	23/3/5	11.01	27	12	1540	169	19.00	50	**64**	**27**
		23/3/12	9.14	25	9	1279	140	17.59	50	**59**	**25**
		23/3/19	10.71	26	11	1499	164	19.16	50	**65**	**27**
		23/3/26	10.61	24	10	1484	163	17.08	50	**58**	**24**
	Avg.		9.88	23.25	9.875	1383	152	16.97	50	**57**	**24**

Secondly, the total travel time of all dollies is longer than that of all tractors, due to the capacity limitation of dollies and the multi-trailer capacity of tractors. However, electric auto-dollies are more advantageous in terms of energy and cost savings, the reasons may include the following: 1) the electric motor efficiency of electric vehicles is usually higher than the internal combustion engine efficiency of fuel vehicles; 2) Hong Kong has a strong ability of power generation, about 3/4 of the electricity is locally supplied; 3) The Hong Kong Government is actively promoting the process of carbon neutrality and encouraging the use of clean energy through electricity subsidy policy; 4) The diesel oil price in Hong Kong has shown a slow and steady rising trend in the past two years, as shown in Fig. 5.5.

The experimental results show different potential and advantages of the two modes in different aspects. Generally, it is recommended to use the tractor-trailer mode in daytime, since tractor-trailer operations require skilled ground staff, daytime shifts align with staff availability, ensuring efficient manual coordination during busy periods. Besides, electric auto-dolly mode is more suitable to adopt at night, as there will be a lower electricity price at night than during the day. We could also restrict dollies to high-density zones to ensure the electricity supply while covering critical routes. To bridge both modes' strengths, it may also be a good choice to replace diesel tractors with electric tractors, retaining multi-trailer capacity while cutting emissions.

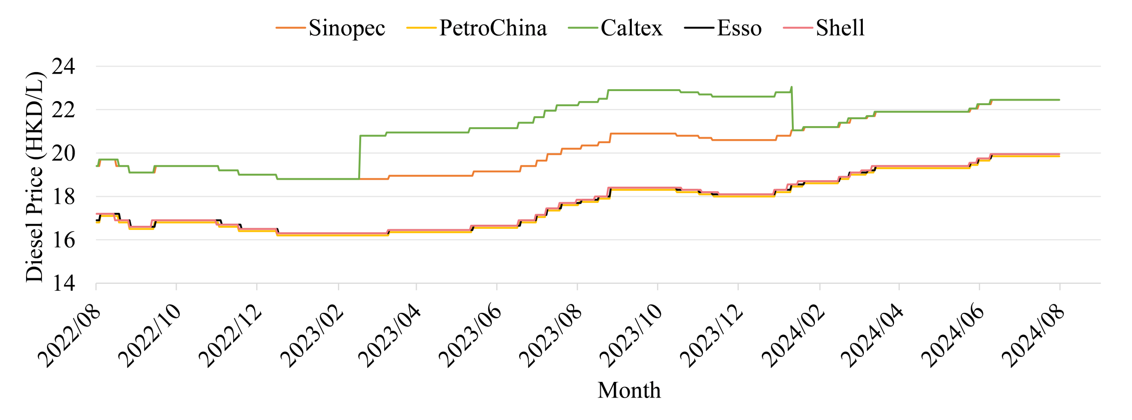


Figure 5.5: The diesel oil price trend of five companies in Hong Kong.

In addition to the above analysis, there are many other factors that may affect the practical application of the two modes. For example, the new electric auto-dolly, which is equipped with multiple sensors and robotic arms, is likely to be more expensive and maintain than the traditional fuel tractor and trailer. The labor cost of the two modes is reflected in different aspects. The use of tractors and trailers requires hiring external airport ground handling staff, like tractor drivers, workers for baggage loading/unloading, etc., while using electric auto-dolly requires more management talent and engineers, and comes with higher training costs. Finally, our experiment and analysis are mainly based on the scenario of the HKIA. The differences of different airports in scale, ground road network, busyness, and local economic level may lead to different suitable modes.

5.2.4 Energy consumption prediction of baggage transport service

According to the analysis in the above section, it is found that the electric auto-dolly mode has the advantages of low operating cost, low energy consumption, and low carbon emission, which meet the needs of airport intelligence and sustainable development. Therefore, we analyze the future energy consumption of baggage transport service based on the electric auto-dolly mode here.

We first generated cases on a daily basis, conducted experiments on all the collected flight data, and then obtained seven sets of monthly dolly operating times. Based on this, a linear regression analysis is conducted on the monthly number of flights F_m and the monthly dolly travel time T_m. The results are shown in Fig. 5.6 and the following formula:$T_m = 0.02F_m - 117.07$ (h).

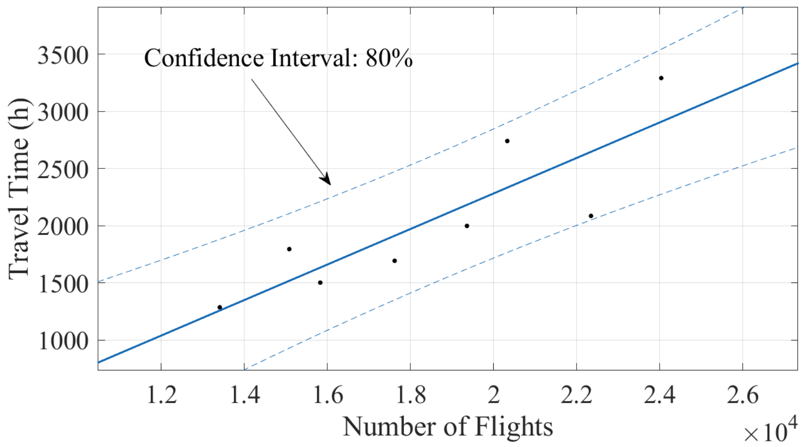


Figure 5.6: The linear regression analysis on the monthly number of flights and monthly dolly travel time.

Next, we use Seasonal Auto-regressive Integrated Moving Average (SARIMA) to forecast the monthly flight number from January 2025 to December 2026 based on the monthly flight number at HKIA from January 2021 to December 2024. SARIMA is the extension of ARIMA for analyzing time series data with seasonal patterns [103]. It combines seasonal differencing, auto-regressive (AR), and moving average (MA) terms to model both non-seasonal (p, d, q) and seasonal (P, D, Q, S) parameters. This method is widely used for forecasting data with trends and recurring cycles, such as monthly sales or temperature variations. The optimized SARIMA parameters using Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are as follows: $p = 0, d = 1, q = 0, P = 1, D = 1, Q = 0, S = 12$. The prediction result is shown in Fig. 5.7. The Mean Absolute Percentage Error (MAPE) of flight number prediction is 10.27%.

According to the above analysis, the predicted flight number and energy consumption of airport baggage transport service from January 2025 to December 2026 under the electric auto-dolly mode is shown in Fig. 5.8. As we forecast that the air traffic of HKIA will continue to grow, the operating costs and carbon emissions of the airport baggage transport service will also continue to increase. However, the impact from this growth can be mitigated by further optimizing the electric auto-dolly scheduling strategy.

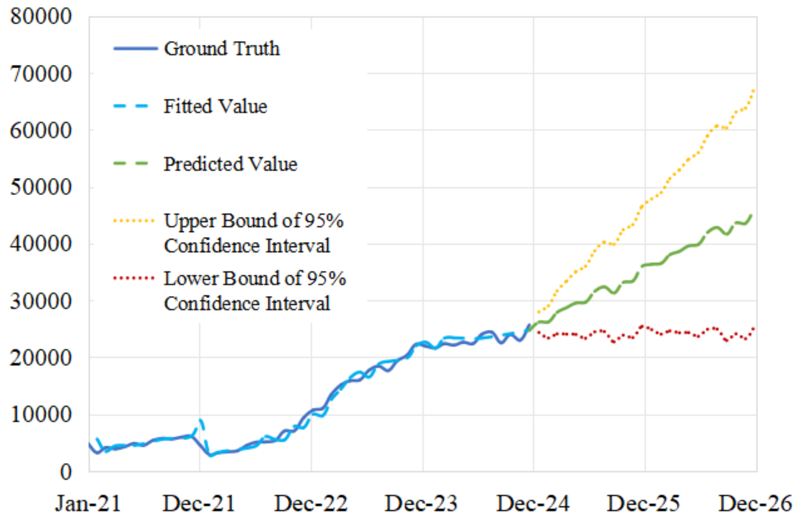


Figure 5.7: The fitted and forecasted number of flights by SARIMA.

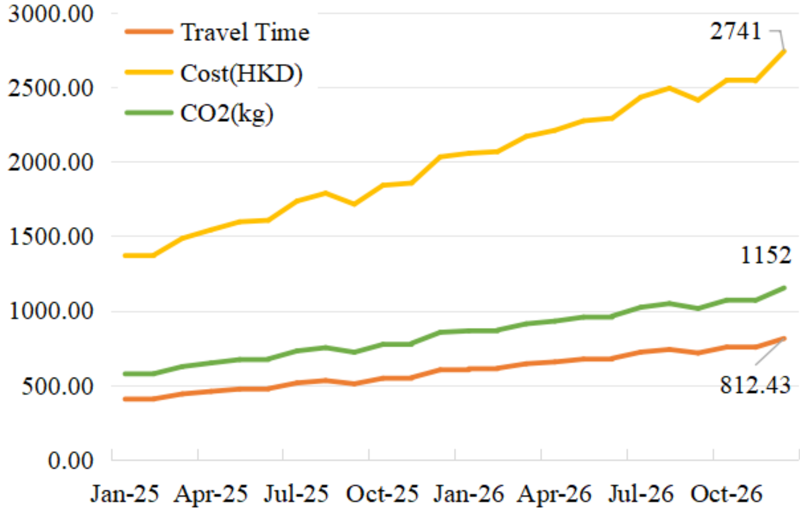


Figure 5.8: The predicted monthly auto-dolly travel time, operation costs, and emissions from Jan 2025 to Dec 2026.

6 CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

In order to keep up with the needs of the booming aviation industry, the smart and sustainable development of airports around the world has been gradually promoted. Efficient airport baggage transport service plays a key role in enhancing airport operational efficiency and service quality. Traditional baggage transport is managed through the collaboration of tractors and trailers operating under the drop-and-pull mode. Recently, new electric auto-dollies have been gradually introduced in airport baggage transport service to foster the intelligent and sustainable development of airports. In this context, it is of great significance to investigate the baggage transport vehicle scheduling problem under two operating modes, tractor-trailer mode and electric auto-dolly mode, to promote the development of airport ground handling. However, the coordinated scheduling of tractors and trailers poses significant challenges due to the complex interactions among tractors, trailers, and flights. Besides, the difficulties of scheduling electric auto-dollies mainly come from the demand for charging and the large problem scale. Additionally, airport ground handling is highly dynamic and uncertain, especially at busy hub airports.

To optimize the baggage transport vehicle scheduling under the two modes and analyze the sustainable development prospect of airport baggage transport service, this study mainly includes the following three aspects. The multi-trailer drop-and-pull baggage transport problem is first investigated. A two-stage scheduling model for tractors and trailers is developed, along with an efficient hybrid intelligence-based solution algorithm.

Specifically, the Adaptive Large Neighborhood Search forms the foundation of the algorithm, enhanced with carefully designed operators. Additionally, two key methods are introduced to boost the algorithm's efficiency: a K-means clustering-based initialization method and a topological sort-based solution evaluation method. The validity of the above methods and operators is verified by simulation experiments.

Secondly, the electric auto-dolly scheduling problem for airport baggage transport is researched. To solve this, a simplified scheduling model is established, which is then formulated into a Markov Decision Process with a heuristic improvement policy. Then, a deep reinforcement learning-based scheduling algorithm is developed, in which the structures of the encoder and decoder are based on the model of the Transformers variant. In addition, the solution features about service time and vehicle power are specifically added to the problem embeddings, thereby improving the algorithm's convergence speed. The performance of our proposed algorithm has been verified through the comparison experiment with other algorithms and the generalization experiment with different distributions and different scale cases.

Finally, empirical analysis is conducted on the baggage transport service under these two modes in the scenario of the Hong Kong International Airport (HKIA). The experiments were carried out on the established integrated airport ground vehicle scheduling simulation platform. Besides, we modeled the energy consumption and carbon emissions of vehicles to support the analysis of the operating cost and sustainable development of the airport baggage transport service. Experiment results show that the electric auto-dolly mode has obvious advantages in terms of operating costs and carbon emissions, but it may require higher vehicle purchase costs and labor costs. Moreover, the monthly flight volume for the second half of 2024, as well as the monthly energy consumption and carbon emissions of airport baggage transport services under the electric auto-dolly mode, to guide airports and related businesses.

In conclusion, this study solved the problem of scheduling airport baggage transport vehicles under the tractor-trailer mode and electric auto-dolly mode, and carried out numerous simulation experiments and detailed empirical analysis. However, there are still some limitations and deficiencies in this study. Firstly, the proposed two-stage algorithm for tractor and trailer based on ALNS shows difficulties in solving large-scale cases, such as providing vehicle scheduling schemes for full-day flights. Therefore, reinforcement learning algorithms can be considered to realize integrated scheduling for tractors and trailers in the future. Secondly, this study only considers the two scenarios of pure fuel and electric fleet, but the mixed vehicle scheduling problem may be an important issue, which may be encountered in the transition phase before the realization of airport full electrification. Finally, based on the existing static scheduling results, it is also a possible important future topic to dynamically adjust the scheduling scheme of baggage transport vehicles using dynamic planning to cope with the high uncertainty and dynamics of flight arrival and departure.

References

[1] M. Studic, A. Majumdar, W. Schuster, and W. Y. Ochieng, "A systemic modelling of ground handling services using the functional resonance analysis method," *Transportation Research Part C: Emerging Technologies*, vol. 74, pp. 245–260, 2017.

[2] J. Chen, M. Weiszer, G. Locatelli, S. Ravizza, J. A. Atkin, P. Stewart, and E. K. Burke, "Toward a more realistic, cost-effective, and greener ground movement through active routing: A multiobjective shortest path approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3524–3540, 2016.

[3] S. Wandelt and K. Wang, "Towards solving the airport ground workforce dilemma: A literature review on hiring, scheduling, retention, and digitalization in the airport industry," *Journal of the Air Transport Research Society*, p. 100 004, 2024.

[4] D. A. Tabares, F. Mora-Camino, and A. Drouin, "A multi-time scale management structure for airport ground handling automation," *Journal of Air Transport Management*, vol. 90, p. 101 959, 2021.

[5] I. A. T. Association, *Iata annual safety report*, 2022. [Online]. Available: `https://www.iata.org/safety-report`.

[6] **Wang, X.**, Y. Long, G. Xu*, and Y. Liu, "Integrated tractor and trailer scheduling for airport baggage transport service," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2023, pp. 3894–3899.

[7] P. P. Oliveira, "Digital twin development for airport management," *Journal of Airport Management*, vol. 14, no. 3, pp. 246–259, 2020.

[8] H. Han, H. G. Olya, E.-N. Untaru, A. Ispas, J. J. Kim, and W. Kim, "Impact of airport green atmospherics on mental health value, image, and loyalty among visitors and workers," *Business Strategy and the Environment*, vol. 29, no. 3, pp. 1186–1198, 2020.

[9] CAAC, *Airport self-driving equipment application roadmap (2021-2025)*, `http://www.caac.gov.cn/XWZX/MHYW/202112/t20211208_210450.html`, 2021.

[10] G. Chen, "Robotics applications at airports: Situation and tendencies," in *2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, IEEE, 2022, pp. 536–539.

[11] C. G. *et al.*, "Driverless technology application in airports," *Academic Journal of Engineering and Technology Science*, vol. 6, no. 12, pp. 45–50, 2023.

[12] F. Greer, J. Rakas, and A. Horvath, "Airports and environmental sustainability: A comprehensive review," *Environmental Research Letters*, vol. 15, no. 10, p. 103 007, 2020.

[13] R. Zhang, D. Wang, and J. Wang, "Multi-trailer drop-and-pull container drayage problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5708–5720, 2020.

[14] Y. Wu, J. Zhou, Y. Xia, X. Zhang, Z. Cao, and J. Zhang, "Neural airport ground handling," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[15] X. Han, P. Zhao, Q. Meng, S. Yin, and D. Wan, "Optimal scheduling of airport ferry vehicles based on capacity network," *Annals of Operations Research*, vol. 295, pp. 163–182, 2020.

[16] L. Lu, Z. Deng, C. Shao, and W. Shen, “A variable neighborhood search algorithm for airport ferry vehicle scheduling problem,” *Transportation Research Part C: Emerging Technologies*, vol. 154, p. 104 262, 2023.

[17] S. Zhu, H. Sun, and X. Guo, “Cooperative scheduling optimization for ground-handling vehicles by considering flights’ uncertainty,” *Computers & Industrial Engineering*, vol. 169, p. 108 092, 2022.

[18] Y. Liu, J. Wu, J. Tang, W. Wang, and X. Wang, “Scheduling optimisation of multi-type special vehicles in an airport,” *Transportmetrica B: Transport Dynamics*, vol. 10, no. 1, pp. 954–970, 2022.

[19] W. Quan, C. Chen, Z. Shao, and K. Meng, “Dynamic scheduling for airport special vehicles based on a multi-strategy hybrid algorithm,” in *2022 41st Chinese Control Conference (CCC)*, IEEE, 2022, pp. 1916–1921.

[20] X. Han, P. Zhao, D. Kong, *et al.*, “Two-stage optimization of airport ferry service delay considering flight uncertainty,” *European Journal of Operational Research*, vol. 307, no. 3, pp. 1103–1116, 2023.

[21] J. M. Brevoord, “Electric vehicle routing problems: The operations of electric towing trucks at an airport under uncertain arrivals and departures,” M.S. thesis, University of Twente, 2021.

[22] P. Zhao, W. Gao, X. Han, W. Luo, *et al.*, “Bi-objective collaborative scheduling optimization of airport ferry vehicle and tractor,” *International Journal of Simulation Modelling*, vol. 18, no. 2, pp. 355–365, 2019.

[23] X. Feng, H. Zuo, and Q. Sun, “Research on collaborative scheduling of aircraft ground service vehicles based on simple temporal network,” in *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, IEEE, 2021, pp. 263–269.

[24] C. Tilk, N. Bianchessi, M. Drexl, S. Irnich, and F. Meisel, "Branch-and-price-and-cut for the active-passive vehicle-routing problem," *Transportation Science*, vol. 52, no. 2, pp. 300–319, 2018.

[25] S. Chen, H. Wang, and Q. Meng, "Autonomous truck scheduling for container transshipment between two seaport terminals considering platooning and speed optimization," *Transportation Research Part B: Methodological*, vol. 154, pp. 289–315, 2021.

[26] Z. Xue, C. Zhang, W. Lin, L. Miao, and P. Yang, "A tabu search heuristic for the local container drayage problem under a new operation mode," *Transportation Research Part E: Logistics and Transportation Review*, vol. 62, pp. 136–150, 2014.

[27] H. Cui, S. Chen, R. Chen, and Q. Meng, "A two-stage hybrid heuristic solution for the container drayage problem with trailer reposition," *European Journal of Operational Research*, vol. 299, no. 2, pp. 468–482, 2022.

[28] D. Bao, J. Zhou, Z. Zhang, Z. Chen, and D. Kang, "Mixed fleet scheduling method for airport ground service vehicles under the trend of electrification," *Journal of Air Transport Management*, vol. 108, p. 102 379, 2023.

[29] S. van Oosterom, M. Mitici, and J. Hoekstra, "Dispatching a fleet of electric towing vehicles for aircraft taxiing with conflict avoidance and efficient battery charging," *Transportation Research Part C: Emerging Technologies*, vol. 147, p. 103 995, 2023.

[30] D. Gulczynski, B. Golden, and E. Wasil, "The split delivery vehicle routing problem with minimum delivery amounts," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 5, pp. 612–626, 2010.

[31] A. Norin, T. A. Granberg, P. Värbrand, and D. Yuan, "Integrating optimization and simulation to gain more efficient airport logistics," in *Eighth USA/Europe Air Traffic Management Research and Development Seminar*, 2009, p. 10.

[32] A. Norin, D. Yuan, T. A. Granberg, and P. Värbrand, "Scheduling de-icing vehicles within airport logistics: A heuristic algorithm and performance evaluation," *Journal of the Operational Research Society*, vol. 63, no. 8, pp. 1116–1125, 2012.

[33] Y. Liu, J. Zhang, C. Ding, and J. Bi, "Modeling and heuristic algorithm of ground ferry vehicle scheduling in large airports," in *CICTP 2019*, 2019, pp. 159–170.

[34] Y. Du, Q. Zhang, and Q. Chen, "Aco-ih: An improved ant colony optimization algorithm for airport ground service scheduling," in *2008 IEEE International Conference on Industrial Technology*, IEEE, 2008, pp. 1–6.

[35] J. Du, O. Brunner Jens, and R. Kolisch, "Planning towing processes at airports more efficiently," *Transportation Research Part E: Logistics and Transportation Review*, vol. 70, pp. 293–304, 2014.

[36] W. Ip, D. Wang, and V. Cho, "Aircraft ground service scheduling problems and their genetic algorithm with hybrid assignment and sequence encoding scheme," *IEEE Systems Journal*, vol. 7, no. 4, pp. 649–657, 2012.

[37] Z. Zeng and L. Jia, "Research on airport special vehicle scheduling problem," in *2020 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, IEEE, 2020, pp. 1076–1078.

[38] M. Weiszer, J. Chen, and G. Locatelli, "An integrated optimisation approach to airport ground operations to foster sustainability in the aviation sector," *Applied Energy*, vol. 157, pp. 567–582, 2015.

[39] Z. Zhang, Y. Che, and Z. Liang, "Split-demand multi-trip vehicle routing problem with simultaneous pickup and delivery in airport baggage transit," *European Journal of Operational Research*, vol. 312, no. 3, pp. 996–1010, 2024.

[40] S. Wang, Y. Che, H. Zhao, and A. Lim, "Accurate tracking, collision detection, and optimal scheduling of airport ground support equipment," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 572–584, 2020.

[41] W. Guo, P. Xu, Z. Zhao, L. Wang, L. Zhu, and Q. Wu, "Scheduling for airport baggage transport vehicles based on diversity enhancement genetic algorithm," *Natural Computing*, vol. 19, pp. 663–672, 2020.

[42] J. Zhou, Y. Wu, Z. Cao, W. Song, J. Zhang, and Z. Chen, "Learning large neighborhood search for vehicle routing in airport ground handling," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[43] T. Clausen and D. Pisinger, "Dynamic routing of short transfer baggage," *DTU Management Engineering, Tech. Rep. 10*, 2010.

[44] S. Padrón, D. Guimarans, J. J. Ramos, and S. Fitouri-Trabelsi, "A bi-objective approach for scheduling ground-handling vehicles in airports," *Computers & Operations Research*, vol. 71, pp. 34–53, 2016.

[45] A.-K. Rothenbächer, M. Drexl, and S. Irnich, "Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows," *Transportation Science*, vol. 52, no. 5, pp. 1174–1190, 2018.

[46] L. Accorsi and D. Vigo, "A hybrid metaheuristic for single truck and trailer routing problems," *Transportation Science*, vol. 54, no. 5, pp. 1351–1371, 2020.

[47] X. Yang, H. A. Daham, and A. Salhi, "Combined strip and discharge delivery of containers in heterogeneous fleets with time windows," *Computers & Operations Research*, vol. 127, p. 105 141, 2021.

[48] D. Wang, I. Moon, and R. Zhang, "Multi-trip multi-trailer drop-and-pull container drayage problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 19 088–19 104, 2022.

[49] Y. Song, J. Zhang, Z. Liang, and C. Ye, "An exact algorithm for the container drayage problem under a separation mode," *Transportation Research Part E: Logistics and Transportation Review*, vol. 106, pp. 231–254, 2017.

[50] M. Schiffer, M. Schneider, G. Walther, and G. Laporte, "Vehicle routing and location routing with intermediate stops: A review," *Transportation Science*, vol. 53, no. 2, pp. 319–343, 2019.

[51] M. Moghaddam, R. H. Pearce, H. Mokhtar, and C. G. Prato, "A generalised model for container drayage operations with heterogeneous fleet, multi-container sizes and two modes of operation," *Transportation Research Part E: Logistics and Transportation Review*, vol. 139, p. 101 973, 2020.

[52] J.-M. Clairand, P. Guerra-Terán, X. Serrano-Guerrero, M. González-Rodríguez, and G. Escrivá-Escrivá, "Electric vehicles for public transportation in power systems: A review of methodologies," *Energies*, vol. 12, no. 16, p. 3114, 2019.

[53] J. Van Duin, L. A. Tavasszy, and H. Quak, "Towards e (lectric)-urban freight: First promising steps in the electric vehicle revolution," *European Transport-Trasporti Europei*, no. 54, 2013.

[54] Q. Zhang, S. Wang, and L. Zhen, "Yard truck retrofitting and deployment for hazardous material transportation in green ports," *Annals of Operations Research*, pp. 1–32, 2022.

[55] R. A. de Mello Bandeira, G. V. Goes, D. N. S. Gonçalves, D. Márcio de Almeida, and C. M. de Oliveira, "Electric vehicles in the last mile of urban freight transportation: A sustainability assessment of postal deliveries in rio de janeiro-brazil," *Transportation Research Part D: Transport and Environment*, vol. 67, pp. 491–502, 2019.

[56] J. Pasha, B. Li, Z. Elmi, A. M. Fathollahi-Fard, Y.-y. Lau, A. Roshani, T. Kawasaki, and M. A. Dulebenets, “Electric vehicle scheduling: State of the art, critical challenges, and future research opportunities,” *Journal of Industrial Information Integration*, p. 100 561, 2024.

[57] I. Kucukoglu, R. Dewil, and D. Cattrysse, “The electric vehicle routing problem and its variations: A literature review,” *Computers & Industrial Engineering*, vol. 161, p. 107 650, 2021.

[58] G. Macrina, L. D. P. Pugliese, and F. Guerriero, “The green-vehicle routing problem: A survey,” *Modeling and optimization in green logistics*, pp. 1–26, 2020.

[59] A. Montoya, C. Guéret, J. E. Mendoza, and J. G. Villegas, “The electric vehicle routing problem with nonlinear charging function,” *Transportation Research Part B: Methodological*, vol. 103, pp. 87–110, 2017.

[60] S. Shao, W. Guan, and J. Bi, “Electric vehicle-routing problem with charging demands and energy consumption,” *IET Intelligent Transport Systems*, vol. 12, no. 3, pp. 202–212, 2018.

[61] S. Rastani, T. Yüksel, and B. Çatay, “Effects of ambient temperature on the route planning of electric freight vehicles,” *Transportation Research Part D: Transport and Environment*, vol. 74, pp. 124–141, 2019.

[62] W. Jie, J. Yang, M. Zhang, and Y. Huang, “The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology,” *European Journal of Operational Research*, vol. 272, no. 3, pp. 879–904, 2019.

[63] R. Raeesi and K. G. Zografos, “Coordinated routing of electric commercial vehicles with intra-route recharging and en-route battery swapping,” *European Journal of Operational Research*, vol. 301, no. 1, pp. 82–109, 2022.

[64] J. Yang and H. Sun, “Battery swap station location-routing problem with capacitated electric vehicles,” *Computers & operations research*, vol. 55, pp. 217–232, 2015.

[65] W. Li-ying and S. Yuan-bin, “Multiple charging station location-routing problem with time window of electric vehicle.,” *Journal of Engineering Science & Technology Review*, vol. 8, no. 5, 2015.

[66] T. Zhang, W. Chen, Z. Han, and Z. Cao, “Charging scheduling of electric vehicles with local renewable energy under uncertain electric vehicle arrival and grid power price,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 6, pp. 2600–2612, 2013.

[67] H. Qin, X. Su, T. Ren, and Z. Luo, “A review on the electric vehicle routing problems: Variants and algorithms,” *Frontiers of Engineering Management*, vol. 8, pp. 370–389, 2021.

[68] G. Hiermann, R. F. Hartl, J. Puchinger, and T. Vidal, “Routing a mix of conventional, plug-in hybrid, and electric vehicles,” *European Journal of Operational Research*, vol. 272, no. 1, pp. 235–248, 2019.

[69] M. M. Alipour, S. N. Razavi, M. R. Feizi Derakhshi, and M. A. Balafar, “A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem,” *Neural Computing and Applications*, vol. 30, pp. 2935–2951, 2018.

[70] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takác, “Reinforcement learning for solving the vehicle routing problem,” *Advances in neural information processing systems*, vol. 31, 2018.

[71] W. Kool, H. Van Hoof, and M. Welling, “Attention, learn to solve routing problems!” *arXiv preprint arXiv:1803.08475*, 2018.

[72] Y. Ma, J. Li, Z. Cao, W. Song, L. Zhang, Z. Chen, and J. Tang, "Learning to iteratively solve routing problems with dual-aspect collaborative transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 096–11 107, 2021.

[73] A. Bogyrbayeva, M. Meraliyev, T. Mustakhov, and B. Dauletbayev, "Learning to solve vehicle routing problems: A survey," *arXiv preprint arXiv:2205.02453*, 2022.

[74] P. Wan, G. Xu, J. Chen, and Y. Zhou, "Deep reinforcement learning enabled multi-uav scheduling for disaster data collection with time-varying value," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[75] H. Hu, X. Jia, Q. He, S. Fu, and K. Liu, "Deep reinforcement learning based agvs real-time scheduling with mixed rule for flexible shop floor in industry 4.0," *Computers & Industrial Engineering*, vol. 149, p. 106 749, 2020.

[76] A. Bdeir, S. Boeder, T. Dernedde, K. Tkachuk, J. K. Falkner, and L. Schmidt-Thieme, "Rp-dqn: An application of q-learning to vehicle routing problems," in *KI 2021: Advances in Artificial Intelligence: 44th German Conference on AI, Virtual Event, September 27–October 1, 2021, Proceedings 44*, Springer, 2021, pp. 3–16.

[77] M. Sewak and M. Sewak, "Deep q network (dqn), double dqn, and dueling dqn: A step towards general artificial intelligence," *Deep reinforcement learning: frontiers of artificial intelligence*, pp. 95–108, 2019.

[78] B. Lin, B. Ghaddar, and J. Nathwani, "Deep reinforcement learning for the electric vehicle routing problem with time windows," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 528–11 538, 2021.

[79] Q. Cappart, T. Moisan, L.-M. Rousseau, I. Prémont-Schwarz, and A. A. Cire, "Combining reinforcement learning and constraint programming for combinato-

rial optimization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 3677–3687.

[80] K. Lei, P. Guo, Y. Wang, X. Wu, and W. Zhao, "Solve routing problems with a residual edge-graph attention neural network," *Neurocomputing*, vol. 508, pp. 79–98, 2022.

[81] C. K. Joshi, T. Laurent, and X. Bresson, "An efficient graph convolutional network technique for the travelling salesman problem," *arXiv preprint arXiv:1906.01227*, 2019.

[82] Y. Wu, W. Song, Z. Cao, J. Zhang, and A. Lim, "Learning improvement heuristics for solving routing problems," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 9, pp. 5057–5069, 2021.

[83] Y. Ma, Z. Cao, and Y. Chee, "Learning to search feasible and infeasible regions of routing problems with flexible neural k-opt," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[84] Y. Ma, J. Li, Z. Cao, W. Song, H. Guo, Y. Gong, and Y. Chee, "Efficient neural neighborhood search for pickup and delivery problems," *arXiv preprint arXiv:2204.11399*, 2022.

[85] A. Hottung and K. Tierney, "Neural large neighborhood search for the capacitated vehicle routing problem," in *ECAI 2020*, IOS Press, 2020, pp. 443–450.

[86] X. Chen and Y. Tian, "Learning to perform local rewriting for combinatorial optimization," *Advances in neural information processing systems*, vol. 32, 2019.

[87] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[88] J. Li, Y. Ma, Z. Cao, Y. Wu, W. Song, J. Zhang, and Y. Chee, “Learning feature embedding refiner for solving vehicle routing problems,” *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[89] L. Duan, Y. Zhan, H. Hu, Y. Gong, J. Wei, X. Zhang, and Y. Xu, “Efficiently solving the practical vehicle routing problem: A novel joint learning approach,” in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 3054–3063.

[90] Y. Min, Y. Bai, and C. P. Gomes, “Unsupervised learning for solving the travelling salesman problem,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[91] Y. Kwon, J. Choo, B. Kim, I. Yoon, Y. Gwon, and S. Min, “Pomo: Policy optimization with multiple optima for reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 188–21 198, 2020.

[92] C. E. Miller, A. W. Tucker, and R. A. Zemlin, “Integer programming formulation of traveling salesman problems,” *Journal of the ACM (JACM)*, vol. 7, no. 4, pp. 326–329, 1960.

[93] X. Shi, F. Deng, M. Guo, J. Zhao, L. Ma, B. Xin, and J. Chen, “A novel fulfillment-focused simultaneous assignment method for large-scale order picking optimization problem in rmfs,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.

[94] S. T. W. Mara, R. Norcahyo, P. Jodiawan, L. Lusiantoro, and A. P. Rifai, “A survey of adaptive large neighborhood search algorithms and applications,” *Computers & Operations Research*, vol. 146, p. 105 903, 2022.

[95] S. Ropke and D. Pisinger, “An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows,” *Transportation science*, vol. 40, no. 4, pp. 455–472, 2006.

[96] X. Wang, T. Choi, Z. Li, and S. Shao, "An effective local search algorithm for the multidepot cumulative capacitated vehicle routing problem," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 12, pp. 4948–4958, 2019.

[97] J. Wang and M. Meng, "Real-time decision making and path planning for robotic autonomous luggage trolley collection at airports," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 4, pp. 2174–2183, 2021.

[98] X. Xu and Z. Wei, "Dynamic pickup and delivery problem with transshipments and lifo constraints," *Computers & Industrial Engineering*, vol. 175, p. 108 835, 2023.

[99] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[100] HKIA, *Arrival flights in the hong kong international airport*, `https://www.hongkongairport.com/en/flights/departures/passenger.page`, 2023.

[101] C. Fiori, K. Ahn, and H. A. Rakha, "Power-based electric vehicle energy consumption model: Model development and validation," *Applied Energy*, vol. 168, pp. 257–268, 2016.

[102] X. Wu, D. Freese, A. Cabrera, and W. A. Kitch, "Electric vehicles' energy consumption measurement and estimation," *Transportation Research Part D: Transport and Environment*, vol. 34, pp. 52–67, 2015.

[103] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.