

Article

AGV Scheduling for Optimizing Irregular Air Cargo Containers Handling at Airport Transshipment Centers

Jie Li ¹, Mingkai Zou ¹, Yaqiong Lv ^{1,*}  and Di Sun ²

¹ School of Transport and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China; ljij@whut.edu.cn (J.L.); zoumingkai@whut.edu.cn (M.Z.)

² S.F. Express Group Co., Ltd., Shenzhen 518103, China; sarrysundi@163.com

* Correspondence: y.q.lv@whut.edu.cn

Abstract: Airport transshipment centers play a pivotal role in global logistics networks, enabling the swift and efficient transfer of cargo, which is essential for maintaining supply-chain continuity and reducing delivery times. The handling of irregularly shaped air cargo containers presents new constraints for automated guided vehicles (AGVs), as these shapes can complicate loading and unloading processes, directly impacting overall operational efficiency, turnaround times, and the reliability of cargo handling. This study focuses on optimizing the scheduling of AGVs to enhance cargo-handling efficiency at these hubs, particularly for managing irregular air cargo containers. A mixed-integer linear programming (MILP) model is developed, validated for feasibility with the Gurobi solver, and designed to handle large-scale operations. It incorporates a novel approach by integrating a simulated annealing optimized genetic algorithm (GA). The experimental results demonstrate that the designed algorithm can solve models of considerable size within 8 s, offering superior time efficiency compared to the solver, and an average solution quality improvement of 12.62% over the genetic algorithm, significantly enhancing both the model's efficiency and scalability. The enhanced AGV scheduling not only boosts operational efficiency but also ensures better integration within the global logistics framework. This research provides a robust foundation for future advancements in logistics technology, offering both theoretical and practical insights into optimizing complex transportation networks.



Citation: Li, J.; Zou, M.; Lv, Y.; Sun, D. AGV Scheduling for Optimizing Irregular Air Cargo Containers Handling at Airport Transshipment Centers. *Mathematics* **2024**, *12*, 3045. <https://doi.org/10.3390/math12193045>

Academic Editor: Andrea Scozzari

Received: 17 August 2024

Revised: 16 September 2024

Accepted: 23 September 2024

Published: 28 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: airport transshipment center; AGV scheduling; container; simulated annealing genetic algorithm

MSC: 90-11

1. Introduction

Air cargo holds an indispensable role in global logistics and supply-chain management, offering efficient and swift transportation services that meet the high standards of timeliness and reliability demanded by the globalized economy [1]. The high timeliness requirements and the complexity of handling irregular containers make air cargo an ideal subject for research on scheduling optimization. Through the study of airport cargo operations, we aim to address its unique challenges and provide viable solutions for other freight centers that also face high timeliness and operational complexity. As a critical node within the air logistics network, the operational efficiency of airport transshipment centers has a decisive impact on the overall performance of the supply chain. Notable airport transshipment centers around the world, such as Memphis International Airport in the United States, Hong Kong International Airport in China, and Dubai International Airport in the United Arab Emirates, have propelled the growth of global trade through their exceptional cargo-handling capabilities. In modern airport transshipment centers, transshipment efficiency is a key indicator of their core competitiveness [2]. An efficient transshipment process not

only accelerates the flow of goods but also reduces operational costs and enhances customer satisfaction. In this context, unit load devices (ULDs) [3], hereafter referred to as containers, play a crucial role. During the transportation process between the origin airports and the hubs, cargo must be loaded into containers for transshipment. Given the considerable size and weight of containers, manual handling is inefficient, making the introduction of automated guided vehicles (AGVs) an inevitable choice [4]. Through precise scheduling and path optimization, AGVs facilitate seamless integration in the handling and loading of containers, significantly improving operational efficiency.

In airport transshipment centers, the handling of containers involves three key areas, namely the cargo aircraft apron, the buffer, and the container yard (Figure 1). Handling irregularly shaped containers in these zones requires specific AGVs equipped with advanced capabilities to manage the unique challenges these containers present. These specialized AGVs are programmed to handle varying dimensions and weight distributions, which is crucial for maintaining efficiency during the loading and unloading processes. The specialization often involves AGVs with customized gripper designs and enhanced sensory and navigation systems to navigate the complexities of moving irregular containers from the aircraft to the buffer and then to the container yard. This adaptation extends to the AGVs' scheduling algorithms, which must be sophisticated enough to coordinate the intricate movements of containers across the different areas of the transshipment center. By deploying AGVs that are specifically tailored for these tasks, the transshipment centers can ensure the precise handling of irregular containers, enhancing overall operational efficiency and reliability in the cargo flow process.

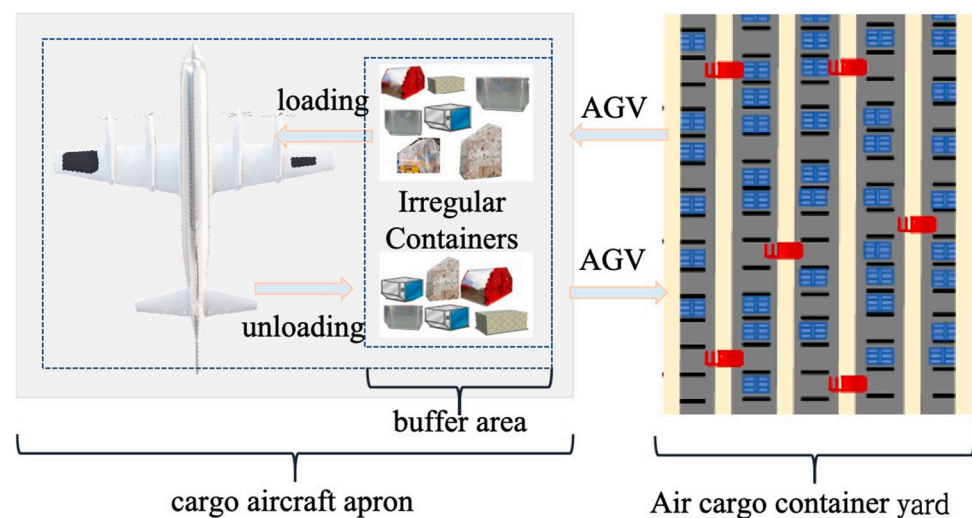


Figure 1. Transshipment hub dispatch layout diagram.

To minimize various costs within the transshipment center under limited scheduling resources and fixed time windows, efficient AGV scheduling is crucial. However, due to the unique characteristics of the air logistics industry, there is currently little research on airport transshipment centers, especially regarding the handling of irregular air containers. These oversized and overweight items, which require more complex loading and unloading processes, are often manually handled. Current studies predominantly focus on the handling of regularized cargo, while irregular air cargo, due to its inherent complexity and dynamic nature, has not received adequate attention in the existing literature [5,6]. The scheduling optimization of automated guided vehicles (AGVs) in the air cargo industry, particularly for irregular cargo, presents a unique set of challenges, including improving operational efficiency and controlling costs. This research gap highlights the significant academic and practical value of exploring the application of AGVs in irregular cargo handling. To address this gap, this study aims to design a refined AGV scheduling scheme for airport transshipment centers, proposing a mixed-integer programming model that seeks to

minimize aircraft-delay costs and transshipment center operating costs within constrained scheduling resources and fixed time windows. The model is solved for small-scale scenarios using the Gurobi solver and a genetic algorithm, with a comparison of their performance. For large-scale problems, both the genetic algorithm and a simulated annealing genetic algorithm developed in this study are employed, with a comparative analysis of these two approaches. The contributions of this study include:

- (1) This paper focuses on the complex scheduling problem of AGV handling irregular containers in airport scenarios, which is a study of significant importance for solving practical issues, addressing a gap in current research;
- (2) A novel optimization framework is introduced, focusing on the scheduling and matching of AGVs. The framework reduces the emphasis on path autonomy in AGV scheduling, concentrating instead on the core tasks of container handling, such as load-bearing constraints. A mixed-integer programming model is developed with the objective of minimizing aircraft-delay costs and transshipment center operating costs;
- (3) Based on model analysis, an improved simulated annealing genetic algorithm is designed to solve the model. The effectiveness of the model and the reliability of the algorithm are validated through comparisons with Gurobi's exact solutions in both small-scale and large-scale experiments.

The remainder of the paper is as follows. The literature is reviewed in Section 2. A detailed description of the problem is given in Section 3. The model and solution approach are described in Section 4, and computational experiments are conducted in Section 5. Finally, Section 6 concludes the paper and discusses future research directions.

2. Literature Review

This study focuses on the scheduling and allocation of automated guided vehicles (AGVs) for irregular containers in airport transit hubs. Compared to standardized containers, irregular containers pose greater challenges in terms of weight, shape, and space utilization, leading to a lack of extensive research in this area. However, research on the scheduling of standardized containers is more prevalent and can provide some insights and methods for addressing the issues discussed in this study. This paper systematically analyzes the existing research to explore how AGVs can be effectively scheduled to transport containers within the constraints of limited AGV resources and fixed time windows. Furthermore, this section attempts to map existing scheduling models to other related scenarios to explore their broader application potential.

In the context of air cargo, only a few studies have examined the scheduling of standardized containers. The research conducted by Huang et al. (2020) [7] primarily focuses on the transportation routes of vehicles in intermodal air-land container operations and the strategies for container consolidation. They considered the cargo routes of airlines and the cost of container utilization to reduce the economic costs of freight transportation. This study is significant for understanding the synergies between different modes of transportation. Virginie Lurkin et al. (2015) [8] explored optimizing the onboard load distribution to reduce fuel consumption, paying particular attention to issues of weight and balance during the journey, as well as the sequence of container pickup and delivery by aircraft. However, their research did not delve into the role of AGVs in container handling, a critical factor for enhancing the efficiency of airport transshipment centers. Hua-An Lu et al. (2011) [9] addressed the issue of container inventory scheduling, developing a spatiotemporal network based on a weekly cycle schedule to formulate a mathematical programming model that minimizes the amount of equipment needed for empty container preparation. Felix Brandt et al. (2019) [10] provided a comprehensive review of the air cargo-loading planning problem, considering both combination carriers and dedicated freight carriers. The study elaborated on the varying shapes of air cargo containers and the loading/unloading processes, with detailed descriptions of container stacking in yards and loading onto aircraft. It also emphasized the challenges of complex scheduling in current loading research, including how to more efficiently utilize container space and

tackle subsequent transportation issues, offering valuable insights for our research. Felix T.S. Chan et al. (2006) [11] approached the problem of differing shapes and specifications of air cargo pallets as a three-dimensional, non-uniform, multi-box packing problem. They proposed a two-stage intelligent decision support system (DSS) for the air cargo loading problem (ACLP). The first stage aimed to minimize the cost by using the weight and volume of pallets as the objective function in a linear programming (LP) model to determine the minimum cost of loading. In the second stage, they developed a heuristic algorithm to generate loading plans for each pallet. However, their study only considered the efficiency of three-dimensional cargo loading without addressing the scheduling of subsequent handling processes. Similarly, studies by Thitipong Jamrus et al. (2016) [12] and Ching-Hui Tang (2011) [13] concentrated on maximizing container loading based on cargo demand, aiming to reduce fuel costs by optimizing container and aircraft space utilization. This differs significantly from our focus on AGV scheduling and loaded container management. Overall, the existing research on airport containers largely emphasizes container loading issues, providing valuable strategies for optimizing container configurations, yet neglecting the scheduling of containers via AGVs.

For the optimization of AGV container scheduling in airport transshipment centers, this study extends to a broader examination of AGV scheduling strategies, including an in-depth exploration of the applicability and optimization methods of these strategies across various scenarios. Zhang et al. (2010) [14] provided a crucial perspective on how to conduct effective scheduling in resource-constrained environments. By employing reactive tabu search (RTS), they investigated the optimal container transportation sequence under limited resources, offering a potential technical pathway for solving AGV scheduling problems. Chen et al. (2020) [15] further emphasized the coordination between AGVs and Automated Stacking Cranes (ASCs) in automated container terminals. They explored the multi-robot coordinated scheduling problem by integrating the scheduling of rail-mounted gantry cranes and AGVs, and their method of iterative adjustment costs offers an alternative solution framework. Radhia Zaghdoud et al. (2016) [16] proposed a hybrid approach combining Dijkstra's algorithm, genetic algorithms (GA), and other heuristic methods to solve the problem of assigning vessel containers to AGVs. The innovation of this method lies in its ability to synthesize the advantages of different algorithms to provide effective solutions for complex scheduling problems. Yu et al. (2023) [17] studied the bidirectional transportation model in AGV scheduling, rigorously considering equipment coordination, bidirectional conflict-free routing, and the requirements of import and export container tasks. They proposed a bi-level mixed-integer programming model and developed a bi-level differential evolution algorithm based on dynamic routing methods to minimize the completion time of all tasks. Additionally, Wang et al. (2024) [18] explored the integrated scheduling of container transfers between ASCs and AGVs in direct mode (without human intervention), buffer mode (involving temporary site storage of containers), and hybrid mode, developing a flexible genetic algorithm to address these scheduling challenges. Furthermore, Han et al. (2024) [19] investigated the integrated optimization problem of flexible job shop scheduling (FJSP) and AGVs with the objective of minimizing makespan, proposing a novel mixed-integer linear programming (MILP) model, and designing a dual-population cooperative genetic algorithm (DCGA) to solve the problem. Li et al. (2024) [20] further explored the characteristics of FJSP-AGVs and proposed a multi-strategy-driven genetic algorithm to optimize the makespan criterion.

Based on the analysis of the aforementioned literature, we can understand the current research on container scheduling at air cargo transfer centers. By extending scheduling models into other scenarios, we gain a better understanding of various research methods and the progress of scheduling AGVs under time-window constraints and resource limitations. The focus of this study is on the scheduling of AGVs for unloading and loading containers from the buffer to the container yard within a given time window at air cargo transfer centers. The characteristics of the problem studied are the following. (1) The study assumes known flight information and details about the containers being loaded

and unloaded. It does not consider path selection for AGVs but focuses on their dynamic operation. (2) The problem considers both the movement of unloaded containers from the buffer to the container yard and the movement of containers from the container yard to the buffer for loading onto airplanes, maximizing the use of limited resources. (3) A genetic algorithm is employed to solve the model, and the effectiveness of the model and the reliability of the algorithm are validated using the Gurobi solver.

3. Problem Statement

The paper takes the transshipment center of an international airport as a case study, focusing on the coordination and scheduling of automated guided vehicles (AGVs) during the handling of containers before and after aircraft arrival at the apron. The aim is to achieve seamless integration and efficient operations in air logistics through the meticulous design of AGV scheduling strategies. The proposed scheduling scheme solely addresses the matching and coordination of AGV movements between the buffer and the container yard without considering the sequence of loading and unloading containers onto the aircraft or the specific routes taken by the AGVs during container transportation.

3.1. Scene Description

This study focuses on AGV scheduling for the transmission of irregular containers within the context of an airport transit center, providing a detailed description of the scenario and the entire process. As illustrated in the scene diagram in Figure 2, after landing, aircraft are parked in designated zones within the cargo aircraft apron, and the container storage area is divided into multiple rows, with a container yard in compartments on each level. The AGVs move along designated pathways. On the right side, the interface of the logistics control center, which is part of the collaborative scheduling system of the transit center, is displayed, along with the dashboard that shows the relationships of the problem's elements. The following is a detailed explanation.

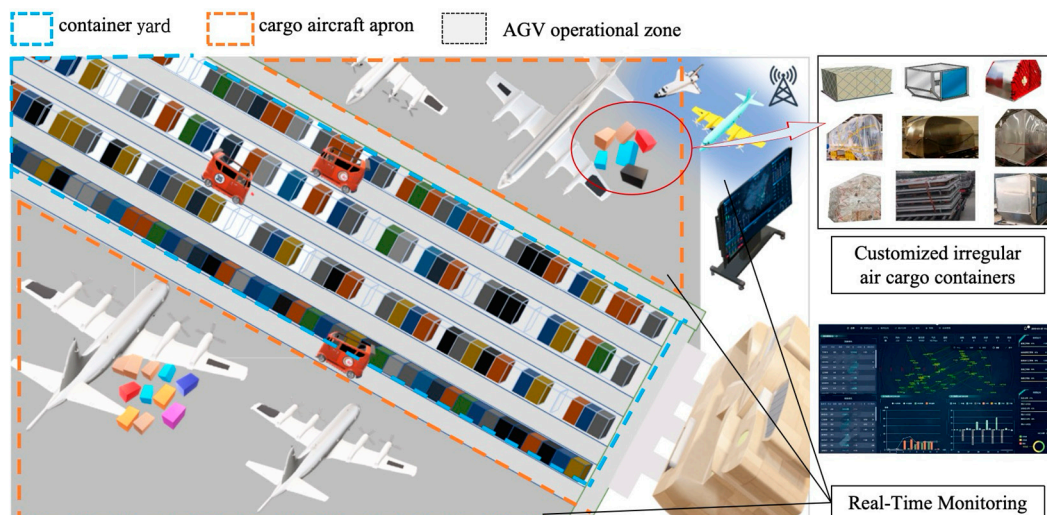


Figure 2. Real-world scenario diagram.

For the coordinated scheduling system logistics control center, in this paper, the coordinated scheduling system comprises four main components, namely the air traffic control information system, the airport information system, the transfer center high-level control system, and the customs information system. This system, functioning at a higher level, integrates these four subsystems to provide comprehensive information, including the flight details and the current AGV scheduling status. By analyzing real-time data, the system facilitates detailed management of transfer center operations and enables prompt adjustments in response to unforeseen events, such as flight delays or AGV malfunctions.

For the cargo aircraft apron, the primary function of the apron is to provide a landing area for aircraft. In this paper, it serves as an endpoint for AGV scheduling. Due to the variations in aircraft types and their internal structures, transferring containers up and down proves to be highly challenging. Therefore, the scheduling premise considered here is that, upon aircraft arrival, all containers to be unloaded are first transferred to a buffer on the apron. They will await AGV transport to the container storage area. Similarly, containers to be loaded onto the aircraft must first be transported to the apron's buffer by AGVs before being loaded onto the aircraft by other means. This paper assumes that the apron is sufficiently large to accommodate the designated number of aircraft.

For the AGV, an intelligent handling vehicle equipped with a lifting cabin is designed for container transportation. The AGVs considered in this paper have varying load capacities, and therefore, scheduling plans must account for these constraints. The speed and charging issues of AGVs are not considered.

For the AGV operational area, the operational area for AGVs is predetermined and fixed, allowing movement only within this designated safety zone. It is assumed that the AGV operational area is sufficiently spacious to prevent collisions or rear-end incidents.

For the container yard, the stack yard is specifically designated for container storage. Due to the varied shapes of containers, each container has a fixed storage position to ensure an efficient use of space. This paper assumes that the container numbering system also represents its fixed position within the stack yard.

Prior to the arrival of the aircraft, the coordinated scheduling system logistics control center in the transfer center integrates modules, such as the flight node dashboard and the apron visual recognition system, to obtain comprehensive flight details, including the flight number, arrival and departure times, container numbers for unloading, and container numbers for loading. This information forms a detailed data foundation, enabling the transfer center's staff to develop precise resource scheduling strategies for container management and accurately plan and execute AGV tasks. For each container, its number corresponds to its fixed position in the container yard, which, when known, provides the endpoint locations and travel times. Additionally, due to the unique requirements of air cargo, including space utilization and weight limits, container shape, and weight information are planned in advance. Since this paper primarily addresses scheduling issues, it focuses only on container weight information. The AGVs used for transporting containers must be capable of supporting the weights of the containers, necessitating the acquisition of AGV load capacity data.

After the aircraft arrives and all unloading containers are transported to the apron, the on-site AGVs execute the scheduling plan according to the dispatch scheme. As depicted in Figure 3, once the flight arrives at the apron, the first step is to unload the containers. In this scenario, the unloading time for all flights is fixed and does not vary with the number of containers being unloaded (represented by the light blue section). Considering the subsequent need to transfer containers from the stack yard to the buffer for loading onto the aircraft, a portion of time is reserved for this operation (represented by the green section). The dark-blue section in the middle represents the acceptable time window for handling the containers for the flight. The actual handling time is indicated by the red outline, which cannot commence before the dark-blue section, as the flight has either not arrived or is still unloading. If the actual handling time extends beyond the dark-blue section, the green section will shift to the right, with the white-filled portion representing the aircraft-delay time.

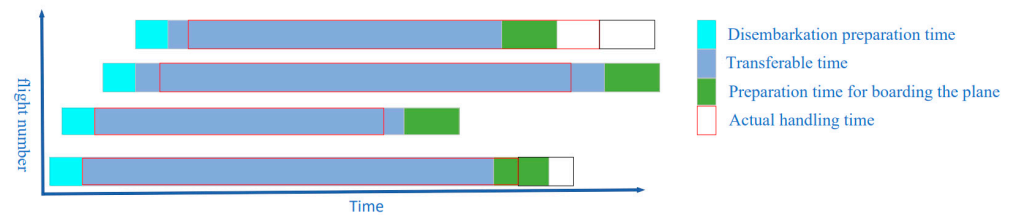


Figure 3. Time distribution chart.

3.2. Problem Description

This article focuses on the coordination and scheduling challenges of AGVs during container handling operations before and after the aircraft arrives at the apron. By gathering information from the control center and the actual site, a definitive AGV scheduling plan must be devised within the time window to ensure that the operation is carried out at the lowest possible cost. The scheduling process is illustrated in Figure 4.

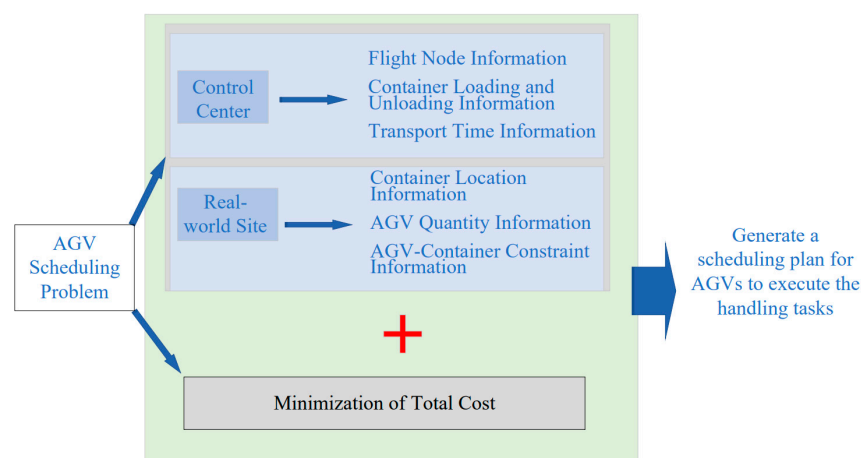


Figure 4. Dispatch process description.

First, the control center at the transfer hub receives information from the aircraft flight, which allows it to view the various time points of the flight, the details of the containers to be handled after landing, the scheduling status of the containers in the ground storage area, and a high-level overview of the aircraft parking situation on the apron, including the parking spot for the next flight.

Subsequently, based on the received information, the boarding staff at the transfer center assigns numbers to the containers being prepared for transport and communicates the container boarding information to the control center. This allows for timely adjustments based on the urgency of the aircraft or any unexpected events.

All preparation work for the containers will be completed before the aircraft arrives. The transfer center will then schedule the AGVs for container handling based on the timely arrival of the aircraft and the readiness of the containers for loading. The optimal scheduling plan will be sent to the ground staff to assist with the AGVs in the loading and unloading process.

After the flight arrives, all containers to be unloaded will be promptly removed and temporarily stored on the buffer. Before this process is completed, the specific AGV scheduling plan for container handling will be fully generated, and the AGVs will carry out their tasks according to the generated plan.

The following diagram illustrates the process of scheduling AGV (automated guided vehicle) for container handling in an airport scenario. Under the conditions of consolidating and understanding various types of information, including online data from the control center and actual offline scenario information from different storage yards and buffer areas, the scheduling model designed in this paper generates specific scheduling tasks with the

objective of minimizing the total costs. Ultimately, the transfer center schedules operations based on the generated scheduling tasks.

4. Models and Methods

4.1. Mathematical Model

In this section, a single-objective optimization model is established with the goal of minimizing costs, including flight delay costs and the operational costs of the transfer center. The model incorporates AGV load constraints, aiming to identify the most efficient scheduling solution.

(1) The sets and parameters

Based on the scenario described in this paper, the model requires the use of seven sets, including the set of AGVs (J), the set of all containers to be transported in the scheduling plan (I), the set of containers to be loaded and unloaded from flights (I_{on} , I_{of}), the set of all flights (F), the set of feasible transportation times (Y), and the set of actual transportation times (T). In this model, the time for AGVs to move without a load will not be considered.

J : set of AGVs on site.

I : set of containers.

I_{of} : the set of all unloading containers, that is, the set of containers that need to be transported from the buffer to the container yard.

I_{on} : the set of all loading containers, that is, the set of containers that need to be transported from the container yard to the buffer.

F : the set of aircraft flights.

Y : the feasible handling time set.

T : the actual handling time set.

The union of sets I_{of} and I_{on} constitutes set I . The feasible handling set pertains specifically to each flight, commencing upon the completion of unloading all unloading containers after arrival and concluding with a reserved period prior to departure for the loading of containers. Therefore, the departure time minus the reserved time marks the end of the feasible handling period. In this work, this set is defined as an interval set $[arrival + 15, departure - 20]$, which represents an ideal scenario. Actual handling times may extend beyond this interval due to insufficient numbers of AGVs and constraints on AGV container handling. Considering practical circumstances, handling cannot commence before the arrival of the flight. Hence, any extension beyond the interval will necessarily result in a delay at the interval's right endpoint.

$[a_i, d_i]$: the time window during which the i -th container can be transported.

W_i : the weight of the i -th container task.

C_j : the maximum load capacity of the j -th AGV.

U_i : the time required to transport the i -th container task.

M : the cost per minute of delay.

P : the operational cost per minute for the transshipment center.

(2) The variables are:

t_{ij} : the commencement time for the j -th AGV to begin transporting the i -th container task.

L_i : the duration of the i -th container task is handled beyond the right endpoint of its time window.

$$L_i = t_{ij} + U_i + 1 - d_i \quad (1)$$

X_{ijt} : binary variable indicating whether the i -th container task is being transported by j -th AGV at time t , with 1 if true and 0 otherwise.

O : The minimum–maximum completion time

$$O = \min(\max(t_{ij} + U_i + 1)) \quad (2)$$

(3) Objective function

The total cost, encompassing the aggregate delay cost and operational cost, for all flights within the computation period is minimized.

$$F = \min(\sum_{i \in I} \sum_{j \in J} (L_i \times M) + P \times O) \quad (3)$$

(4) Constraint

An AGV can transport only one container at a time.

$$\sum_{i \in I} \sum_{j \in J} X_{ijt} = 1, \forall i \in I \quad (4)$$

The interval between consecutive handling tasks for the same AGV must be no less than the handling duration plus a relaxation period (which may be interpreted as the time required for container loading and unloading).

$$t_{ij} + U_i + 1 \leq t_{kj}, \forall j \in J, \forall i \in I, \forall k \in I, i \neq k \quad (5)$$

The container handling constraint is any given container that shall be transported by a single AGV exclusively.

$$\sum_{i \in I} X_{ijt} = 1, \forall j \in J, \forall t \in T \quad (6)$$

For the AGV load capacity constraint, for any AGV, the weight of the container being transported must fall within its designated load capacity.

$$S_j = \sum_{j \in J} \sum_{t \in T} (X_{ijt} \times W_i), \forall i \in I \quad (7)$$

$$S_j \leq C_j, \forall j \in J \quad (8)$$

4.2. Solution Approach

4.2.1. Technical Route

Combination optimization problems are notoriously challenging to solve due to their vast and unevenly distributed solution spaces. This paper focuses on two heuristic algorithms and provides a description of their concepts. Simulated annealing (SA) [21] is a stochastic optimization method inspired by the annealing process in solid-state physics, aimed at finding the global or near-optimal solution in large-scale complex systems. Genetic algorithm (GA) [22], on the other hand, is a bio-inspired optimization technique that mimics the process of natural selection and genetic mechanisms to search for optimal solutions. The former avoids local optima by gradually lowering the temperature, while the latter improves solution quality through natural selection and mutation. Both methods excel in combination optimization tasks [23,24], particularly in AGV scheduling problems [25]. This paper presents a hybrid approach that integrates simulated annealing with genetic algorithms, leveraging the strengths of both to improve the efficiency and quality of scheduling solutions while exploring the underlying principles and application outcomes.

The proposed simulated annealing genetic algorithm (SAGA) commences by setting parameters, such as the AGV ensemble, container set, AGV load capacity, and container weights. Based on the problem at hand, the initial conditions, including the population size, number of iterations, mutation rate, and initial annealing temperature, are established, followed by the generation of an initial population of container assignment schemes. The genetic algorithm loop then ensues, beginning with the formulation of a comprehensive fitness evaluation based on the objective function and constraints. Chromosomes are selected using a roulette wheel method, and to avoid selecting local optima, a simulated annealing operator is employed to probabilistically replace inferior solutions, thus ensuring

global optimization and maintaining diversity within the population, which is conducive to the propagation of the group and the discovery of the global optimum. Subsequently, crossover and mutation operations are performed to increase population diversity, with the temperature parameter from the simulated annealing process regulating the breadth and depth of the search. As the number of iterations increases, the temperature is gradually reduced, enhancing the algorithm's ability to concentrate on superior solutions.

Upon completion of these phases, a condition check determines whether the iteration limit has been reached. If so, the loop is exited, the fitness of all individuals in the population is assessed, and the best solution found is recorded. Otherwise, the process returns to the initialization stage, and the optimization computations are executed anew, as illustrated in Figure 5.

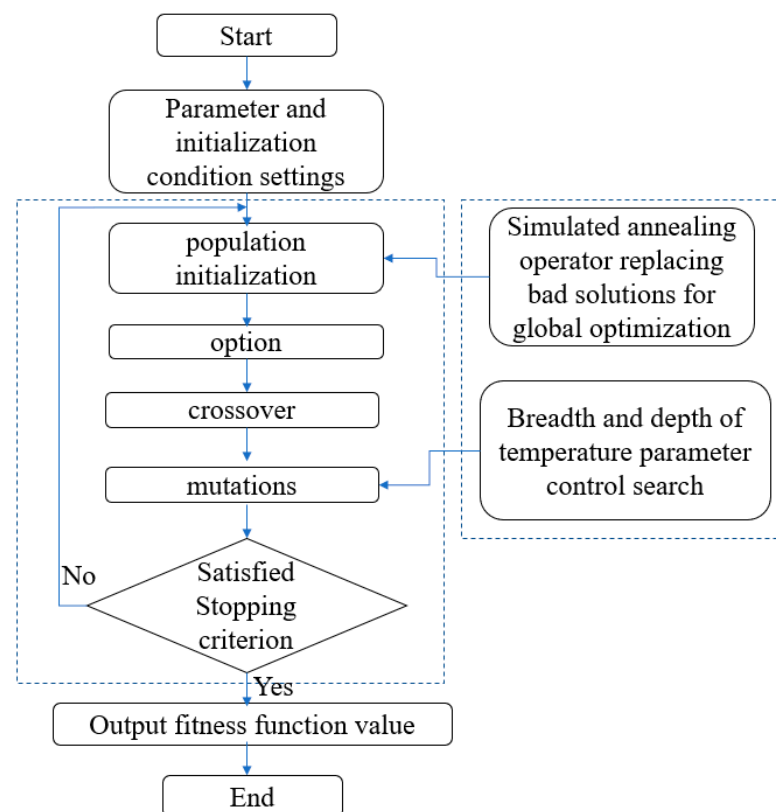


Figure 5. Overview of the simulated annealing genetic algorithm.

4.2.2. Operation Steps

The computational process of a genetic algorithm typically encompasses four principal steps, namely selection of the encoding scheme, population initialization, core loop of the genetic algorithm, and optimal solution screening and recording [26]. To address the problem of assigning appropriate AGVs for the handling of containers during aircraft loading and unloading at an airfreight transshipment center, traditional genetic algorithms require enhancement. This paper introduces the integration of a simulated annealing algorithm into the conventional genetic algorithm, resulting in a novel genetic simulated annealing algorithm. Compared to GA, SAGA employs a simulated annealing mechanism that allows for a certain probability of accepting worse solutions during the selection process, which helps to escape local optima and, thus, facilitates the search for the global optimum. Moreover, the hybrid approach enables a more extensive exploration of the solution space and, with an increase in the number of iterations, gradually focuses on higher-quality solutions. The specific operational steps are outlined in the following pseudocode (Algorithm 1):

Algorithm 1: SAGA

```

01: Data:I-containers, J-AGV, P-Population, population_size = 100, generations = 1000,
mutation_rate = 0.01, temp = 10,000, alpha = 0.99, min_temp = 1
02: Output:the best solution and minimal cost
03: procedure
04: Set Count = 0
05: Population P randomized matching i & j from I & J under constraints
06: while (Stopping criteria are not satisfied) do
07: New Population  $\hat{P} \leftarrow$  selection mechanisms (GA, SA)
08: Crossover operators (GA) and dual mutation operation (GA, SA)
09: If Count = Restare then
10:   Calculate div(P)
11:   If div(P) < threshold then
12:     If similarity > boundary then
13:       execute the restart strategy
14:     end if
15:   end if
16:   Count  $\leftarrow$  0
17: else
18:   Count ++
19: end if
20: P  $\leftarrow$   $\hat{P}$ 
21: end while

```

Step One: Selection of the Encoding Scheme

The task allocation encoding in this work employs a dictionary-based data structure. Each automated guided vehicle (AGV) serves as a key within the dictionary, with the assigned containers constituting the corresponding values. This representation is both intuitive and suitable for addressing complex scheduling problems.

Step Two: Population Initialization

Population initialization is achieved through the random assignment of containers, taking into account the payload constraints of the AGVs. This method ensures the feasibility of the initial solutions while providing diversity within the population, thereby enhancing the search space for subsequent iterations.

Step Three: Core Loop of the Genetic Algorithm**(1) Fitness function design**

The objective is to minimize both the cost of flight delays and the operating cost of the transshipment center. Accordingly, fitness functions are formulated for these two metrics, incorporating the established time-window constraints. A composite fitness metric is then constructed based on these two criteria to assess the quality of the solutions.

(2) Selection

Selection is performed using the roulette wheel selection method. To prevent the chromosome from becoming trapped in a local optimum, a simulated annealing algorithm is introduced. During selection, bad solutions are accepted with a certain probability of promoting global optimization and maintaining population diversity, thus enhancing the likelihood of finding the global optimum.

(3) Crossover

Crossover involves the exchange of gene segments at the same position between two chromosomes. In this work, crossover operations must adhere to the following conditions:

- (a) Maximize the retention of high-quality genes from parent chromosomes;
- (b) Ensure that there are no repeated gene values within offspring chromosomes;
- (c) Offspring chromosomes must comply with stringent constraint conditions.

This design ensures that offspring chromosomes inherit high-quality genes while adhering to the limitations imposed by the practical problem.

(4) Mutation

Given that gene values cannot be repeated within a chromosome, a swap mutation strategy is employed, involving the random exchange of two genes. This method is referred to as a two-point swap mutation. Additionally, the temperature parameter from simulated annealing is utilized to control the breadth and depth of the search. As the number of iterations increases, the temperature gradually decreases, thereby sharpening the algorithm's focus on high-quality solutions.

Step Four: Termination of the Loop and Optimal Solution Selection

Upon reaching the predetermined number of iterations, the algorithm selects and records the current best solution. This process ensures that the final output represents the optimal solution found throughout the entire search process.

5. Computational Experiments

The data for the illustrative cases in this study originate from the operational records of an airfreight transshipment center, serving to evaluate the efficacy of the model and algorithm proposed herein. The analysis and discussion are based on the optimization outcomes. Given the complexity of the research problem, small-scale instances were resolved using the Gurobi solver, while large-scale cases were addressed through the simulated annealing genetic algorithm, with the results of the smaller instances compared accordingly. The experiments were executed on a personal computer equipped with an Intel Core i5-9300H CPU running at 2.40 GHz and 8.00 GB of RAM. The simulated annealing genetic algorithm was coded in Python 3.9, with Gurobi 11.0.3 selected as the solver.

5.1. Instance Generation

5.1.1. Parameters' Setting

The time allocated for handling each container's loading and unloading is set at one minute. The cost of flight delay is established at CNY 3000 per minute, and the operating cost of the transshipment center is fixed at CNY 200 per minute. To make the contrasts in the results more pronounced, the flight intervals are assumed to be as narrow as possible.

In the context of the genetic algorithm, the population size is designated as 100, with 1000 iterations, and a mutation probability of 0.01.

The initial temperature for the embedded simulated annealing local search algorithm is set at 10,000 degrees Celsius, with a cooling rate of 0.99, and the minimum temperature is established at 1 degree Celsius.

5.1.2. Test Cases

A small-scale experiment was conducted using a scenario where two AGVs and 20 sets of container tasks from three flights were considered. Each flight has different arrival and departure times. Taking into account the time required for loading and unloading containers, the feasible handling time is adjusted based on the actual time. The left boundary of the time interval is set 15 min after the flight arrives at the apron, while the right boundary is 20 min before the flight's departure. As described in Section 3.1, the tables containing the known information for this small-scale case are provided in Tables A1–A3.

Using the model developed in this study and the known data, the experiment utilized Gurobi to obtain an exact solution. The results indicate an objective value of CNY 250,400, and the specific scheduling plan is as follows (Table 1):

Table 1. Gurobi exact scheduling results.

Container ID	Transport AGV	Start Transport Time	Container ID	Transport AGV	Start Transport Time
i01	j01	1:58	i11	j02	1:31
i02	j01	1:32	i12	j02	1:26
i03	j02	1:02	i13	j02	1:21
i04	j02	1:35	i14	j02	1:15
i05	j02	1:46	i15	j01	1:08
i06	j01	1:02	i16	j01	1:37
i07	j01	1:22	i17	j02	1:53
i08	j01	1:15	i18	j01	1:43
i09	j02	1:07	i19	j01	1:50
i10	j01	1:27	i20	j02	1:41

5.2. Small-Scale Experiments

Scenarios with fewer than 30 containers are classified as small-scale cases. To evaluate the effectiveness of the algorithm designed in this study, 10 small-scale experiments were conducted, solved using both the Gurobi and the genetic algorithm. Each algorithm was run five times, with the average value of the results taken. Although this research considers the weight of the AGVs and the containers, this section primarily aims to compare the performance of the two algorithms under identical settings, allowing for random variations in container weights and AGV load capacities. According to the relevant information [27–29], combined with the scale of the experiments used in this paper, as well as the needs of the actual operating environment, set the upper bound of the calculation time of Gurobi as 300 s, increase the number of containers uniformly, randomly select different groups to the number of AGVs, and set it to be the largest scale of the small-scale example when the calculation time of Gurobi for the further expansion of the example exceeds the effective calculation time. The results of the small-scale experiments are presented in Table 2.

Table 2. Small-scale experimental results.

Scale		Gurobi		GA		F1
Containers' QTY	AGVs' QTY	Obj Value	Computing Time (min)	Obj Value	Computing Time (min)	
10	2	6800	2	7000	1.06	0.029
15	2	39,400	8	44,200	2.08	0.122
20	2	250,400	43	305,800	4.01	0.221
	3	11,800	36	13,400	3.09	0.136
25	2	682,200	146	728,400	2.10	0.068
	3	14,000	183	17,800	1.81	0.271
	4	8400	47	10,200	3.12	0.214
30	2	1,394,600	190	1,798,800	2.03	0.290
	3	\	\	767,400	4.32	\
	4	91,000	217	102,400	3.12	0.125

F1 = (Target value of GA – Target value of Gurobi)/Target value of Gurobi.

Based on the results in the table, it is evident that when the number of AGVs remains constant and only the number of containers is increased, the objective value grows, and Gurobi's solving time exhibits exponential growth. When the number of containers remains constant and the number of AGVs is increased, the objective value decreases, though Gurobi's solving time may either increase or decrease. The pattern is not entirely evident from the experiments conducted. When the data scale exceeds a certain threshold, Gurobi's computation time surpasses the preset upper limit, as shown in the table when the number of AGVs is three and the number of containers is 30. Compared to Gurobi, the genetic

algorithm yields a difference in objective values within 30%, while in any scenario, the genetic algorithm's computation time is significantly shorter, becoming more pronounced as the scale increases.

5.3. Large-Scale Experiments

Based on actual operational requirements, the scale of containers was adjusted. In this experiment, 35 containers and three AGVs were randomly selected as the minimum experimental data for large-scale operations. Subsequently, the number of containers was gradually increased by five units. For a fixed number of containers, the number of AGV groups was randomly selected until 10 experimental groups were obtained. These large-scale experiments were solved using both genetic algorithms (GA) and simulated annealing genetic algorithms (SAGA). Each scenario was computed five times, recording the minimum, maximum, and average values, along with the average computational time for both algorithms, which are displayed in Table 3.

Table 3. Large-scale experimental results.

Scale Containers' QTY	AGVs' QTY	Max Obj Value	GA Mean Obj Value	Computing Time (min)	Min Obj Value	Max Obj Value	Mean Obj Value	SAGA Min Obj Value	Computing Time (min)	H1	H2	H3
35	3	1,539,200	1,427,040	3.26	1,337,000	1,478,200	1,283,320	1,133,200	6.54	0.041	0.112	0.180
40	3	2,322,000	2,207,360	5.13	2,022,400	2,318,400	2,132,840	1,976,200	7.13	0.002	0.035	0.023
	4	933,400	826,720	4.39	733,400	844,200	731,800	611,200	6.87	0.106	0.130	0.200
	5	552,000	476,440	4.68	411,400	497,400	413,880	356,200	6.52	0.110	0.151	0.155
45	3	3,532,400	3,162,620	4.53	2,827,800	3,116,200	2,927,200	2,756,200	7.09	0.134	0.080	0.026
	4	1,719,600	1,618,800	3.97	1,538,200	1,538,800	1,475,440	1,402,400	7.23	0.117	0.097	0.097
	5	931,600	812,220	4.26	728,400	797,200	718,420	608,200	7.54	0.169	0.131	0.198
50	6	641,200	524,640	4.33	468,200	531,800	461,240	395,600	6.99	0.206	0.137	0.184
	7	5,072,000	4,607,520	4.26	4,407,200	4,373,800	3,949,120	3,644,200	7.15	0.160	0.167	0.209
	8	2,520,000	2,440,160	5.18	2,364,400	2,234,200	1,997,320	1,742,200	7.22	0.128	0.222	0.357

H1 = (GA's maximum – SAGA's maximum)/SAGA's maximum. H2 = (GA's mean – SAGA's mean)/SAGA's mean. H3 = (GA's minimum – SAGA's minimum)/SAGA's minimum.

In the small-scale experiments, the Gurobi solver's computation time exceeded the effective time limit as the problem instance size increased. The computation times of GA and SAGA were significantly less than the set effective time, with GA exhibiting notably shorter computation times compared to SAGA. However, the solution quality of SAGA was better than that of GA, a trend that is also evident in the large-scale experiments. According to the aforementioned large-scale results, we observe that the average processing time of SAGA is slightly longer than that of GA, yet the processing time remains well below the set effective time limit, indicating SAGA's superior time performance. Furthermore, by examining the objective values and H values, it can be seen that the three objective results—maximum, minimum, and average—from five runs of SAGA are all better than those of GA, demonstrating the feasibility and accuracy of the SAGA algorithm. Additionally, both algorithms indicate that, when the number of AGVs is constant, an increase in the number of containers leads to a higher objective value. Conversely, when the number of containers is fixed, increasing the number of AGVs results in a decrease in the objective value. Through large-scale experiments and effective analysis, it is further demonstrated that the model exhibits strong robustness and stability in the scheduling process under different AGV load conditions, task numbers, and time windows. Therefore, as the scheduling scale expands, the model's scalability is capable of supporting more complex scenarios, including hundreds of AGVs and more intricate cargo transportation needs.

5.4. Analysis and Discussions

5.4.1. The Sensitivity Analysis of AGV Load Bearing

For the identical container handling tasks, the payload capacity of AGVs influences the outcomes of container assignment. The results were computed for the varying payload capacities of AGVs, with the number of containers set at 20, 30, and 40, respectively. Each scenario was calculated five times, and the average values were taken as the final result. Figure 6 illustrates that, when handling the same number of container tasks, increasing

the AGV's payload capacity can significantly reduce the completion time. This is because aviation containers, unlike standard containers, have different shapes and weights. A higher payload capacity for AGVs means that a greater variety of AGV types can handle each container, resulting in a more balanced number of containers handled by each AGV and reduced handling time. Consequently, containers can be moved within an effective timeframe, reducing aircraft-delay costs. However, once the AGV's payload capacity reaches a level that can accommodate all containers with varying shapes, further increases in payload capacity will not significantly improve the task completion time, as payload limitations are no longer the bottleneck. On the other hand, increasing the AGV's payload also raises costs. This implies that there is an optimal theoretical payload capacity that balances operational needs with reasonable cost expenditure. As shown in the figure below, the curve results for different payload capacities of [40,45,50] do not vary significantly for the same number of containers. Determining the specific optimal capacity requires further analysis in conjunction with actual manufacturing costs.

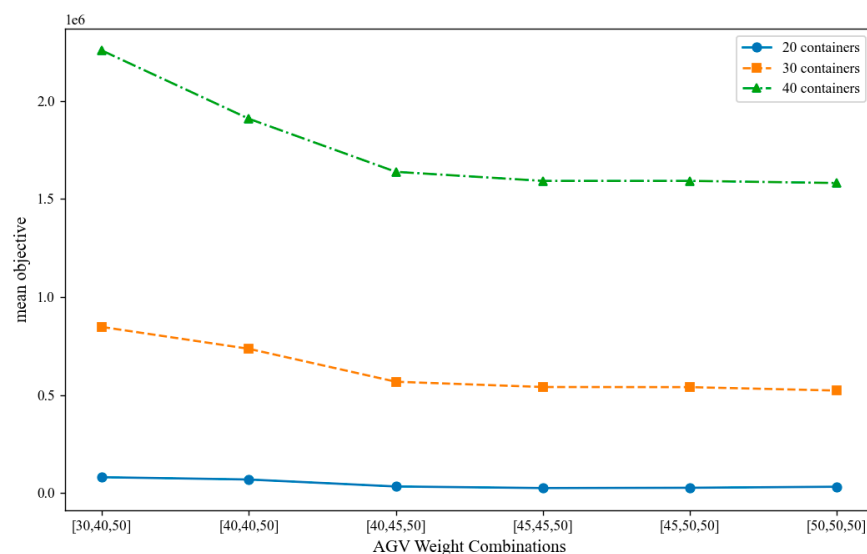


Figure 6. Results for different AGV load capacities.

5.4.2. The Sensitivity Analysis of the Number of AGVs

In scheduling tasks, the number of AGVs plays a crucial role in task allocation. Calculate the objective values for different numbers of AGVs when the number of container groups is 20, 30, and 40. Given the varying load capacities of the AGVs in this study, we will consider increasing the number of AGVs while also increasing their load capacity to the average of the original AGV capacities. Specifically, assuming that the load capacities of the first two AGVs are 40 kg and 50 kg, respectively, all additional containers will have a load capacity of 45 kg. Each scenario will be computed five times, with the average value taken as the result (Figure 7).

From the results above, we can observe that, when the number of container tasks remains constant, increasing the number of AGVs initially causes a significant decrease in the target value, leading to a substantial improvement in system performance. This is because more AGVs can distribute the tasks, reducing the workload on each individual AGV and, consequently, shortening the time required to complete all tasks. However, once the number of AGVs reaches a certain threshold, the marginal benefit of adding additional AGVs diminishes, meaning each additional AGV contributes less to the overall performance improvement. On one hand, the curve showing the reduction in the target value resembles a decaying exponential curve. As the number of AGVs increases, the extent of the reduction in the target value decreases until, when the ratio of AGVs to containers is one, the curve levels off, indicating that adding more AGVs contributes nothing further. On the other hand, the cost of using AGVs must also be considered. The more AGVs used,

the higher the cost. Due to limitations in scope and capacity, this aspect is not explored further here.

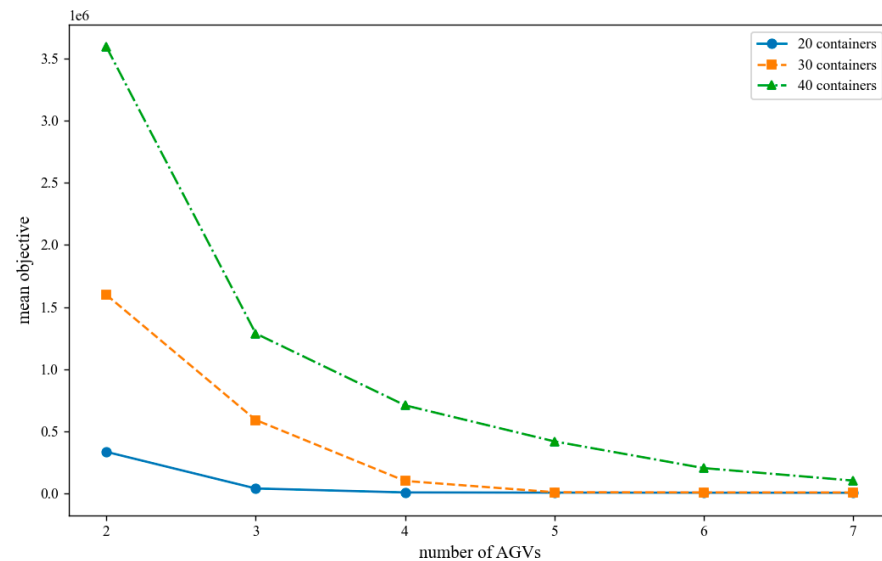


Figure 7. Results for different numbers of AGVs.

5.4.3. The Sensitivity Analysis of Transportable Time Intervals

For the scenario involving two AGV tasks, we consider modifying the feasible time intervals. We analyze six cases, namely $\{[2,30] [5,35]\}$, $\{[2,35] [5,35]\}$, $\{[2,40] [5,35]\}$, $\{[2,40] [5,40]\}$, $\{[2,40] [5,45]\}$, and $\{[2,45] [5,45]\}$. To present the results in detail, we focus on the case with 20 containers and compute each scenario five times using the average as the result. Research shows that, as the feasible processing time interval expands, the target value decreases. However, when the time interval is further extended, the rate of decrease in the target value gradually diminishes until it eventually becomes zero. This indicates that relaxing time constraints within a certain range can improve scheduling efficiency, but beyond a certain limit, the potential for efficiency gains becomes limited and eventually reaches zero. We can also conclude that, in practice, for tasks with the same level of urgency, strategies such as prioritizing tasks with smaller time windows and using heuristic algorithms can be employed to optimize these time intervals (Figure 8).

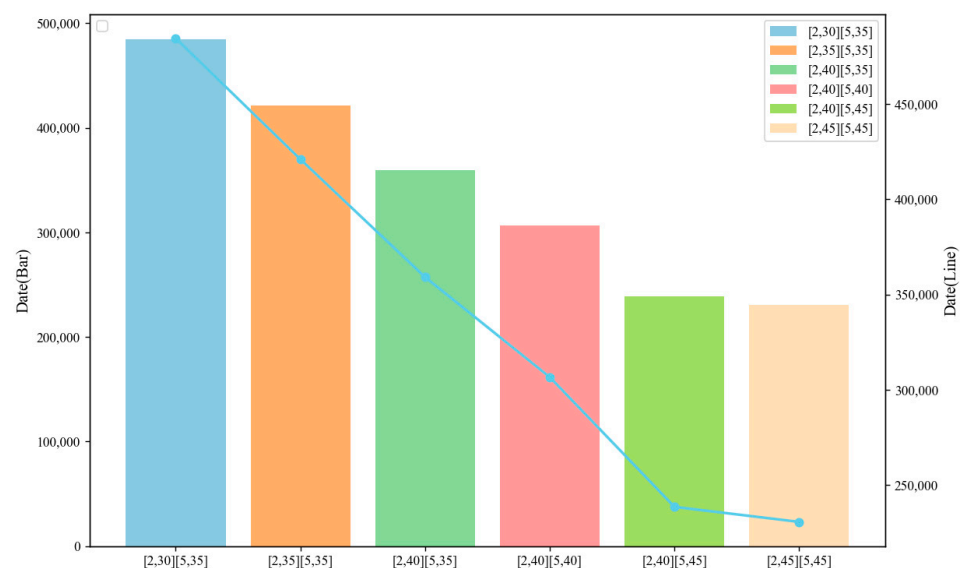


Figure 8. Results for different handling ranges.

6. Conclusions

The efficient scheduling of AGVs is pivotal to the smooth operation of airport transfer centers. This paper thoroughly investigates the scheduling issues of AGVs within such centers. Upon the arrival of a flight at the apron, the focus is on the scheduling of weight-bearing AGVs within the time window between container unloading and loading. In developing the model, considerations were given to the weight constraints of the AGVs and the time interval constraints for single AGV transporting containers. The goal is to minimize the aircraft-delay time and the makespan, thereby reducing both the delay costs and the operational costs of the transfer center. A heuristic algorithm, based on an improved genetic algorithm, was designed to solve the task allocation problem for AGV container transportation. In small-scale experiments, the genetic algorithm was compared with the commercial solver Gurobi, and the results indicate that the genetic algorithm can produce acceptable schedules in a short time. For large-scale scenarios, due to time constraints, the Gurobi solver becomes impractical, and a comparative experiment between the genetic algorithm and the simulated annealing genetic algorithm designed in this study demonstrates that the latter provides superior results.

Through the application of a mixed-integer programming model and an improved genetic algorithm, this study not only provides new theoretical insights but also demonstrates significant advantages in practical operations. In actual airport transfer operations, the proposed model helps address the large-scale irregular container scheduling problem by optimizing AGV scheduling. This approach significantly reduces aircraft delays and operational costs, improving the overall performance of the transfer center. Future research could consider integrating container-loading issues within time windows and container palletizing problems to further enhance the comprehensive performance of airport transfer centers. Additionally, with advancements in technology and the development of new algorithms, incorporating real-time optimization and dynamic scheduling to handle unexpected events using real-time data, as well as integrating cutting-edge intelligent algorithms such as deep learning and reinforcement learning to overcome current technological limitations, will be key to improving aviation logistics efficiency. The methods developed in this study have broad applicability beyond airport transfer centers and can be extended to other logistics and supply chain scenarios. In particular, the AGV scheduling optimization methods developed in this study are applicable to any logistics scenario involving automated handling equipment. They perform exceptionally well in managing large-scale, multi-constraint complex scenarios, such as warehousing and sorting centers, and can be applied to optimization scheduling across various logistics nodes, having a broader impact on the global freight logistics field. Although the SAGA algorithm proposed in this paper performs well, limitations such as computation time may affect the effectiveness of the proposed method. Future research could explore optimization under better computational conditions and enhance hardware computing capabilities. Additionally, introducing adaptive mechanisms could lead to further innovations in the algorithm.

Author Contributions: Writing—methodology and original draft preparation, J.L.; validation, M.Z. review and supervision, Y.L.; resources, D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: Author Di Sun was employed by the S.F. Express Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

Table A1. Case study flight information.

Sequence Number	Flight Identification	Arrival Time	Departure Time	AGV Feasible Handling Time Window	Unloading Container Identification	Loading Container Identification
1	f01	00:47	2:02	[1:02,1:42]	i01\i02\i03	i04\i05\i06
2	F02	00:50	1:56	[1:05,1:36]	i07\i08\i09\i10\i11	i12\i13\i14\i15
3	F03	00:54	2:03	[1:09,1:43]	i16\i17\i18	i19\i20

Table A2. Container weight and handling time information.

Container Identification	Transportation Time (min)	Weight (kg)	Container Identification	Transportation Time (min)	Weight (kg)
i01	8	35	i11	3	45
i02	4	20	i12	4	40
i03	4	45	i13	4	45
i04	5	40	i14	5	30
i05	6	45	i15	6	25
i06	5	30	i16	5	30
i07	4	25	i17	7	50
i08	6	30	i18	6	25
i09	7	50	i19	7	20
i10	4	25	i20	4	45

Table A3. AGV load capacity data.

AGV Identification	Load Capacity (kg)
j01	35
j02	50

References

1. Van Bockstaele, V.; Buyle, S.; Dewulf, W. Solving the mystery of discrepancies and double counting in air cargo through demand and supply big data analysis. *J. Air Transp. Res. Soc.* **2023**, *1*, 81–100. [\[CrossRef\]](#)
2. Cheung, T.K.Y.; Wong, C.W.H.; Lei, Z. Assessment of hub airports' connectivity and Self-Connection Potentials. *Transp. Policy* **2022**, *127*, 250–259. [\[CrossRef\]](#)
3. Tang, C.-H.; Yen, C.-Y. Airline unit load device dispatching considering service level and violation days. *J. Air Transp. Manag.* **2019**, *79*, 101685. [\[CrossRef\]](#)
4. Alonso, M.T.; Alvarez-Valdes, R.; Iori, M.; Parreño, F.; Tamarit, J. Mathematical models for multicontainer loading problems. *Omega* **2017**, *66 Pt A*, 106–117. [\[CrossRef\]](#)
5. Bai, Y.; Lv, Y.; Zhang, J. Smart mobile robot fleet management based on hierarchical multi-agent deep Q network towards intelligent manufacturing. *Eng. Appl. Artif. Intell.* **2023**, *124*, 106534. [\[CrossRef\]](#)
6. Efegan, V.; Temiz, İ. Changes in efficiency and physical size of container ports: An integration of genetic matching and stochastic data envelopment analysis. *Res. Transp. Bus. Manag.* **2024**, *54*, 101125. [\[CrossRef\]](#)
7. Huang, K.; Lee, Y.-T.; Xu, H. A routing and consolidation decision model for containerized air-land intermodal operations. *Comput. Ind. Eng.* **2020**, *141*, 106299. [\[CrossRef\]](#)
8. Lurkin, V.; Schyns, M. The Airline Container Loading Problem with pickup and delivery. *Eur. J. Oper. Res.* **2015**, *244*, 955–965. [\[CrossRef\]](#)
9. Lu, H.-A.; Chen, C.-Y. A time-space network model for unit load device stock planning in international airline services. *J. Air Transp. Manag.* **2011**, *17*, 94–100. [\[CrossRef\]](#)
10. Brandt, F.; Nickel, S. The air cargo load planning problem—a consolidated problem definition and literature review on related problems. *Eur. J. Oper. Res.* **2019**, *275*, 399–410. [\[CrossRef\]](#)
11. Chan, F.T.S.; Bhagwat, R.; Kumar, N.; Tiwari, M.K.; Lam, P. Development of a decision support system for air-cargo pallets loading problem: A case study. *Expert Syst. Appl.* **2006**, *31*, 472–485. [\[CrossRef\]](#)
12. Jamrus, T.; Chien, C.-F. Extended priority-based hybrid genetic algorithm for the less-than-container loading problem. *Comput. Ind. Eng.* **2016**, *96*, 227–236. [\[CrossRef\]](#)
13. Tang, C.-H. A scenario decomposition-genetic algorithm method for solving stochastic air cargo container loading problems. *Transp. Res. Part E Logist. Transp. Rev.* **2011**, *47*, 520–531. [\[CrossRef\]](#)

14. Zhang, R.; Yun, W.Y.; Moon, I.K. Modeling and optimization of a container drayage problem with resource constraints. *Int. J. Prod. Econ.* **2011**, *133*, 351–359. [[CrossRef](#)]
15. Chen, X.; He, S.; Zhang, Y.; Tong, L.; Shang, P.; Zhou, X. Yard crane and AGV scheduling in automated container terminal: A multi-robot task allocation framework. *Transp. Res. Part C Emerg. Technol.* **2020**, *114*, 241–271. [[CrossRef](#)]
16. Zaghdoud, R.; Mesghouni, K.; Dutilleul, S.C.; Zidi, K.; Ghedira, K. A Hybrid Method for Assigning Containers to AGVs in Container Terminal. *IFAC-PapersOnLine* **2016**, *49*, 96–103. [[CrossRef](#)]
17. Cao, Y.; Yang, A.; Liu, Y.; Zeng, Q.; Chen, Q. AGV dispatching and bidirectional conflict-free routing problem in automated container terminal. *Comput. Ind. Eng.* **2023**, *184*, 109611. [[CrossRef](#)]
18. Wang, Y.-Z.; Hu, Z.-H.; Tian, X.-D. Scheduling ASC and AGV considering direct, buffer, and hybrid modes for transferring containers. *Comput. Oper. Res.* **2024**, *161*, 106419. [[CrossRef](#)]
19. Han, X.; Cheng, W.; Meng, L.; Zhang, B.; Gao, K.; Zhang, C.; Duan, P. A dual population collaborative genetic algorithm for solving flexible job shop scheduling problem with AGV. *Swarm Evol. Comput.* **2024**, *86*, 101538. [[CrossRef](#)]
20. Li, W.; Li, H.; Wang, Y.; Han, Y. Optimizing flexible job shop scheduling with automated guided vehicles using a multi-strategy-driven genetic algorithm. *Egypt. Inform. J.* **2024**, *25*, 100437. [[CrossRef](#)]
21. Yang, X.-S. Chapter 5-Simulated Annealing. In *Nature-Inspired Optimization Algorithms Biology*, 2nd ed.; Yang, X.-S., Ed.; Academic Press: Cambridge, MA, USA, 2021; pp. 83–90.
22. Tsai, C.-W.; Chiang, M.-C. Uncertainty, Computational Techniques, and Decision Intelligence. In *Handbook of Metaheuristic Algorithms*; Tsai, C.-W., Chiang, M.-C., Eds.; Academic Press: Cambridge, MA, USA, 2023; Chapter Seven-Genetic; pp. 111–138.
23. Tian, P.; Ma, J.; Zhang, D.-M. Application of the simulated annealing algorithm to the combinatorial optimisation problem with permutation property: An investigation of generation mechanism. *Eur. J. Oper. Res.* **1999**, *118*, 81–94. [[CrossRef](#)]
24. Huang, Y.; Sheng, B.; Fu, G.; Luo, R.; Lu, Y. Multi-objective simulated annealing algorithm for robotic mixed-model two-sided assembly line balancing with setup times and multiple constraints. *Appl. Soft Comput.* **2024**, *156*, 111507. [[CrossRef](#)]
25. Wang, Z.; Xu, H.; Wang, Y. Core flow distribution optimization of a natural circulation reactor using genetic algorithm, simulated annealing and characteristic statistic algorithm. *Prog. Nucl. Energy* **2023**, *165*, 104904. [[CrossRef](#)]
26. Rolf, B.; Reggelin, T.; Nahhas, A.; Lang, S.; Müller, M. Assigning dispatching rules using a genetic algorithm to solve a hybrid flow shop scheduling problem. *Procedia Manuf.* **2020**, *42*, 442–449. [[CrossRef](#)]
27. Meng, F.; Chu, D.; Li, K.; Zhou, X. Multiple-class multidimensional knapsack optimisation problem and its solution approaches. *Knowl. -Based Syst.* **2019**, *166*, 1–17. [[CrossRef](#)]
28. Zhu, Z.; Chen, Y.; Wahab, M. An exact algorithm for simultaneous pickup and delivery problem with split demand and time windows. *Comput. Oper. Res.* **2024**, *170*, 106761. [[CrossRef](#)]
29. Yang, K.; Wang, R.; He, H.; Yang, X.; Zhang, G. Multi-supply multi-capacitated p-median location optimization via a hybrid bi-level intelligent algorithm. *Comput. Ind. Eng.* **2021**, *160*, 107584. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.