Transportation Research Part E 197 (2025) 104066

Contents lists available at ScienceDirect

Transportation Research Part E

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

Less-than-container cargo scheduling for China Railway Express along belt and road initiative routes[☆]

Yanjie Zhou[a], Zhanwen He[a], Chengcheng Liu[a,*], Jingrong Zhang[a], Yumin Li[a], Yan Wang[b]

[a] School of Management, Zhengzhou University, Zhengzhou 450001, China
[b] Shanghai Zhenhua Heavy Industries Company Limited, 200125, Shanghai, China

ARTICLE INFO

Keywords:
China Railway Express
Less-than-container-load cargo
Optimization
Hybrid genetic algorithm
Iterative local search

ABSTRACT

With the rapid expansion of global trade, the use of LCL(Less than container load) transportation in international trade is becoming increasingly widespread. This study explores the application of LCL transportation in the context of China Railway Express (CR Express). Addressing the challenges of low cargo loading efficiency and complex container scheduling in CR Express LCL services, we aim to maximize customer satisfaction and develop a multi-objective mixed-integer programming model. The model aims to minimize the number of containers used and the maximum transportation time. To effectively tackle large-scale instances, we have designed an efficient genetic algorithm enhanced with an iterative local search (ILS-GA). Computational experiments across small, medium, and large instances reveal that ILS-GA identifies optimal solutions in small-scale instances. ILS-GA discovers the optimal solution within an average runtime of 5.45 s, which is 95.56% faster than CPLEX's 180 s, demonstrating its high solution efficiency. In medium and large instances, compared to CPLEX and SA, ILS-GA provides better solutions with higher computational efficiency, significantly outperforming the SA algorithm in terms of global search capability and optimization efficiency. Additionally, we analyze the initialization and local iterative search strategies through experiments, verifying the proposed strategies' effectiveness in improving the ILS-GA solutions.

1. Introduction

In recent years, with the development of cross-border e-commerce (Wang et al., 2020), transportation problems play an important role in the field of logistics (Wang et al., 2022b). The rapid growth of cross-border e-commerce has made China Railway Express (CR Express) a key facilitator in developing foreign trade (Li et al., 2023). According to China State Railway Group, in 2024, CR Express operated 16,145 trains and transported 1,749,000 TEUs of cargo, marking a 7% and 19% year-on-year increase, respectively.[1] The cargo types have expanded to 53 categories, encompassing over 50,000 kinds of commodities, with Less-than-container-load (LCL) cargoes from small and medium-sized enterprises (SMEs) constituting a growing share. As the LCL cargo customer base grows, CR Express operators have introduced a more diversified service: LCL transportation. Unlike Full-container-load (FCL) transportation,

☆ This paper has been selected as the Best Paper at the International Joint Conference of the 6th Yangtze-River Research and Innovation Bele (Y-RIB) and UNESCAP held on 7th June – 8th June, 2024 in Zhejiang University, Zhoushan, China.

* Corresponding author.Chengcheng Liu, chengyulcc@gs.zzu.edu.cn
E-mail addresses: ieyjzhou@zzu.edu.cn (Y. Zhou), 13526676947@163.com (Z. He), chengyulcc@gs.zzu.edu.cn (C. Liu), zjr@zzu.edu.cn (J. Zhang), liyumin@zzu.edu.cn (Y. Li), wangyan_wy@zpmc.com (Y. Wang).

1 These data are provided by the http://www.china-railway.com.cn/.

https://doi.org/10.1016/j.tre.2025.104066
Received 24 January 2025; Received in revised form 3 March 2025; Accepted 4 March 2025
Available online 18 March 2025

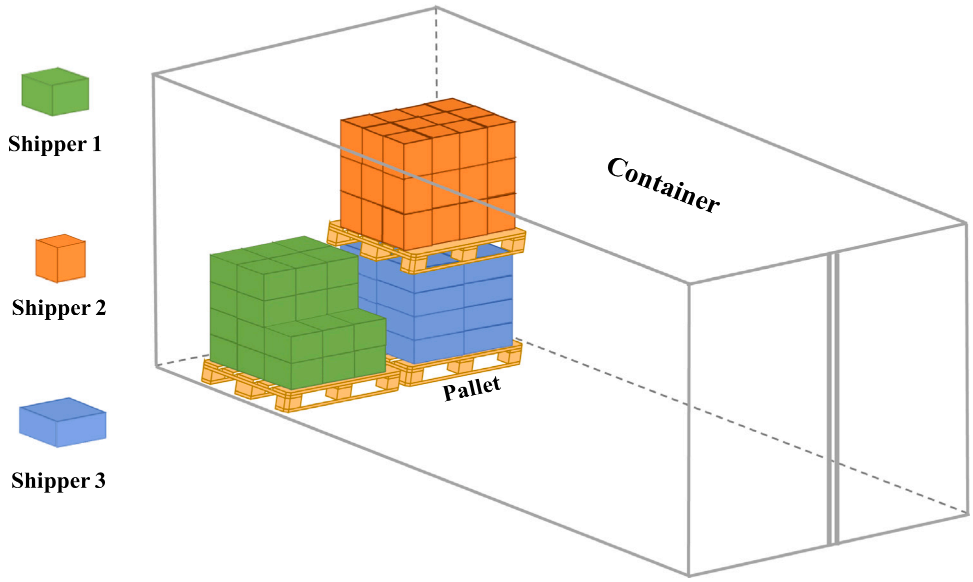


Fig. 1. Example of less-than-container loading (LCL).

LCL transportation allows operators to consolidate cargoes from various shippers into a single container to realize container space sharing (as shown in Fig. 1). Consequently, cargo owners are charged only for the space they use, which enhances container space utilization and significantly reduces shippers' transportation costs.

As demand for LCL cargo transportation increases (Tiwari et al., 2021), the LCL business of CR Express is experiencing rapid growth. Fig. 2 shows the detailed process of the LCL business of Zhengzhou International Hub Development and Construction Co., Ltd.(ZIH). CR Express operators collect LCL cargo from various shippers at the warehouse, where staff categorizes them by shipper and destination. After sorting, cargoes with the same owner and destination are palletized, loaded into containers, and scheduled for dispatch within a rational cycle. However, as the LCL cargo customer base expands and business volume grows, the reliance on manual labor for tasks such as sorting, packing, container loading, and scheduling shipping cycles has become increasingly complex and challenging. To enhance the transport efficiency of LCL operations, CR Express operators must solve two core problems. Firstly, they must determine how to classify, palletize, and load cargo from various shippers and destinations to maximize loading efficiency, which is usually considered the 3D loading problem. Second, given the limited number of LCL containers per dispatch cycle, container dispatch cycles must be arranged rationally based on the timeliness requirements of various shippers, which can be interpreted as a container scheduling problem. Together, these challenges comprise a comprehensive optimization issue for CR Express's LCL operations. However, whether it is the loading strategy of LCL cargo or the scheduling arrangement of containers, achieving the optimal solution for the comprehensive optimization problem among different shippers, destinations, cargo volumes, sizes, and timeliness requirements is challenging with current manual experience and intuitive judgment alone. Consequently, there is an urgent need for CR Express operators to develop efficient mathematical models and algorithms to enhance the operational efficiency of the LCL business.

Currently, in research on the LCL problem, scholars usually pay more attention to enhancing container space utilization (Jamrus and Chien, 2016), with scant attention given to the specific application scenarios of LCL and the resolution of issues arising within those scenarios. This study focuses on the cargo loading and container scheduling issues within CR Express's LCL business scenario, aiming to enhance the operational efficiency of the CR Express operators' entire LCL process (including cargo sorting - loading - transportation, excluding the distribution stage after arriving at the destination). Specifically, the contribution points of this study include the following. (1) New scenario construction and problem definition for LCL problems. Based on the CR Express LCL transportation business scenario, this study, for the first time, proposes two core issues railway operators face in the two aspects of LCL cargo loading and container scheduling. (2) Construction of a multi-objective mixed integer programming model. A multi-objective mixed integer programming model that aims to minimize the number of containers used and minimize the maximum cargo transportation time was established to decide the specific location of each shipper's cargo within pallets and containers and the container dispatch cycle. (3) Designed and implemented an improved genetic algorithm. A genetic algorithm (ILS-GA) with iterative local search was customized for the comprehensive optimization problem of cargo consolidation. A two-stage encoding and decoding scheme was tailored for LCL cargo's pallet loading and container loading. The search efficiency and solution quality were significantly improved by designing operators and incorporating iterative local search algorithms to expand the search space. (4)

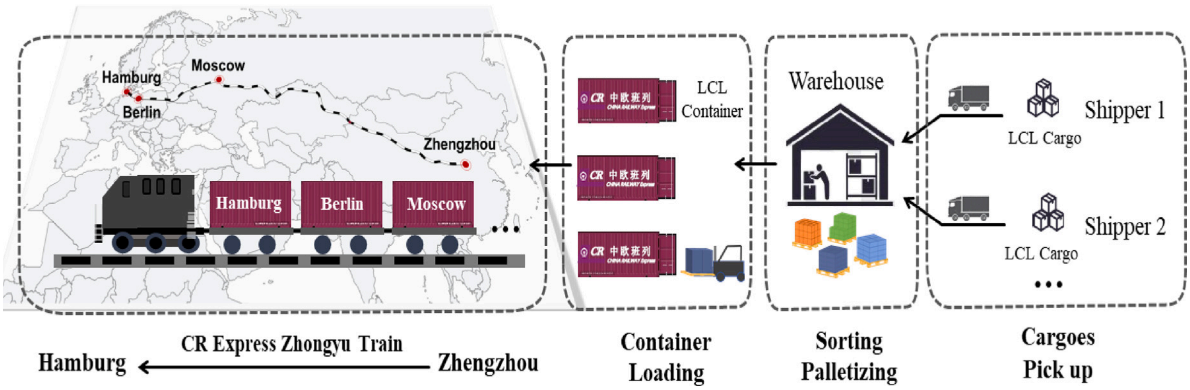


Fig. 2. Operation process of CR Express Zhongyu Train LCL.

Instances analysis of model effectiveness and algorithm performance. The effectiveness of the mixed integer programming model and the efficiency of the algorithms developed in this study were verified through small-scale, medium-scale, and large-scale instances. The results show that the algorithms outperformed the CPLEX solver and traditional simulated annealing algorithms regarding solution efficiency.

The remainder of this paper is organized as follows. Section 2 shows the literature review. Section 3.1 presents the studied problem. The mathematical model is presented in Section 3. Section 4 proposes a two-stage LCL loading method and genetic algorithm. Section 5 shows the experimental results. Finally, Section 6 concludes this paper.

2. Literature review

2.1. Related research

Due to its flexibility and cost-effectiveness, the Less-than-Container Load (LCL) transportation mode is often favored by small and medium-sized enterprises. It has been widely applied in international trade (Wei et al., 2019). Xiao (2011) explored the role of LCL transportation in international trade and evaluated the conditions under which foreign trade containers choose LCL transportation. Currently, research on LCL business is relatively scarce and mostly focuses on LCL transportation services at ports. For instance, addressing the issue of low efficiency in the LCL business process at port container terminals, (Tan et al., 2018) suggested using blockchain to establish a LCL export platform (LEP) to optimize the LCL business process. Regarding the problem of low efficiency in LCL transshipment at port container intermodal terminals, (Wei et al., 2019) discussed the impact of LCL cargo transshipment efficiency on customer satisfaction. This study focuses on the scenario of LCL transportation at inland international land ports, providing specific transportation scheduling solutions for the issues of low cargo loading efficiency and container scheduling in the LCL business of China Railway Express.

Regarding the bin packing problem, especially the three-dimensional bin packing problem (Wang et al., 2022a), container loading is an important subfield (Zhao et al., 2016). For the loading problem of a single container, (Huang and He, 2007) designed a new algorithm using the principle of "maximum collapse", which improves the tightness of item placement and achieves an average container capacity utilization of 94.6%. Upadhyay et al. (2017) expanded the study to the more complex case of double-deck containers, and proposed a mathematical model that takes into account the constraints of containers of different types, weights, and heights from the perspective of an exact solution, and conducted a numerical case study on a train operator in India. The results showed that their mathematical model could reduce the transportation cost by 3%. For the 3D container loading problem, (Deidson Vitorio Kurpel and others, 2020) also proposed a mathematical model considering box orientation, stability, and cargo separation constraints from an exact solution point of view and validated it with well-known benchmark instances. The results show that the proposed new mathematical model improves the results of the known instances. Similarly, (Junqueira et al., 2012) also proposed mixed-integer linear programming models to optimize the container loading rate. However, since the container loading problem is a typical high-complexity (NP-hard) problem, the exact solution methods face scalability limitations. Therefore, some scholars have investigated the use of heuristic methods to solve these container loading problems (Iwasawa et al., 2016; Zhou and Liu, 2017; Alonso et al., 2014). However, the performance and effectiveness of these heuristics are difficult to prove. To solve this problem, this study proposes an exact method for solving small and medium-sized instances and customizes the GA, incorporating an iterative local search strategy to deal with large-sized instances; at the same time, the exact solution can also verify the validity of the algorithm results.

Currently, in railway transportation, to improve loading and transportation efficiency, pallet loading is a more common method (Zhou, 2018; Wu et al., 2022). To address the secondary pallet loading problem, (Moura and Bortfeldt, 2017) considers loading goods onto pallets as the first stage and placing pallets into containers as the second stage. These two stages of loading are treated as independent processes, each solved using different methods. The research mentioned above treats the two-stage loading with pallets as separate entities, which deviates from real-world needs and significantly simplifies the problem. To remedy this deficiency, this study designs a two-stage loading model for LCL cargo. It develops customized heuristic methods to optimize the entire loading process, resulting in loading solutions that are more aligned with practical requirements.

Unlike the standard container loading problem, LCL loading requires consideration of the different cargo requirements of various shippers during the actual loading and transportation process, to optimize the utilization of container space from different shippers. There is relatively little research on LCL loading that addresses practical needs. To our knowledge, the only attempt to explore this issue is Jamrus and Chien (2016), which aims to maximize the utilization of LCL container space. They designed an extended priority-based hybrid genetic algorithm and verified its practical feasibility through instances. However, their research limits the problem to LCL loading within a single container. This study further extends the work of Jamrus et al. considering the LCL loading problem involving multiple containers.

Multi-container loading aims to load cargoes into the minimum number of containers while satisfying various types of constraints (Alonso et al., 2019). In the research on loading multiple containers, (Alonso et al., 2019) established an integer linear programming model to minimize the number of containers used. The results from real-world examples indicate that their proposed model can obtain optimal solutions in most cases. Similarly, (Alonso et al., 2020) optimized the multi-container loading problem by designing a greedy random adaptive search procedure to minimize the number of containers used. Che et al. (2011) transformed the objective of minimizing the number of containers used into minimizing the cost of multi-container loading and used a linear integer programming method to solve it. However, considering LCL transportation, in addition to the multi-container loading problem, LCL transportation often impacts the timeliness of cargo. In the transportation of the China Railway Express, due to the limited number of LCL containers available within each dispatch cycle, cargoes from different shippers heading to different destinations will face varying waiting times. To improve the service satisfaction of shippers, we introduce another optimization objective: minimizing the maximum transportation time of cargoes, thereby controlling the overall transportation time. This study also provides a scheduling optimization solution for multiple containers, which further extends the current research on multi-container loading problems.

2.2. Research gap and contribution

This study focuses on a new research scenario for LCL transportation: the CR Express, further expanding the research issues and scenarios in LCL transportation. Specifically, the contributions of this study are as follows:

(1) Addressing the issue of low loading efficiency in LCL cargo operations on the CR Express, optimizes the two-stage loading of LCL cargo.

(2) To enhance shippers' overall satisfaction, a time optimization objective function was introduced. This resulted in the design of a multi-container scheduling optimization scheme, filling the gap in research on multi-container scheduling in LCL transportation.

(3) Exact methods for solving small and medium-sized instances and customized heuristic methods for handling large-scale instances were designed.

Specifically, the comparison of this study with other current research is shown in Table 1.

3. Mathematical model

3.1. Problem statement

As the name implies, LCL transportation consolidates cargo from various shippers into a single container for shipping. However, this process entails numerous business operations. As each CR Express container has a predetermined final destination before departure, to maximize container space utilization, it is typically essential to consolidate cargo destined for the same destination into one container whenever possible. However, different shippers within the same destination have distinct final distribution points. This necessitates cargo from other shippers to be placed on separate pallets to ease distribution. Therefore, once the cargo has been collected and centralized at the warehouse, the operator must first direct sorters to categorize the cargo based on the shippers and destinations. Upon completion of sorting, cargo destined for the same shipper and destination will be consolidated into one or more pallets (this applies to all shippers). Following palletization, cargoes with identical destinations will be loaded into the same or multiple containers, thereby completing the LCL cargo loading process (as shown in Fig. 3). However, as cargo volumes increase and destination diversification grows, manual completion of these tasks becomes time-consuming, labor-intensive, and efficiency-limited.

Besides cargo loading issues, given the necessity to consolidate with other shippers, LCL also affects the timeliness of cargo transportation. In LCL operations, if shipper A's cargo arrives early at the warehouse, it must wait for other shippers' cargoes for co-packing. If cargoes bound for destination A are not dispatched in the initial departure cycle, they must await the subsequent departure cycle. Consequently, cargoes from various shippers to different destinations will experience varying waiting times (as shown in Fig. 4). To optimize total cargo transportation time, we first ascertain the maximum transportation duration (comprising waiting and transit times) for cargo from a particular owner to a particular destination, then aim to minimize this duration to manage overall transportation times, reduce the transportation cycle, and enhance customer satisfaction.

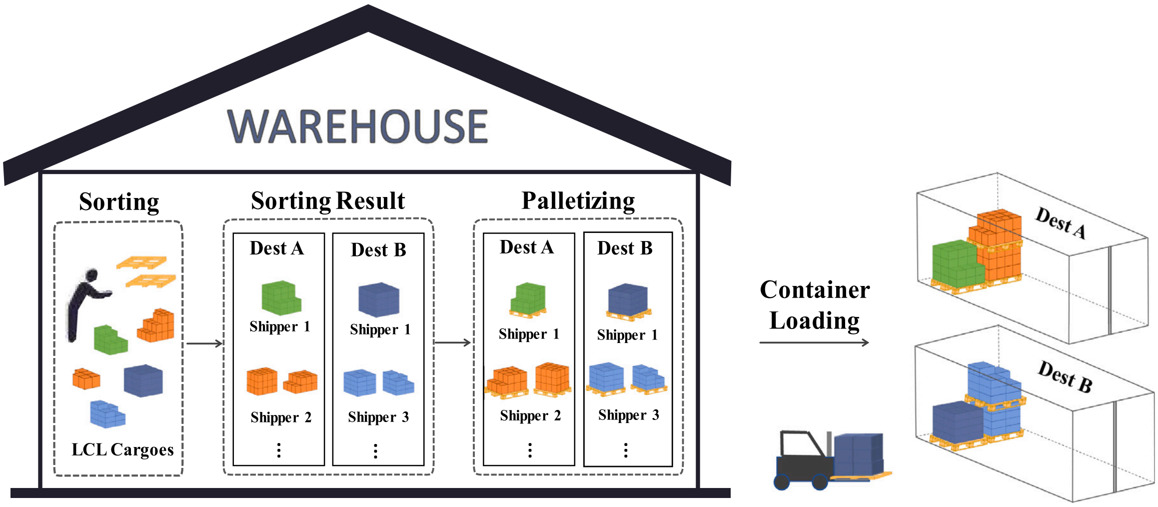


Fig. 3. Operation process of LCL cargo sorting-palletizing-loading.

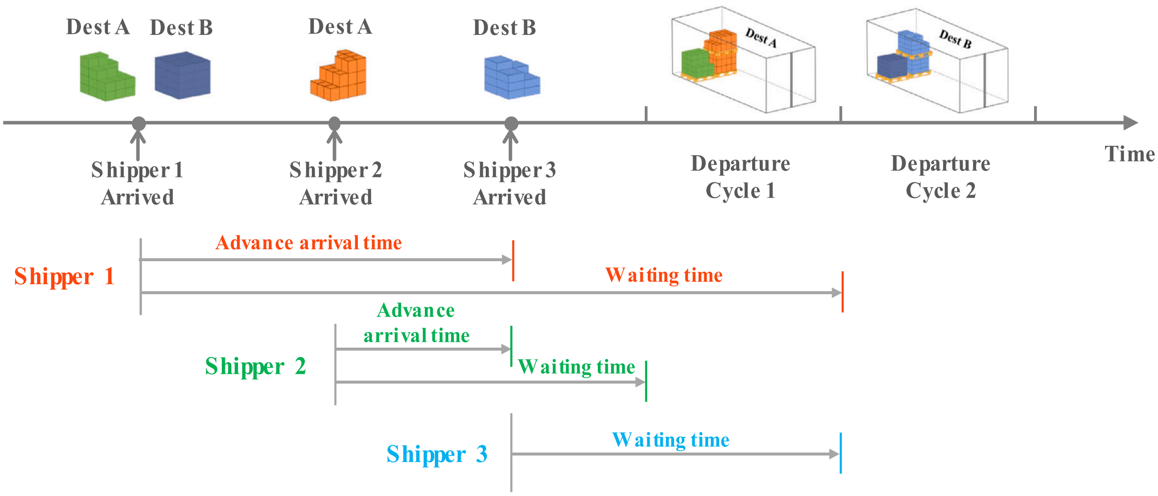


Fig. 4. Impact of LCL on the timeliness of different shippers.

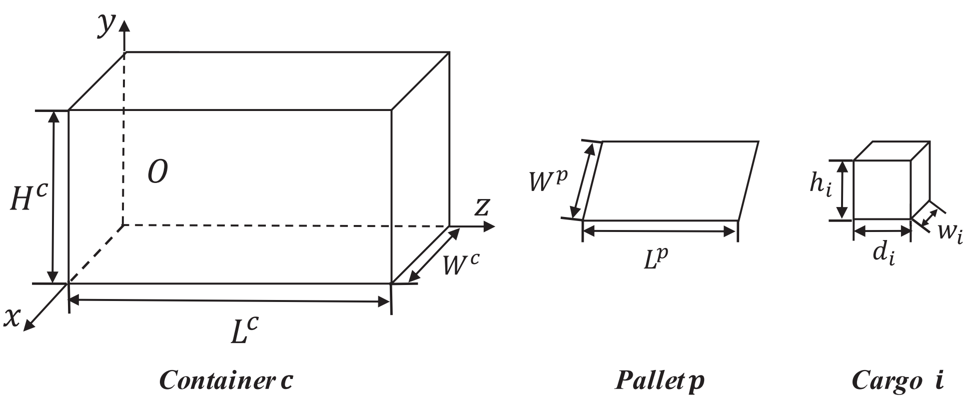


Fig. 5. Three-dimensional dimensions of containers, pallets, and LCL cargo.

Table 1
The differences between this study and the current research.

Reference	Problem studied	Objective function		Constraint		Solution method
		Minimize container usage or maximize loading rate	Minimize maximum transportation time	Multi Container	Two-stage loading	
Junqueira et al. (2012)	3D container loading problems	✓				A mixed integer linear programming models
Jamrus et al. (2016)	A less-than-container loading problem	✓				An extended priority-based hybrid genetic algorithm
Moura et al. (2016)	The two-stage loading problem for containers	✓			✓	A tree search algorithm
Kurpel et al. (2020)	3D container loading problems	✓				A branch and bound exact algorithm
Gajda et al. (2022)	A container loading problem	✓		✓		A randomized constructive heuristic
Alonso et al. (2018)	The two-stage loading problem for containers	✓		✓	✓	An Integer linear programming model
Alonso et al. (2019)	The two-stage loading problem for containers	✓		✓	✓	A Greedy Randomized Adaptive Search algorithm
This study	Multi-container two-stage LCL loading and scheduling	✓	✓	✓	✓	A MILP model and an improved genetic algorithm

In this study, we developed a multi-objective mixed-integer programming model to minimize container usage and maximum cargo transportation time. The model aims to decide the specific placement of cargoes on pallets and pallets on containers and the departure cycle of containers, providing the CR Express operators with a full-process operation plan for LCL business, from cargo loading to container dispatching.

We define the set of shippers as M, the set of destinations as D, and the set of cargoes sent by each shipper to each destination as S. Specifically, we divide the process of packing and loading LCL cargoes into two stages: (1) the pallet packing stage of LCL cargoes and (2) the container loading stage of pallets. The pallets each shipper sends to each destination are P in the first stage, while the set of containers used in the second stage is denoted as C. Given the limited number of LCL containers available per departure cycle of the CR Express, we define the set of departure cycles as T and plan the departure cycles of loaded containers rationally. The related sets are shown in the following table.

Sets	
M	The set of shippers (consignors) of LCL cargo
S	The set of cargo sent by each shipper to each destination
D	The set of destinations of LCL cargo
P	The set of pallets sent by each shipper to each destination in the first stage
C	The set of containers used in the second stage
T	The set of departure cycles

The 3D loading optimization problem for cargo involves geometric constraints. To illustrate these geometric constraints clearly, we establish a 3D coordinate system with the left rear bottom point of the container as the origin O. The container, pallet, and cargo dimensions are annotated as shown in Fig. 5.

Model assumptions

In exploring the issue of LCL transportation on the CR Express, we found that each cargo has a different shipper, and the cargoes are of different shapes and sizes. To maintain the general applicability of the study, we assume that:

(1) Multiple consignments of LCL cargoes from different shippers can be consolidated into one full container for transportation (Jamrus and Chien, 2016).

(2) Irregularly shaped LCL cargoes need to be repacked into regular rectangular shapes. For uniform handling, we consider all LCL cargoes standard rectangular shapes.

3.2. Symbol description

Indexes	
m	Index of shippers
i	Index of cargo
p	Index of pallets
c	Index of containers
Parameters	
W^c	The width of the container
H^c	The height of the container
L^c	The length of the container
W^p	The width of the pallet
L^p	The length of the pallet
H^p	The height of the pallet itself
w_{mdi}	The width of the cargo i sent by shipper m to destination d, $d \in D, m \in M, i \in S$
h_{mdi}	The height of the cargo i sent by shipper m to destination d, $d \in D, m \in M, i \in S$
l_{mdi}	The length of the cargo i sent by shipper m to destination d, $d \in D, m \in M, i \in S$
TA_{mdi}	Advance arrival time of the cargo i sent by shipper m to destination d at the consolidation centre, $d \in D, m \in M, i \in S$
TS_d	Transport time required for cargo to be transported from the consolidation centre to the destination d, $d \in D$
TD_t	Waiting time for cargo to be transported in the t departure cycles, $t \in T$
E	Number of LCL containers dispatched per departure cycle
Decision variables	
$(x_{mdi}, y_{mdi}, z_{mdi})$	Packing coordinates of cargo i sent by shipper m to destination d on the pallet, $d \in D, m \in M, i \in S$
$(x_{mdp}, y_{mdp}, z_{mdp})$	Packing coordinates of pallet p sent by shipper m to destination d on the container, $d \in D, m \in M, i \in S$
l_{mdij}	Be equal to 1 if cargo i sent by shipper m to destination d is to the left of j and 0 otherwise, $d \in D, m \in M, i, j \in S$
b_{mdij}	Be equal to 1 if cargo i sent by shipper m to destination d is to the behind of j and 0 otherwise, $d \in D, m \in M, i, j \in S$
u_{mdij}	Be equal to 1 if cargo i sent by shipper m to destination d is on the top of j and 0 otherwise, $d \in D, m \in M, i, j \in S$
l_{dmnpq}	Be equal to 1 if at the same destination pallet p of shipper m is to the left of pallet q of shipper n and 0 otherwise, $d \in D, p, q \in P, m, n \in M$
b_{dmnpq}	Be equal to 1 if at the same destination pallet p of shipper m is to the behind of pallet q of shipper n and 0 otherwise, $d \in D, p, q \in P, m, n \in M$
u_{smnpq}	Be equal to 1 if at the same destination pallet p of shipper m is on the top of pallet q of shipper n and 0 otherwise, $d \in D, p, q \in P, m, n \in M$
f_{mdip}	Be equal to 1 if cargo i sent by shipper m to destination d is packed on the pallet p and 0 otherwise, $d \in D, p \in P, m \in M, i \in S$
a_{dmpct}	Be equal to 1 if pallet p sent by shipper m to destination d is packed on the container c at the departure cycle t and 0 otherwise, $p \in P, c \in C, t \in T, d \in D, m \in M$
z_{ct}	Be equal to 1 if the container c be used at the departure cycle t, $c \in C, t \in T$
Derived variables	
ph_{dmp}	The height of the pallet p sent by shipper m to destination d, $p \in P, d \in D, m \in M$

3.3. Mathematical model

3.3.1. Objective function

Considering the challenges in cargo loading and container scheduling within LCL transportation, the model aims to minimize container usage and minimize the maximum cargo transportation time. The specific objective functions are as follows.

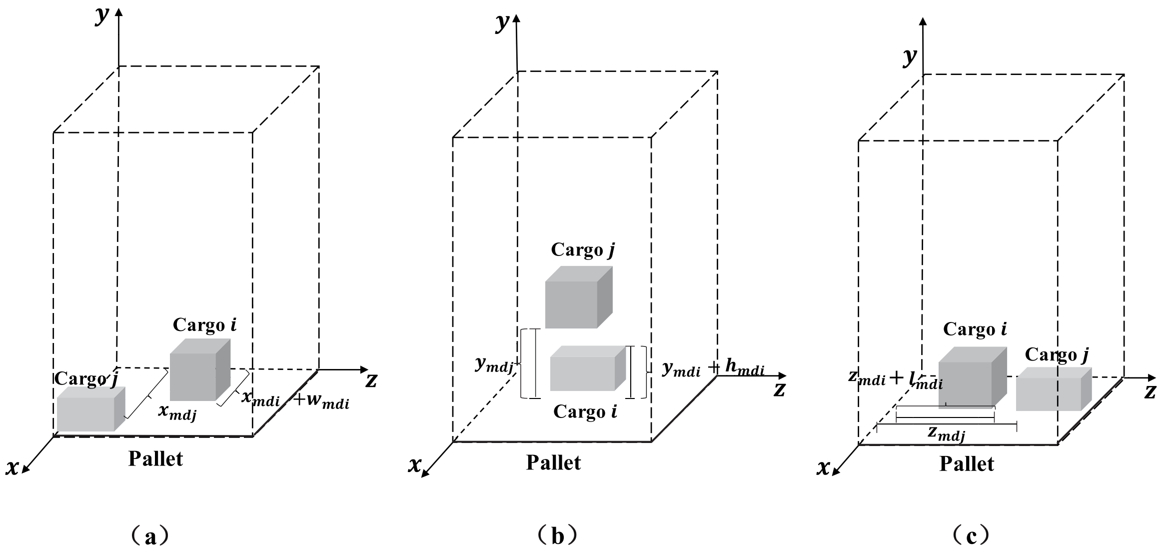


Fig. 6. LCL Cargo Location Relationships On the Pallet.

(1) Optimization Objective 1: Minimize the total container usage U_1.

$$U_1 = \min \sum_{c \in C} \sum_{t \in T} z_{ct} \tag{1}$$

(2) Optimization Objective 2: Minimize the longest transportation time(comprising waiting and transit times) U_2.

$$U_2 = \min \left[\max \left(TA_{mdi} + TS_d + TD_t\right) \times f_{mdip} \times a_{dmpct}\right] \quad \forall i \in S, \forall p \in P, \forall d \in D, \forall m \in M, \forall t \in T \tag{2}$$

Using dynamic programming, the objective function (2) is transformed into the following objective function and constraints:

$$U_2 = \min T_{\max} \tag{3}$$

$$T_{\max} \geq \left(TA_{mdi} + TS_d + TD_t\right) \times f_{mdip} \times a_{dmpct} \quad \forall i \in S, \forall p \in P, \forall d \in D, \forall m \in M, \forall t \in T \tag{4}$$

For two objective functions, to streamline the optimization process, we introduce variable δ as a linear weight, ranging from 0 to 1, and achieve unified optimization of the two objective functions through linear weighting. However, due to the different units of container usage and transportation time, it is necessary to standardize the units of the two objective functions. This can be accomplished by calculating the optimal values U_1^* and U_2^* for each single objective function and eliminating the unit disparity. The unified optimization objective is as follows:

$$\mathrm{Min}\, U_3 = \delta \times \frac{U_1}{U_1^*} + (1 - \delta) \times \frac{U_2}{U_2^*} \tag{5}$$

3.3.2. Constraints

(1) No Overlapping Constraints

$$l_{mdij} + l_{mdji} + b_{mdij} + b_{mdji} + u_{mdij} + u_{mdji} + 1 - f_{mdip} + 1 - f_{mdjp} \geq 1 \quad \forall i, j \in S, i < j, \forall p \in P, \forall d \in D, \forall m \in M \tag{6}$$

The Eq. (6) represents the non-overlapping constraint for consolidating LCL cargo during the pallet packing process. Drawing inspiration from Pisinger and Sigurd (2005) for handling the non-overlapping constraints, the positional relationships of LCL cargo are constrained in the x, y, z axes. To ensure that LCL cargo within the same pallet does not overlap spatially, it is required that cargo i are positioned completely above, below, left, right, front, or back of cargo j, as illustrated in Fig. 6.

$$l_{dmnpq} + l_{dmnqp} + b_{dmnpq} + b_{dmnqp} + u_{dmnpq} + u_{dmnqp} \geq a_{dmpct} + a_{dnqct} - 1 \quad \forall p, q \in P, \forall m, n \in M, m = n, \forall d \in D, \forall c \in C, \forall t \in T \tag{7}$$

$$l_{dmnpq} + l_{dmnqp} + b_{dmnpq} + b_{dmnqp} + u_{dmnpq} + u_{dmnqp} \geq a_{dmpct} + a_{dnqct} - 1 \quad \forall p, q \in P, \forall m, n \in M, m < n, \forall d \in D, \forall c \in C, \forall t \in T \tag{8}$$

The Eqs. (7) and (8) represent the non-overlapping constraint for pallets during the container loading process. Similar to the spatial constraints of LCL cargo, it is necessary to ensure that a pallet p from shipper m is positioned completely above, below, to the left, right, front, or back of a pallet q from shipper n. However, due to the possibility of placing pallets from different shippers within the same container, the constraints must ensure non-overlapping placement not only between pallets from the same shipper ($m = n$) but also between pallets from different shippers ($m < n$).

(2) Geometric Constraint

$$x_{mdi} - x_{mdj} + W^p \times b_{mdij} \le W^p - w_{mdi} \quad \forall i, j \in S, \forall d \in D, \forall m \in M \tag{9}$$

$$x_{mdi} \le W^p - w_{mdi} + \left(1 - f_{mdip}\right) \times W^p \quad \forall i \in S, \forall p \in P \forall d \in D, \forall m \in M \tag{10}$$

$$x_{mdp} - x_{ndq} + W^c \times b_{dmnpq} \le W^c - W^p \quad \forall p, q \in P, \forall d \in D, \forall m, n \in M \tag{11}$$

$$x_{mdp} \le W^c - W^p + \left(1 - a_{dmpct}\right) \times W^c \quad \forall p \in P, \forall d \in D, \forall m \in M, \forall c \in C, \forall t \in T \tag{12}$$

$$y_{mdi} - y_{mdj} + H^c \times u_{mdij} \le H^c - h_{mdi} \quad \forall i, j \in S, \forall d \in D, \forall m, n \in M \tag{13}$$

$$y_{mdi} \le ph_{dmp} - h_{mdi} - H^p + (1 - f_{mdip}) \times H^c \quad \forall i \in S, \forall p \in P, \forall d \in D, \forall m \in M \tag{14}$$

$$y_{mdp} - y_{ndq} + H^c \times u_{dmnpq} \le H^c - ph_{dmp} \quad \forall p, q \in P, \forall d \in D, \forall m, n \in M \tag{15}$$

$$y_{mdp} \le H^c - ph_{dmp} + (1 - a_{dmpct}) \times H^c \quad \forall p \in P, \forall d \in D, \forall m \in M, \forall c \in C, \forall t \in T \tag{16}$$

$$z_{mdi} - z_{mdj} + L^p \times l_{mdij} \le L^p - l_{mdi} \quad \forall i, j \in S, \forall d \in D, \forall m \in M \tag{17}$$

$$z_{mdi} \le L^p - l_{mdi} + (1 - f_{mdip}) \times L^p \quad \forall i \in S, \forall p \in P, \forall d \in D, \forall m \in M \tag{18}$$

$$z_{mdp} - z_{ndq} + L^c \times l_{dmvpq} \le L^c - L^p \quad \forall p, q \in P, \forall d \in D, \forall m, n \in M \tag{19}$$

$$z_{mdp} \le L^c - L^p + (1 - a_{dmpct}) \times L^c \quad \forall p \in P, \forall d \in D, \forall m \in M, \forall c \in C, \forall t \in T \tag{20}$$

Eqs. (9)–(12) represent the geometric constraints for the loading of LCL cargo onto pallets and the subsequent loading of pallets into containers on the x-axis direction. Taking the example of LCL cargo pallet packing, if the shipper m's cargo i to the same destination d is behind the cargo j, indicated by the condition $b_{mdij} = 1$, then, $x_{mdi} + w_{mdi} \leqslant x_{mdj}$, it is necessary to ensure that the total width occupied by the cargo does not exceed the width range of the pallet, as illustrated in Fig. 6(a). The pallet loading width range is set to be consistent with the maximum width of the pallet, denoted as W^p, as shown in Eqs. (9)–(10). When loading pallets into a container, if the shipper m's pallet p to the same destination d is behind the shipper n's pallet q, indicated by the condition $b_{dmnpq} = 1$, then, $x_{mdp} + W^p \leqslant x_{ndq}$, it is essential to ensure that the total width occupied by the pallets does not exceed the width range of the container W^c. The container loading width range is set to be consistent with the maximum width of the container.as shown in Eqs. (11)–(12).

Similar to Eqs. (9)–(12), Eqs. (13)–(16) represent the geometric constraints for the loading of LCL cargo onto pallets and the subsequent loading of pallets into containers on the y-axis direction. Eqs. (17)–(20) represent the geometric constraints for the loading of LCL cargo onto pallets and the subsequent loading of pallets into containers on the z-axis direction.

(3) Logical Constraint

$$\sum_{p \in P} f_{mdip} \ge 1 \quad \forall i \in S, \forall d \in D, \forall m \in M \tag{21}$$

$$f_{mdip} \le \sum_{c \in C} \sum_{t \in T} a_{dmpct} \quad \forall i \in S, \forall p \in P, \forall d \in D, \forall m \in M \tag{22}$$

$$a_{dmpct} + a_{d'nqct} \le 1 \quad \forall p, q \in P, \forall m, n \in M, \forall d, d' \in D, d < d', \forall c \in C, \forall t \in T \tag{23}$$

$$\sum_{c \in C} \sum_{t \in T} a_{dmpct} \le 1 \quad \forall p \in P, \forall m \in M, \forall d \in D \tag{24}$$

$$a_{dmpct} \le z_{ct} \quad \forall p \in P, \forall d \in D, \forall c \in C, \forall t \in T, \forall m \in M \tag{25}$$

$$\sum_{c \in C} z_{ct} \le E \forall t \in T \tag{26}$$

$$\sum_{t \in T} z_{ct} \le 1 \forall c \in C \tag{27}$$

$$f_{mdip}, z_{ct}, a_{dmpct} \in \{0, 1\} \quad \forall i \in S, \forall p \in P, \forall m \in M, \forall d \in D, \forall c \in C, \forall t \in T \tag{28}$$

Cargo ID	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Destination	A					B						
Shipper	a					b			c			
Cargo Encoding	3	5	4	2	1	8	6	7	11	9	12	10

Fig. 7. Encoding the Loading Sequence of LCL Cargoes.

$$l_{mdij}, b_{mdij}, u_{mdij} \in \{0, 1\} \quad \forall i, j \in S, \forall d \in D, \forall m \in M \tag{29}$$

$$l_{dmnpq}, b_{dmnpq}, u_{dmnpq} \in \{0, 1\} \quad \forall p, q \in P, \forall d \in D, \forall m, n \in M \tag{30}$$

$$ph_{dmp} \geq 0 \quad \forall p \in P, \forall d \in D, \forall m \in M \tag{31}$$

$$x_{mdp}, y_{mdp}, z_{mdp} \geq 0 \quad \forall p \in P, \forall d \in D, \forall m \in M \tag{32}$$

$$x_{mdi}, y_{mdi}, z_{mdi} \geq 0 \quad \forall i \in S, \forall d \in D, \forall m \in M \tag{33}$$

Eq. (21) ensures that cargo i destined for destination d shipper m must be loaded onto a pallet. Eq. (22) ensures that pallet p must be loaded into a container c with departure cycle t if pallet p to destination d shipper m is loaded with cargo. Eq. (23) ensures that only pallets with the same destination can be packed into the same container. Eq. (24) ensures that a pallet p destined for destination d shipper m can be loaded into at most one container c with departure cycle t. Eq. (25) ensures that a container is used if it contains a pallet. Eq. (26) ensures that the number of containers per departure cycle does not exceed E. Eq. (27) ensures that each container can only depart in a single departure cycle. Eqs. (28)–(33) are decision variable constraints, including 0–1 variables and integer variables.

4. Iterative local search based genetic algorithm

This study addresses the LCL cargo loading problem and optimizes container scheduling concurrently. From an operations research optimization standpoint, this problem is classified as NP-hard. Considering the challenge of obtaining precise solutions for large-scale instances, this study introduces an iterative local search strategy, leveraging a genetic algorithm to enhance the entire LCL business process.

4.1. Encoding

The loading process of LCL cargo involves two critical phases: palletization of cargoes and containerization of pallets. This study proposes a two-stage coding strategy for these phases. In the first stage, we focus on coding the loading sequence for LCL cargoes. In the second stage, we focus on the loading sequence for pallets. The coding strategy for both phases employs an integer coding method.

We established a loading order for LCL cargoes in the first stage (as shown in Fig. 7). This process accounts for varying destinations and shippers, and we generate a coding scheme for each destination and shipper. For instance, let us consider five LCL cargoes destined for destination A and shipper a. a sample sequence might be 3-5-4-2-1, this method is applied uniformly across different destinations and shippers. Proceeding to the second stage (as shown in Fig. 8), we refine the pallet loading sequence from the first stage. Once the first loading stage is concluded, Pallet 1 might contain shipments coded 1, 2, and 3, whereas Pallet 2 contains a shipment coded 4, and so forth, 12 consolidated shipments are distributed across 7 pallets. Subsequently, the pallets are sorted and coded based on destination. For instance, we assigned a random sequence like 3-2-1 to three pallets headed for Destination A. This coding approach was also applied to pallets bound for other destinations.

4.2. Decoding

Based on the two stages encoding strategy, the corresponding decoding process is detailed below: In the first phase, as shown in Fig. 7, we decode the LCL cargo's loading sequence. We begin by loading cargo coded 1, shipped by cargo owner A to destination A, onto the pallet, and then update the loadable points before proceeding to cargo coded 2. If the loading height exceeds the container's height, the loading of the current pallet is completed, and a new pallet is opened. Continuing this process, cargoes shipped by shipper b will be loaded to destination B, and cargoes shipped by shipper c will be loaded to destination B until all are palletized, recording each pallet's height and destination data. We decode the pallet loading sequence in the second stage, as shown in Fig. 8. the pallet with destination A, coded 1, is loaded first, and then the loadable points and container destinations are. Next, pallet coded as 2 is loaded, if it cannot fit in the current container, a new one is opened. Then, proceed with pallets for destination B until all destination pallets are loaded.

Pallet ID	P1	P2	P3	P4	P5	P6	P7
Cargo Encoding	1,2,3	4	5	6,7	8	9,10	11,12
Shipper	a			b		c	
Destination	A			B			
Pallet Encoding	3	2	1	6	5	4	7

Fig. 8. Encoding the Loading Sequence of Pallets.

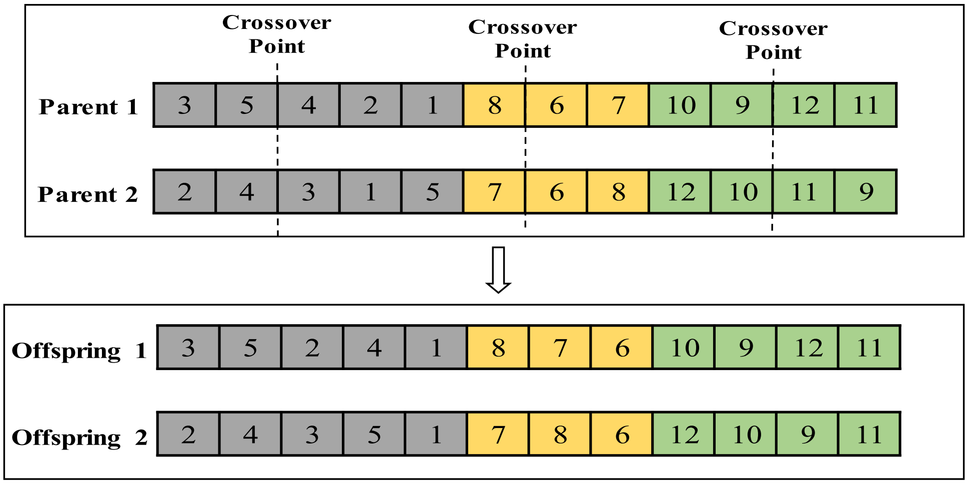


Fig. 9. Single-point Crossover.

4.3. Population initialization

According to the actual loading practices of CR Express's LCL operations, this study designs various sorting methods to generate an initial population, including random order, volume, length, height, width, and arrival time arrangements. Specifically, in a random order arrangement, once cargoes are categorized by destination and shipper, the loading sequence is randomly assigned using the Rand function for cargoes belonging to the same shipper and destined for the same destination.

4.4. Fitness function

The objective of this study is to minimize the number of containers and minimize the maximum shipping time, which is a minimization problem. Therefore, the fitness function should assign higher fitness to individuals with smaller objective function values, indicating better performance. This study reverses the objective function values by converting the reciprocal into fitness values. The specific fitness function is described in Eq. (34).

$$Fitness = \frac{1}{U_3} \tag{34}$$

4.5. Crossover operation

In this study, we utilize single-point and two-point crossover methods for genetic crossover operations on the coding of LCL cargo for the same destination and the same shipper.

(1) Single-point Crossover. Two highly fit individuals are randomly selected from the population to serve as Parent 1 and Parent 2, and a crossover point is randomly determined. The gene fragments preceding the crossover point are inherited unchanged by Offspring 1 and Offspring 2. Subsequently, genes in Parent 2 identical to those in Parent 1 pre-crossover are removed, and the remaining genes form the post-crossover gene fragments for Offspring 1, Offspring 2's gene fragments are created using the same approach. As shown in Fig. 9, this crossover strategy is also applied to the cargo coding sequences of other shippers.

(2) Two-point Crossover. Two highly fit individuals are randomly selected from the population to serve as Parent 1 and Parent 2, and two crossover points are randomly determined. The genes between the crossover points are inherited unchanged by Offspring 1

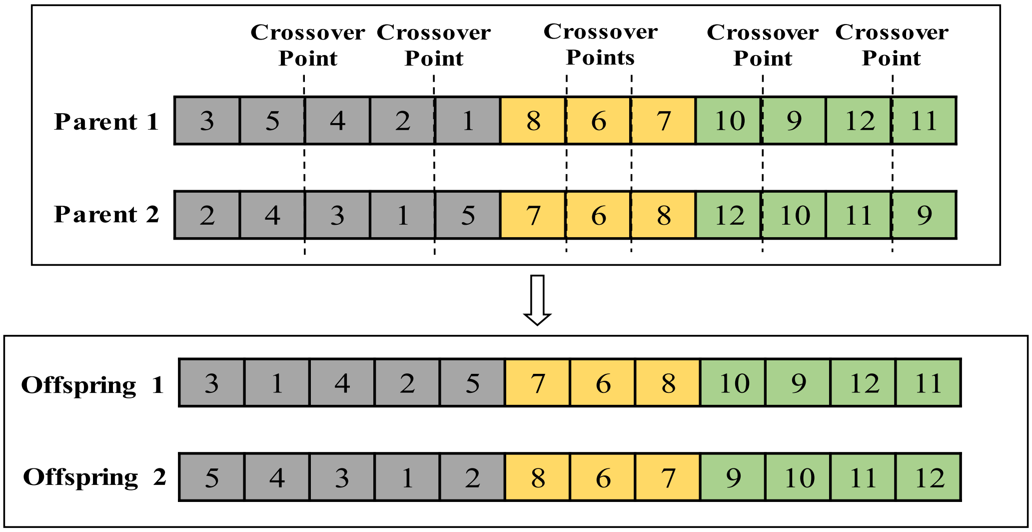


Fig. 10. Two-point Crossover.

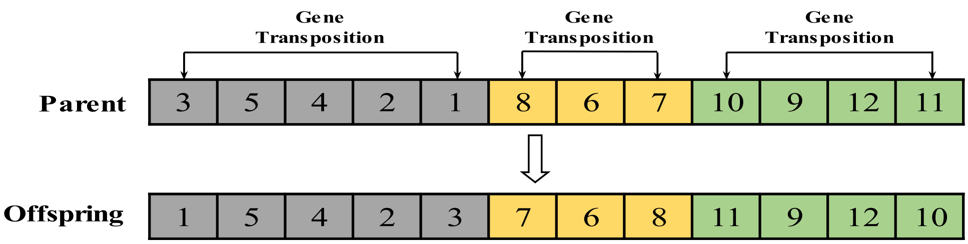


Fig. 11. Two-point Swap.

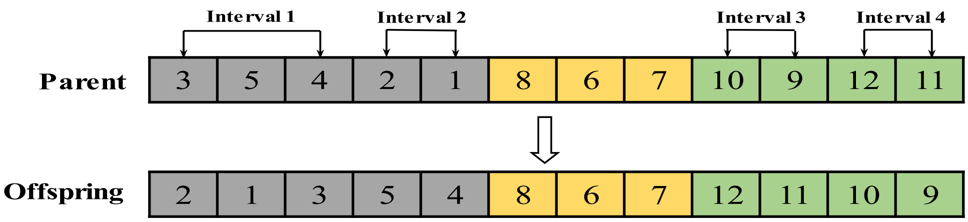


Fig. 12. Interval Reversal.

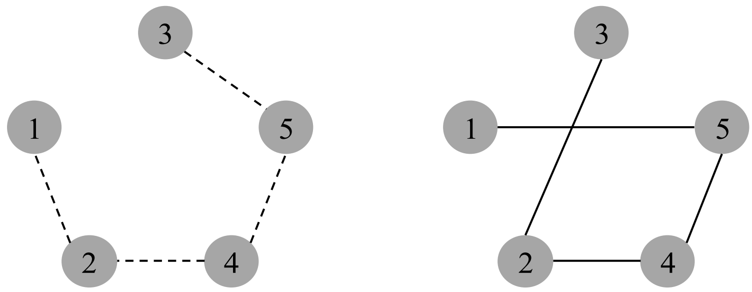
and Offspring 2. Subsequently, gene fragments in Parent 2 identical to those between the crossover points in Parent 1 are removed, and the remaining genes are sequentially allocated to Offspring 1. Offspring 2's gene fragments are created using the same approach. As shown in Fig. 10, this crossover strategy is also applied to the cargo coding sequences of other shippers.

4.6. Mutation operation

This study introduces two mutation strategies, two-point swapping, and interval reversal, to encode cargoes with the same destination for the same shipper. These strategies efficiently search the solution space and prevent reliance on crossover operations that may result in gene duplication and entrapment in local optima.

(1) Two-point Swap. For the gene sequence of cargo with the same destination and shipper, two swap points are randomly generated, and the genes between the two points are exchanged, refer to Fig. 11.

(2) Interval Reversal. For the gene sequence of cargo with the same destination and shipper, two intervals are randomly generated, and the gene segments within the two intervals are reversed as a whole, refer to Fig. 12.



(a) The solution before disturbance (b) The solution after disturbance

Fig. 13. Double-bridge Move Disturbance Strategy.

4.7. Iterative local search

Local search algorithms aim to find better solutions near a given solution. Techniques like variable neighborhood search, Tabu search, Beam search, and simulated annealing are recognized as effective local search methods for 3D bin packing problems (Anibal Tavares de Azevedo and others, 2014; Bi-Chao Bang, 2011). The performance of these algorithms is influenced by factors including algorithm design, neighborhood size, and search strategy. Hence, designing a local search strategy for the specific solution structure is essential to achieve high-quality solutions.

This study introduces a local search mechanism to enhance the quality of the initial population and the new offspring solutions. Various local search strategies are developed from the initial solution by applying a disturbance strategy to perturb it, aiming to discover better solutions in the solution's neighborhood. Additionally, a termination condition is established to halt the search upon meeting specific criteria and to output the optimal solution identified thus far. Continuous iteration of this method increases search diversity, aiding in escaping local optima and approaching the global optimal solution. The pseudo-code of the algorithm is shown in Algorithm 1.

Algorithm 1: Iterative Local Search Algorithm.

```
Function ILSA():
    i = 0;
    S* = S_0;                          // Initialize the initial solution to the optimal solution
    while i < L do
        S_1 = Disturbance(S*);                         // Disturbing the current solution
        S_2 = LocalSearch(S_1);                        // Local search for the Disturbed solution
        // if find a better solution, set it as the current optimal solution
        if (FitFun(S_2) > FitFun(S*)) then
            S* = S_2;
        end
        i = i + 1;                                     // Increase the number of iterations i + 1
    end
    return S* ;                                        // return the optimal solution
```

4.7.1. Disturbance strategy

During the iterative local search process, a disturbance strategy must be incorporated to prevent the solution from getting stuck in local optima. However, the magnitude of the disturbance applied to the solution should be manageable. When the disturbance is too small, it is prone to getting trapped in local optima. When the disturbance is too large, the randomness introduced to the troubled solution is too significant, resulting in lower solution quality.

This study adopts the classic double-bridge move disturbance strategy, commonly used in the Traveling Salesman Problem. The strategy perturbs the cargo of the same shipper with the same destination. The basic idea is randomly selecting four nodes: a, b, c, and d. Then, the connections between $(a, a+1)$, $(b, b+1)$, $(c, c+1)$, and $(d, d+1)$ are severed, and new connections between $(a, c+1)$, $(c, a+1)$, $(b, d+1)$, and $((d, b+1)$ are made. To illustrate, consider five cargoes with the encoding order of 35421, destined for destination A and belonging to shipper a. The connections $(3, 5)$, $(5, 4)$, $(4, 2)$, and $(2, 1)$ are cut, and connections $(3, 2)$, $(4, 5)$, $(5, 1)$, and $(2, 4)$ are made. As a result, the perturbed encoding order becomes 3 2 4 5 1, as shown in Fig. 13.

4.7.2. Local search strategy

In the local search phase, this study designs five local search methods for the encoding of pallets, encoding of cargo within pallets, and encoding of the shipping cycle of containers.

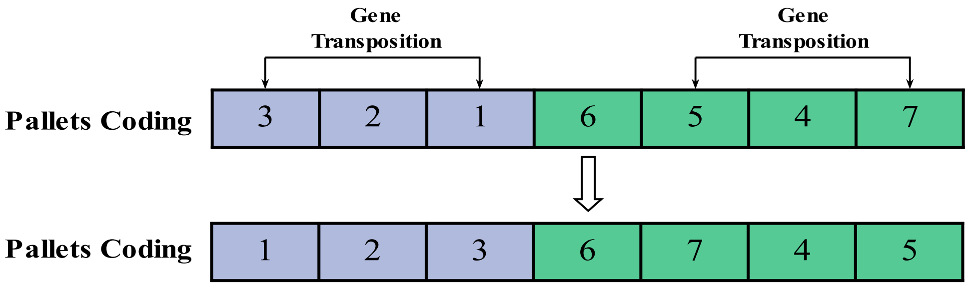
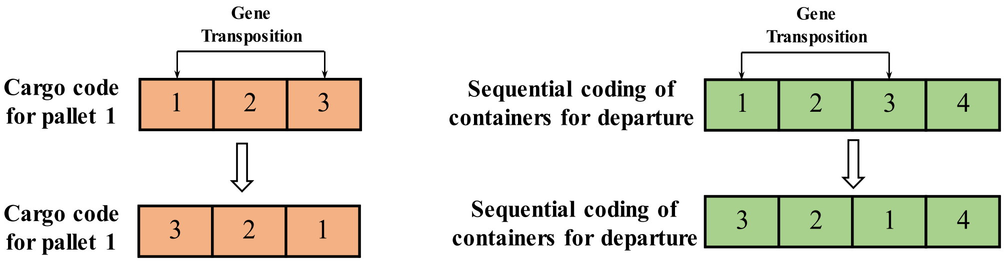


Fig. 14. Gene Transposition Between Two Pallets.



(a) Gene Transposition of Cargo Within the Same Pallet   (b) Gene Transposition of Container Departure Sequence

Fig. 15. Gene Transposition of Cargo Within the Same Pallet and Container.

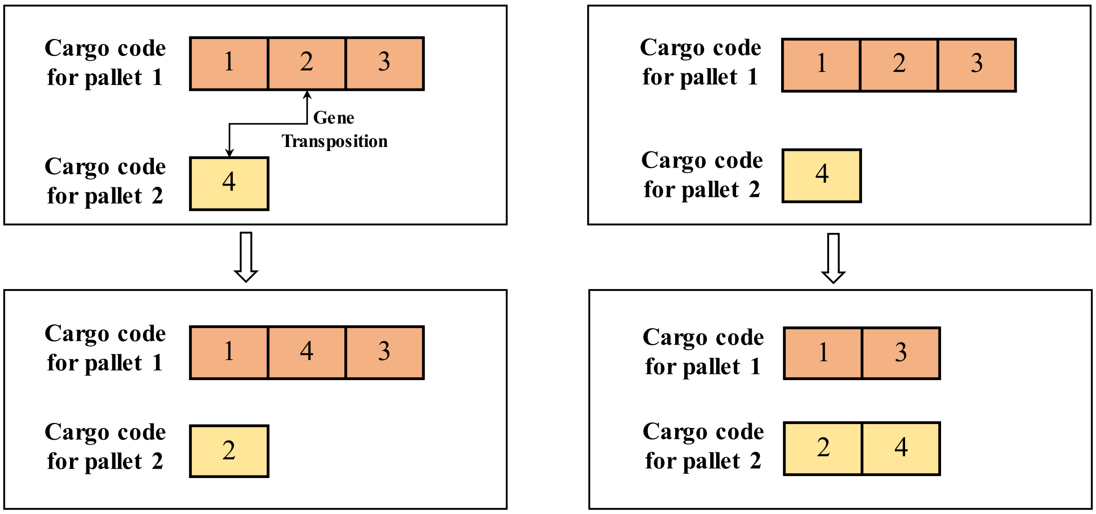


Fig. 16. Gene Transposition of Cargo Between Two Pallets.

(1) Exchange the loading sequence between pallets. The loading sequence of pallets with the same destination is changed through a transposition method. As shown in Fig. 14, the pallets with codes 3 and 1 destined for destination A are exchanged, and the pallets with codes 5 and 7 destined for destination B are exchanged.

(2) Exchange the order of cargo within the same pallet. As shown in Fig. 15 (*a*), the cargo order with codes 1 and 3 within pallet 1 is exchanged.

(3) Exchange the cargo order between two pallets for the same destination and shipper. As shown in Fig. 16, taking the cargo for destination A and shipper a as an example, the cargo with code 2 in pallet 1 is exchanged with the cargo with code 4 in pallet 2.

(4) Transfer cargo from one pallet to another pallet for the same destination and shipper. As shown in Fig. 16, the cargo with code 2 in pallet 1 is placed at the beginning of the cargo in pallet 2.

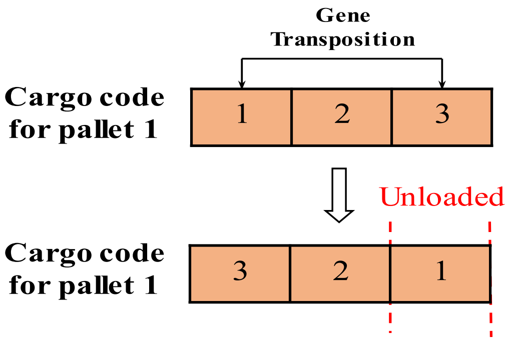


Fig. 17. Infeasible Solution due to Rearrangement.

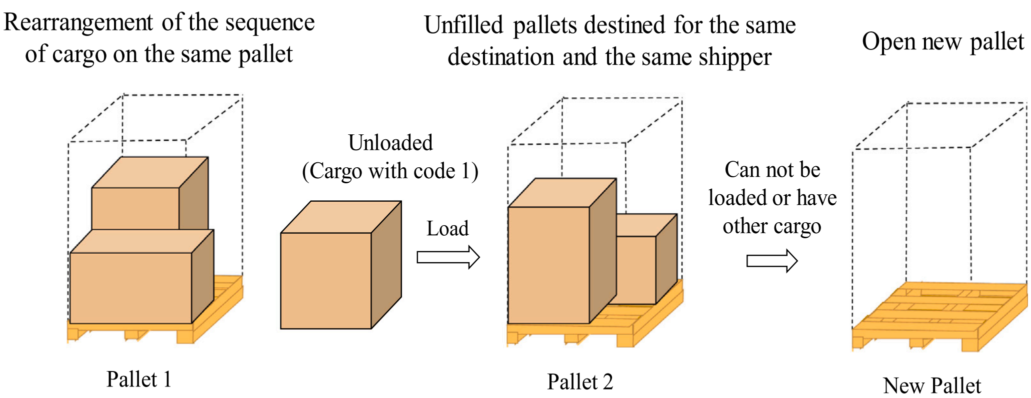


Fig. 18. Schematic of the Repair Strategy.

(5) Exchange the shipping cycle of containers, as shown in Fig. 15(*b*).

4.8. Repair strategy

During the local search process, partially infeasible new solutions may be generated. For example, for cargo destined for the same destination and the same shipper, the pallet will repack these cargo when the loading sequence of the cargo within the same pallet or between two pallets is exchanged. However, the repackaging process may not satisfy the loading constraints, resulting in infeasible solutions.

Taking Fig. 17 as an example, pallet 1 initially loaded cargo with the numbers 1, 2, and 3; after rearrangement, the desired loading sequence for pallet 1 becomes 3, 2, 1, and the change in sequence causes cargo 1 to be unable to load into the pallet. In such cases, it is necessary to develop a repair strategy to transform these infeasible solutions into feasible ones. In this study, we consider allocating the unloaded cargo to other unfilled pallets destined for the same destination and the same shipper. A new pallet will be opened if there is remaining cargo, as shown in Fig. 18.

Taking the loading of cargo in pallet 1, if cargo with the number 1 cannot be loaded into pallet 1 due to the rearrangement, it will be allocated to unfilled pallet 2. If it cannot be loaded into pallet 2, a new pallet will be opened. The pseudo-code for the specific repair strategy can be found in 2.

Algorithm 2: Repair Strategy.

```
Function RS():
    S = Unloaded cargos;                        // Generate the set S of unloaded cargos
    while i ∈ S do
        while p ∈ P do
            DBLF(i, p);                         // Loading of cargo i onto a pallet p with completed loading
            if (CompleteLoading) then
                break;
            end
            if (LoadingNotAchieved) then
                DBLF(i, NewPallet);             // Open a new pallet
            end
        end
    end
    return TheHeightsOfPallets ;                // return the heights of pallets
```

4.9. ILS-GA algorithm

This study introduces an Iterative Local Search Genetic Algorithm (ILS-GA) for a two-stage Less than Container Load (LCL) cargo loading process. The algorithm begins by inputting detailed specifications of containers and pallets and the dimensions and information about the cargo to be loaded. It then configures key parameters of the genetic algorithm, such as population size, crossover rate, mutation rate, and the number of iterations. Once the initial population is generated, the algorithm enters the main loop. Within this loop, the fitness of the population is assessed, and each individual undergoes crossover and mutation operations to enhance genetic diversity. Subsequently, the Iterative Local Search Algorithm (ILSA) is applied to each individual to refine the quality of the solutions. If the ILSA identifies a solution with higher fitness than the current individual, it replaces the original, facilitating the continuous evolution of the population. The population is then selected for the next generation based on fitness, and the cycle repeats until a termination condition is met. At this point, the algorithm concludes and returns the optimal loading plan. The specific pseudo code is as follows:

Algorithm 3: ILS-GA Algorithm.

```
Function ILS-GA():
    Input details of containers, pallets, cargoes
    Input algorithm related parameters;         // Setting algorithm-related parameters
    GenerateInitialPopulation(P);               // Generating the initial population
    while not Termination condition do
        EvaluatePopulation(P);                  // Calculate the fitness function
        while i ∈ P do
            PerformCrossoverAndMutation(i);     // crossover, mutation operations
            S* = ILSA(i);                       // local search strategy
            if (FitFun(S*) > FitFun(i)) then
                ReplaceIndividual(i, S*);
            end
        end
        SelectNewPopulation(P);                 // Selection of a new generation of populations
    end
    return BestSolution(P) ;                    // Return the optimal loading scheme
```

To present the key steps and the overall process of the ILS-GA, we divide the overall process of the algorithm into five parts: the setting of algorithm parameters, population initialization, the main loop of the algorithm, the iterative local search algorithm, and the repair strategy, and have drawn a detailed flowchart(as shown in Fig. 19). The flowchart clearly shows the logical relationships among these parts. It intuitively displays how the local search mechanism is integrated into the genetic algorithm framework in each iteration, which is also one of the leading innovative points of our algorithm research.

5. Experimental result

5.1. Benchmark data set

In this study, we take the core route of “Zhengzhou-Hamburg” of CR Express Zhongyu Train as an example. We calculate three kinds of scales, small, medium, and large instances, to validate the model and algorithm’s correctness, validity, and optimization.

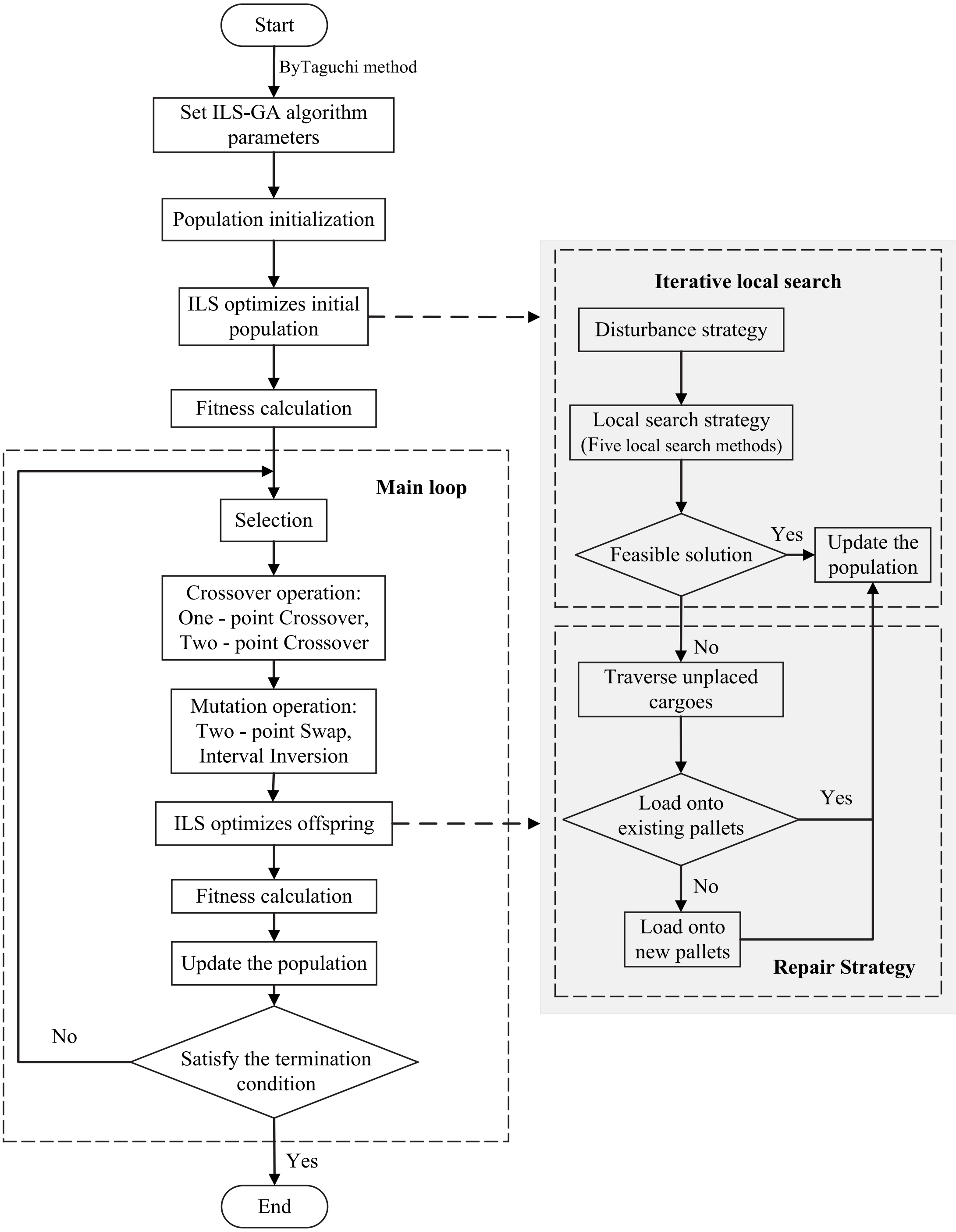


Fig. 19. Flowchart of ILS-GA.

The "Zhengzhou - Hamburg" train starts from Zhengzhou Putian Station, exits from Alashankou, and passes through 4 stations in Almaty, Moscow, Klaipeda, and Hamburg in turn, with a departure cycle of 1 day, a full journey of 10,245 kilometers, and a total running time of about 16 days (the specific arrival time at each station is shown in Table 2).

The standards of pallet and container specifications used in the CR Express are shown in Table 3, and this study is based on the 20-foot container specifications, for example, analysis. Table 4 shows the range of sizes and quantities for three scale instances of LCL cargo (the specific sizes of LCL cargo were obtained through on-site investigation and collation).

Table 2
Arrival time of the "Zhengzhou-Hamburg" at each station.

The origin	Stations	Time of arrival/Day
Zhengzhou	Almaty	6
	Moscow	11
	Klaipeda	13
	Hamburg	16

Table 3
Pallet and container size.

	Types	Length/cm	Width/cm	Height/cm
Pallet		120	80	14
Container	20 ft.	569	213	218
	40 ft.	1180	213	218

Table 4
The range of sizes and quantities for three instances LCL cargo.

	Length/cm	Width/cm	Height/cm	Quantity/pc
Small scale	$[80, 120]$	$[60, 80]$	$[100, 200]$	$[12, 20]$
Medium scale	$[50, 100]$	$[40, 80]$	$[80, 150]$	$[25, 45]$
Large scale	$[30, 120]$	$[30, 80]$	$[90, 200]$	$[80, 120]$

Table 5
Influence factor levels values.

Levels	Influencing factor				
	P	P_c	P_m	$Iter$	N
1	40	0.5	0.05	50	10
2	50	0.6	0.1	70	15
3	60	0.7	0.15	90	20
4	80	0.8	0.2	110	25

5.2. Experimental environment

The following instances are conducted on a computer with an Intel(R) Core(TM) i7-7500U CPU @ 2.70 GHz 2.90 GHz, the operating system Windows 10 Enterprise, and 8.0G of RAM. The proposed mixed-integer planning model is solved by the IBM ILOG CPLEX 12.8. The proposed genetic algorithm is implemented in C++ programming language with the GCC compiler.

5.3. Parameter tuning

This study is based on Taguchi's experimental design method to set the parameter combination of the genetic algorithm. Taguchi experiment is a quality engineering method of cost reduction. It has been used in many optimization problems (Xin et al., 2024; Zhou and Lee, 2020), and its core idea is that the performance of the algorithm is affected by the controllable factors and noise factors. The combination of the levels of the controllable factors is selected to reduce the sensitivity to the noise factors and reduce the fluctuation of the algorithm's performance. Taguchi's instances utilize the signal-to-noise ratio (S/N) to measure the algorithm's robustness. The larger the signal-to-noise ratio for problems with 'larger the better' quality characteristics, the better the robustness, and the smaller the signal-to-noise ratio for issues with 'smaller the better' quality characteristics, the better the robustness. The method does not require instances on all combinations of parameter configurations; only specific parameter combinations are set by orthogonal tables.

Factors affecting the performance of genetic algorithms include population size (P) - a small population size will lead to insufficient gene diversity and poor quality initial solutions, while a large population size will generate more poor quality genes and reduce the operational efficiency of the algorithm; crossover probability (P_c) and mutation probability (P_m) - when the crossover and mutation probabilities are small, the algorithm will not be able to effectively update the population, easy to fall into the local optimum, and when the probability is larger, easy to destroy high-quality genes; the number of iterations ($Iter$), - too few iterations will lead to the algorithm cannot converge, the number of times too much will reduce the operating efficiency of the algorithm; the number of iterative local search (N) - for the initial solution and the new solution for the iterative local search, the number of times too much will increase the running time, and too few times lead to poor quality of the searched solution. Based on this, the experimental data was tested by setting multiple influencing factor level values as shown in Table 5. Four levels were set for each influencing factor, and an orthogonal matrix table was set up to design the experiment using Taguchi's method, as shown in Table 6.

Table 6
Influence factor levels values.

No.	P	P_c	P_m	$Iter$	N
1	40	0.5	0.05	50	10
2	40	0.6	0.1	70	15
3	40	0.7	0.15	90	20
4	40	0.8	0.2	110	25
5	50	0.5	0.1	90	25
6	50	0.6	0.05	110	20
7	50	0.7	0.2	50	15
8	50	0.8	0.15	70	10
9	60	0.5	0.15	110	15
10	60	0.6	0.2	90	10
11	60	0.7	0.05	70	25
12	60	0.8	0.1	50	20
13	80	0.5	0.2	70	20
14	80	0.6	0.15	50	25
15	80	0.7	0.1	110	10
16	80	0.8	0.05	90	15

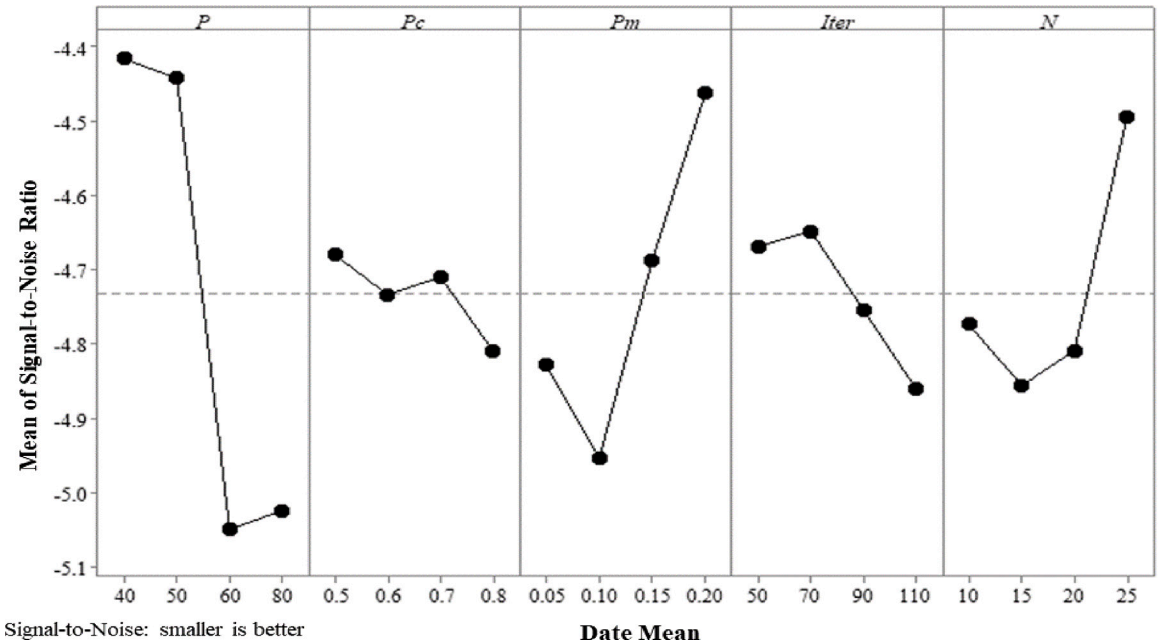


Fig. 20. Main Effect Plot for Signal-to-Noise Ratio.

Different combinations of parameters affect the performance of the algorithm, to select the optimal combination of parameters so that the algorithm calculates the optimal target value in a shorter time, this study uses two indicators: the objective value (OBJ) and the running time (CT), to express the performance of the algorithm. The smaller the objective value and the running time are, the better. Based on the large-scale arithmetic data set, 16 parameter combinations are experimented with. The performance function of the algorithm is constructed as in Eq. (35) (due to the different units of the objective value and the running time, they are respectively divided by the average value of each of their respective values to eliminate the units). The'smaller the better' signal-to-noise ratio (S/N) in Taguchi's method (as in Eq. (36)) is utilized for calculation, and the main effect plot of the (S/N) ratio is plotted (shown in Fig. 20).

$$UF = \frac{OBJ}{(\sum OBJ)/n} + \frac{CT}{(\sum CT)/n} \tag{35}$$

$$\mathrm{S/N} = -10log_{10}(\frac{\sum UF^2}{n}) \tag{36}$$

The problem of this study is a problem with'smaller the better' quality characteristics, so the parameters are taken as the minimum value at each level. Finally, the parameters of the genetic algorithm are set as shown in Table 7.

Table 7
Genetic algorithm parameter setting.

The parameters of genetic algorithm	Values
Population size (P)	60
Crossover probability (P_c)	0.8
Mutation probability (P_m)	0.1
the number of iterations ($Iter$)	110
The number of iterative local search (N)	15

Table 8
Data information of the small-scale instance.

Cargo ID	Destination	Shipper	Length/cm	Width/cm	Height/cm	Advance arrival time $/day$
1	Moscow	1	103	78	194	9
2	Moscow	1	117	77	200	9
3	Moscow	1	104	62	187	9
4	Moscow	1	112	69	183	9
5	Moscow	2	115	79	187	5
6	Moscow	2	110	72	190	5
7	Moscow	2	114	74	143	5
8	Moscow	2	102	76	203	5
9	Hamburg	3	119	60	199	8
10	Hamburg	3	112	74	174	8
11	Hamburg	3	108	61	152	8
12	Hamburg	3	104	73	188	8
13	Hamburg	4	114	79	83	7
14	Hamburg	4	106	67	112	7
15	Hamburg	4	106	62	189	7
16	Hamburg	4	112	70	179	7

Table 9
LCL loading scheme (small-scale).

Container ID	Pallet ID	Cargo ID	Length of pallet /cm	Departure cycle
1	1	3	201	1
	2	1	208	
	3	4	197	
	4	2	214	
	5	6	204	
	6	5	201	
	7	7	157	
	8	8	217	
2	9	12	202	
	10	11	166	
	11	10	188	
	12	9	213	
	13	13 and 14	209	
	14	15	203	
	15	16	193	

5.4. Comparison between mathematical model and proposed algorithm

5.4.1. Small-scale instances

To verify the correctness of the mixed integer programming model, this study is based on a small-scale instance, which is solved accurately by using the Cplex solver. 4 shippers with a total of 16 LCL cargoes need to be shipped to Moscow and Hamburg, the number of containers allowed to be LCL transportation in each cycle is set to be 2. The corresponding data information of the instance is shown in Table 8.

Solving for the single objectives of container usage and transportation time, respectively, $U_1^* = 2, U_2^* = 24$. A total of 15 pallets are used to load all LCL cargoes, and a total of 2 containers are used to load all pallets, all of which are transported in departure cycle 1. Table 9 shows the specific loading scheme.

To validate the effectiveness of the genetic algorithm with iterative local search(ILS-GA), 20 small-scale instances were considered. First, the instances were solved using CPLEX to obtain exact solutions. Then, the ILS-GA was independently run 10 times for each experiment. The usage of containers U_1, transportation time U_2, and objective values U_3 were recorded for both CPLEX and ILS-GA solutions. The optimal solutions obtained from the ILS-GA were compared with the exact solutions obtained from CPLEX, considering the Gap_1 of objective and Gap_2 runtime. $Gap_1 = \left(l_1 - l_2\right)/l_1 \cdot 100\%$ Let l_1 and l_2 represent the objective values obtained by CPLEX and ILS-GA, respectively. $Gap_2 = \left(t_1 - t_2\right)/t_1 \cdot 100\%$ Let t_1 and t_2 represent the runtime values obtained

Table 10
Comparison of Cplex and ILS-GA solving performance for small-scale instances.

$Instance$ $No.$	$CPLEX$			$ILS-GA$				Gap_1/%	Gap_2/%
	C /pc	TT /day	RT /s	C /pc	TT /day	RT /s	SV		
1	2	26	36	2	26	4	0	0	89.26
2	3	22	95	3	22	3	0	0	96.41
3	3	24	17	3	24	4	0	0	74.85
4	2	26	74	2	26	5	0	0	92.84
5	2	28	89	2	28	6	0	0	93.82
6	2	24	95	2	24	4	0	0	95.79
7	3	24	120	3	24	6	0	0	95.25
8	3	26	260	3	26	8	0	0	96.92
9	3	24	246	3	24	5	0	0	98.09
10	3	25	175	3	25	7	0	0	96.12
11	2	28	225	2	28	7	0	0	97.11
12	2	27	150	2	27	6	0	0	96.00
13	3	26	205	3	26	5	0	0	97.56
14	3	28	303	3	28	5	0	0	98.42
15	3	24	263	3	24	7	0	0	97.34
16	2	26	217	2	26	5	0	0	97.74
17	2	23	260	2	23	5	0	0	98.27
18	3	25	278	3	25	6	0	0	97.70
19	2	25	226	2	25	6	0	0	97.57
20	3	26	265	3	26	5	0	0	98.19

by CPLEX and ILS-GA, respectively. The standard deviation (SV) of the objective values obtained by the ILS-GA for the 10 runs was also calculated. The analysis results are presented in Table 10 ($C - theUsageOfContainers$; $TT - TransportationTime$; $RT - Runtime$).

From Table 10, it can be observed that the ILS-GA discovered the optimal solutions within a certain time limit, as the objective values obtained by ILS-GA matched the exact solutions obtained by CPLEX (Gap_1 with a value of 0). This indicates that the genetic algorithm designed in this study is correct and effective in solving the problem. Furthermore, the SV of the objective values obtained by ILS-GA for each experiment was 0, indicating that the algorithm found the optimal solution in all 10 runs for each experiment. It is further demonstrated that the algorithm has cargo stability and robustness. The average runtime of CPLEX was 180 s, while the average runtime of ILS-GA was only 5.45 s, which is approximately 95.56% lower. This significant difference suggests that, compared to exact solutions, the ILS-GA algorithm achieves better optimization efficiency in terms of runtime.

5.4.2. Medium-scale instances

To verify the convergence of the ILS-GA in solving medium-scale instances, Table 11 presents the information on the medium-scale LCL cargoes. There are 10 shippers with a total of 25 LCL cargoes that need to be transported to Almaty, Moscow, Klaipeda, and Hamburg. The maximum number of containers allowed for LCL Transportation per cycle is set to 2. The ILS-GA algorithm is repeatedly used to solve the single-objective problems of container utilization and transportation time, resulting in a value of $U_1^* = 4, U_2^* = 25$, respectively. A total of 20 pallets are used to load all the LCL cargo, and 4 containers are used to load all the pallets. In the first cycle, containers with sequence numbers 3 and 4 are transported; in the second cycle, containers with sequences 1 and 2 are transported. The specific loading program is shown in Table 12.

To analyze the changes in the population's worst, average, and best objective values with the number of iterations, the iteration curve of the ILS-GA for solving this instance is plotted in Fig. 21. It can be observed that during the iteration process, the worst objective value stabilizes around the $35th$ generation, and the best objective value stabilizes around the $17th$ generation. When the worst individual is completely replaced by the best individuals, the average objective value stabilizes, and all three curves converge to the global optimum, verifying the cargo convergence of the algorithm.

To validate the optimality of the ILS-GA, this study compares the high-quality solutions obtained by CPLEX with those obtained by the ILS-GA based on a dataset of 20 medium-scale instances. Due to the large number of LCL cargoes, it is challenging for the CPLEX solver to achieve an exact solution within a reasonable time. Therefore, a runtime of 2 h is set for CPLEX to compute high-quality approximate solutions. The genetic algorithm is run independently ten times for each instance, and usage of containers U_1, transportation time U_2, and objective values U_3 obtained by ILS-GA are recorded. The objective values Gap_1 and runtime values Gap_2 of CPLEX and ILS-GA are calculated separately, $Gap_1 = \left(l_1 - l_2\right)/l_1 \cdot 100\%$ Let l_1 and l_2 represent the objective values obtained by CPLEX and ILS-GA, respectively. $Gap_2 = \left(t_1 - t_2\right)/t_1 \cdot 100\%$ Let t_1 and t_2 represent the runtime values obtained by CPLEX and ILS-GA, respectively. The results are shown in Table 13 ($C - theUsageOfContainers$; $TT - TransportationTime$; $RT - Runtime$).

From Table 13, it can be observed that regarding the optimization objectives, for objective 1 (container utilization), ILS-GA has the same number of containers as CPLEX in 13 instances, outperforming CPLEX in 7 instances. For objective 2 (transportation time), the transportation time obtained by ILS-GA is less than or equal to that obtained by CPLEX in all instances. The average objective values of CPLEX and GA are 6.78% apart, indicating that ILS-GA provides better solutions. In terms of runtime, the runtime of CPLEX exceeds 2 h for all instances, while the average runtime of GA is only 27.25 s, demonstrating higher computational efficiency.

Table 11
Data information of the medium-scale instance.

Cargo ID	Destination	Shipper	Length/cm	Width/cm	Height/cm	Advance arrival time /*day*
1	Almaty	1	67	64	97	5
2	Almaty	1	58	76	86	5
3	Almaty	2	96	70	85	6
4	Almaty	2	67	68	88	6
5	Almaty	2	83	54	96	6
6	Moscow	3	40	78	84	7
7	Moscow	3	82	62	83	7
8	Moscow	3	99	65	82	7
9	Hamburg	4	91	71	87	10
10	Hamburg	4	100	78	144	10
11	Hamburg	5	81	62	93	5
12	Hamburg	5	72	63	99	5
13	Klaipeda	6	63	50	94	11
14	Klaipeda	6	50	60	86	11
15	Klaipeda	6	98	58	83	11
16	Klaipeda	7	100	50	148	7
17	Klaipeda	7	92	51	95	7
18	Klaipeda	7	94	76	135	7
19	Hamburg	8	97	79	145	9
20	Hamburg	8	61	66	107	9
21	Hamburg	8	88	65	137	9
22	Hamburg	9	92	73	145	8
23	Hamburg	9	63	50	137	8
24	Hamburg	10	82	74	123	7
25	Hamburg	10	93	74	107	7

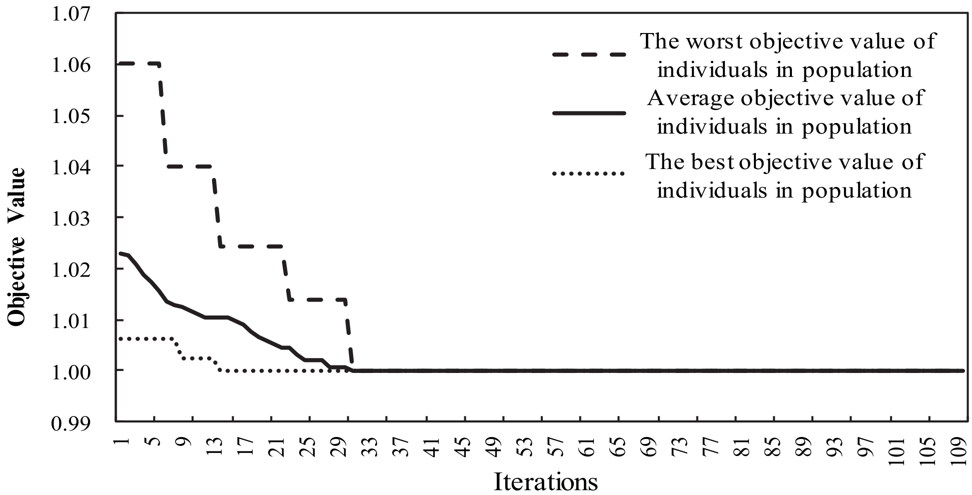


Fig. 21. Plot of Iteration Curves for Medium-Scale instances using ILS-GA.

5.4.3. Large-scale instances

If faced with a large-scale demand for LCL containers on the CR Express, the CPLEX solver cannot obtain high-quality approximate optimal solutions within a reasonable time. To verify the optimality of the algorithm in solving large-scale instances, this study refers to the loading algorithm designed by Liao et al. (2023) and based on a two-stage loading strategy, design a Simulated Annealing (SA) algorithm for solving the problem. The approximate optimal solutions obtained by the SA algorithm are compared with those obtained by the ILS-GA. The Gap_1 of objective 1 (container quantity) and Gap_2 of objective 2 (transportation time) values obtained by ILS-GA and SA are calculated, respectively. The results are shown in Table 14.

The GAP values of the two optimization objectives are plotted in a bar chart, as shown in Fig. 22. For objective 1 (container quantity), the ILS-GA algorithm has eight instances with fewer containers than the SA solution, and the rest are equal to SA, indicating that ILS-GA achieves a higher loading rate. For objective 2 (transportation time), SA outperforms ILS-GA in only 4 instances, demonstrating that ILS-GA has better optimization effects on transportation time.

Due to the different optimization processes of the genetic algorithm (GA) and the simulated annealing (SA) algorithm, GA optimizes the population, while SA optimizes individual solutions. To compare the convergence and optimization performance of the

Table 12
LCL loading scheme (medium-scale).

Container ID	Pallet ID	Cargo ID	Length of pallet /cm	Departure cycle
1	1	1 and 2	197	2
	2	3	99	
	3	4 and 5	198	
2	4	6 and 8	180	
	5	7	97	
	6	9	101	
	7	10	158	
	8	11 and 12	206	
3	9	13	108	1
	10	14 and 15	183	
	11	16	162	
	12	17	209	
4	13	18	149	
	14	19	159	
	15	20	121	
	16	21	151	
	17	22	159	
	18	23	151	
	19	24	137	
	20	25	121	

Table 13
Comparison of Cplex and ILS-GA solving performance for medium-scale instances.

Instance No.	*CPLEX*			*ILS − GA*			Gap_1/%	Gap_2/%
	C /pc	*TT /day*	*RT /s*	*C /pc*	*TT /day*	*RT /s*		
1	5	28	7231	5	26	31	3.70	99.57
2	4	29	7232	4	25	23	7.41	99.68
3	6	26	7222	5	26	26	9.09	99.64
4	5	23	7230	5	22	26	2.22	99.64
5	5	28	7219	4	25	32	15.61	99.56
6	5	25	7230	5	25	35	0	99.52
7	5	23	7224	4	22	27	12.87	99.63
8	4	30	7258	4	28	21	3.45	99.71
9	6	26	7235	6	24	35	4	99.52
10	6	32	7230	5	28	32	14.63	99.56
11	6	27	7225	6	24	23	5.88	99.68
12	4	26	7244	4	26	28	0	99.61
13	4	27	7257	4	24	21	5.88	99.71
14	5	31	7225	5	28	30	5.08	99.58
15	5	27	7218	4	27	23	11.11	99.68
16	5	25	7218	5	22	35	6.38	99.52
17	4	23	7217	4	23	27	0	99.63
18	6	32	7242	5	29	27	13.17	99.63
19	5	25	7224	5	23	22	4.17	99.70
20	6	23	7222	5	22	21	10.93	99.71

two optimization algorithms during the iterative process, this study uses the fitness evaluation times as the x-axis and the optimal objective value as the y-axis to plot the iteration curves of ILS-GA and SA, The iterative curves for the first five instances are shown in Figs. 23 (as shown in Appendix A).

Based on the information provided in Figs. 23(a), 23(b), and 23(d), The SA algorithm converges to the optimal solution around 950, 1100, and 2164 evaluation times, respectively, but gets trapped in local optima. In contrast, after more evaluation times, the ILS-GA converges to the optimal solution, but the convergent objective values are better than those of the SA algorithm. This indicates that during the iterative process, the SA algorithm prematurely converges without searching for the global optimum, while the ILS-GA can search for and find higher-quality optimal solutions within a certain time frame, demonstrating better optimization performance. In Figs. 23(c) and 23(e), both the SA and ILS-GA converge to the same optimal solution, but the ILS-GA achieves this with fewer evaluation times, showing higher optimization efficiency. In summary, the results suggest that the ILS-GA outperforms the SA algorithm regarding global search capability and optimization efficiency, even though the SA algorithm may converge faster in some cases.

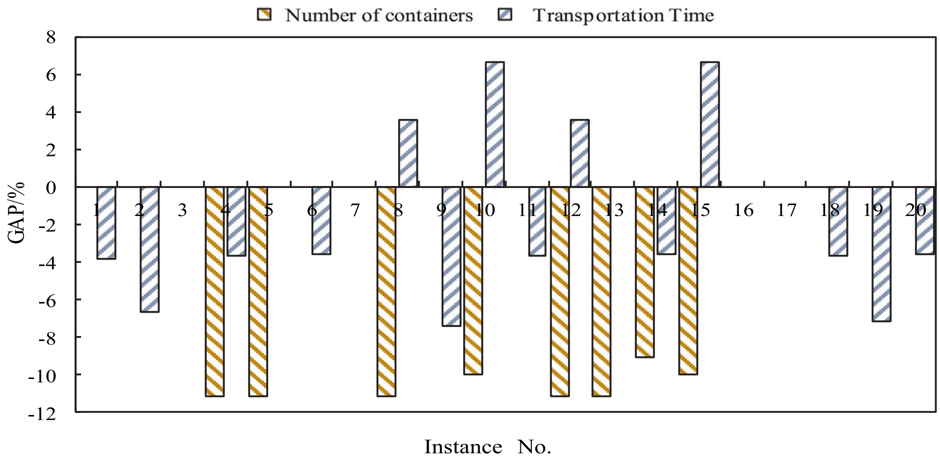


Fig. 22. GAP histogram of ILS-GA vs. SA objective values.

Table 14
Comparison of SA and ILS-GA solving performance for large-scale instances.

Instance No.	*ILS − GA*		*SA*		Gap_1/%	Gap_2/%
	C /pc	*TT /day*	*C /pc*	*TT /day*		
1	9	26	9	27	0	−3.85
2	10	30	10	32	0	−6.67
3	10	31	10	31	0	0
4	9	27	10	28	−11.11	−3.70
5	9	29	10	29	−11.11	0
6	8	28	8	29	0	−3.75
7	10	28	10	28	0	0
8	9	28	10	27	−11.11	3.57
9	10	27	10	29	0	−7.41
10	10	30	11	28	−10	6.67
11	10	27	10	28	0	−3.70
12	9	28	10	27	−11.11	3.57
13	9	28	10	28	−11.11	0
14	11	28	12	29	−9.09	−3.57
15	10	30	11	28	−10	6.67
16	12	28	12	28	0	0
17	10	28	10	28	0	0
18	11	27	11	28	0	−3.70
19	13	28	13	30	0	−7.14
20	11	28	11	29	0	−3.57

5.5. Algorithm analysis

5.5.1. Initialization strategies analysis

To verify the improvement in the solving ability of the ILS-GA with different initialization strategies, this paper conducted a comparative study based on 12 instances, using six initialization strategies and a completely random initialization strategy. The calculation of the objective value GAP before and after adding the various initialization strategies is as follows: $\text{Gap} = (l_1 - l_2)/l_1 \cdot 100\%$, Where l_1 and l_2 represent the objective values obtained by the ILS-GA with the completely random initialization strategy and the ILS-GA with the added initialization strategies, respectively. The solving results are shown in Table 15. The average objective value obtained by the ILS-GA using the completely random initialization strategy is 1.138. The average objective value obtained by the ILS-GA using the various initialization strategies is 1.088. The average GAP (objective value gap) between the random and different initialization strategies is 4.27%. These results indicate that adding the various initialization strategies has enriched the diversity of the initial population, which in turn has improved the quality of the solutions obtained. Compared with the ILS-GA using the random initialization method, the ILS-GA with the initialization strategy can find more optimal solutions and improve performance.

This study designed six different initialization strategies for the genetic algorithm: Sorting items by volume, length, width, height, arrival order, and random order. To validate which initialization strategy provides the best improvement in solution quality, we

Table 15
Initialization strategy and random strategy for GA solution values.

Instance $No.$	Objective Values		Gap%
	Random strategy	Initialization strategy	
1	1.156	1.023	11.51
2	1.123	1.082	3.65
3	1.116	1.083	2.96
4	1.158	1.125	2.85
5	1.094	1.077	1.55
6	1.126	1.103	2.04
7	1.187	1.109	6.57
8	1.096	1.048	4.38
9	1.188	1.142	3.87
10	1.109	1.064	4.06
11	1.187	1.126	5.14
12	1.11	1.08	2.70
Average Value	1.138	1.088	4.27

added the six strategies to an instance and ran it five times. Then the iteration curves were plotted as shown in Fig. 24 (as shown in Appendix B). Among these strategies, sorting items by arrival order emerged as the most effective, achieving the best mean objective value. This outcome is attributed to organizing items based on their arrival sequence adhering to the first-come-first-served principle, thereby reducing the waiting time for items that arrive earlier. This approach directly contributes to the optimization of transportation time, which is the study's primary objective.

5.5.2. Local search strategies analysis

To verify the improvement effect of the designed iterative local search (ILS) algorithm on the genetic algorithm (GA) solution, based on 12 instances, this study compares the objective function values of GA before and after adding ILS. The GA without ILS is referred to as the traditional GA, and the GA combined with the simulated annealing algorithm is called the hybrid GA, where the annealing strategy of simulated annealing is used as the local search strategy of GA, as shown in the pseudo-code in Algorithm 4.

To verify the performance of the ILS-GA algorithm proposed in this study compared to the traditional genetic algorithm (GA) and the hybrid genetic algorithm (Hybrid GA), The following three GAP indicators were calculated: $Gap_1 = (l_1 - l_2)/l_1 \cdot 100\%$ (the GAP of the objective values between the traditional GA and the Hybrid GA), $Gap_2 = (l_1 - l_3)/l_1 \cdot 100\%$ (the GAP of the objective values between the traditional GA and the ILS-GA), $Gap_3 = (l_2 - l_3)/l_2 \cdot 100\%$ (the GAP of the objective values between the Hybrid GA and the ILS-GA). Where l_1, l_2 and l_3 represent the objective values of the traditional GA, Hybrid GA, and the ILS-GA proposed in this study, respectively. As shown in Table 16, the average objective value of the ILS-GA is 1.063, which is better than the results of 1.1 for the GA and 1.078 for the Hybrid GA. The average Gap_1 between the GA and the Hybrid GA is 1.99%, and the average Gap_2 between the GA and the ILS-GA is 3.39%, indicating that ILS has an improving effect on the GA solution. The average Gap_3 between the Hybrid GA and the ILS-GA is 1.42%, showing that this study's ILS strategy has better optimization performance than the simulated annealing-based local search strategy.

Algorithm 4: Simulated Annealing Local Search.

Function SALS():
  $S^* = S_0$; // Initialize the initial solution to the optimal solution
  while $T_0 > T_{min}$ **do**
    $S_1 = LocalSearch(S_0)$; // Local search for the current solution
    $doubleDE = S_0 - S_1$
    // Determine if a neighborhood solution is accepted
    if $(DE < 0)||(exp - (DE/startT) > rand(0,1))$ **then**
      $S^* = S_1$;
    end
    if $(NoChangeInSeveralConsecutiveObjectiveValues)$ **then**
      break;
    end
    $T_0 = T_0 \cdot \delta$
  end
  return S^* ; // return the optimal solution

The iteration curves of the best individual values of the three algorithm populations were plotted to verify the convergence and optimization performance of the three optimization algorithms.

Table 16
Comparison of objective values of three genetic algorithms.

Instance *No.*	Objective Values			$Gap_1\%$	$Gap_2\%$	$Gap_3\%$
	GA	Hybrid GA	ILS-GA			
1	1.065	1.032	1.032	3.1	3.1	0
2	1.123	1.123	1.107	0	1.42	1.42
3	1.114	1.102	1.065	1.08	4.40	3.36
4	1.114	1.097	1.08	1.53	3.05	1.55
5	1.086	1.016	1.006	6.45	7.37	0.98
6	1.12	1.097	1.097	2.05	2.05	0
7	1.069	1.069	1.069	0	0	0
8	1.145	1.112	1.112	2.88	2.88	0
9	1.094	1.083	1.078	1.01	1.46	0.46
10	1.103	1.086	1.033	1.54	6.35	4.88
11	1.042	1.033	1.028	0.86	1.34	0.48
12	1.125	1.086	1.043	3.47	7.29	3.96
Average Value	1.1	1.078	1.063	1.99	3.39	1.42

Since the Hybrid GA and the ILS-GA added local search strategies, the iteration time for each initial solution and new offspring population in each generation is longer than the traditional genetic algorithm. Therefore, comparing the number of iterations to the x-axis is relatively unfair. Instead, the x-axis represents the fitness evaluation times, and the y-axis represents the objective value, The iterative curves for the first five instances are shown in Fig. 25 (as shown in Appendix C), the ILS-GA was able to obtain the optimal solution in each group, compared to the GA and the Hybrid genetic algorithm. The GA and the Hybrid GA both exhibited premature convergence to local optimal solutions. When the GA or the Hybrid GA converged to the same objective value as the ILS-GA in this study, they required more computational iterations. This indicates that the ILS strategy designed in this study expanded the search range and improved the search quality of the solutions, demonstrating better optimization performance and convergence.

5.6. Management insights

This study focuses on optimizing the LCL loading process for LCL cargoes on the CR Express, providing multifaceted and valuable practical guidance for its operation and management.

(1) With the LCL cargo loading scheme provided by this study, the operating enterprises of CR Express can accurately plan the palletizing and container-loading process of cargoes. Based on information such as the size, weight, and destination of the cargoes, they can reasonably arrange the layout of the cargoes on pallets and in containers, improve the space utilization rate of containers, and achieve economic benefit growth.

(2) This study's optimization scheme for the departure cycle of LCL cargo containers allows enterprises to flexibly adjust the issuance quantity of LCL containers and the departure cycle of containers in each period according to the actual cargo volume and transportation demand, maximizing shippers overall satisfaction.

(3) With this study's algorithm design, CR Express's operating enterprises can integrate the algorithm into the logistics management information system, realizing the automatic generation and optimization of loading schemes, reducing the risks of low-efficiency and error-prone traditional manual loading planning methods, and achieving digital and intelligent operation.

6. Conclusions and discussions

6.1. Conclusions

Focusing on the LCL transportation scenario of CR Express, this study tackles two key challenges that railway operators face regarding LCL cargo loading and container scheduling. The study incorporates a time-optimization objective function to enhance shipper satisfaction into a multi-objective mixed-integer programming model. This model aims to minimize the number of containers used and the longest cargo transportation time. The model determines the specific loading plans for pallets and containers in LCL cargo and the scheduling plans for containers. The accuracy and effectiveness were confirmed through validation using small-scale instances. An improved genetic algorithm was designed and implemented to address large-scale problems. Additionally, a customized genetic algorithm with an iterative local search (ILS-GA) was developed to optimize cargo LCL comprehensively. A two-stage encoding and decoding scheme was tailored for LCL cargo's pallet and container loading. By designing operators and integrating iterative local search algorithms to expand the search space, this approach demonstrated higher solution efficiency and quality in medium and large-scale instances compared to solvers and traditional simulated annealing algorithms. Experiments were also conducted to analyze initialization and local iterative search strategies, verifying the improvement effects of the proposed strategies on ILS-GA solutions in this study.

In summary, with the development of information technology, the logistics industry is moving in the direction of intelligence (Wang et al., 2024). This study develops mathematical models and optimization algorithms to overcome the limitations of manual LCL cargo loading and scheduling in CR Express, providing operators with an integrated solution throughout the process. This approach achieves high efficiency in LCL cargo and container scheduling, enhances customer satisfaction, and promotes the digital and intelligent development of CR Express's LCL business.

6.2. Discussions

Compared to existing research, this study addresses the limitations of manual loading and planning scheduling for LCL cargo on the CR Express. Designing a mixed-integer programming model and heuristic algorithms greatly improves cargo loading efficiency and customer satisfaction, enabling intelligent transportation of LCL cargo on the CR Express and filling the research gap in intelligent LCL cargo consolidation.

However, this study still has some limitations. In establishing the two-stage mixed-integer programming model for LCL cargo loading, the model only considers the geometric constraints of non-overlapping and placement of cargo to improve loading efficiency. However, in reality, the balance of container weight is also a crucial factor that CR Express operators pay close attention to, as it affects the stability of cargo transportation and is closely related to cargo damage and transport safety. In future research, we will focus on the following aspects: Firstly, considering that the balance of cargo during transportation is crucial for reducing cargo damage and ensuring transportation safety, and is a significant concern for operators, we will incorporate cargo balance constraints into our mathematical modeling in future studies. Secondly, in the context of the less-than-container-load (LCL) problem, each piece of cargo comes from different shippers and varies in shape and size. To maintain the general applicability of our research, we have assumed cargo to be of regular shapes. Moving forward, to better align with actual LCL situations, we plan to explore the consolidation of irregularly shaped cargo in future studies.

CRediT authorship contribution statement

Yanjie Zhou: Supervision. **Zhanwen He:** Software, Methodology. **Chengcheng Liu:** Writing – review & editing, Writing – original draft, Formal analysis. **Jingrong Zhang:** Supervision, Funding acquisition. **Yumin Li:** Funding acquisition. **Yan Wang:** Resources, Investigation.

Declaration of competing interest

The author(s) declared no potential conflicts of interest concerning this article's research, authorship, and/or publication.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (72201252), the Shanghai Special Project for Promoting High-Quality Industrial Development, China (212102), the Major Project in Fundamental of Philosophy and Social Sciences in Universities of Henan Province, China (2024-JCZD-20), and Key Soft Science Project in Henan Province, China (242400411138).

Appendix A

In large-scale instance experiments, To compare the convergence and optimization performance of the two optimization algorithms during the iterative process, this study uses the fitness evaluation times as the x-axis and the optimal objective value as the y-axis to plot the iteration curves of GA and SA, as shown in Fig. 23.

Appendix B

Regarding the initialization strategies of the algorithm, six strategies were designed in this study: sorting by volume, length, width, height, arrival order, and random order. To verify which initialization strategy has the best effect on improving the solution quality, five instances of each strategy were analyzed, and the iteration curves were plotted, as shown in Fig. 24.

Appendix C

To explore the effect of the iterative local search strategy on the algorithms' performance, the performance of the traditional genetic algorithm, the simulated annealing algorithm, and the ILS-GA algorithm proposed in this study are compared. The iteration curves of the best individual values of the three algorithm populations were plotted, as shown in Fig. 25.

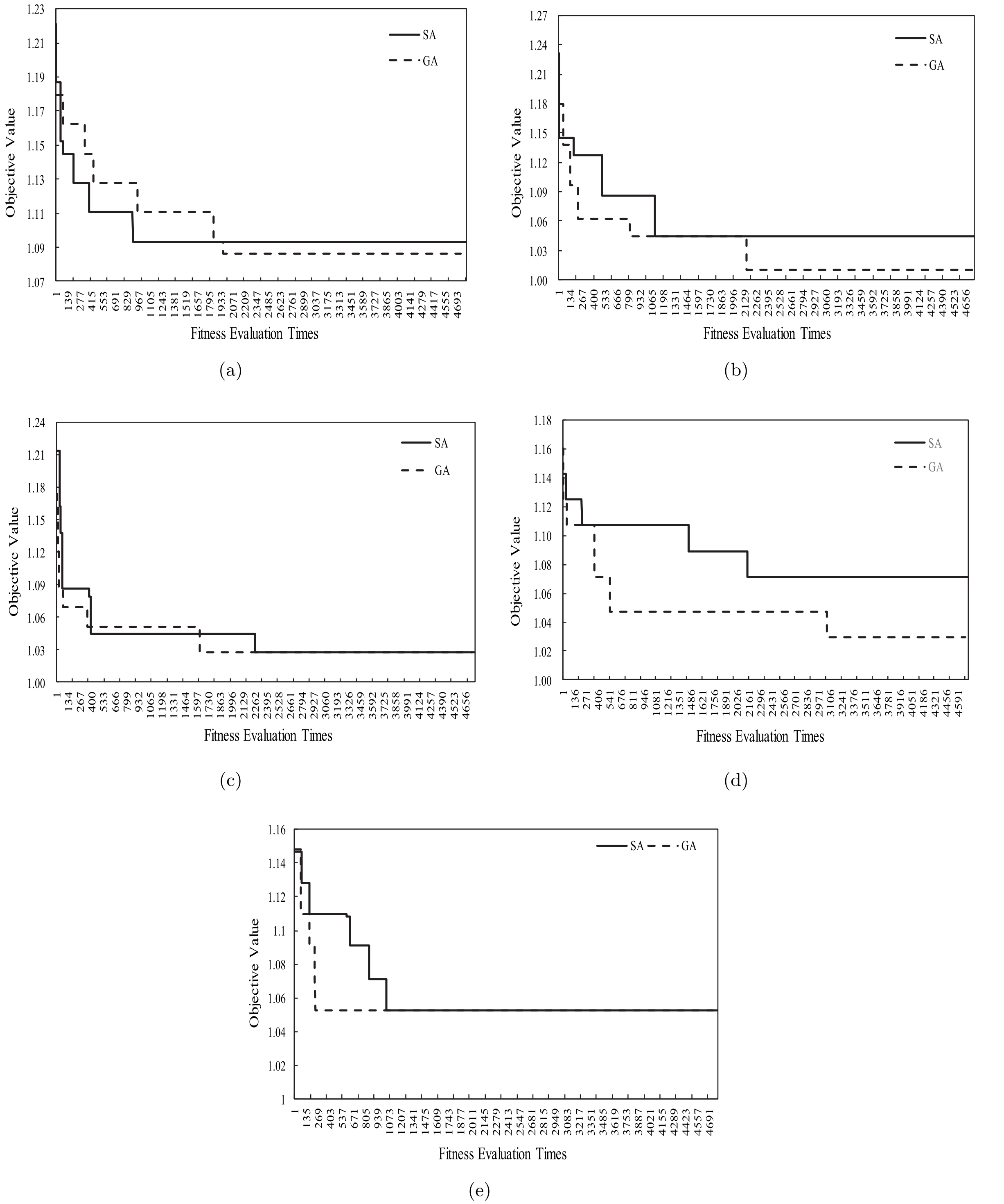


Fig. 23. Iterative Plot of GA vs. SA Solving Instances.

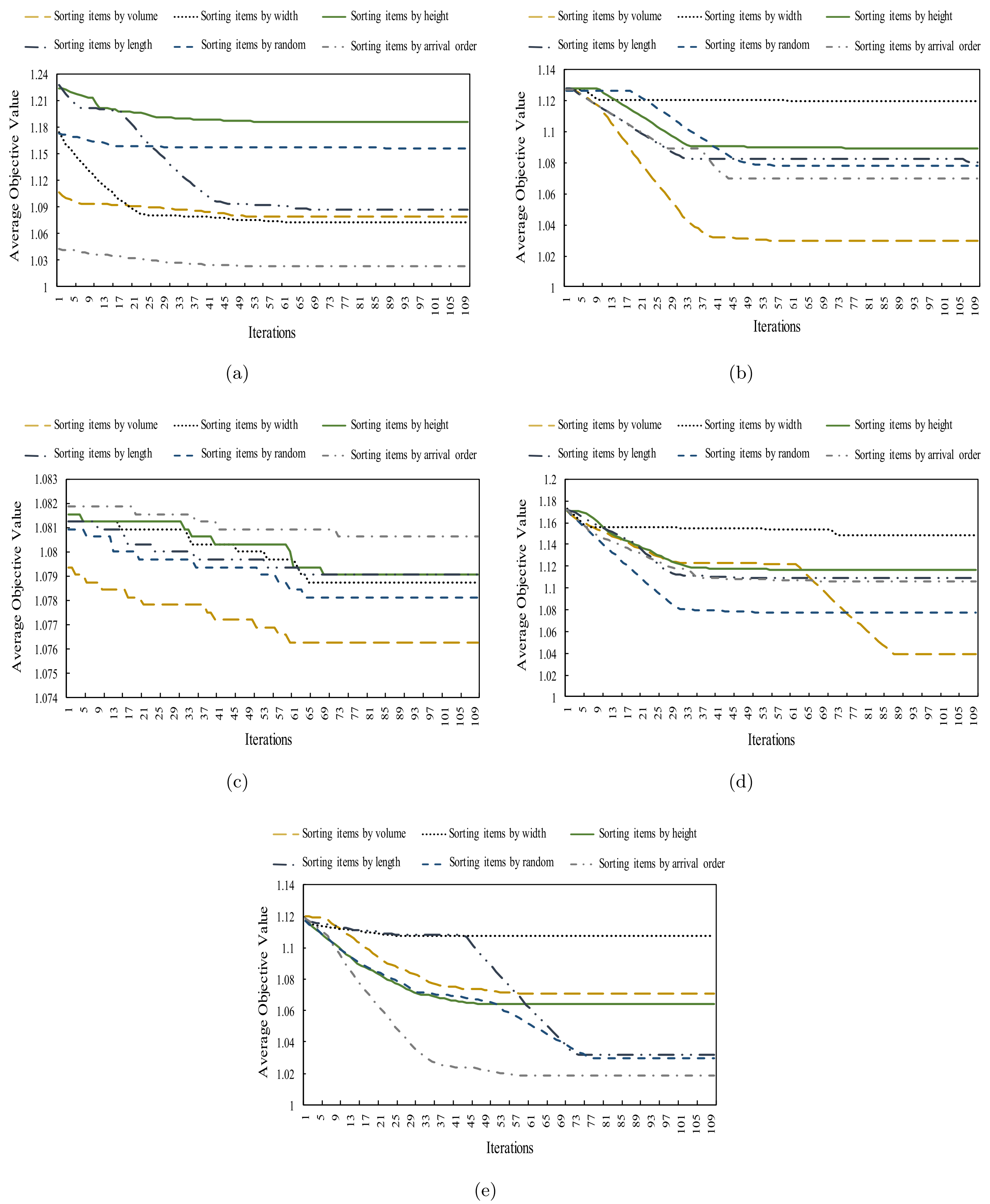


Fig. 24. Multiple Initialization Strategy GA Iteration Curve.

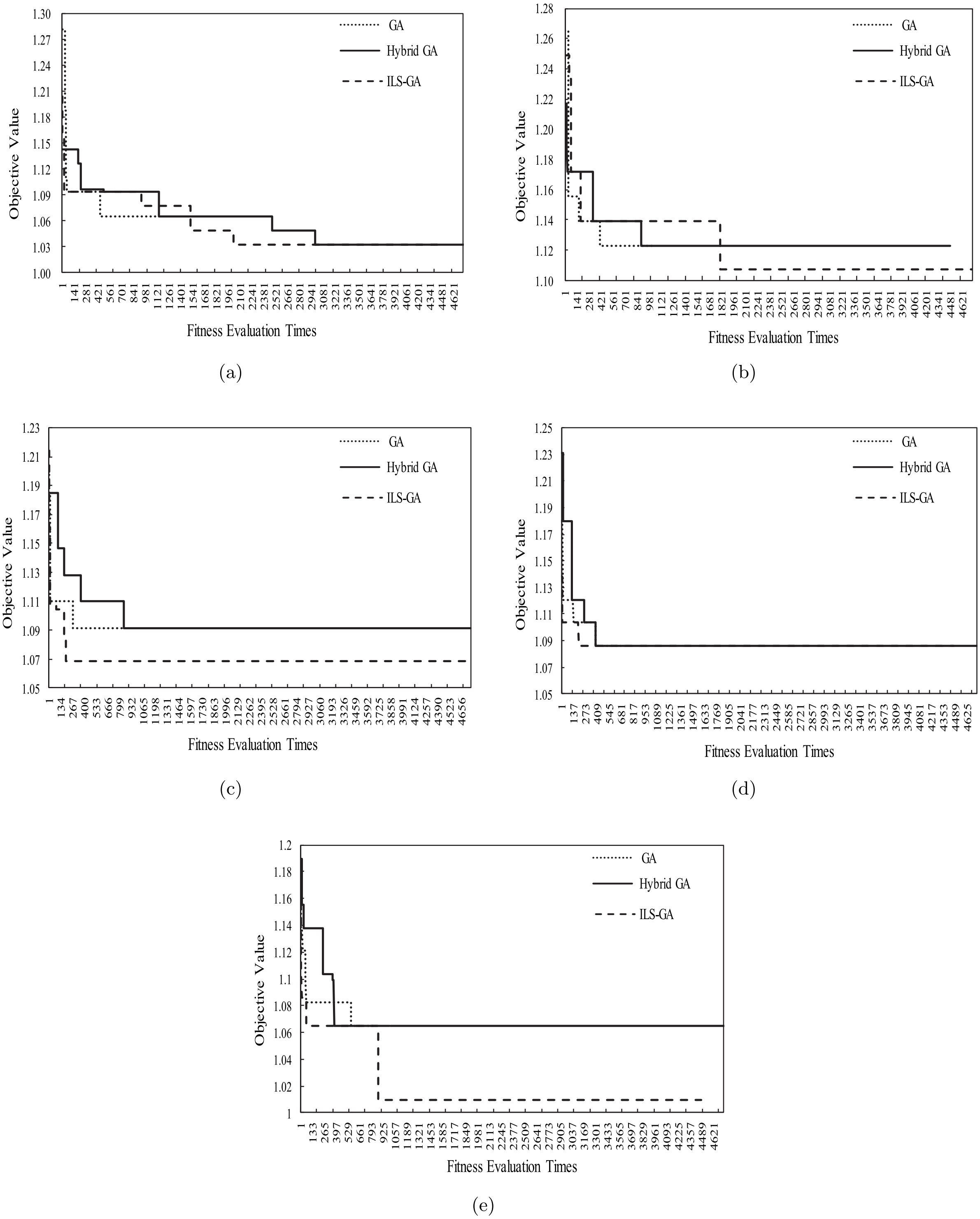


Fig. 25. Convergence and Optimization Performance of Three Optimization Algorithms.

References

Alonso, M.T., Alvarez-Valdes, R., ParreAo, F., 2014. A reactive GRASP algorithm for the container loading problem with load-bearing constraints. Eur. J. Ind. Eng. 8, 669–694, URL https://api.semanticscholar.org/CorpusID:109266094.

Alonso, M.T., Alvarez-Valdés, R., Parreño, F., 2019. Mathematical models for multi container loading problems with practical constraints. Comput. Ind. Eng. 127, 722–733, URL https://api.semanticscholar.org/CorpusID:29331162.

Alonso, M.T., Alvarez-Valdés, R., Parreño, F., 2020. A GRASP algorithm for multi container loading problems with practical constraints. 4OR 18, 49–72, URL https://api.semanticscholar.org/CorpusID:126889909.

Anibal Tavares de Azevedo, R.C.M., et al., 2014. Solving the 3D container ship loading planning problem by representation by rules and meta-heuristics, 6. pp. 228–260, URL https://api.semanticscholar.org/CorpusID:9864330.

Bi-Chao Bang, Y.-W., 2011. Research on vehicle routing problem with 3D loading constraints based on tabu search. Comput. Eng. 37, 190–191,194, URL https://www.ecice06.com/EN/10.3969/j.issn.1000-3428.2011.11.065.

Che, C.H., Huang, W., Lim, A., Zhu, W., 2011. The multiple container loading cost minimization problem. European J. Oper. Res. 214, 501–511, URL https://api.semanticscholar.org/CorpusID:2753469.

Deidson Vitorio Kurpel, C.T.S., et al., 2020. The exact solutions of several types of container loading problems. 284. pp. 87–107, URL https://api.semanticscholar.org/CorpusID:212738299.

Huang, W., He, K., 2007. An efficient algorithm for solving the container loading problem. In: Combinatorics, Algorithms, Probabilistic and Experimental Methodologies. pp. 396–407. http://dx.doi.org/10.1007/978-3-540-74450-4_36, URL https://api.semanticscholar.org/CorpusID:38688260.

Iwasawa, H., Hu, Y., Hashimoto, H., Imahori, S., Yagiura, M., 2016. A heuristic algorithm for the container loading problem with complex loading constraints. J. Adv. Mech. Des. Syst. Manuf. 10, URL https://api.semanticscholar.org/CorpusID:114522495.

Jamrus, T., Chien, C.-F., 2016. Extended priority-based hybrid genetic algorithm for the less-than-container loading problem. Comput. Ind. Eng. 96, 227–236. http://dx.doi.org/10.1016/j.cie.2016.03.030, URL https://www.sciencedirect.com/science/article/pii/S036083521630105X.

Junqueira, L.V., Morabito, R., Yamashita, D.S., Yanasse, H.H., 2012. Optimization models for the three-dimensional container loading problem with practical constraints. URL https://api.semanticscholar.org/CorpusID:117460493.

Li, B., Jiang, S., Zhou, Y., Xuan, H., 2023. Optimization of train formation plan based on technical station under railcar demand fluctuation. J. Ind. Prod. Eng. 40 (6), 448–463.

Liao, Y., Shan, H., Song, W., 2023. Container loading optimization based on hybrid heuristic algorithm. Manuf. Autom. 45 (5), 124–128.

Moura, A., Bortfeldt, A., 2017. A two-stage packing problem procedure. Int. Trans. Oper. Res. 24, 43–58, URL https://api.semanticscholar.org/CorpusID:205220361.

Pisinger, D., Sigurd, M., 2005. The two-dimensional bin packing problem with variable bin sizes and costs. Discrete Optim. 2 (2), 154–167. http://dx.doi.org/10.1016/j.disopt.2005.01.002.

Tan, A.W.K., Zhao, Y., Halliday, T., 2018. A blockchain model for less container load operations in China. Int. J. Inf. Syst. Supply Chain Manag. 11, 39–53, URL https://api.semanticscholar.org/CorpusID:51868629.

Tiwari, S., Wee, H.M., Zhou, Y., Tjoeng, L., 2021. Freight consolidation and containerization strategy under business as usual scenario & carbon tax regulation. J. Clean. Prod. 279, 123270.

Upadhyay, A., Gu, W., Bolia, N.B., 2017. Optimal loading of double-stack container trains. Transp. Res. Part E- Logist. Transp. Rev. 107, 1–22, URL https://api.semanticscholar.org/CorpusID:115787618.

Wang, Y., Luo, S., Fan, J., Zhen, L., 2024. The multidepot vehicle routing problem with intelligent recycling prices and transportation resource sharing. Transp. Res. Part E: Logist. Transp. Rev. URL https://api.semanticscholar.org/CorpusID:268813311.

Wang, Y., Peng, S., Zhou, X., Mahmoudi, M., Zhen, L., 2020. Green logistics location-routing problem with eco-packages. Transp. Res. Part E- Logist. Transp. Rev. 143, 102118, URL https://api.semanticscholar.org/CorpusID:225115341.

Wang, Y., Wei, Y., Wang, X., Wang, Z., Wang, H., 2022a. A clustering-based extended genetic algorithm for the multidepot vehicle routing problem with time windows and three-dimensional loading constraints. Appl. Soft Comput. 133, 109922, URL https://api.semanticscholar.org/CorpusID:254661508.

Wang, Y., Zhe, J., Wang, X., Fan, J., Wang, Z., Wang, H., 2022b. Collaborative multicenter reverse logistics network design with dynamic customer demands. Expert Syst. Appl. 206, 117926, URL https://api.semanticscholar.org/CorpusID:250028885.

Wei, Q., Xu, Y., Li, C., Zhang, Y., 2019. Efficiency evaluation of LCL transshipment at port railway container intermodal terminal. J. Coast. Res. 83, 456–464, URL https://api.semanticscholar.org/CorpusID:134851325.

Wu, H., Chen, S., Lu, M., Cui, W., 2022. Forecast and analysis of railway logistics pallet sharing demand based on regression model. In: Other Conferences. URL https://api.semanticscholar.org/CorpusID:246663279.

Xiao, H., 2011. Model and calculation for foreign trade container shipping mode. In: 2011 International Conference on Computer Science and Service System. CSSS, pp. 2331–2334, URL https://api.semanticscholar.org/CorpusID:2937198.

Xin, J., Yuan, Q., D'Ariano, A., Guo, G., Liu, Y., Zhou, Y., 2024. Dynamic unbalanced task allocation of warehouse AGVs using integrated adaptive large neighborhood search and kuhn–munkres algorithm. Comput. Ind. Eng. 195, 110410.

Zhao, X., Bennell, J.A., Bektaş, T., Dowsland, K.A., 2016. A comparative review of 3D container loading algorithms. Int. Trans. Oper. Res. 23, 287–320, URL https://api.semanticscholar.org/CorpusID:26214439.

Zhou, K., 2018. The pallet loading method of single category cargo based on railway containerized transport. In: Proceedings of the 2018 10th International Conference on Computer and Automation Engineering. URL https://api.semanticscholar.org/CorpusID:4902669.

Zhou, Y., Lee, G.M., 2020. A bi-objective medical relief shelter location problem considering coverage ratios. Int. J. Ind. Eng. 27 (6), 971–988..

Zhou, Q., Liu, X., 2017. A swarm optimization algorithm for practical container loading problem. In: IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society. pp. 5690–5695, URL https://api.semanticscholar.org/CorpusID:24649749.