systems

Article

Pickup and Delivery Problem of Automobile Outbound Logistics Considering Trans-Shipment among Distribution Centers

Yu Wang *, Renrong Zheng, Yan Zhao and Chengji Liang

Logistics Research Center, Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China; 202030510287@stu.shmtu.edu.cn (R.Z.); 201930510064@stu.shmtu.edu.cn (Y.Z.); liangcj@shmtu.edu.cn (C.L.)
* Correspondence: wangyu@shmtu.edu.cn; Tel.: +86-021-3828-4620

Abstract: This paper considers a pickup and delivery problem in automobile logistics. In the daily operations of a third-party logistics company (3PL), decisions must be made for two kinds of demands: delivering finished automobiles from an outbound warehouse to distribution centers (DCs) and transferring automobiles among the DCs according to specific customer orders. The problem is to assign a set of automobiles to a set of heterogeneous auto-carriers and deliver them to their destinations considering the outbound and transfer demands. Each automobile is assigned a value indicating its urgency level to be handled and a car type: small, medium, or large. Each of the auto-carriers has a specific number of slots with different types indicating the largest size of an automobile that can be loaded into the slot. An integer programming (IP) model is formulated for the problem to maximize the total loaded value and minimize the total transportation cost depending on the routing of the carriers. An improved adaptive large neighborhood search algorithm is developed to solve the problem efficiently, where a heuristic generates an initial solution, and a series of operators update the solution iteratively. Experimental results based on multi-scale instances show that the proposed algorithm can generate near-optimal solutions in an acceptable amount of time, and outperforms solving the IP model directly by CPLEX to a large extent. The algorithm can help 3PL companies make efficient and economical decisions in daily operations.

Keywords: automobile outbound logistics; pickup and delivery; trans-shipment among distribution centers; adaptive large neighborhood search

Citation: Wang, Y.; Zheng, R.; Zhao, Y.; Liang, C. Pickup and Delivery Problem of Automobile Outbound Logistics Considering Trans-Shipment among Distribution Centers. *Systems* **2023**, *11*, 457. https://doi.org/10.3390/systems11090457

Academic Editor: Vladimír Bureš

Received: 8 August 2023
Revised: 30 August 2023
Accepted: 1 September 2023
Published: 3 September 2023

1. Introduction

The automotive industry plays an essential role in global economic development. In 2022, the global sales of passenger cars exceeded 57 million [1]. To meet the increasing and diversified customer demands, automotive manufacturers have always had a massive outbound network of multi-distribution centers. Usually, finished automobiles are transported from an outbound warehouse near the manufacturing plant to different distribution centers through the network using special transporters known as auto-carriers. Each auto-carrier has a specific number of slots, each of which is capable of carrying one automobile.

In daily operations, a third-party logistics company (3PL) makes decisions for automobile distributions considering two kinds of demands as follows. The first ones are basic outbound distribution demands, in which a set of automobiles in the manufacturer's outbound warehouse needs to be assigned to a set of heterogeneous auto-carriers and delivered to the corresponding distribution centers (DCs). The second kind of demand arises from actual situations in which a group of automobiles has to be transferred from one distribution center to another such that some specific customer orders could be satisfied as soon as possible. The amount of transferring demands among DCs is usually smaller than that of the outbound demands from the warehouse but with higher priority. The 3PLs always handle the two kinds of demands separately in practice, resulting in a high empty rate of auto-carriers during transportation and high logistics costs. The outbound demands and transfer demands should be considered simultaneously since the origins of the transfer demands may also be the destinations of the outbound demands. Therefore, this paper studies a pickup and delivery problem of automobile logistics considering both basic outbound demands and the transfer demands among distribution centers. Based on our investigation of 3PL companies in China, the problem involves the following three aspects.

First, the 3PL company must determine which automobiles should be loaded onto auto-carriers preferentially. In realistic conditions, the number of auto-carriers available each day is limited, meaning that not all finished automobiles can be loaded simultaneously. Therefore, each automobile, from an outbound warehouse or a distribution center, has a given value indicating its level of urgency to be loaded. The higher the value is, the more urgently the automobile should be delivered. In daily operations, the 3PL company should prioritize loading automobiles with higher values onto the available auto-carriers as much as possible.

Second, the total transportation cost of auto-carriers must be taken into account. The transportation cost of an auto-carrier consists of a fixed cost if it is used, and a traveling cost depending on the routing distance of the auto-carrier. The routing distance of each auto-carrier is defined by the visiting sequence among the origins and destinations of the automobiles that are loaded onto it. Based on the demand type, the origin of each automobile could be either the outbound warehouse or a distribution center. Different loading combinations of automobiles assigned to each auto-carrier will thus lead to various routes resulting in different transportation costs.

Third, a downward compatible loading structure (DCLS) constraint should be met for each auto-carrier. As the examples shown in Table 1, without loss of generality, automobiles are classified into three types, small, medium, and large, according to their appearance and size. Accordingly, each slot on the auto-carriers is also indexed as one of the three types, indicating the largest car type that can be loaded into the slot. Namely, a large slot can accommodate all three types of automobiles, a medium slot can load medium or small automobiles, and a small slot can only load small automobiles. Figure 1 shows two different loading patterns for a simple auto-carrier as an example based on DCLS constraints. Obviously, there could be many feasible loading patterns even for an auto-carrier with a given slot configuration.

Table 1. Examples of three types of automobiles.

Type	Standards			Examples				
	Length (mm)	Width (mm)	Height (mm)	Manufacturer	Brand	Length (mm)	Width (mm)	Height (mm)
Small	≤4650	≤2000	≤1550	SAIC	MG3	3999	1728	1517
				GM	Aveo	4039	1735	1517
				Volkswagen	Polo	4176	1650	1465
Medium	4650–4850	≥2000	1550–1650	Volkswagen	Santana2000	4680	1700	1423
				Volkswagen	Santana2000	4687	1700	1450
				GM	Epica	4808	1807	1450
Large	≥4850	≥2000	≥1650	Volkswagen	Touran	4411	1794	1670
				SAIC	R950	4996	1857	1502
				GM	Buick GL8	5213	1847	1750

To sum up, our problem is to pick up a set of automobiles at their origins and deliver them to the corresponding destinations such that the total value of the assigned automobiles is maximized and the total transportation cost of the occupied auto-carriers is minimized, subject to a special DCLS constraint of a limited number of heterogeneous auto-carriers with different slot configurations. The problem becomes very complex because the outbound demands and transfer demands are considered simultaneously. In order to make the decisions efficiently, we design an improved adaptive large neighborhood search algorithm to solve the problem. The remainder of this paper is organized as follows: Section 2 is the literature review, and Section 3 presents a precise problem description and an integer programming model. Section 4 describes the improved adaptive large neighborhood search (IALNS) algorithm. Section 5 shows the experimental results based on different scales of instances, and Section 6 concludes the paper.

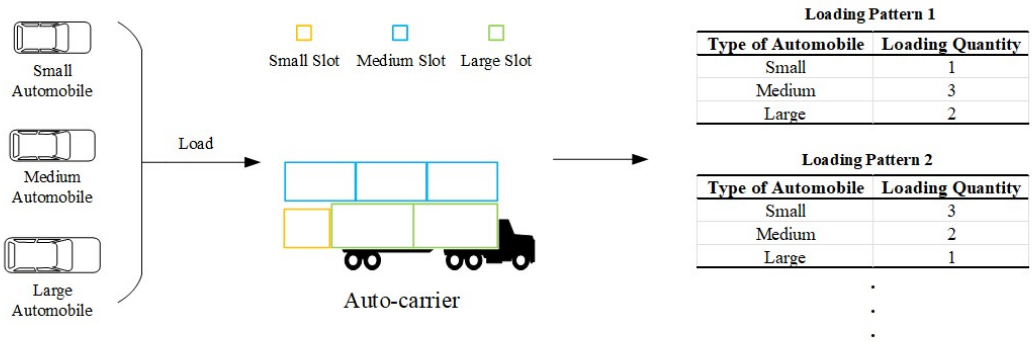

Figure 1. Examples of loading patterns with DCLS constraints.

2. Literature Review

Researchers have proposed numerous models and methods for solving pickup and delivery problems but rarely considered the characteristics of automobile distribution, and none has considered the outbound and transfer demands simultaneously. We review relevant work on classic pickup and delivery problems and automobile transportation, respectively.

2.1. Pickup and Delivery Problems

The pickup and delivery problem (PDP) is a variant of the classic vehicle routing problem (VRP), which was proposed in 1959 by Danzig and Ramser [2]. The first study on the PDP dates back to Min (1989) [3], and the PDPs have attracted attention from numerous perspectives.

Some scholars consider the PDP with different time constraints, and many heuristics have been proposed for specific scenarios. Stefan and David [4] studied a pickup and delivery problem with a time window, and an adaptive large neighborhood search heuristic is proposed. Yuan and Jonathan [5] studied a similar problem, where the trans-shipments between nodes are considered. Can et al. [6] and Olcay et al. [7] used variable neighborhood search algorithms based on the ant colony algorithm and disturbance to solve PDP with a time limit. Chao et al. [8] studied a PDP with time windows and designed a local hybrid search that effectively combined a simulated annealing algorithm to solve this problem. Zheyu Wang et al. [9] considered a PDP with a hard time window and time-dependent travel times, and an exact branch-cut-and-price algorithm was proposed in their study.

Some scholars emphasized different loading constraints for PDPs. Emmanouil et al. [10] considered a two-dimensional loading constraint for a PDP and explained how the relocation would influence the optimal routing. Moura et al. [11] extended the problem proposed by Emmanouil et al. to three-dimensional loading constraints, and a matheuristic approach was designed. The PDPs are also considered in many specific scenarios. Christian and Ralf [12] studied a vehicle routing problem with trans-shipment facilities, where time windows and heterogeneous fleets are taken into consideration, and an adaptive large neighborhood search was proposed. Lais et al. [13] investigated a novel problem from the urban delivery system integrating the routing of vans and porters. An iterated heuristic combing local search was designed. More PDP variants have been studied considering different objectives, methods, and application scenarios [14–19].

Although many mathematical models and methods have been studied to solve different pickup and delivery problems, none of them has considered the unique loading structure of auto-carriers and could not be directly used in our problem.

2.2. Automobile Transportation

There are a handful of studies in existing literature focusing on automobile transportation. Agbegha et al. (1998) [20] is the earliest study focusing on the loading problem of auto-carriers for automobile distribution. They considered the particular structure of the auto-carriers in the early days and focused more on the reduction in reloads when some automobiles must be temporarily unloaded before reaching their destination for a given route of the auto-carrier. Tadei et al. (2002) [21] studied an automobile delivery problem without involving constraints such as loading precedence and compatibility between specific vehicles and slots, and a local search heuristic was proposed by dividing the destinations into small clusters. Dell'Amico et al. (2014) [22] considered a complex capacity constraint but applied a last-in-first-out loading strategy for the automobile distribution problem. They considered that the auto-carriers are suitable enough for delivering all the automobiles and proposed an iterated local search algorithm to minimize the total traveling distance of all auto-carriers. The problem was extended into a dynamic version on a rolling horizon with more constraints [23].

With the development of auto-carriers and third-party logistics (3PL) companies, several studies have focused on specific scenarios. Juárez Pérez et al. (2019) [24] investigated a real-world distribution problem for delivering cars and vans in Mexico. Capacity and time window constraints were considered and a two-phase heuristic algorithm was implemented to solve the problem. Bonassa et al. (2019) [25] proposed a mixed-integer programming formulation to solve a variation of the dynamic multi-period auto-carrier transportation problem. The objective was to find the best combination of vehicles to be loaded on auto-carriers over a multiple-day planning horizon, such that the total transportation cost is minimized and the delivery deadlines are fulfilled to the greatest possible extent. The model is later revised and implemented into real situations in Brazil, and a multi-start local search heuristic is proposed to solve large-scale instances [26].

Currently, automobiles of different sizes and shapes are produced, and the auto-carriers have become standardized. Several studies have also been performed that consider loading patterns and special loading structures. Hu et al. (2015) [27] introduced loading patterns to specify possible ways of loading various types of vehicles onto the auto-carriers, and studied a problem to select from a given set of loading patterns and generate traveling routes based on the selected pattern. Yu Wang et al. (2018) [28] proposed the downward compatible loading structure of auto-carriers according to the actual operation of current automobile outbound logistics. They studied a pickup and delivery problem from multiple warehouses to multiple dealers with known orders and assumed that all pickups have to be complete before any delivery in each shipment. A column generation-based algorithm was designed to solve the problem. A similar loading structure was used in the study proposed by Feng Chen and Yu Wang (2020) [29], where inter-set costs among dealers were considered instead of the routing of auto-carriers.

Although many efforts have been made on solving the loading and routing problems for automobile transportation and on other PDPs, our study differs from the previous studies in the following ways.

(a) A pickup and delivery problem of automobiles is considered with the urgency level of the automobiles and the special downward compatible loading structure (DCLS) of modern auto-carriers. The objective and loading constraints in our problem add more complexity, since different combinations of assigned automobiles would result in varying loading values and delivery routes.

(b) From the perspective of 3PL companies, the outbound demands from the assembly plant and the transfer demands among distribution centers are considered simultaneously. This allows for simultaneous pickup and delivery at some of the DCs during each shipment of the auto-carriers, which makes the decisions more complex but will improve the utility of auto-carriers.

(c) The proposed problem is formulated as an integer programming model, and an improved adaptive large neighborhood search algorithm is proposed with carefully designed operators to efficiently solve large-scale instances in practice.

3. Problem Description and Formulation

In this section, the pickup and delivery of automobiles considering both outbound demands and transfer demands among DCs are described precisely, as well as the DCLS constraints, and the problem is formulated as an integer programming model.

3.1. Problem Description

The pickup and delivery problem of automobiles considering trans-shipment among distribution centers can be described precisely as follows. A 3PL company has J available auto-carriers, denoted by set $\mathbb{J} = \{1, 2, \ldots, J\}$. There are two sets of automobiles that need to be assigned to the auto-carriers. Let set $\mathbb{I}^o = \{1, 2, \ldots, I^o\}$ be the set of automobiles stored in an outbound warehouse and to be delivered to certain distribution centers. Let set $\mathbb{I} = \{1, 2, \ldots, I\}$ be the set of automobiles where each of which is required to be transferred from its origin DC to another. Let $\mathbb{D} = \{1, 2, \ldots, D\}$ be the set of distribution centers involved in the problem, namely the destinations of automobiles in $\mathbb{I}^o$ and the origins and destinations of those in $\mathbb{I}$ all belong to $\mathbb{D}$. Let $\mathbb{I}^* = \{1, 2, \ldots, I^*\} = \mathbb{I}^o \bigcup \mathbb{I}$ be the set of all automobiles.

The automobiles are classified into T types, let $\mathbb{T} = \{1, 2, \ldots, T\}$ indicate the set of different types. For any $t_1, t_2 \in \mathbb{T}$, $t_1 < t_2$ means that type t_1 is smaller than t_2. Similarly, each of the slots on the auto-carriers is also indexed with one of the T types. According to the downward compatible loading structure (DCLS), for any slot of type $t^* \in \mathbb{T}$ on an auto-carrier, all automobiles with a type $t \leq t^*$ can be loaded onto the slot. Each automobile $i \in \mathbb{I}^*$ has a given value $v_i \in \mathbb{V}$ indicating its urgency level to be handled, a type $t_i \in \mathbb{T}$, an origin to be picked up, and a destination $d_i \in \mathbb{D}$ to be delivered to. For auto-carrier $j \in \mathbb{J}$, let p_{jt} be the number of slots of type $t \in \mathbb{T}$. For any two nodes $a, b \in \{\mathbb{D} \bigcup \{O\}\}$, where the O is the outbound warehouse, the traveling cost between a and b is denoted as C_{ab}.

Our problem aims to select a set of automobiles from $\mathbb{I}^*$ with maximized total value such that they can be loaded onto the set of auto-carriers subject to the DCLS constraints, and can be picked up and delivered with a minimized total transportation cost, which includes a traveling cost based on routing decisions and a fixed cost for each auto-carrier used.

3.2. Integer Programming Model

First, we introduce the following sets, parameters, and decision variables.

Sets and Indexes:

- $\mathbb{I}^o$ set of automobiles in the outbound warehouse, $\mathbb{I}^o = \{1, 2, \ldots, I^O\}$.
- $\mathbb{I}$ set of automobiles to be transferred, $\mathbb{I} = \{1, 2, \ldots, I\}$.
- $\mathbb{I}^*$ set of all automobiles, $\mathbb{I}^* = \{1, 2, \ldots, I^*\} = \mathbb{I}^o \cup \mathbb{I}$.
- $\mathbb{J}$ available auto-carriers, $\mathbb{J} = \{1, 2, \ldots, J\}$.
- $\mathbb{D}$ all distribution centers, $\mathbb{D} = \{1, 2, \ldots, D\}$.
- $\overline{\mathbb{D}}$ distribution centers with transfer demands, $\overline{\mathbb{D}} = \{1, 2, \ldots, \overline{D}\}$.
- $\mathbb{D}^*$ the set of all nodes, $\mathbb{D}^* = \mathbb{D} \cup \{O\}$.
- $\mathbb{T}$ types of automobiles and slots, $\mathbb{T} = \{1, 2, \ldots, T\}$.
- $\mathbb{V}$ values of automobiles, $\mathbb{V} = \{1, 2, \ldots, V\}$.

Parameters:

d_i	the destination of automobile $i \in \mathbb{I}^*$, $d_i \in \mathbb{D}^*$.
v_i	the value of automobile $i \in \mathbb{I}^*$, $v_i \in \mathbb{V}$.
t_i	the type of automobile $i \in \mathbb{I}^*$, $t_i \in \mathbb{T}$.
w_{it}	$w_{it} = 1$ if $t_i \geq t, \forall i \in \mathbb{I}^*, t \in \mathbb{T}$, otherwise $w_{it} = 0$
p_{jt}	the number of slots of type $t \in \mathbb{T}$ in auto-carrier $j \in \mathbb{J}$.
C_{ab}	the traveling cost between nodes $a, b \in \mathbb{D}^*$.
f_{id}	$f_{id} = 1$ if the destination of i is d, $\forall i \in \mathbb{I}^*, d \in \mathbb{D}^*$, otherwise $f_{id} = 0$.
f'_{id}	$f'_{id} = 1$ if the origin of i is d, $\forall i \in \mathbb{I}^*, d \in \mathbb{D}^*$, otherwise $f'_{id} = 0$.
α	weight parameter of the total traveling cost.
β	weight parameter of the total value of selected automobiles.
γ	weight parameter of the fixed cost of used auto-carriers.
M	sufficiently large constant.

Decision variables:

$x_{i^o j}$	$x_{i^o j} = 1$ if automobile $i^o \in \mathbb{I}^o$ stored at the outbound warehouse is assigned to auto-carrier $j \in \mathbb{J}$, otherwise $x_{i^o j} = 0$.
y_{ij}	$y_{ij} = 1$ if automobile $i \in \mathbb{I}$ stored at one of the distribution centers is assigned to auto-carrier $j \in \mathbb{J}$, otherwise $y_{ij} = 0$.
u_{dj}	$u_{dj} = 1$ if auto-carrier $j \in \mathbb{J}$ visits node $d \in \mathbb{D}^*$, otherwise $u_{dj} = 0$.
l_{abj}	$l_{abj} = 1$ if auto-carrier $j \in \mathbb{J}$ directly visits node $b \in \mathbb{D}^*$ from node $a \in \mathbb{D}^*$, $a \neq b$, otherwise $l_{abj} = 0$.
z_j	$z_j = 1$ if auto-carrier $j \in \mathbb{J}$ is used, otherwise $z_j = 0$.
Q^t_{jd}	the number of automobiles from the outbound warehouse of type $t \in \mathbb{T}$ in . auto-carrier $j \in \mathbb{J}$ after visiting node $d \in \mathbb{D}^*$.
q^t_{jd}	the number of automobiles of transfer demands with type $t \in \mathbb{T}$ in . auto-carrier $j \in \mathbb{J}$ after visiting node $d \in \mathbb{D}^*$.

Based on the above sets, parameters, and variables, the problem can be formulated as the following integer programming (IP) model ([4,5,10]).

$$(\text{IP}) \quad \max \quad \beta\Big(\sum_{j\in\mathbb{J}}\sum_{i^o\in\mathbb{I}^o} v_{i^o}\cdot x_{i^o j} + \sum_{j\in\mathbb{J}}\sum_{i\in\mathbb{I}} v_i \cdot y_{ij}\Big) - \alpha \sum_{j\in\mathbb{J}}\sum_{a,b\in\mathbb{D}^*} C_{ab}\cdot l_{abj} - \gamma\sum_{j\in\mathbb{J}} z_j \tag{1}$$

$$\text{s.t.} \quad \sum_{j\in\mathbb{J}} x_{i^o j} \leq 1, \ \forall i^o \in \mathbb{I}^o \tag{2}$$

$$\sum_{i^o\in\mathbb{I}^o} w_{i^o t} x_{i^o j} \leq \sum_{t'=t}^{T} p_{jt'}, \ \forall j \in \mathbb{J}, t \in \mathbb{T} \tag{3}$$

$$\sum_{i^o\in\mathbb{I}^o} x_{i^o j} \leq M z_j, \ \forall j \in \mathbb{J} \tag{4}$$

$$y_{ij} \leq z_j, \ \forall i \in \mathbb{I}, j \in \mathbb{J} \tag{5}$$

$$\sum_{j\in\mathbb{J}} y_{ij} \leq 1, \ \forall i \in \mathbb{I} \tag{6}$$

$$\sum_{i\in\mathbb{I}} w_{it} y_{ij} f'_{ib} \leq \sum_{t'=t}^{T} p_{jt'} - \sum_{t'=t}^{T} Q^{t'}_{jb} - \sum_{t'=t}^{T} q^{t'}_{ja} + \sum_{i\in\mathbb{I}} w_{it} y_{ij} f_{ib} + M\Big(1 - \sum_{a\in\mathbb{D}} l_{abj}\Big), \ \forall b \in \overline{\mathbb{D}}, j \in \mathbb{J}, t \in \mathbb{T} \tag{7}$$

$$\sum_{t\in\mathbb{T}} Q_{jb}^t + \sum_{t\in\mathbb{T}} q_{jb}^t \leq \sum_{t\in\mathbb{T}} p_{jt},\ \forall d \in \mathbb{D}^*, j \in \mathbb{J} \tag{8}$$

$$\sum_{a\in\mathbb{D}} l_{oaj} = \sum_{a\in\mathbb{D}} l_{aoj} \leq 1,\ \forall j \in \mathbb{J} \tag{9}$$

$$\sum_{a\in\mathbb{D}} l_{abj} \leq 1,\ \forall b \in \mathbb{D}, j \in \mathbb{J} \tag{10}$$

$$\sum_{a\in\mathbb{D}^*} l_{abj} = \sum_{c\in\mathbb{D}^*} l_{bcj},\ \forall b \in \mathbb{D}, j \in \mathbb{J} \tag{11}$$

$$x_{i^o j} \cdot f_{i^o b} = u_{i^o j},\ \forall i^o \in \mathbb{I}^o, b \in \mathbb{D}, j \in \mathbb{J} \tag{12}$$

$$y_{ij} \cdot f_{ib} = u_{ij},\ \forall i \in \mathbb{I}, b \in \mathbb{D}, j \in \mathbb{J} \tag{13}$$

$$y_{ij} \cdot f'_{ib} = u_{ij},\ \forall i \in \mathbb{I}, b \in \mathbb{D}, j \in \mathbb{J} \tag{14}$$

$$\sum_{t\in\mathbb{T}} Q_{ja}^t - \sum_{t\in\mathbb{T}} Q_{jb}^t \geq \sum_{i^o\in\mathbb{I}^o} x_{i^o j} f_{i^o b} + M(1 - l_{abj}),\ \forall a \in \mathbb{D}^*, b \in \mathbb{D}, j \in \mathbb{J} \tag{15}$$

$$\sum_{d\in\mathbb{D}} \sum_{t\in\mathbb{T}} Q_{jd}^t f'_{id} - \sum_{b\in\mathbb{D}} \sum_{t\in\mathbb{T}} Q_{jd}^t f_{ib} \geq M(1 - y_{ij}),\ \forall i \in \mathbb{I}, j \in \mathbb{J} \tag{16}$$

$$\sum_{d\in\mathbb{D}} u_{dj} f'_{id} - \sum_{b\in\mathbb{D}} u_{bj} f_{ib} \leq M(1 - y_{ij}),\ \forall i \in \mathbb{I}, j \in \mathbb{J} \tag{17}$$

$$u_{aj} - u_{bj} \leq D - 1 - D \cdot l_{abj},\ \forall a, b \in \mathbb{D}, j \in \mathbb{J} \tag{18}$$

$x_{i^o j}$, y_{ij}, u_{dj}, l_{abj}, z_j are binary, Q_{jd}^t and q_{jd}^t are non-negative integers, $\forall i^o \in \mathbb{I}^o$,
$i \in \mathbb{I},\ j \in \mathbb{J},\ a, b, d \in \mathbb{D}^*,\ t \in \mathbb{T}$

In the mathematics IP model, the objective value function (1) maximizes the weighted total value minus the weighted total transportation cost, including the traveling costs and the fixed costs. Constraint (2) ensures that each automobile from the outbound warehouse can be assigned to at most one auto-carrier. Constraint (3) means that the loading combination of each auto-carrier should satisfy the DCLS constraints. Constraint (4) ensures $z_j = 1$ if any automobile is assigned to auto-carrier $j \in \mathbb{J}$. Constraint (5) means that auto-carrier $j \in \mathbb{J}$ can be assigned with transfer demands if and only if the auto-carrier is used. Constraint (6) ensures each to-be-transferred automobile can be assigned to at most one auto-carrier. Constraints (7) and (8) indicate that each auto-carrier should satisfy the constraint of DCLS constraints at any point during each shipment. Constraints (9) to (11) are the flow balance constraints for routing decisions. Constraints (12) to (14) mean that auto-carrier $j \in \mathbb{J}$ must visit the origin and the destination of the automobile $i \in \mathbb{I}^*$ if i is assigned to j. Constraint (15) is the change in loading combination of each auto-carrier after visiting each node. Constraints (16) to (18) ensure that each automobile has to be picked up before it can be delivered.

4. Improved Adaptive Large Neighborhood Search (IALNS) Algorithm

The proposed pickup and delivery problem considering both outbound demands and transfer demands is obviously NP-hard since routing decisions are involved. To meet the needs of 3PL companies in practice, inspired by the references [4,5,30,31], this section proposes an improved adaptive large neighborhood search (IALNS) algorithm to solve the problem efficiently.

We first propose a heuristic to generate an initial solution in Section 4.1, which is an essential step for ALNS-based algorithms. Starting with the initial solution, an improved ALNS algorithm is presented, searching for better solutions iteratively by applying remove operators and repair operators adaptively in each iteration. Based on the characteristics of the proposed problem, we design three remove operators and four repair operators in Section 4.2. To improve the ability to jump out of local optimal, inspired by the idea of the simulated annealing (SA) algorithm, we introduce a solution-acceptance criterion in Section 4.3. Section 4.4 discusses the adaptively adjusted rule to update the weights of different operators. The flow of the proposed IALNS is shown in Figure 2.

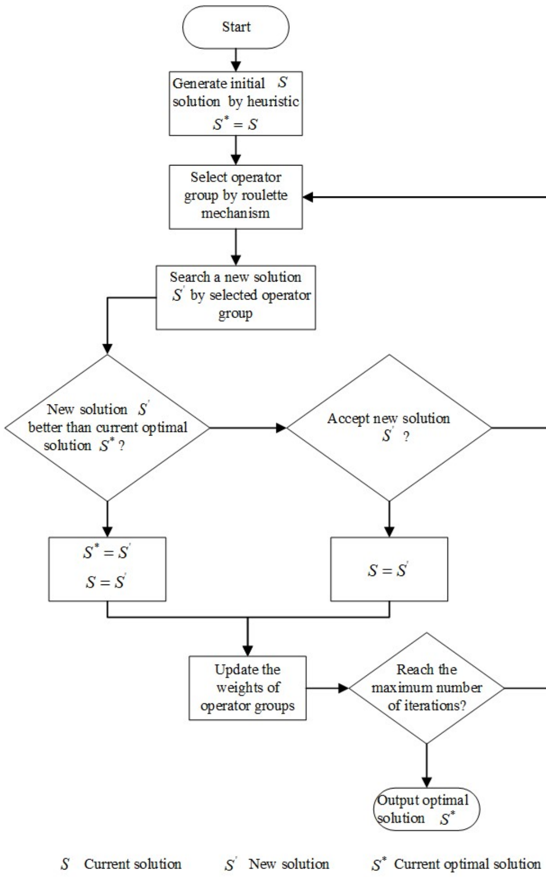

Figure 2. The flow of proposed IALNS algorithm.

4.1. Initial Solution Generation

The heuristic contains two steps described in the following.

Step1. Generating initial routes for the automobiles of outbound demands from manufacturers. Algorithm 1 is the pseudocode of Step 1, where sets $\mathbb{L}_j$ and $\mathbb{R}_j$ are the set of automobiles loaded by auto-carrier $j \in \mathbb{J}$ and the corresponding route. Let $\mathbb{A}_d$ denote the subset of automobiles that need to be handled in the outbound warehouse with a destination equal to $d \in \mathbb{D}$.

Algorithm 1 Initial routing for automobiles of outbound demands

Input: Sets $\mathbb{I}^o$, $\mathbb{D}$, $\mathbb{J}$
Output: $\mathbb{R}_j$ and $\mathbb{L}_j$
for $j = 1$ **to** J **do**
  Sort the DCs in $\mathbb{D}$ by descending order of the total value $\sum_{a=1}^{|\mathbb{A}_d|} v_a$ of corresponding automobiles, let $\mathbb{W}$ be the new sorted set
for $w = 1$ **to** $|\mathbb{W}|$ **do**
  Try loading automobiles onto auto-carrier j subject to DCLS
  $\mathbb{R}_j \leftarrow \mathbb{R}_j \bigcup \{w\}$, update $\mathbb{L}_j$, $\mathbb{A}_d$
  if $|\mathbb{L}_j| < \sum_{t \in \mathbb{T}} p_{jt}$ **then**
    continue
  else
    break

Step2. Insert the to-be-transferred automobiles into the initial routes generated by Step 1. Sort all the automobiles of transfer demands by the decreasing order of their values. Then, based on the initial routes, try to insert the sorted automobiles and update the routes accordingly following the rules and loading constraints below, as the examples shown in Figure 3.

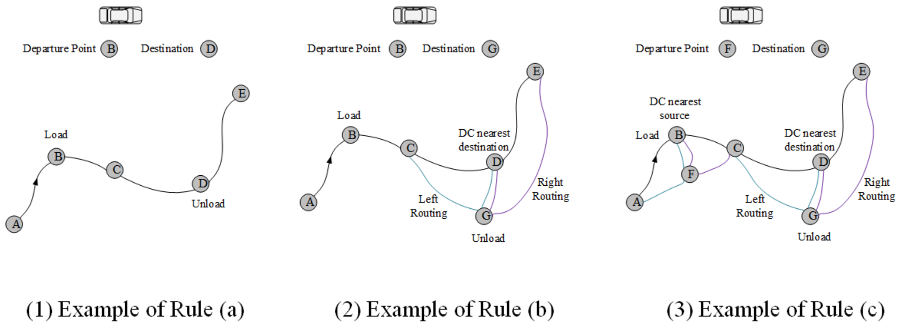

Figure 3. Examples of three insertion rules.

(a) If the origin and the destination of the incumbent automobile are both on one of the current routes, and the origin of the automobile is visited before the destination in the corresponding route, then the automobile can be assigned to the corresponding auto-carrier if the DCLS constraint can be satisfied. The routes will not be changed if this rule is applied, as the example shown in Figure 3(1).

(b) If only the origin or the destination of the incumbent automobile is on one of the current routes, then try to load the automobile by modifying the corresponding route with the smallest cost. As the example shown in Figure 3(2), the origin of the automobile, point B, is on one of the current routes while the destination, point G, is not on the route. Find the point on the route that is nearest to destination G. If the DCLS constraint can be satisfied, then the automobile can be assigned to the corresponding auto-carrier, and the route can be changed by inserting destination point G into the route. After insertion, let the left route represent the case in which point G is visited before D, and let the right route represent the other case in which point G is visited after D. Compare the new routes in the two cases and choose the one with a smaller total length to update the route.

(c) For the incumbent automobile, if neither the origin nor the destination is on one of the current routes, then select a route in which one of the visited distribution centers is the nearest point to the origin of the automobile, and the DCLS constraints can be satisfied. The route, if it exists, can be updated following rule (b) above. Figure 3(3) shows an example of updating the route in this case.

4.2. Remove and Repair Operators

In this section, we design three remove operators and four repair operators to update the initial solution. As described below, rm1–rm3 are the remove operators, and rp1–rp4 are the repair operators.

(rm1) Randomly remove from outbound demands: Randomly select p assigned automobiles of the outbound demands, and remove them from the current solution.

(rm2) Remove by smallest cost: In each iteration, for each DC involved in each route, calculate the reduction in objective value if the automobiles related to the DC are removed from the current route. Compare the reductions, choose the smallest one, and remove the DC and corresponding automobiles from the route.

(rm3) Randomly remove from transfer demands: Randomly select p automobiles of the transfer demands, and remove them from the current solution if the corresponding route will not be changed.

(rp1) Randomly repair for the outbound demands: For any automobile of the outbound demand that is removed by the above rules, randomly insert it into a place into the current route.

(rp2) Greedy repair for the outbound demands: For each removed automobile of the outbound demand, reload the automobile to the current route if it contains the destination of the automobile directly. Otherwise, try to insert the destination DC into the current route by rule (b) of Step 2 in Section 4.1.

(rp3) Repair for the transfer demands without changing route: If the origin and destination are on the current route, then try to reload the removed automobile to the auto-carrier.

(rp4) Greedy repair for the transfer demands: For each removed automobile of the transfer demands, try to insert them into the current route using the three rules described in Section 4.1 to update the route.

4.3. Solution-Acceptance Criterion

In each iteration, after a new solution is generated by the remove and repair operators, we use a solution-acceptance criterion based on the idea of the simulated annealing (SA) algorithm. There are three rules to determine whether the new solution can be accepted or not:

(sc1) If the objective value of the new solution S' is better than that of the current solution S, then update the current solution by $S = S'$.

(sc2) If the objective value of the new solution S' is equal to that of the current solution S, then reject the new solution.

(sc3) If the objective value of the new solution S' is worse than that of the current solution S, then accept the new solution with a possibility $e^{(-[f(S')-f(S)]/G)}$. G is the temperature of the current iteration, and the temperature is discounted by λ in each iteration, where $\lambda \in (0,1)$ is the cooling factor.

4.4. Adaptive Updates of Operator Group Weights

In each iteration, the proposed IALNS algorithm will select one operator to generate a new feasible solution, and the selection of operators determines the efficiency of the entire procedure. To make the algorithm implementable for large-scale practical problems, the operators are clustered as four different operator groups: rm1, rm3, rp1, rp3; rm1, rm3, rp2, rp3; rm2, rp1, rp4; and rm2, rp2, rp4. As described below, the roulette mechanism is applied to select one operator from the groups, and the weight parameters assigned to the groups will be updated adaptively.

Let set $\mathbb{G}$ denote the groups of the operators. For each group $g \in \mathbb{G}$, let w_g be the weight parameter related to the possibility that the group is chosen in each iteration. Define a given K number of iterations as a phase. Let π_g and θ_g be the score and the number of use of group g during each phase. Let σ_1, σ_2 and σ_3 be three parameters corresponding to the three rules of solution-acceptance criterion described in Section 4.3, respectively, and $\sigma_1 + \sigma_2 + \sigma_3 = 1$. The weight parameter w_g will be updated at the end of each phase as described in Algorithm 2.

Algorithm 2 The adaptive searching process

Input: Solution S in current iteration, σ_1, σ_2, σ_3, $\mathbb{G}$
Output: w_g, $\forall g \in \mathbb{G}$

$w_1 = w_2 = \ldots w_{|\mathbb{G}|}$, $\sum\limits_{g \in \mathbb{G}} w_g = 1$
for $p = 1$ **to** P **do**
  initialize $\pi_g = 0$ and $\theta_g = 0$
  if the operator group g^* is used in each iteration of this phase **then**
    $\theta_{g^*} \leftarrow \theta_{g^*} + 1$, and $\pi_{g^*} \leftarrow \pi_{g^*} + \sigma_u$
  $\pi_g \leftarrow \pi_g / \theta_g$
  let $\overline{\pi}_g = \dfrac{\pi_g}{\sum\limits_{g \in \mathbb{G}} \pi_g}$
  $w_g \leftarrow (1-\epsilon) \cdot w_g + \epsilon \cdot \overline{\pi}_g$

5. Numerical Experiments

To show the effectiveness and efficiency of the proposed IALNS algorithm in solving the pickup and delivery problem of automobiles with simultaneous outbound and transfer demands, this section shows the randomly generated real-scale instances and the experimental results. All instances are tested by directly solving the IP model with the commercial solver CPLEX 12.8 and the proposed IALNS algorithm, respectively. The model and algorithm are implemented on a computer with 16 GB memory and an AMD Ryzen 5460oH processor.

5.1. Instances Generation and Parameter Setting

Based on the practical operations of a 3PL company in China, we generate three groups of random instances of a small, medium, and large scale, respectively, each of which consists of 10 instances. For the small-scale group, $I^o = 20$, $D = 10$, $I = 5$, $J = 3$. For the medium-scale group, $I^o = 40$, $D = 20$, $I = 10$, $J = 6$. For the large-scale group, $I^o = 60$, $D = 30$, $I = 15$, $J = 10$. The slot configuration of each auto-carrier, the location of each distribution center, and the outbound warehouse are randomly generated in a given range. Accordingly, the value, type, origin, and destination of each automobile are also randomly generated in a given range. The hyperparameters of the algorithm include the maximum number of iterations, number of to-be-removed automobiles p, cooling factor λ, three acceptance criteria weights $\sigma_1, \sigma_2, \sigma_3$, and adaptive weight updates ϵ. We first determine the two most likely values for each hyperparameter [31]. These values were randomly combined, resulting in 2^7 combinations. Each hyperparameter combination is pretested on different-scale instances, and the best combinations of hyperparameter values are determined by the average of the test results. Table 2 shows related parameters in the IP model and the hyperparameters of IALNS.

Table 2. The setting of parameters and hyperparameters.

Parameters	Values
α, β, γ	0.7, 0.2, 0.1
Maximum iteration	200
Number of to-be-removed automobiles p	A random integer in $[\frac{1}{4}\lvert\mathbb{I}^*\rvert, \frac{1}{2}\lvert\mathbb{I}^*\rvert]$
Initial temperature G	100
Cooling factor λ	0.99
$\sigma_1, \sigma_2, \sigma_3$	0.7, 0.1, 0.2
ϵ	0.5

5.2. Experimental Results of Different-Scale Instances

First, for each group of instances, we use the commercial solver CPLEX and our proposed IALNS algorithm to solve them. Since obtaining the optimal solution using CPLEX would take a very long time for large-scale instances, we set up a limit of 1800 s to the calculation time. Let Z_{CPLEX} be the best objective value generated by solving the IP model directly by CPLEX within the time limit. Let Z_{IALNS} and $t_I ALNS$ be the objective value and computational time solved by IALNS. Table 3 shows the performance of the proposed IALNS algorithm compared to CPLEX. The first column shows the different scales of instances. The second column shows the average relative gap between Z_{IALNS} and Z_{CPLEX}, which is defined as $(Z_{CPLEX} - Z_{IALNS})/Z_{CPLEX} \cdot 100\%$. The third and fourth columns are the average calculation time of the IALNS algorithm and solving the IP model by CPLEX, respectively.

Based on the results in Table 3, our proposed IALNS algorithm can generate near-optimal solutions with a much shorter calculation time. For small-scale instances, where CPLEX can generate the optimal solution within the time limit, the average computation time of IALNS is 20 s on average, while the average computation time of CPLEX is 622 s, and the average relative gap is 9.18%. For medium- and large-scale instances, CPLEX cannot generate the optimal solutions within 1800 s, while the average computation time of IALNS is 197 s, and the average relative gap is 9.23%.

Table 3. Experimental results of IALNS and CPLEX.

Instances Scale	Average IALNS-CPLEX (%)	Average Computation Time—IALNS (s)	Average Computation Time—CPLEX (s)
Small	9.18	20	622
Medium	10.71	110	>1800
Large	7.73	284	>1800
Average	7.84	138	-

We use instance 1 of small scale to demonstrate the optimization process and computation results of the proposed IALNS algorithm. Figure 4 shows the converging curve of solving instance 1 by IALNS, where the abscissa axis represents the number of iterations, and the vertical axis represents the fitness value, which is defined as the negative of the objective function in the IP model. It can be observed that the objective function value starts to converge when the IALNS algorithm iterates around 38 times. Figure 5 shows the pickup and delivery routes for three auto-carriers in instance 1.

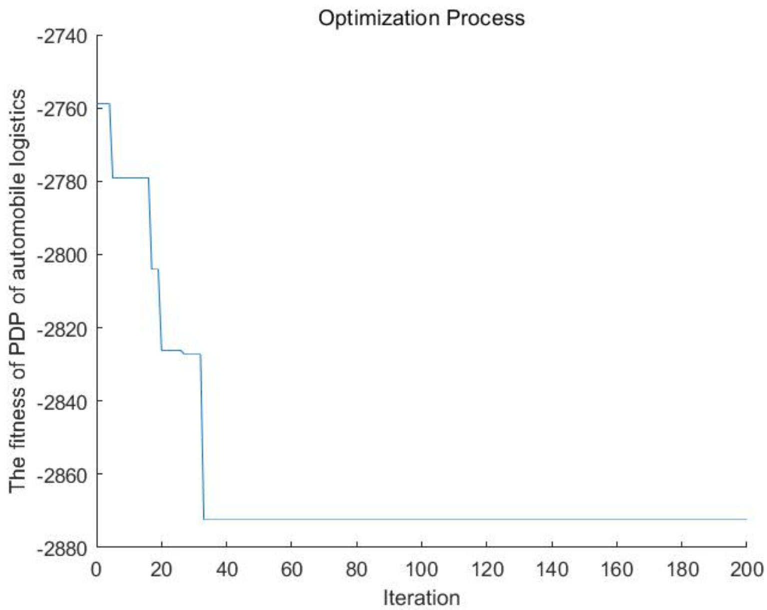

Figure 4. The optimization process of small−scale instance 1.

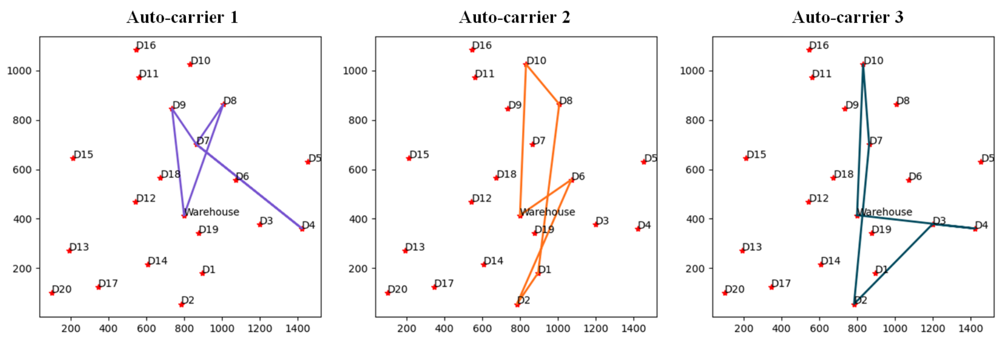

Figure 5. The pickup and delivery routing of different auto-carriers of small−scale instance 1.

Second, we compare the results of the proposed IALNS algorithm over all the instances to three heuristics, the initial heuristic (IH) proposed in Section 4.1, large-scale neighborhood search (LNS) and a local search (LS) algorithm. The LNS algorithm uses the same operators as those in IALNS, while the LS algorithm keeps only one pair of the operators. Let Z_{IH} be the objective value obtained by initial heuristic, and the average computational time of the IH is not shown in the results since solving each instance takes less than 1 s. Let Z_{LNS} and t_{LNS} be the average objective value and the average computational time of solving the instances by LNS, and let Z_{LS} and t_{LS} be the average objective value and the average computational time by LS, respectively. Table 4 shows the experimental results comparing different algorithms. The first column shows the scale of the instances, and the second column shows the average relative gap between the objective values of IH and IALNS, which equals $\frac{Z_{IH}-Z_{IALNS}}{Z_{IH}} \cdot 100\%$. The third column is the average relative gap between the results of IALNS and LNS, where $Gap_{obj} = \frac{Z_{IALNS}-Z_{LNS}}{Z_{LNS}} \cdot 100\%$, and $Gap_{time} = \frac{t_{IALNS}-t_{LNS}}{t_{LNS}}$. The fourth column is the average relative gap between the results of IALNS and LS, where $Gap_{obj} = \frac{Z_{IALNS}-Z_{LS}}{Z_{LS}} \cdot 100\%$, and $Gap_{time} = \frac{t_{IALNS}-t_{LS}}{t_{LS}}$.

Table 4. Experimental results of different algorithms.

Instances Scale	IALNS-IH (%)	IALNS-LNS (%)		IALNS-LS (%)	
		Gap_{obj}	Gap_{time}	Gap_{obj}	Gap_{time}
Small	25.02	−0.73	−9.02	4.22	−49.82
Medium	30.99	−0.14	−7.84	8.02	−50.43
Large	40.05	−0.12	−9.70	6.77	−46.28
Average	32.02	−0.33	−8.85	6.34	−48.84

As shown in Table 4, the proposed IALNS algorithm has a significant improvement of the objective value compared to IH, reaching an average relative gap of 32.02%. Compared to the LNS algorithm, the proposed IALNS algorithm achieves an average gap of −0.33% in terms of the objective value, but it exhibits a notable improvement of 8.85% in average computational time. With the increase in the instance scale, the relative gap between objective values decreases from −0.73% for small-scale instances to −0.12% for large-scale instances. When comparing IALNS to LS, the proposed IALNS algorithm shows more advantages. Across three groups of instances with varying scales, IALNS improves the objective value by an average of 6.34% compared to LS. Moreover, IALNS significantly reduces the average computation time by 48.84%.

5.3. Sensitivity Analysis on Weight Parameters

This section tests the proposed IALNS algorithm with different values for the weight parameters α, β, and γ. We select five instances from the large-scale group, and each instance is solved by initial heuristic, IALNS, and CPLEX, respectively. Table 5 shows the results, in which the first column shows different combinations of the weight parameters, the second column shows the instance index, and the remaining columns are the same as those in Tables 3 and 4.

According to the results in Table 5, CPLEX cannot find the optimal solution within the time limit under the different weighted parameters, and the proposed IALNS always generates satisfactory solutions much more efficiently. When α decreased from 0.8 to 0.5, the average relative gap between IALNS and IH increased from 36.47% to 39.87%. Similarly, there was a notable increase in the average relative gap between IALNS and CPLEX, going from 2.6% to 7.04%. On the other hand, the computation time of IALNS did not exhibit a discernible trend, fluctuating within the range of 250–300 s. These experimental results demonstrate that as the value of α increases, the proposed IALNS has advantages over IH and solving the IP model using CPLEX. The proposed algorithm shows strong robustness and can generate near-optimal solutions to the problem within a reasonable time under different weight parameters.

Table 5. Experimental results of different weighted values of α, β, and γ.

Weights	Instance	IALNS-IH (%)	Computation Time—IALNS (s)	Computation Time—CPLEX (s)	IALNS-CPLEX (%)
$\alpha = 0.8$ $\beta = 0.1$ $\gamma = 0.1$	1	30.83	285	>1800	5.69
	2	45.32	214	>1800	3.45
	3	14.19	284	>1800	0.95
	4	26.33	297	>1800	2.35
	5	65.65	226	>1800	0.54
	Average	36.47	261	-	2.60
$\alpha = 0.7$ $\beta = 0.2$ $\gamma = 0.1$	1	30.44	281	>1800	5.45
	2	45.76	243	>1800	3.6
	3	14.21	300	>1800	2.37
	4	26.57	359	>1800	2.39
	5	63.98	265	>1800	0.91
	Average	36.19	290	-	2.96
$\alpha = 0.6$ $\beta = 0.3$ $\gamma = 0.1$	1	−30.96	284	>1800	7.69
	2	46.80	230	>1800	5.38
	3	14.53	274	>1800	3.75
	4	26.91	292	>1800	4.73
	5	67.12	240	>1800	4.18
	Average	37.26	264	-	5.15
$\alpha = 0.5$ $\beta = 0.4$ $\gamma = 0.1$	1	−32.12	298	>1800	9.75
	2	50.42	281	>1800	8.68
	3	15.28	358	>1800	5.6
	4	29.01	335	>1800	6.61
	5	67.53	265	>1800	4.56
	Average	39.87	308	-	7.04

6. Conculsions

This paper studies a pickup and delivery problem for automobile logistics considering the outbound demands from the assembly plant warehouse and the transfer demands among distribution centers simultaneously. Based on the investigation of 3PL companies' practices, the urgent level of handling the automobiles, the special downward compatible loading structure of auto-carriers, and the transportation cost based on the vehicle routing are considered in the problem. An integer programming model is formulated for the problem, and an improved adaptive large neighborhood search (IALNS) algorithm is proposed to solve the problem efficiently. The IALNS starts from an initial solution generated by a heuristic algorithm and can find near-optimal solutions after iterations with elaborately designed operator groups and adaptive weights. The experimental results show that the proposed IALNS algorithm can significantly improve the solutions compared to the initial heuristic, and outperform directly solving the IP model by CPLEX to a large extent.

The mathematical model and algorithm proposed in this paper can be implemented easily in the daily decision-making of 3PLs, which will improve the efficiency of automobile logistics operations, increase the responsiveness of handling transfer demands, raise the utility of the relatively scarce auto-carrier resources, and reduce transportation costs. In future research, one can extend the problem to multiple outbound warehouses or multi-period dynamic environments. One can also consider multi-modal transportation of automobile logistics based on our problem.

Author Contributions: Y.W.: Conceptualization, Methodology, Writing—review and editing, Investigation, Validation, Funding acquisition; R.Z.: Methodology, Investigation, Validation; Y.Z.: Methodology, Investigation, Software, Writing—original draft; C.L.: Conceptualization, Writing—review and editing, Funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Shanghai Sailing Program (21YF1416400) and Soft Science Research Project of Shanghai Science and Technology Innovation Action Plan (22692111200).

Data Availability Statement: The data are available from the corresponding author on reasonable request.

Acknowledgments: This work was supported by the Shanghai Sailing Program (21YF1416400) and the Soft Science Research Project of the Shanghai Science and Technology Innovation Action Plan (22692111200).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Organisation Internationale des Constructeurs d'Automobiles (OICA). Global Sales Statistics 2019–2022. 2023. Available online: www.oica.net (accessed on 12 May 2023).
2. Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [CrossRef]
3. Min, H. The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transp. Res. Part A Gen.* **1989**, *23*, 377–386. [CrossRef]
4. Stefan Ropke, D.P. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **2006**, *40*, 455–472. [CrossRef]
5. Qu, Y.; Bard, J.F. A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Comput. Oper. Res.* **2012**, *39*, 2439–2456. [CrossRef]
6. Kalayci, C.B.; Kaya, C. An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Syst. Appl.* **2016**, *66*, 163–175. [CrossRef]
7. Polat, O.; Kalayci, C.B.; Kulak, O.; Günther, H.O. A perturbation based variable neighborhood search heuristic for solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery with Time Limit. *Eur. J. Oper. Res.* **2015**, *242*, 369–382. [CrossRef]
8. Wang, C.; Mu, D.; Zhao, F.; Sutherland, J.W. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows. *Comput. Ind. Eng.* **2015**, *83*, 111–122. [CrossRef]
9. Wang, Z.; Dessouky, M.; Woensel, T.V.; Ioannou, P. Pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times. *EURO J. Transp. Logist.* **2023**, *12*, 100099. [CrossRef]
10. Zachariadis, E.E.; Tarantilis, C.D.; Kiranoudis, C.T. Vehicle routing strategies for pick-up and delivery service under two dimensional loading constraints. *Oper. Res.* **2017**, *17*, 115–143. [CrossRef]
11. Moura, A.; Pinto, T.; Alves, C.; de Carvalho, J.V. A Matheuristic Approach to the Integration of Three-Dimensional Bin Packing Problem and Vehicle Routing Problem with Simultaneous Delivery and Pickup. *Mathematics* **2023**, *11*, 713. [CrossRef]
12. Friedrich, C.; Elbert, R. Adaptive large neighborhood search for vehicle routing problems with transshipment facilities arising in city logistics. *Comput. Oper. Res.* **2022**, *137*, 105491. [CrossRef]
13. Wehbi, L.; Bektaş, T.; Iris, Ç. Optimising vehicle and on-foot porter routing in urban logistics. *Transp. Res. Part D Transp. Environ.* **2022**, *109*, 103371. [CrossRef]
14. Avci, M.; Topaloglu, S. A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Expert Syst. Appl.* **2016**, *53*, 160–171. [CrossRef]
15. Rodrigues, V.P.; Morabito, R.; Yamashita, D.; Silva, B.; Ribas, P. Ship Routing with Pickup and Delivery for a Maritime Oil Transportation System: MIP Model and Heuristics. *Systems* **2016**, *4*, 31. [CrossRef]
16. Hornstra, R.P.; Silva, A.; Roodbergen, K.J.; Coelho, L.C. The vehicle routing problem with simultaneous pickup and delivery and handling costs. *Comput. Oper. Res.* **2020**, *214*, 119118. [CrossRef]
17. Aziez, I.; Côté, J.F.; Coelho, L.C. Exact algorithms for the multi-pickup and delivery problem with time windows. *Eur. J. Oper. Res.* **2020**, *284*, 906–919. [CrossRef]
18. Xu, W.; Zhang, C.; Cheng, M.; Huang, Y. Electric Vehicle Routing Problem with Simultaneous Pickup and Delivery: Mathematical Modeling and Adaptive Large Neighborhood Search Heuristic Method. *Energies* **2022**, *15*, 9222. [CrossRef]
19. Lyu, Z.; Pons, D.; Zhang, Y. Emissions and Total Cost of Ownership for Diesel and Battery Electric Freight Pickup and De-livery Trucks in New Zealand: Implications for Transition. *Sustainability* **2023**, *15*, 7902. [CrossRef]
20. Agbegha, G.Y.; Ballou, R.H.; Mathur, K. Optimizing auto-carrier loading. *Transp. Sci.* **1998**, *32*, 174–188. [CrossRef]
21. Tadei, R.; Perboli, G.; Della Croce, F. A heuristic algorithm for the auto-carrier transportation problem. *Transp. Sci.* **2002**, *36*, 55–62. [CrossRef]
22. Dell'Amico, M.; Falavigna, S.; Iori, M. Optimization of a real-world auto–carrier transportation problem. *Transp. Sci.* **2014**, *49*, 402–419. [CrossRef]
23. Cordeau, J.F.; Dell'Amico, M.; Falavigna, S.; Iori, M. A rolling horizon algorithm for auto-carrier transportation. *Transp. Res. Part B Methodol.* **2015**, *76*, 68–80. [CrossRef]
24. Juárez Pérez, M.A.; Pérez Loaiza, R.E.; Quintero Flores, P.M.; Atriano Ponce, O.; Flores Peralta, C. A Heuristic Algorithm for the Routing and Scheduling Problem with Time Windows: A Case Study of the Automotive Industry in Mexico. *Algorithms* **2019**, *12*, 111. [CrossRef]

25. Bonassa, A.C.; Cunha, C.B.; Isler, C.A. An exact formulation for the multi-period auto-carrier loading and transportation problem in Brazil. *Comput. Ind. Eng.* **2019**, *129*, 144–155. [CrossRef]
26. Bonassa, A.C.; da Cunha, C.B.; Isler, C.A. A multi-start local search heuristic for the multi-period auto-carrier loading and transportation problem in Brazil. *Eur. J. Oper. Res.* **2023**, *307*, 193–211. [CrossRef]
27. Hu, Z.H.; Zhao, Y.; Tao, S.; Sheng, Z.-H. Finished-vehicle transporter routing problem solved by loading pattern discovery. *Ann. Oper. Res.* **2015**, *234*, 37–56. [CrossRef]
28. Wang, Y.; Chen, F.; Chen, Z.L. Pickup and delivery of automobiles from warehouses to dealers. *Transp. Res. Part B Methodol.* **2018**, *117*, 412–430. [CrossRef]
29. Chen, F.; Wang, Y. Downward compatible loading optimization with inter-set cost in automobile outbound logistics. *Eur. J. Oper. Res.* **2020**, *287*, 106–118. [CrossRef]
30. Ghilas, V.; Demir, E.; Van Woensel, T. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Comput. Oper. Res.* **2016**, *72*, 12–30. [CrossRef]
31. Iris, Ç.; Pacino, D.; Ropke, S. Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *105*, 123–147. [CrossRef]