

Article

# Solving Dynamic Full-Truckload Vehicle Routing Problem Using an Agent-Based Approach

Selin Çabuk \* and Rızvan Erol

Department of Industrial Engineering, Faculty of Engineering, Cukurova University, Sarıcam, 01330 Adana, Turkey; rerol@cu.edu.tr

\* Correspondence: selincabuk@cu.edu.tr

**Abstract:** In today's complex and dynamic transportation networks, increasing energy costs and adverse environmental impacts necessitate the efficient transport of goods or raw materials across a network to minimize all related costs through vehicle assignment and routing decisions. Vehicle routing problems under dynamic and stochastic conditions are known to be very challenging in both mathematical modeling and computational complexity. In this study, a special variant of the full-truckload vehicle assignment and routing problem was investigated. First, a detailed analysis of the processes in a liquid transportation logistics firm with a large fleet of tanker trucks was conducted. Then, a new original problem with distinctive features compared with similar studies in the literature was formulated, including pickup/delivery time windows, nodes with different functions (pickup/delivery, washing facilities, and parking), a heterogeneous truck fleet, multiple trips per truck, multiple trailer types, multiple freight types, and setup times between changing freight types. This dynamic optimization problem was solved using an intelligent multi-agent model with agent designs that run on vehicle assignment and routing algorithms. To assess the performance of the proposed approach under varying environmental conditions (e.g., congestion factors and the ratio of orders with multiple trips) and different algorithmic parameter levels (e.g., the latest response time to orders and activating the interchange of trip assignments between vehicles), a detailed scenario analysis was conducted based on a set of designed simulation experiments. The simulation results indicate that the proposed dynamic approach is capable of providing good and efficient solutions in response to dynamic conditions. Furthermore, using longer latest response times and activating the interchange mechanism have significant positive impacts on the relevant costs, profitability, ratios of loaded trips over the total distance traveled, and the acceptance ratios of customer orders.

**Keywords:** dynamic optimization; vehicle routing; full-truck load; multi-agent modeling

**MSC:** 390B06



**Citation:** Çabuk, S.; Erol, R. Solving Dynamic Full-Truckload Vehicle Routing Problem Using an Agent-Based Approach. *Mathematics* **2024**, *12*, 2138. <https://doi.org/10.3390/math12132138>

Academic Editors: Xue-Ming Yuan, Hao Yu and Andrea Scozzari

Received: 7 April 2024

Revised: 21 June 2024

Accepted: 26 June 2024

Published: 7 July 2024



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

## 1. Introduction

The logistics sector is experiencing rapid global development. Effective logistics management is increasingly critical due to the growing complexity of logistics systems and the escalating share of transportation costs within total expenses [1]. One of the most prevalent transportation challenges is the vehicle routing problem (VRP), first introduced by [2] to address the task of finding the best set of routes for a fleet of vehicles to meet freight orders of a set of customers in a network. The VRP encompasses various variants based on the considered constraints, including a limited or unlimited vehicle capacity, homogeneous or heterogeneous fleets, single or multiple vehicles, and time windows, among others [3].

One variant of the VRP is the full-truckload vehicle routing problem (FTVRP), which entails a truck being loaded near its full capacity with identical or similar products from one or multiple customers and directly transporting the freight from the pickup to the delivery

point without intermediate transshipment, often termed as direct delivery service [4]. Despite its significance, the FTVRP has not received as much research attention compared with other VRP variants.

This study addressed a special dynamic full-truckload vehicle assignment and routing problem. The problem was defined through a comprehensive analysis of a prominent Turkish logistics firm, which serves its clientele by overseeing a substantial fleet comprising 600 long-haul trucks dedicated to full-truckload trips. The trucks are composed of two primary components, the trailer and the tractor (tow truck), with some capable of interchanging trailers while others are not. Freight orders comprise specific load types transportable using a range of appropriate trailer configurations, accommodating single or multiple trips. Given the dynamic arrival of freight orders, static optimization methods are not suitable for addressing our problem.

In the context of load pickup and delivery, time windows are established in accordance with customer requirements. Nodes within the distribution network serve either as loading points, unloading points, or both, depending on the incoming demand. Trailers undergo washing procedures as needed before carrying a new load. At specific nodes, tractors interchange trailers based on the load type for pickup. An additional challenge arises when vehicles travel to pickup points without freight, incurring empty fuel costs. To minimize these expenses and optimize profitability, maximizing vehicle utilization involves assigning the most suitable vehicles and routing orders to achieve full capacity.

Traditional static optimization algorithms entail excessively long computation times owing to the dynamic nature of operations and the variable influx of requests. Assigning the optimal order to the available fleet of trucks poses a particularly difficult computational task with traditional operation research algorithms, given the multitude of unpredictable parameters associated with incoming orders.

In today's technological landscape, acquiring real-time solutions that can adapt to the continuously evolving conditions of real-world problems is imperative. Throughout the planning phase, modifications in existing information and the dynamic emergence of new data result in substantial system changes. For logistics management systems that necessitate immediate responses to dynamic demand fluctuations, decisions informed by real-time data are paramount. Given the inherently distributed data and independently operating structure of the logistics and transportation sector, decentralized decision-making systems prove highly advantageous.

Agent-based modeling and simulation represent a recent approach to modeling intricate systems comprising interacting, autonomous 'agents.' These agents exhibit behaviors typically governed by simple rules and engage in interactions with other agents [5]. Multi-agent technology is regarded as an intelligent solution offering autonomous and adaptable problem-solving abilities in dynamic environments. Considering the decentralized, distributed, and dynamic characteristics of the transportation sector, agent-based systems are well-suited for addressing its challenges [6].

Given the definitions and characteristics of agent-based systems, they are highly versatile tools for research in the transportation and logistics field. The logistics sector typically has a decentralized, distributed, and dynamic structure. In logistics systems, agents can embody various entities such as distributors, intermediaries, senders, receivers, carriers, cargo handlers, and more. They may also represent transportation demands, cargoes, or loading units, alongside higher-level organizational units. By modeling these entities as agent types, we can exploit the capabilities of agent-based modeling to reach optimal or near-optimal solutions for dynamic problems.

The contributions of this study can be expressed as follows:

- i. This study emphasizes the importance of addressing various characteristics such as time windows, heterogeneous fleets, capacity constraints, and multiple depots collectively during problem formulation for full-load transportation companies. It presents a comprehensive problem formulation that incorporates a significant

- portion of these requirements, unlike previous studies that have often addressed these characteristics separately.
- ii. This study transforms the problem into a dynamic optimization challenge by considering factors such as demand dynamics, order arrival times, response times, order sizes, and freight types. It uniquely integrates practical aspects like heterogeneous fleets, capacity constraints, and time windows, which have not been collectively addressed in the existing literature.
  - iii. Unlike previous studies that have relied on genetic algorithms, particle swarm optimization, or savings algorithms, this study adopts an agent-based approach to model and solve the dynamic full-truckload vehicle assignment and routing problem. This approach offers autonomous, flexible, and efficient solutions.
  - iv. This study conducted a novel analysis of response times, congestion rates, and multi-trip demand rates of the algorithms within the agent-based system under varying conditions. It also investigated the impact of the developed interchange mechanism, providing a comprehensive analysis of its performance across different scenarios.

The remaining sections of this paper are structured as follows: Section 2 provides a detailed literature review, highlighting the distinctions from our study. The problem formulation is described in Section 3. Section 4 introduces the proposed agent-based modeling framework. Section 5 discusses the simulation experiments and their results. Finally, Section 6 focuses on the conclusions and outlines avenues for future work.

## 2. Literature Review

Despite its significance, the full-truckload vehicle routing problem (FTVRP) has not received as much attention in the research compared with other variants of the vehicle routing problem (VRP). Only a few studies have delved into this specific issue. The motivation for this study arose from the scarcity of literature on the FTVRP, despite its widespread application in various domains.

The full-truckload vehicle routing problem (FTVRP), also referred to as the full-truckload multi-depot pickup and delivery problem (FTMDPDP), was initially introduced by Ball et al. [7]. In the literature, approaches to solving various variants of the FTVRP can be classified into three main groups: (1) exact algorithms, (2) problem-specific heuristics or metaheuristics, and (3) agent-based solutions. Furthermore, most papers focus on static FTVRPs, in contrast with the stochastic, dynamic problem in our study.

In contrast with Ball et al.'s [7] heuristic approach, Desroisiers et al. [8] presented an exact method for solving the same problem. Arunapuram et al. [9] and Gronalt et al. [10] explored the FTVRP with multiple depots and time windows (TWs), introducing new branch-and-price algorithms and four savings heuristics, respectively. Imai et al. [11] tackled a similar problem to Gronalt et al. [10] but without time windows, proposing a Lagrangian relaxation approach.

Caris and Janssens [12] extended the problem formulation of Imai et al. [11] by adding strict time windows at customer locations and the depot. They employed a two-phase insertion heuristic coupled with local search techniques to enhance the initial solution. Currie and Salhi [13] extended the FTVRP to accommodate heterogeneous goods and trucks. Their approach involved hybrid methods that combined heuristics with neighborhood operators and insertion heuristics to minimize empty truck movements. Zhang et al. [14,15] developed cluster methods and reactive Tabu search algorithms for addressing multi-depot container truck transportation problems. They further extended their research to encompass multiple terminals and depots, utilizing mathematical models based on preparative graphs.

Li and Lu [16] introduced a variant of the FTVRP with multiple delivery points originating from the same pickup point, allowing orders to be carried by the same or different vehicles multiple times. They proposed a hybrid genetic algorithm for its solution. Liu et al. [17] studied a full-truckload multi-depot capacitated vehicle routing problem in carrier collaboration, proposing a mathematical programming model to minimize empty vehicle movements and a two-phase greedy algorithm as an alternative for larger problems.

Nossack and Pesch [18] proposed a two-stage heuristic solution for a full-truckload pickup and delivery problem with time windows. Bai et al. [19] presented a bidirectional multi-shift full-truckload vehicle routing problem with time windows and a single depot, while Braekers et al. [20] studied a similar problem for drayage operations with a single depot and homogeneous vehicles, employing deterministic annealing algorithms.

Maneengam and Udomsakdigool [21] considered full-truckload vehicle routing problems with time windows and split bulk deliveries. This study differs from our study since it has a homogeneous vehicle fleet and a single depot. Maneengam and Udomsakdigool [22] extended their earlier problem formulation into a collaborative bidirectional multi-period multi-depot vehicle routing problem involving multiple shippers and carriers, terminal and transport capability constraints, and time window constraints. They employed a branch-and-bound algorithm to solve this problem, contrasting with our study's focus on a heterogeneous vehicle fleet.

Several studies related to our work were reviewed. Grimault et al. [23] addressed the full-truckload vehicle routing problem with multiple depots, a heterogeneous fleet, and time windows, integrating routing and synchronization decisions. They proposed an adaptive large neighborhood search to solve this, which contrasts with our study's focus on synchronization decisions. Wang et al. [24] concentrated on a variation of the full-truckload pickup and delivery problem with a single depot, using a mixed-integer linear programming formulation for modeling and solving, differing from our homogeneous vehicle setting. Bouyahyious and Bellabdaoui [25] considered the full-truckload selective multi-depot vehicle routing problem with time windows and heterogeneous fleet constraints, assuming a one-to-one correspondence between pickup and delivery points and vehicles being incapacitated. In contrast, our study allows one pickup point to correspond to multiple delivery nodes and one order to be served by different vehicles at different times.

Xue et al. [26] proposed a hybrid column generation with a genetic algorithm and variable neighborhood search method for the FTVRPTW with a homogeneous vehicle fleet, single depot, and multiple shifts, efficient for large-scale real-life FTL problems. Bouyahyious and Bellabdaouia [27] examined a selective full-truckload multi-depot vehicle routing problem with time windows, multi-depot, and selective orders in an empty return scenario, aiming to maximize overall profit using a genetic algorithm. Bouyahyious and Bellabdaoui [1] studied the full-truckload vehicle routing problem with multiple depots and time windows (SFTMDVRPTWs) in an empty return scenario and order selection. Melchiori et al. [3] investigated the full-truckload vehicle routing problem with multiple depots and synchronization decisions for a homogeneous vehicle fleet. These studies differ from ours in terms of homogeneous truck usage, single-depot settings, and the requirement for vehicles to return to the starting node after completing their routes.

Deep learning algorithms have been extensively applied to various vehicle routing problems over the past 15 years. Two recent survey papers by Tsolaki et al. [28] and Bai et al. [29] highlight the growing number of studies on machine learning-assisted vehicle routing algorithms. Most studies utilize artificial neural networks combined with reinforcement learning. However, there are only a few studies that have focused on applying machine learning to full-truckload routing problems.

Agent-based modeling for dynamic vehicle routing problems has been proposed in several studies (Baykasoglu and Kaplanoglu [6]; Martin et al. [30]) with promising results. Most agent-based studies focus on less-than-truckload (LTL) problems. Mes et al. [31] employed agent-based scheduling using look-ahead heuristics for a specific FTL problem that does not include many of the problem characteristics addressed in our study. In a similar FTL problem, Mes et al. [32] proposed a multi-agent model in which shipper agents send jobs to sequential auctions, and vehicle agents bid on these jobs.

### 3. Problem Description

To formulate a meaningful decision problem that meets the practical needs of firms, we first analyze the logistics processes of a leading logistics firm in Turkey, which transports full-truckload (FTL) freights (e.g., liquid foods and chemicals) between pairs of points within a large distribution network. The firm manages its operations through branches across Turkey, utilizing a large fleet of trucks and various types of trailers, including steel, chrome, aluminum, heated, and insulated units. Pickup and delivery operations are conducted daily on a 24 h/7 days a week basis.

Firm data are used to assign values to the parameters in our model, such as freight tonnages, load types, trailer washing times, tractor speed, and trailer capacity. Fuel costs for full and empty travel, washing costs, and trailer replacement costs are obtained from the firm’s accounting records.

Table 1 introduces the notation used in the problem’s definition and methodology.

**Table 1.** Notation definitions.

Sets		Indices	
I	Set of nodes	$i$	Pickup node index of an order ( $i \in I$ )
K	Set of vehicles	$j$	Delivery node index of an order ( $j \in I$ )
D	Set of trailer types	$k$	Vehicle index ( $k \in K$ )
O	Set of order agents	$m$	Order agent index
F	Set of freight types	$f$	Freight type index ( $f \in F$ )
$p^w$	Set of washing nodes	$t$	Trip index
$p^t$	Set of trailer change nodes	$o_m$	Order agent $m$ ( $o_m \in O$ )
$h_f$	Set of suitable trailer types for freight $f$	$o_{mt}$	$T$ .th trip of order agent $m$
Parameters			
$OA_k$	Accepted order list of vehicle agent $k$	$Q$	Capacity of a trailer (ton)
$OA_k^*$	Temporary order list of vehicle agent $k$	$l_k$	Current node of vehicle $k$
$Cost(OA_k)$	Cost function of $oa_k$	$u_k$	Current trailer type of vehicle $k$
$C_k$	Total cost of vehicle $k$	$y_k$	Whether vehicle $k$ has changeable trailer (1: yes; 0: no)
$C_o^k$	Cost of order $o$ in vehicle $k$	$l_w$	Nearest washing node
$bid_o^k$	Cost offer of vehicle $k$ for order $o$	$l_p$	Nearest trailer change node
$d_k$	Current trailer type of vehicle $k$	$w_o$	Weight of order $o$ (ton)
$c^f$	Unit loaded trip cost (TL/km)	$p_o$	Pickup node of order $o$
$c^e$	Unit empty trip cost (TL/km)	$d_o$	Delivery node of order $o$
$c^p$	Trailer change cost	$t_o^{AVL}$	Earliest pickup time of order $o$
$c^s$	Annual fixed cost of a vehicle	$t_o^{ALN}$	Latest pickup time of order $o$
WM	Trailer washing cost matrix	$t_o^{DVL}$	Earliest delivery time of order $o$
WT	Trailer washing time matrix	$t_o^{DLN}$	Latest delivery time of order $o$
TM	Travel time matrix between any two nodes	$t_o^{ARV}$	Request time of order $o$
IM	Revenue matrix between any two nodes	$t_o^{RES}$	Latest time to respond order $o$
DM	Distance matrix between any two nodes		

Freight orders arrive randomly, and upon arrival, there is a specified latest response time by which decisions must be made regarding the assignment of vehicles to an order or its rejection. Acceptance decisions are influenced by system dynamics, including the trailer’s compatibility with the freight type, transportation plans, profitability, and available vehicle capacities. The primary decision variables involve the assignment of trips within an order to vehicles and the sequencing of these trips for a specific vehicle. For instance, consider an order consisting of seven trips. We can assign all trips to a single vehicle or distribute the trips among three vehicles, with the distribution being two trips, three trips, and two trips, respectively. At the vehicle level, the assigned trips must be sequenced to

maximize profitability and meet time window constraints. Thus, our problem encompasses both fleet assignment and routing decisions.

Figure 1 illustrates a typical vehicle route where the vehicle completes trip  $o_{32}$  after finishing trip  $o_{11}$  and then proceeds to node 4 to load trips  $o_{21}$  and  $o_{22}$ . To load a different type of freight for trip  $o_{41}$  at node 7, the vehicle goes to node 6 to wash its trailer. After completing trip  $o_{41}$ , the vehicle proceeds to a trailer change location at node 9 to accommodate the different trailer type required for trip  $o_{52}$ , which starts at node 9.

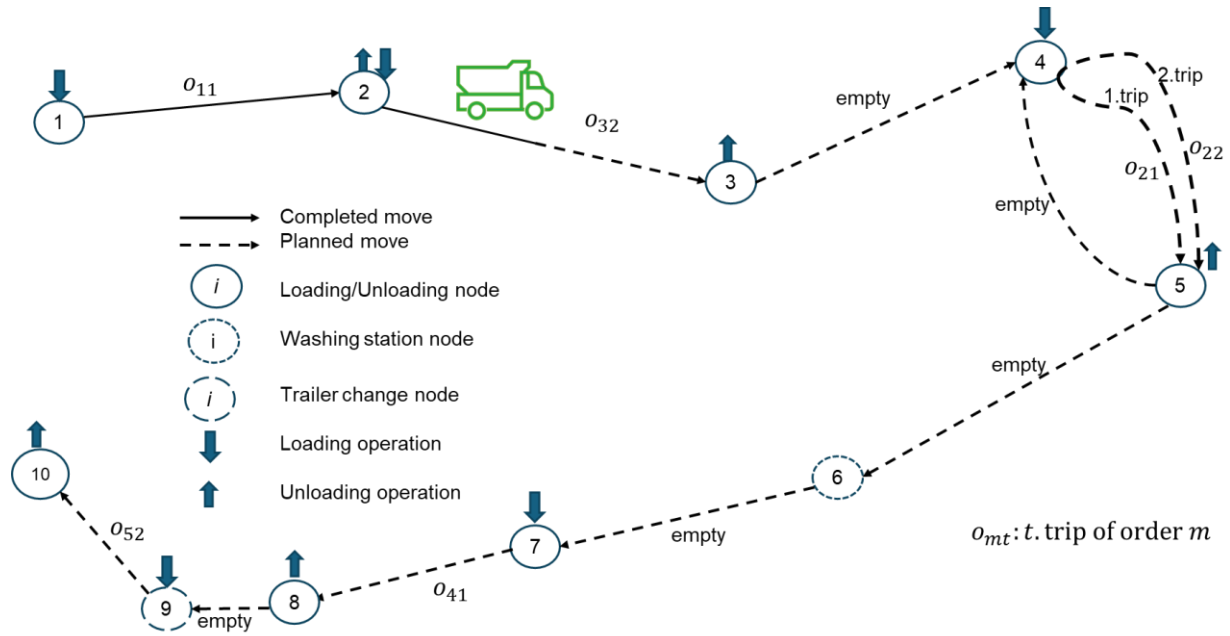


Figure 1. A typical route of a vehicle.

Making vehicle assignment and routing decisions at the arrival of an order and not modifying them thereafter may result in missing better alternatives that could emerge over time due to the dynamic nature of our problem. Therefore, our problem formulation permits revisions to early decisions until the latest response time for an order,  $t_o^{RES}$ , as shown in Figure 2, in relation to the pickup time window of an order. The latest response time can vary for each order and is set by either the customer or the carrier. This time may extend beyond the start time ( $t_o^{AVL}$ ) of the pickup time window, indicating that any assigned but not yet started trip can be considered for the interchange mechanism, which identifies better alternative solutions with lower costs. The execution of the interchange mechanism will be explained in more detail in Section 3.

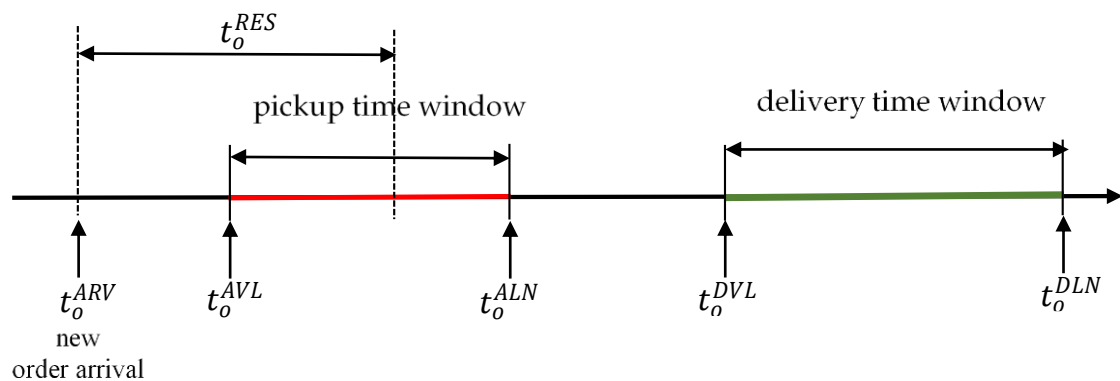


Figure 2. Illustration of time windows and the latest response time.

Based on the aforementioned discussions, the basic characteristics of the problem are summarized as follows:

- i. The transportation network consists of pickup, delivery (warehouses, seaports, and factories), trailer washing, and trailer changing nodes.
- ii. Each order may need multiple trips which can be carried out by one or multiple vehicles.
- iii. Each request contains a single type of freight.
- iv. There are separate time windows for the pickup and delivery of each order.
- v. Vehicles can be either trucks or semi-trucks with removable trailers.
- vi. Certain types of trailers can be used to carry specific types of freight.
- vii. When necessary, removable trailers can be changed to carry specific types of freight at predetermined locations.
- viii. If the freight type changes for the subsequent trip, a trailer washing time and cost may be required depending on the previous type of freight.
- ix. The following cost factors are considered: (1) the unit variable cost per kilometer with freight or empty (including fuel cost, maintenance cost, etc.); (2) the trailer washing cost, if needed; (3) the trailer change cost; and (4) the monthly fixed cost of vehicles.

Although agent-based modeling permits models with fewer assumptions, we make some assumptions to reach a reasonable and practical problem formulation. Most of these assumptions can be easily removed. Our assumptions are as follows:

- i. Environmental factors (e.g., traffic and accidents) are ignored in movement times between nodes, and all travel times are known in advance.
- ii. Order arrival times, freight sizes, freight types, and pickup and delivery nodes are generated from appropriate random distributions.
- iii. There are always enough suitable trailers available at any time.
- iv. There is a sufficient number of drivers to meet the legal working time limits of drivers.
- v. The fleet size does not change over time.
- vi. The vehicles are identical in terms of speed and the types of freight they can carry.
- vii. Transport times are computed based on the average speed of the vehicles.
- viii. While the vehicles are loaded and on the move, they cannot be directed to another node for another order.
- ix. When a new order arrives, all vehicle assignments can be changed to improve the current solutions until the predefined response time for each order.
- x. All unit costs and transportation tariffs between two nodes do not change over time.
- xi. All unit costs and transportation tariffs between two nodes are the same for all vehicles.

#### 4. Methodology

In logistics and industrial operations, problems rarely occur in static environments where all parameters are known in advance. On the contrary, numerous unpredictable events (such as order cancellations, vehicle breakdowns, delayed deliveries, and new customer arrivals) are the fundamental characteristics of real-world problems. The optimization of systems with such a high number of unknowns is the primary concern of dynamic optimization (DO). DO problems are a type of problem in which the feasible solution space and/or the value of the objective function changes due to dynamic events occurring over time.

This study addresses a dynamic optimization problem, given that the interarrival times of orders, freight sizes, freight types, and time windows are all random variables. Furthermore, these random variables may not follow stationary distributions. When a new order arrives in the system, previous vehicle assignments and routes need to be reconsidered, as the feasible solution space changes over time.

Perhaps the first approach that comes to mind to solve DO problems is to re-optimize the problem for the remaining time horizon using the same optimization technique, taking into account the emerging dynamic events. For example, a new transportation request can be responded to as soon as it arrives, or the system can be re-optimized once a certain number of requests have been received. In DO problems, the concept of an immutable optimal solution does not exist. In fact, the re-optimization policy itself can turn into a dynamic optimization problem. Instead of responding instantaneously to emerging dynamic events, responding by accumulating them can lead to better solutions in terms of more efficient use of limited resources (Karami et al. [33]).

Therefore, the goal of methods used to solve dynamic optimization problems is not to find a static optimal solution but to locate high-quality solutions while closely monitoring the time-dependent movement of the feasible solution space. The aim here is to develop optimization policies that can closely track these changes. Traditional methods developed for deterministic optimization do not possess such characteristics. Among meta-heuristics, population-based methods, such as evolutionary algorithms or particle swarm optimization, have been proposed as alternatives that can adapt to this field with the necessary updates and adjustments. However, evolutionary algorithms do not perform well in rapidly changing environments, as they require numerous fitness evaluations, and there is limited time to find satisfactory solutions (Lu et al., [34]).

There is no consensus on how to assess the performance of an algorithm or solution approach for solving a dynamic optimization problem. Karami et al. [33] propose that a dynamic optimization algorithm is considered optimal if its objective function value matches that of its static counterpart. They acknowledge that this is not a reasonable benchmark but serves as a tangible reference. Solving the static counterpart of our problem, even for small problem sizes, is very time-consuming due to the *np*-hard nature of the computational complexity. This study focuses on investigating the influence of the latest response time, along with the developed interchange mechanism, on the solution quality.

Agent-based models addressing dynamic optimization problems have allowed for various optimization mechanisms to track optima moving through the search space. This has led many researchers to utilize agent-based modeling for the optimization of dynamic systems. As noted by Baykasoglu and Kaplanoglu [6], the agent paradigm has proven to be a promising approach for developing intelligent systems due to its features of autonomy, flexibility, and collaborative problem-solving behavior. Multi-agent systems consist of a series of autonomous agents that interact with each other and their environment. The term autonomy here implies that the agents are active entities capable of making their own decisions.

#### 4.1. Agent-Based Framework

##### 4.1.1. Framework Architecture and Operation

This study proposes an auction-based logistics planning system. There are three types of agents in this system: vehicle agents, order agents, and a dispatcher agent. Figure 3 illustrates the interactions between these three types of agents.

##### Order Agent

Order agents represent each dynamic transportation request arriving in the system. The attributes of an order agent include the pickup and delivery nodes, time windows for pickup and delivery, the response time to receive the final bid, the order weight, and the freight type. The order agents remain in the system either throughout the response time ( $t_o^{RES}$ ) or until the end of their transportation time, depending on acceptance by the vehicle agents. At the end of the response time, an order agent is either rejected or accepted.



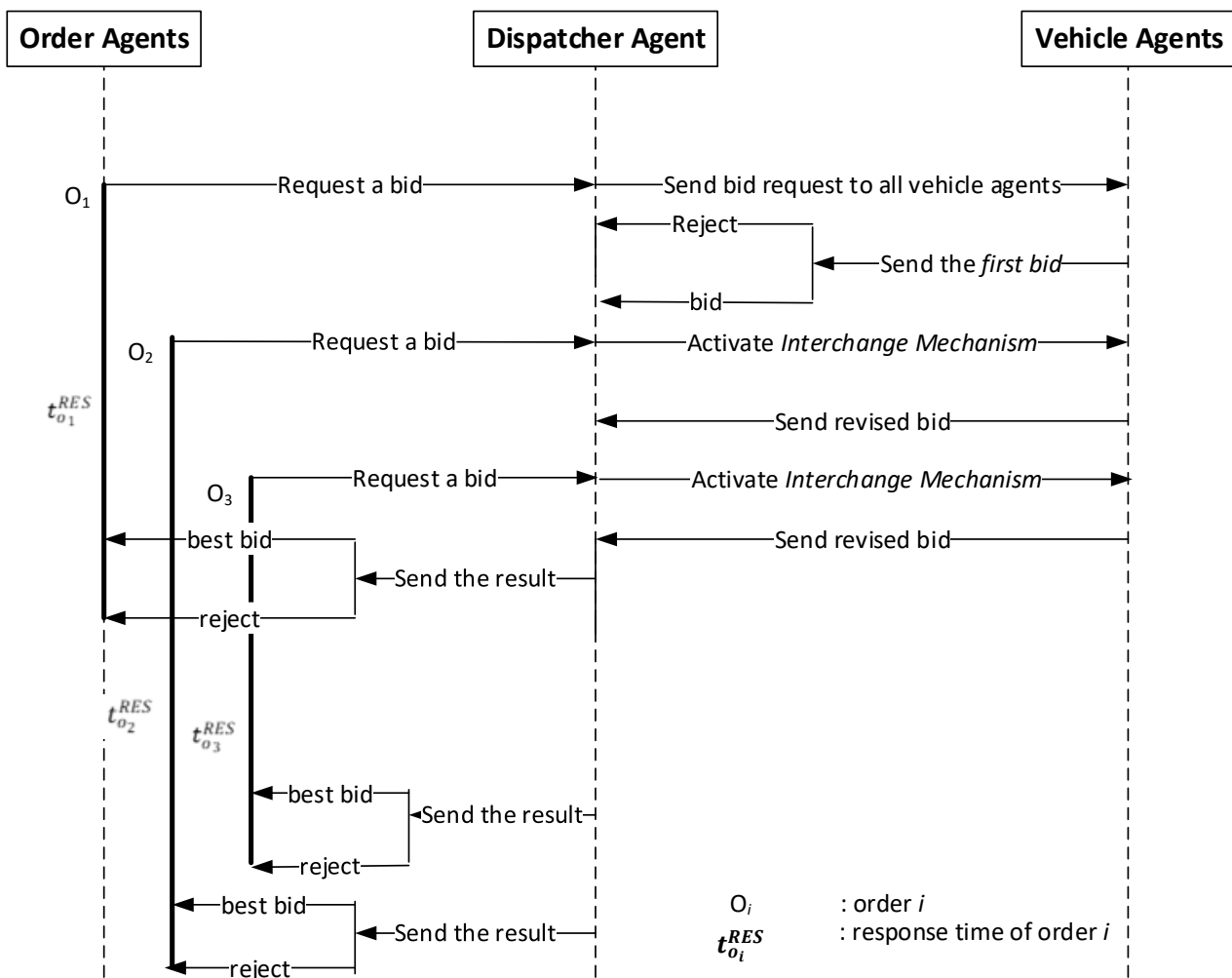


Figure 3. Interaction diagram of the agents.

### Dispatcher Agent

When an order agent enters the system, it directly communicates with the dispatcher agent to initiate the auction-bidding system. The dispatcher agent holds a list of available vehicle agents in the system and determines the suitable vehicle agents when an order agent enters the system. The dispatcher agent is responsible for coordinating vehicle assignment decisions and managing the bidding and negotiation processes. Therefore, the dispatcher is responsible for generating alternative transportation plans with lower costs through an interchange mechanism.

### Vehicle Agent

A vehicle agent has attributes including vehicle number, weight capacity, current location, current freight type, and trailer type. Upon receiving a bid request from the dispatcher agent, the vehicle agent calculates the cost of carrying a new order request and sends its bid to the dispatcher agent. Vehicle agents also revise their bids within the response time of all active orders when requested by the dispatcher agent upon the arrival of a new order.

#### 4.1.2. Agent Algorithms

All agents in the system execute defined algorithms to perform various operations and share information based on the outcomes of these algorithms. The interrelationships between agents are illustrated in Figure 4.

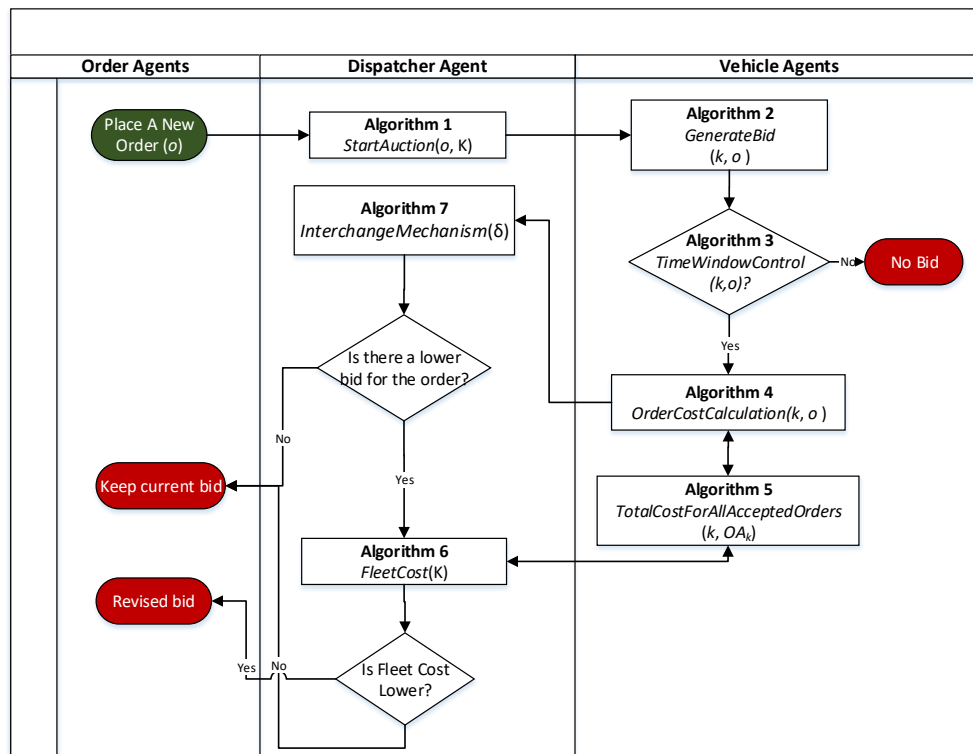


Figure 4. Flowchart of the agent algorithms.

When a new order ( $o$ ) arrives in the system, it initiates Algorithm 1,  $StartAuction(o, K)$ , which collects bids from the vehicles in set  $K$  through Algorithm 2,  $GenerateBid(k, o)$ . Algorithm 1 selects the best bid,  $(bid_o^*)$ , and stores it until the end of the order response time ( $t_o^{RES}$ ).  $GenerateBid(k, o)$  checks the trailer type, pickup/delivery time window constraints using Algorithm 3,  $TimeWindowControl(k, o)$ , and the profitability of carrying the order. If these constraints cannot be met or the order is not profitable for the vehicle, no bid is sent to the dispatcher agent. Otherwise, the bid is added to the list of accepted orders for that vehicle ( $OA_k$ ). Algorithm 6,  $InterchangeMechanism(\delta)$ , is then called to generate a better assignment of previous orders with the addition of the new order, where the term  $\delta$  represents the number of iterations for randomly selected pairs of assignments to interchange. We will explain how  $InterchangeMechanism(\delta)$  works in more detail using an example later in this section.

---

**Algorithm 1**  $StartAuction(o, K)$

---

**input:**  $o; K$

**local:**  $B [(k_1, C_o^{k_1}), \dots, (k_n, C_o^{k_n})]$  list of bids for order  $o$ ;  $t_{now}$  current time;  $bid_o^*$  the best bid for order  $o$   
 $B \leftarrow \emptyset$

**for all**  $k \in K$  **do**

$bid_o^k \leftarrow GenerateBid(k, o)$

$B \leftarrow B \cup \{bid_o^k\}$

**end**

$bid_o^* \leftarrow \text{Min} \{C_o^{k_i}\}$  in  $B$

**wait**  $t_o^{RES} - (t_{now} - t_o^{ARV})$

**if**  $bid_o^* \neq \emptyset$  **then**

$send\ msg('order\ accepted')$

**else**

$send\ msg('order\ rejected')$

**end**

---

**Algorithm 2** GenerateBid ( $k, o$ )

---

```

input:  $o; k$ 
local:  $I_o$ , revenue of order;  $C_o^k$ , order cost
output:  $bid_o^k$ 
if  $y_k = 0$  and  $d_k \notin h_f$  then
     $bid_o^k = \emptyset$ 
else
    if TimeWindowControl ( $k, o$ ) = false then
         $bid_o^k = \emptyset$ 
    else
        OrderCostCalculation ( $k, o$ )
         $bid_o^k \leftarrow \{k, C_o^k\}$ 
        /* profitability control */
         $I_o \leftarrow IM[p_o, d_o]$ 
        if  $I_o < C_o^k$  then
             $bid_o^k = \emptyset$ 
        else
             $OA_k \leftarrow OA_k \cup \{o\}$ 
            call InterchangeMechanism( $\delta$ )
        end
    end
return  $bid_o^k$ 

```

---

**Algorithm 3** TimeWindowControl ( $k, o$ )

---

```

input:  $k \in K$  vehicle;  $o \in O$  order
output: true or false
local:  $OA_k^*$ , temporarily accepted order list of vehicle agent;
 $OA_k^* \leftarrow copy\ OA_k \mid \mid \{\}$ 
if  $OA_k^* = \emptyset$  then
    return true
end
for  $o \leftarrow 0$  to  $\mid OA_k^* \mid$  do
    if  $o = 0$  then
        if  $t_o^{ALN} > t_o^{DVL} + TM[d_o, p_o]$  then
            return true
        end
    else if  $i = \mid OA_k^* \mid$  then
        if  $t_o^{DVL} + TM[d_{o-1}, p_o] < t_o^{ALN}$  then
            return true
        end
    else
        if  $TM[d_{o-1}, p_o] + TM[d_o, p_o] < t_o^{ALN} - t_{o-1}^{DVL}$  and  $t_o^{DVL} < t_o^{ALN} +$ 
 $TM[d_o, p_o]$  and
         $t_o^{ALN} > t_{o-1}^{DVL} + TM[d_{o-1}, p_o]$  then
            return true
        end
    end
end
return false

```

---

In Algorithm 4, *OrderCostCalculation* ( $k, o$ ), the vehicle agent calculates its bid for order  $o$  by determining the difference between the total cost of all currently assigned orders and the total using cost Algorithm 5 after adding the new job to the current list.

**Algorithm 4** OrderCostCalculation ( $k, o$ )

---

**input:**  $k \in K$  vehicle;  $o \in O$  order  
**output:**  $C_o^k$ : cost of order  $o$  in vehicle  $k$   
**local:**  $C_k$ : total cost of vehicle  $k$ ;  $OA_k^* \subseteq O$  accepted orders list of vehicle  $k$   
 $C_k \leftarrow TotalCostForAllAcceptedOrders(k, OA_k)$   
 $OA_k^* \leftarrow OA_k \cup \{o\}$   
 $C_k^* \leftarrow TotalCostForAllAcceptedOrders(k, OA_k^*)$   
 $C_o^k \leftarrow C_k^* - C_k$   
**return**  $C_o^k$

---

**Algorithm 5** TotalCostForAllAcceptedOrders ( $k, OA_k$ )

---

**input:**  $k, OA_k$   
**output:**  $C_k$ , total cost of transportation plan for vehicle  $k$   
**local:**  $u_k^* \subseteq T$ , the last trailer type on vehicle  $k$ ;  $l_k^* \in P$ , the last node of vehicle  $k$ ;  $l_w \in P^w$ , nearest washing point;  $l_p \in P^t$ , the nearest trailer replacement node

$C_k \leftarrow 0$   
 $u_k^* \leftarrow \{u_k\}$   
 $l_k^* \leftarrow l_k$   
 $OA_k \leftarrow sort(OA_k)$   
**for**  $m \leftarrow 1$  **to**  $|OA_k|$  **do**  
    **if**  $m > 1$  **and**  $f_{m-1} \neq f_m$  **then**  
        **if**  $l_k^* \in P^w$  **then**  
             $l_w \leftarrow l_k^*$   
        **else**  
             $l_w \leftarrow NP(p^w, p_o)$   
        **end**  
         $C_k \leftarrow C_k + WM[f_{m-1}, f_m]$   
         $C_k \leftarrow C_k + (DM[l_k^*, l_w] * c^e)$   
         $l_k^* \leftarrow l_w$   
    **end**  
    **if**  $y_k = 1$  **and**  $u_k^* \not\subseteq h_f$  **then**  
        **if**  $l_k^* \in P^t$  **then**  
             $l_p \leftarrow l_k^*$   
        **else**  
             $l_p \leftarrow NP(p^t, p_o)$   
        **end**  
         $C_k \leftarrow C_k + c^p$   
         $C_k \leftarrow C_k + (DM[l_k^*, l_p] * c^e)$   
         $l_k^* \leftarrow l_p$   
         $u_k^* \leftarrow h_f$   
    **end**  
    **if**  $l_k^* \neq p_m$  **then**  
         $C_k \leftarrow C_k + (DM[l_k^*, p_m] * c^e)$   
    **end**  
     $C_k \leftarrow C_k + (DM[p_m, d_m] * c^f)$   
     $l_k^* \leftarrow d_m$   
**end**  
 $C_k \leftarrow C_k + c_k^s$   
**return**  $C_k$

---

## 4.1.3. Interchange Mechanism

Algorithm 6, *InterchangeMechanism* ( $\delta$ ), outlines the steps that are taken when a new order arrives in the system, potentially leading to better assignment solutions. The *FleetCost* ( $K$ ) function in Algorithm 7 computes the cost of all currently assigned orders for the entire fleet of vehicles, establishing a base cost for comparison with alternative assignments. Subsequently, a random vehicle (the incumbent vehicle) and a random order assigned to

this vehicle are selected. The algorithm seeks to find a better solution by performing a pairwise comparison between the fleet cost of keeping the incumbent vehicle’s assignment and the cost of reassigning that order to another vehicle. This random comparison iteration is repeated  $\delta$  times.

---

**Algorithm 6** InterchangeMechanism( $\delta$ )

---

**input:**  $\delta$  number of iterations  
**local:**  $K^*$  temporary copy of the vehicle list,  $K^{o^*}$  order alternative vehicle list  
 $K^* \leftarrow \text{copy } K$   
 $C \leftarrow \text{FleetCost}(K)$   
**for**  $i \leftarrow 1$  **to**  $\delta$  **do**  
     $k^* = \text{random } K^*$   
     $o^* = \text{random } OA_{k^*}$   
     $K^{o^*} \leftarrow \emptyset;$   
     $C_{o^*}^{k^*} \leftarrow \text{OrderCostCalculation}(k^*, o^*)$   
    **for all**  $k \neq k^*$  **do**  
         $C_{o^*}^k \leftarrow \text{OrderCostCalculation}(k, o^*)$   
        **if**  $C_{o^*}^k < C_{o^*}^{k^*}$  **then**  
             $C^* \leftarrow \text{FleetCost}(K^*)$   
            **if**  $C^* < C$  **then**  
                 $OA_k \leftarrow OA_k \cup \{o^*\}$   
                 $OA_{k^*} \leftarrow OA_{k^*} - \{o^*\}$   
                 $K^{o^*} \leftarrow K^{o^*} \cup \{k\}$   
            **end**  
        **end**  
    **end**  
    **if**  $|K^{o^*}| > 0$  **then**  
        **call**  $\text{StartAuction}(o^*, K^{o^*})$   
    **return**  
**end**

---



---

**Algorithm 7** FleetCost ( $K$ )

---

**input:**  $K$   
**output:**  $C$ : total transportation cost of all vehicles in the system  
 $C \leftarrow 0$   
**for all**  $k \in K$  **do**  
     $C_k \leftarrow \text{TotalCostForAllAcceptedOrders}(k, OA_k)$   
     $C \leftarrow C + C_k$   
**end**  
**return**  $C$

---

To illustrate how the interchange mechanism works, Table 2 provides an example dataset for five freight orders. The dataset includes information on freight types, pickup and delivery nodes, pickup and delivery time windows, and the response time for the dispatcher agent to send the final bid to the order agent, respectively.

Figure 5 illustrates the interchange mechanism in two distinctive cases using the example dataset. In Figure 5a; Order  $o_1$ , requiring two trips ( $o_{11}$  and  $o_{12}$ ), is assigned to vehicles 1 and 3, respectively, which offer the lowest bids. When order  $o_2$ , with a single trip  $o_{21}$ , enters the system, no suitable vehicle is available that meets both the freight type and time window requirements. Consequently, order  $o_2$  continues to search for a vehicle by participating in regular auctions. When order  $o_3$  enters the system, it is assigned to vehicle 2, thereby activating the interchange mechanism. As shown in the Gantt chart, trip  $o_{11}$  is transferred from vehicle 1 to vehicle 2 via the interchange mechanism. Trip  $o_{21}$ , re-entering the auction-based system, is then placed in the vacant slot in vehicle 1’s transport plan.

Upon the arrival of order  $o_4$ , which is unprofitable for all vehicles, it is temporarily assigned to vehicle 4, which submitted the lowest cost bid. In Figure 5b; Order  $o_4$  remains in a waiting status until the end of the response time. Subsequently, with the assignment of order  $o_5$  to vehicle 1, the interchange mechanism is activated again, and vehicle 1 submits a new bid for  $o_{41}$ . Thus, as shown in Figure 5, order  $o_4$  moves from vehicle 4, where it is unprofitable and temporarily assigned, to vehicle 1, where it becomes profitable.

Table 2. Freight orders and their attributes.

$o_i$	Freight Type	$p_o$	$d_o$	$w_o$ (ton)	$t_o^{AVL}$	$t_o^{ALN}$	$t_o^{DVL}$	$t_o^{DLN}$	$t_o^{RES}$ (min)
$o_1$	Glue	$N_5$	$N_{15}$	16	Day 1 14:00	Day 3 22:00	Day 1 16:00	Day 2 22:00	240
$o_2$	Petrol	$N_8$	$N_{15}$	8	Day 1 14:00	Day 1 15:00	Day 1 22:00	13.4.23 22:00	120
$o_3$	Glue	$N_7$	$N_5$	8	Day 1 11:00	Day 2 1:00	Day 1 2:00	12.4.23 1:00	030
$o_4$	Cooking oil	$N_2$	$N_6$	8	Day 3 11:00	Day 3 20:00	Day 3 20:00	14.4.23 20:00	180
$o_5$	Cooking oil	$N_{15}$	$N_2$	8	Day 2 8:00	Day 3 17:00	Day 4 17:00	14.4.23 17:00	60

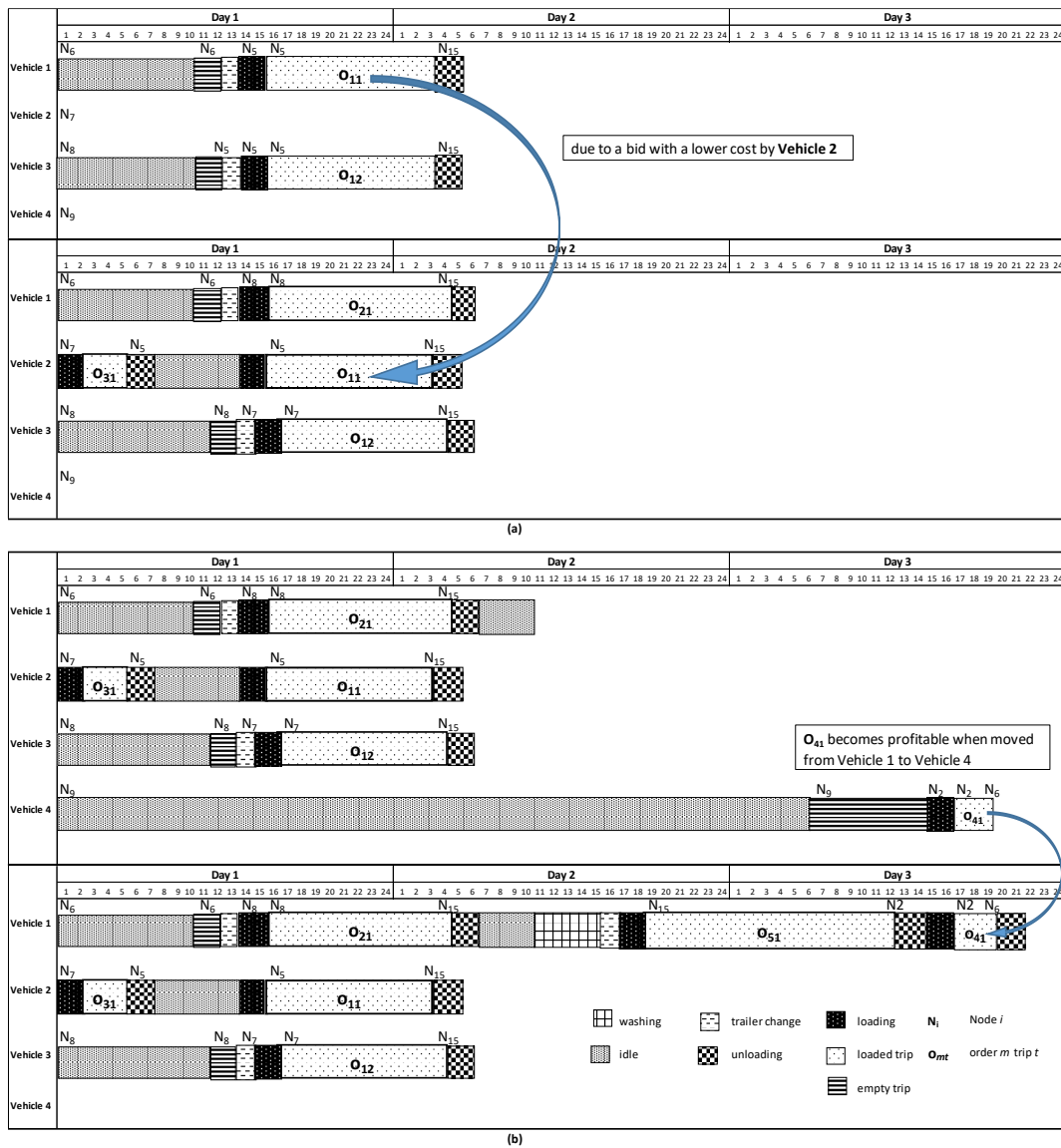


Figure 5. A numerical example of the proposed interchange mechanism (a) search for a feasible vehicle, (b) search for a vehicle with profitable assignment.

## 5. Simulation Experiments

### 5.1. Design of Simulation Experiments

A simulation model was developed to test our multi-agent-based vehicle assignment and routing approach under different conditions. In this model, order weights are generated from a discrete uniform distribution. Order arrival times are generated from a Poisson arrival process. The response time to requests is assumed to follow an exponential distribution. For the simulation tests of the developed system, an application interface was developed using the C# programming language in the Microsoft Asp.Net Core library infrastructure, which ran on one virtual machine (VM) in the Microsoft Azure cloud infrastructure, each with 2.7 GHz, 2 core processors, and an 8.0 GB cache capacity. The combinations of the following parameters were used to generate simulation scenarios.

#### 5.1.1. Multiple-Trip Order Ratio (%)

As our system does not allow the partial fulfillment of orders with multiple trips, it was important to investigate the impact of the multiple-trip order ratio on the performance of the agent-based model. This ratio was varied at three levels: 20%, 30%, and 40%. For example, at a 20% ratio, 80% of orders involve a single trip, while the remaining 20% of orders involve multiple trips uniformly distributed between 2 and 15.

#### 5.1.2. Congestion Factor ( $\rho$ )

The congestion factor is the level of congestion that may be experienced in assigning orders to vehicles, considering the intensity of orders and the number of trips that can be made with the available number of vehicles. When this ratio exceeds 1, it signifies that there are more orders than the system can accommodate with the available vehicle capacity. This ratio is calculated as:

$$\rho = \frac{\lambda_d \times \bar{s}}{K \times \mu} \tag{1}$$

where  $\lambda_d$  is the average number of orders received per week,  $\bar{s}$  is the average number of trips per order,  $\mu$  is the average service rate per vehicle (trips/week), and  $K$  is the number of vehicles in the fleet. The values of 84, 168, and 336 orders/week were taken as  $\lambda_d$ . The average number of trips per order was calculated as 2.5, 3.25, and 4.0 for multi-trip demand rates of 20%, 30%, and 40%, respectively.

The experimental design was based on a 17-node distribution network. Table 3 shows the probability distribution of the order pickup nodes, indicating that most order pickups originated from nodes 8, 12, and 14.

**Table 3.** Probability distribution of order pickup nodes.

Pickup Node Number	Probability ( $w_i$ )
8	0.24
12	0.24
14	0.24
Remaining 14 nodes	0.02

The average distance per one-way trip between pickup and delivery is calculated as shown in Equation (2):

$$\bar{d} = \sum_{i=1}^{17} w_i \left[ \frac{\sum_{j=1, i \neq j}^{17} d_{ij}}{16} \right] \tag{2}$$

Assuming that most trips return to the same pickup node after delivery, the average time per trip can be calculated as  $2\bar{d}/v$ , where  $v$  is the average speed assumed for all vehicles. Assuming that all vehicles operated 24 h a day, 7 days a week, the average service rate per vehicle (trips/week) was 12.74 trips/week in our case. Table 4 displays the congestion factors for different scenarios.

**Table 4.** Congestion rates for different scenarios.

Multi-Trip Rate (%)	20	20	20	30	30	30	40	40	40
$\bar{s}$	2.50	2.50	2.50	3.25	3.25	3.25	4.00	4.00	4.00
$\lambda_d$	336	168	84	336	168	84	336	168	84
$\lambda_s$	840	420	210	1092	546	273	1344	672	336
$\mu$	12.74	12.74	12.74	12.74	12.74	12.74	12.74	12.74	12.74
Congestion Factor ( $\rho$ )	0.94	0.47	0.24	1.22	0.61	0.31	1.50	0.75	0.38

5.1.3. Response Time ( $t_o^{RES}$ )

Response time levels of 30, 60, 120, and 240 min were utilized in the analysis.

5.1.4. Interchange Mechanism

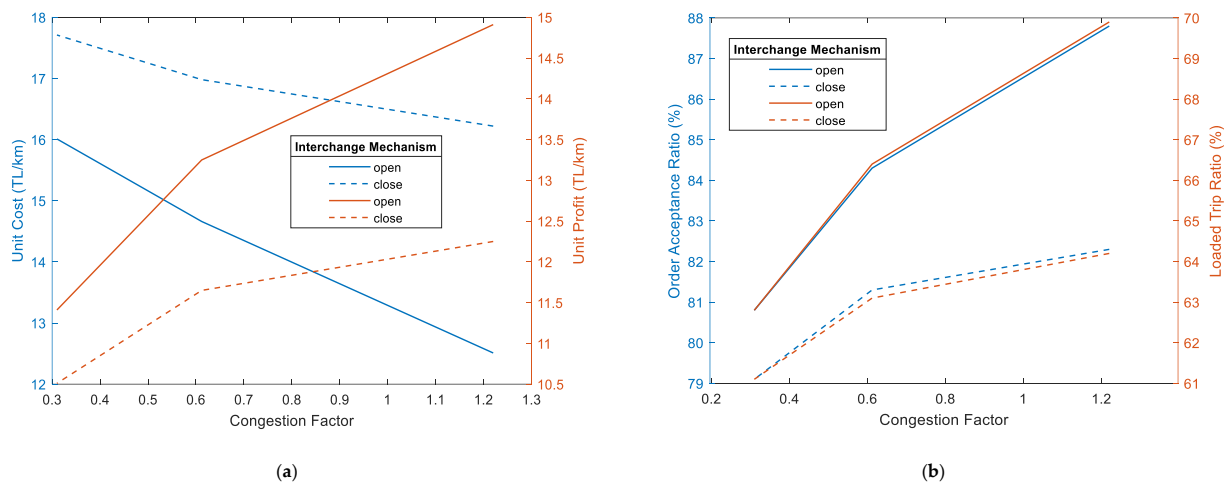
This mechanism is toggled between on and off states in the system, and its impact on system performance metrics was analyzed.

The following key performance indicators were used to evaluate the solutions generated by the agent-based algorithms:

- i. *Unit cost* (TL/km): The ratio of the total system cost to the total distance traveled by vehicles.
- ii. *Unit profit* (TL/km): The ratio of the total profit generated in the system to the total distance traveled by vehicles.
- iii. *Order acceptance ratio* (%): The percentage of orders approved for transportation at the end of the response time.
- iv. *Loaded trip ratio* (%): The percentage of the total distance traveled by all vehicles when loaded, compared with the total distance traveled when loaded or empty.

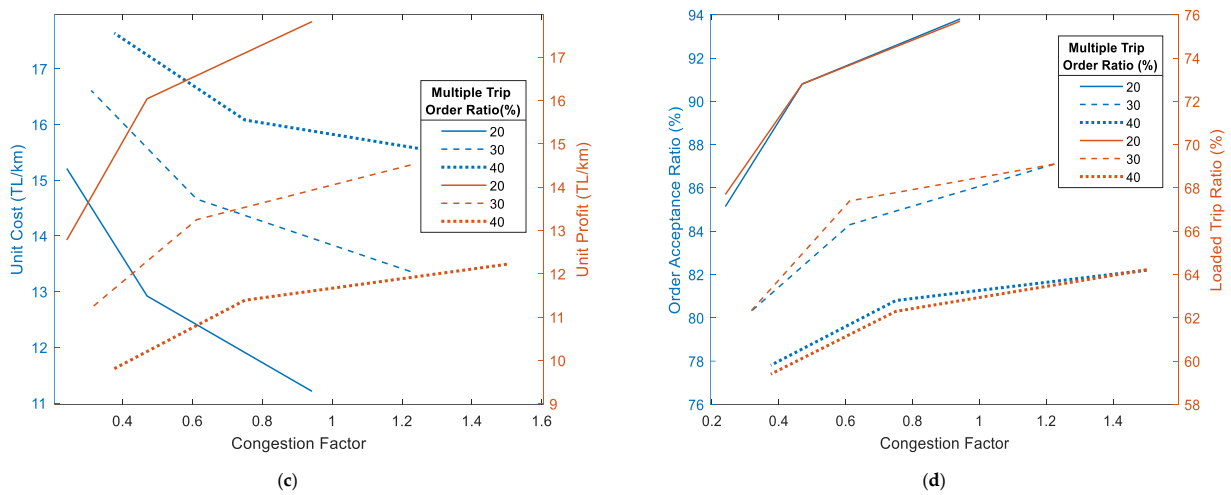
5.2. Analysis of Simulation Results

Figure 6a shows that a higher congestion factor reduces the unit cost, thereby increasing the unit profit per kilometer. This is attributed to simultaneous improvements in the order acceptance ratio and the loaded trip ratio. Activating the interchange mechanism results in a significant reduction in the unit cost, with a maximum reduction of 22.9% and an increase in the unit profit of 21.7%, as depicted in Figure 6b. The interchange mechanism dynamically improves current solutions throughout the response time.



**Figure 6.** Cont.

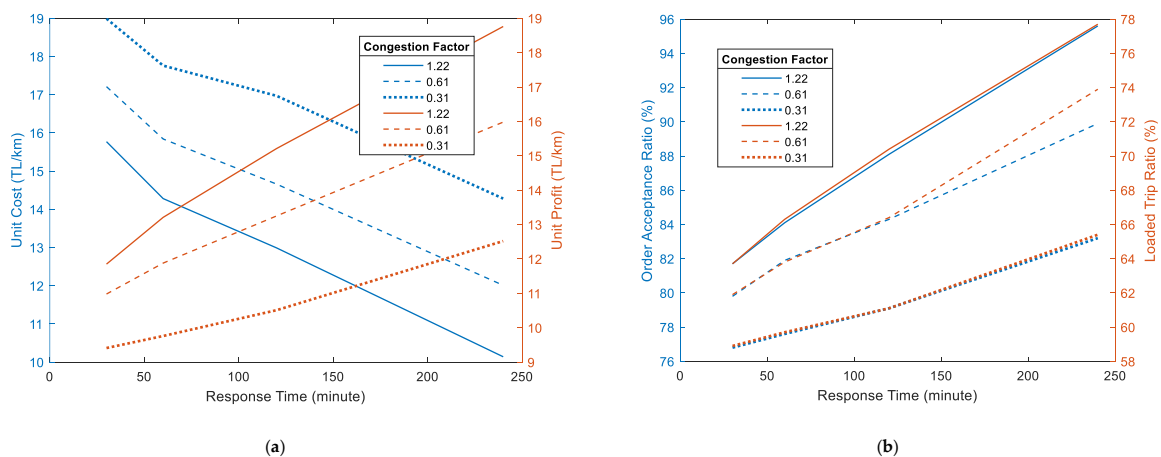




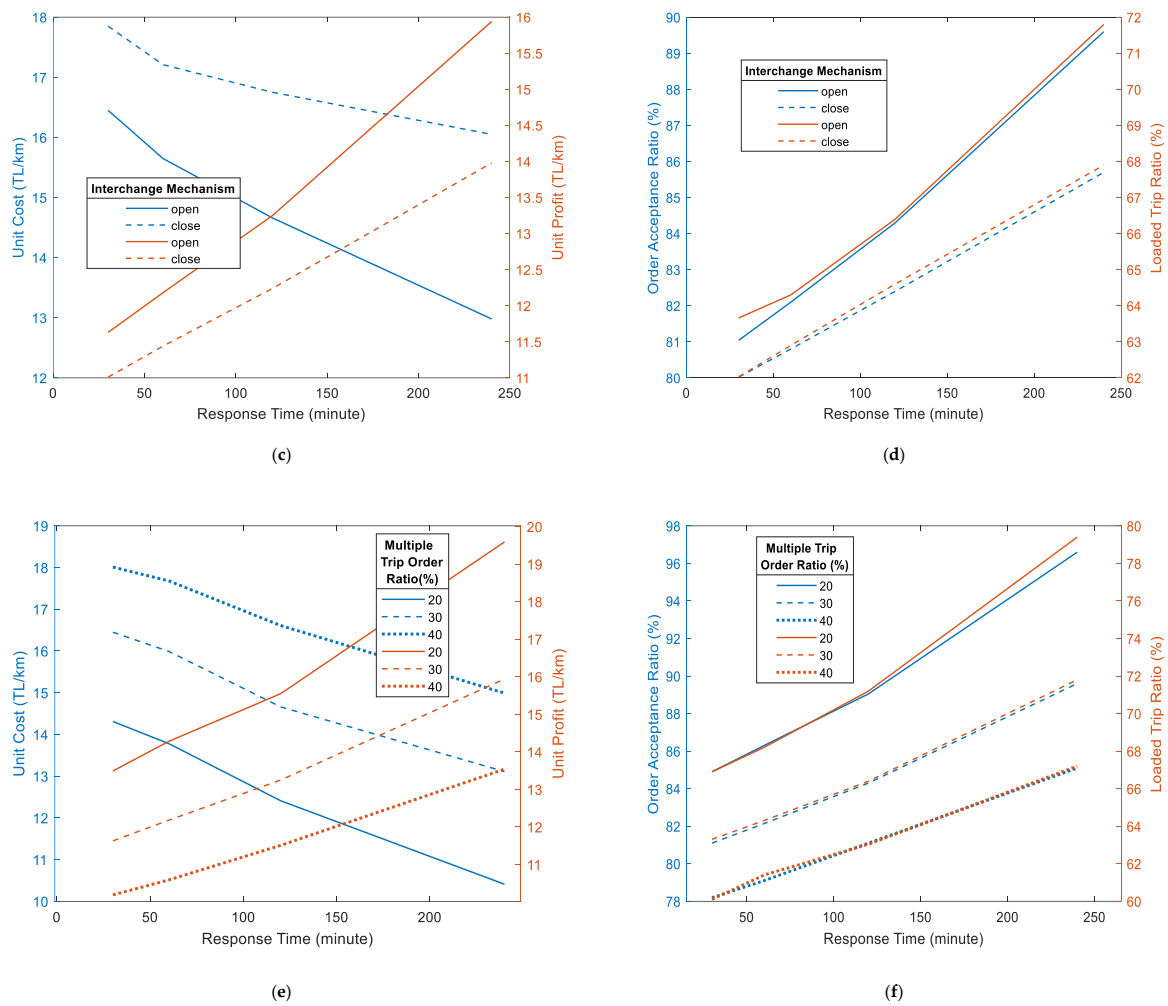
**Figure 6.** The impact of congestion factor on performance measures (a) unit cost and unit profit under interchange mechanism levels, (b) order acceptance ratio and loaded trip ratio under interchange mechanism levels, (c) unit cost and unit profit under multiple trip order ratio levels, (d) order acceptance ratio and loaded trip ratio under multiple trip order ratio levels.

Conversely, Figure 6c demonstrates that an increase in the ratio of orders with multiple trips leads to a rise in the unit cost and a decline in the unit profit. This is because the system prohibits the partial fulfillment of orders with multiple trips, resulting in a higher refusal rate for such orders. For the same reason, the order acceptance ratio and the loaded trip ratio significantly decrease as the ratio of orders with multiple trips increases, as illustrated in Figure 6d.

Figure 7a–f illustrate the impact of the response time on performance measures under varying conditions. Firstly, longer response times consistently have a positive impact on the unit cost, unit profit, order acceptance ratio, and loaded trip ratio in all cases. These positive effects on performance are further enhanced when the interchange mechanism is active, and the congestion factor increases. For example, Figure 7a,b show that at the longest response time of 240 min and the highest congestion factor of 1.22, the unit cost decreases by 35.7%, and the unit profit increases by 36.8%. Additionally, activating the interchange mechanism along with longer response times has a substantial synergistic positive impact on all performance measures, as seen in Figure 7c,d, due to the reasons explained earlier for Figure 6a,b. The adverse effect of a large ratio of orders with multiple trips is significantly reduced by longer response times across all performance measures, as seen in Figure 7e,f.



**Figure 7.** Cont.

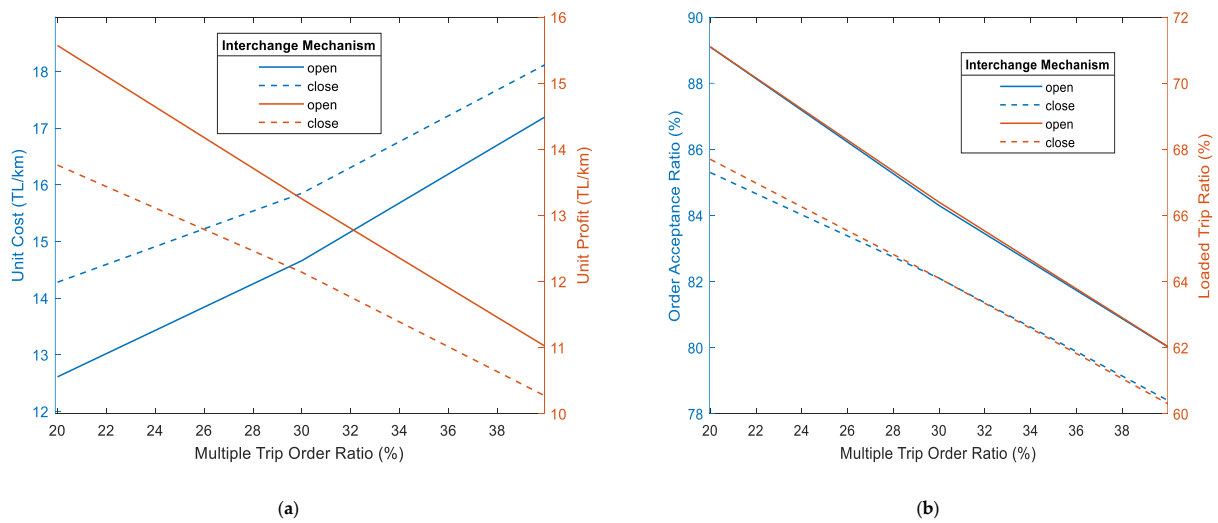


**Figure 7.** The impact of response time on performance measures (a) unit cost and unit profit under congestion levels, (b) order acceptance ratio and loaded trip ratio under congestion levels, (c) unit cost and unit profit under interchange mechanism levels, (d) order acceptance ratio and loaded trip ratio under interchange mechanism levels, (e) unit cost and unit profit under multiple trip order ratio levels, (f) order acceptance ratio and loaded trip ratio under multiple trip order ratio levels.

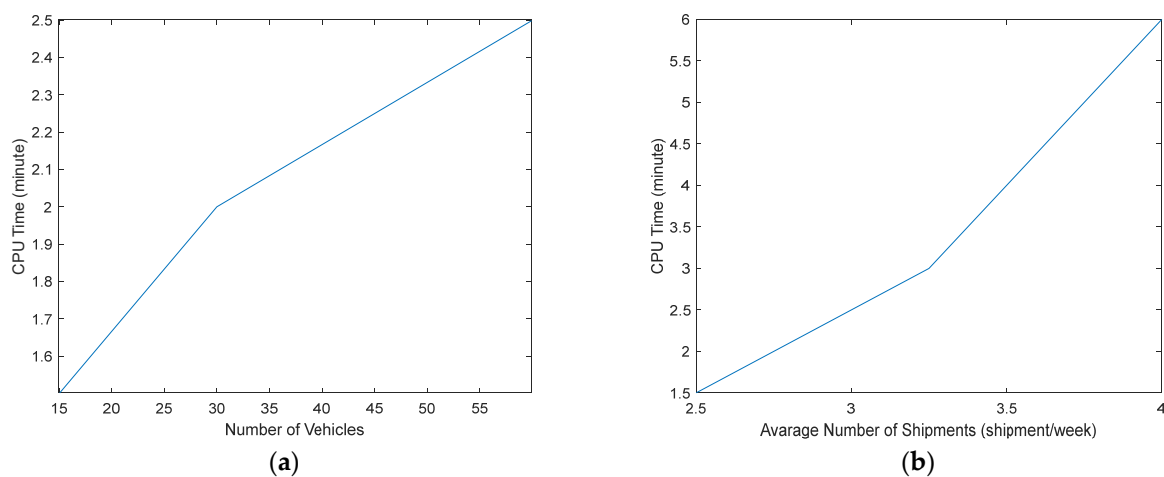
Similarly, the negative impact of an increased ratio of orders with multiple trips can be mitigated to a certain degree by activating the interchange mechanism across all performance measures. For instance, at the highest ratio of orders with multiple trips of 40%, the interchange mechanism can reduce the unit cost by 5% and increase the unit profit by 7.3%. This improvement is due to enhancements in the order acceptance ratio and the loaded trip ratio when the interchange mechanism is active, as seen in Figure 8a,b.

### 5.3. CPU Time Analysis

In an empirical computation time analysis, the CPU time spent by the developed vehicle assignment and routing algorithms in solving problems of different sizes is evaluated. The problem size is related to the number of vehicles and the number of trips that need to be planned per unit of time. Figure 9a shows that for a large fleet of vehicles, more bids need to be collected and more vehicles need to be screened, leading to an increase in the CPU time. However, this increase is not exponential. Figure 9b illustrates that as the number of trips to be assigned per week increases, the number of auctions and the number of trips to be interchanged also increases, leading to an approximately polynomial increase in CPU time.



**Figure 8.** The impact of multiple-trip order ratio on performance measures (a) unit cost and unit profit under interchange mechanism levels, (b) order acceptance ratio and loaded trip ratio under interchange mechanism levels.



**Figure 9.** The impact of the problem size on the CPU time.

### 6. Conclusions

In this study, a new dynamic full-truckload vehicle routing problem was investigated, featuring a more comprehensive problem formulation compared with most previous studies. Due to the stochastic and dynamic nature of the problem, an agent-based approach was adopted to flexibly model and solve it, aiming to achieve autonomous, flexible, and fast solutions. To assess the performance of the proposed approach under varying environmental conditions (e.g., the system congestion factor and the ratio of orders with multiple trips) and different algorithmic parameter levels (e.g., the latest response time to orders and activating the interchange of trip assignments between vehicles), a detailed scenario analysis was conducted based on a set of designed simulation experiments.

The simulation results indicate that the proposed dynamic approach was capable of providing good and efficient solutions in response to dynamic conditions. Additionally, the activation of the interchange mechanism substantially impacted unit costs and unit profitability. An increase in the multi-trip order ratio led to a decrease in order acceptance ratios, resulting in lower profitability and higher costs. Finding alternative solutions to boost acceptance rates at high multiple-trip order ratios becomes challenging. An increase in profitability is possible if a portion of orders with a large number of trips is permitted. In the case of high congestion, the interchange mechanism allows more alternative orders to be

evaluated simultaneously, significantly maximizing profitability. Furthermore, using longer latest response times and activating the interchange mechanism always had a significantly positive impact on the relevant costs, profitability, ratios of loaded trips over the total distance traveled, and the acceptance ratios of customer orders in all cases. The success of multi-agent-based approaches in solving the complex logistics problem addressed in this study was observed with a more detailed and realistic model. The developed vehicle assignment and routing algorithms maintained reasonable CPU times, which increased at an approximate polynomial rate with the number of vehicles and order density.

In future research, alternative solution methods such as genetic algorithms, particle swarm optimization, and ant colony algorithms can be investigated. Artificial intelligence methods can be incorporated into agents' decision-making algorithms. Additionally, incorporating dynamic events like disruptive events (e.g., traffic jams and vehicle breakdowns), order changes, or cancellations can also be considered.

**Author Contributions:** Methodology, formal analysis, writing—original draft, and writing—review and editing, S.Ç. and R.E.; supervision, R.E. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was supported by The Scientific Projects Unit of Cukurova University (grant number: FDK-2018-11037).

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

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Bouyahyiouy, K.E.; Bellabdaoui, A. A mixed-integer linear programming model for the selective full-truckload multi-depot vehicle routing problem with time windows. *Decis. Sci. Lett.* **2021**, *10*, 471–486. [\[CrossRef\]](#)
2. Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [\[CrossRef\]](#)
3. Melchiori, L.; Nasini, G.; Montagna, J.M.; Corsano, G. Resources synchronization in a full truckload pickup and delivery problem: An exact approach. *Comput. Oper. Res.* **2023**, *151*, 106118. [\[CrossRef\]](#)
4. Bouyahyiouy, K.E.; Bellabdaoui, A. An am-TSPTW transformation and a RTS algorithm for commodity selection and vehicle routing planning in full truckload industry. In Proceedings of the 14th International Conference of Logistics and Supply Chain Management, Eljadida, Morocco, 25–27 May 2022.
5. Macal, C.M.; North, M.J. Tutorial on agent-based modelling and simulation. *J. Simul.* **2010**, *4*, 151–162. [\[CrossRef\]](#)
6. Baykasoglu, A.; Kaplanoglu, V. An application oriented multi-agent based approach to dynamic load/truck planning. *Expert Syst. Appl.* **2015**, *42*, 6008–6025. [\[CrossRef\]](#)
7. Ball, M.O.; Golden, B.; Assad, A.; Bodin, L. Planning for truck fleet size in the presence of a commoncarrier option. *Decis. Sci.* **1983**, *14*, 103–120. [\[CrossRef\]](#)
8. Desrosiers, J.; Laporte, G.; Sauve, M.; Soumis, F.; Taillefer, S. Vehicle routing with full loads. *Comput. Oper. Res.* **1988**, *15*, 219–226. [\[CrossRef\]](#)
9. Arunapuram, S.; Mathur, K.; Solow, D. Vehicle routing and scheduling with full truckloads. *Transp. Sci.* **2003**, *37*, 170–182. [\[CrossRef\]](#)
10. Gronalt, M.; Hartl, R.F.; Reimann, M. New savings based algorithms for time constrained pickup and delivery of full truckloads. *Eur. J. Oper. Res.* **2003**, *151*, 520–535. [\[CrossRef\]](#)
11. Imai, A.; Nishimura, E.; Current, J. A Lagrangian relaxation-based heuristic for the vehicle routing with full container load. *Eur. J. Oper. Res.* **2007**, *176*, 87–105. [\[CrossRef\]](#)
12. Caris, A.; Janssens, G. A local search heuristic for the pre- and end-haulage of intermodal container terminals. *Comput. Oper. Res.* **2009**, *36*, 2763–2772. [\[CrossRef\]](#)
13. Currie, R.H.; Salhi, S. A tabu search heuristic for a full-load, multi-terminal, vehicle scheduling problem with backhauling and time windows. *J. Math. Model Algorithm* **2004**, *3*, 225–243. [\[CrossRef\]](#)
14. Zhang, R.; Yun, W.Y.; Moon, I. A reactive tabu search algorithm for the multi-depot container truck transportation problem. *Transp. Res. Part E* **2009**, *45*, 904–914. [\[CrossRef\]](#)
15. Zhang, R.; Yun, W.Y.; Kopfer, H. Heuristic-based truck scheduling for inland container transportation. *OR Spectr.* **2010**, *32*, 787–808. [\[CrossRef\]](#)
16. Li, J.; Lu, W. Full truckload vehicle routing problem with profits. *J. Traffic Transp. Eng.* **2014**, *1*, 146–152.
17. Liu, R.; Fung, R.Y.K.; Chen, F.; Liu, X. Two-phase heuristical algorithms for full truckloads multi-depot capacitated vehicle routing problem in carrier collaboration. *Comput. Oper. Res.* **2010**, *37*, 950–959. [\[CrossRef\]](#)

18. Nossack, J.; Pesch, E. A truck scheduling problem arising in intermodal container transportation. *Eur. J. Oper. Res.* **2013**, *230*, 666–680. [[CrossRef](#)]
19. Bai, R.; Xue, N.; Chen, J.; Roberts, G.W. A set-covering model for a bidirectional multi-shift full truckload vehicle routing problem. *Transp. Res. Part B* **2015**, *79*, 134–148. [[CrossRef](#)]
20. Braekers, K.; Caris, A.; Janssens, G.K. Integrated planning of loaded and empty container movements. *OR Spectr.* **2013**, *35*, 457–478. [[CrossRef](#)]
21. Maneengam, A.; Udomsakdigool, L. Solving the Bidirectional Multi-Period Full Truckload Vehicle Routing Problem with Time Windows and Split Delivery for Bulk Transportation Using a Covering Model. In Proceedings of the 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bangkok, Thailand, 16–19 December 2018.
22. Maneengam, A.; Udomsakdigool, L. Solving the collaborative bidirectional multi-period vehicle routing problems under a profit-sharing agreement using a covering model. *Int. J. Ind. Eng. Comput.* **2020**, *11*, 185–200. [[CrossRef](#)]
23. Grimault, A.; Bostel, N.; Lehuédéa, F. An adaptive large neighborhood search for the full truckload pickup and delivery problem with resource synchronization. *Comput. Oper. Res.* **2017**, *88*, 1–14. [[CrossRef](#)]
24. Wang, A.; Ferro, N.; Majewski, R.; Gounaris, C.E. Mixed-integer linear optimization for full truckload pickup and delivery. *Optim. Lett.* **2021**, *15*, 1–17. [[CrossRef](#)]
25. Bouyahyious, K.E.; Bellabdaoui, A. A new crossover to solve the full truckload vehicle routing problem using genetic algorithm. In Proceedings of the 3rd International Conference on Logistics Operations Management (GOL), Fez, Morocco, 23–25 May 2016.
26. Xue, N.; Bai, R.; Qub, R.; Aickelinc, U. A hybrid pricing and cutting approach for the multi-shift full truckload vehicle routing problem. *Eur. J. Oper. Res.* **2021**, *292*, 500–514. [[CrossRef](#)]
27. Bouyahyious, K.E.; Bellabdaoui, A. The Selective Full Truckload Multi-depot Vehicle Routing Problem with Time Windows Formulation and a Genetic Algorithm. *Int. J. Supply Oper. Manag.* **2022**, *9*, 299–320.
28. Tsolaki, K.; Vafeiadis, T.; Nizamis, A.; Ioannidis, A.D. Utilizing machine learning on freight transportation and logistics applications: A review. *ICT Express* **2023**, *9*, 284–295. [[CrossRef](#)]
29. Bai, R.; Chen, X.; Chen, Z.L.; Cui, T.; Gong, S.; Hea, W.; Jiang, X.; Jina, H.; Jin, J.; Kendall, G.; et al. Analytics and machine learning in vehicle routing research. *Int. J. Prod. Res.* **2023**, *61*, 4–30. [[CrossRef](#)]
30. Martin, S.; Ouelhadj, D.; Beullens, P.; Ozcan, E.; Juan, A.A.; Burke, E.K. A multi-agent based cooperative approach to scheduling and routing. *Eur. J. Oper. Res.* **2016**, *254*, 169–178. [[CrossRef](#)]
31. Mes, M.; Heijden, M.; Harten, A. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *Eur. J. Oper. Res.* **2007**, *181*, 59–75. [[CrossRef](#)]
32. Mes, M.; Heijden, M.; Schuur, P. Interaction between intelligent agent strategies for real-time transportation planning. *Cent. Eur. J. Oper. Res.* **2013**, *21*, 337–358. [[CrossRef](#)]
33. Karami, F.; Vancroonenburg, W.; Berghe, G.W. A periodic optimization approach to dynamic pickup and delivery problems with time windows. *J. Sched.* **2020**, *23*, 711–731. [[CrossRef](#)]
34. Lu, X.; Tang, K.; Menzel, S.; Yao, X. Dynamic optimization in fast-changing environments via offline evolutionary search. *IEEE Trans. Evol. Comput.* **2022**, *26*, 431–445. [[CrossRef](#)]

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