energies

MDPI

Article

Optimizing Multi Cross-Docking Systems with a Multi-Objective Green Location Routing Problem Considering Carbon Emission and Energy Consumption

Ieva Meidute-Kavaliauskiene [1,*], Nihal Sütütemiz [2], Figen Yıldırım [3], Shahryar Ghorbani [4] and Renata Činčikaitė [1]

1 Department of Business Technologies and Entrepreneurship, Vilnius Gediminas Technical University, Sauletekio al. 11, 10223 Vilnius, Lithuania; renata.cincikaite@vilniustech.lt
2 Department of Management Information Systems, University of Sakarya, Sakarya 54050, Turkey; nihals@sakarya.edu.tr
3 Department of International Trade, Istanbul Commerce University, Istambul 34445, Turkey; figen@ticaret.edu.tr
4 Department of Production Management, University of Sakarya, Sakarya 54050, Turkey; mg.shahryar@gmail.com
* Correspondence: ieva.meidute-kavaliaukiene@vilniustech.lt

Abstract: Cross-docking is an excellent way to reduce the space required to store goods, inventory management costs, and customer order delivery time. This paper focuses on cost optimization, scheduling incoming and outgoing trucks, and green supply chains with multiple cross-docking. The three objectives are minimizing total operating costs, truck transportation sequences, and carbon emissions within the supply chain. Since the linear programming model is an integer of zero and one and belongs to NP-hard problems, its solution time increases sharply with increasing dimensions. Therefore, the non-dominated sorting genetic algorithm-II (NSGA-II) and the multi-objective particle swarm optimization (MOPSO) were used to find near-optimal solutions to the problem. Then, these algorithms were compared with criteria such as execution time and distance from the ideal point, and the superior algorithm in each criterion was identified.

Keywords: non-dominated sorting genetic algorithm-II (NSGA-II); multi-objective particle swarm optimization (MOPSO); cross-docking

Citation: Meidute-Kavaliauskiene, I.; Sütütemiz, N.; Yıldırım, F.; Ghorbani, S.; Činčikaitė, R. Optimizing Multi Cross-Docking Systems with a Multi-Objective Green Location Routing Problem Considering Carbon Emission and Energy Consumption. *Energies* **2022**, *15*, 1530. https://doi.org/10.3390/en15041530

Academic Editors: Bernard Ziębicki and Edyta Bielińska-Dusza

Received: 21 December 2021
Accepted: 16 February 2022
Published: 18 February 2022

1. Introduction

Logistics strategy is an essential advantage for supply chain management operations that requires centralized planning and adjustment of procedures to reduce costs and increase customer satisfaction [1,2]. Classic supply chain management for transporting goods from suppliers to the distribution centre requires warehousing for storage [3] and then packaging and shipping to customers [4], according to their demand [5]. This form of supply chain management has high inventory maintenance costs and human resources costs [6,7]. In other words, there are four stages of receiving, storage, order selection, and sending in traditional warehouses [8]. Still, in the cross-dock, the storage stage, one of the main obstacles of inventory systems, is removed [9,10]. Therefore, it reduces inventory and increases customer satisfaction [11]. Cross-dock is a logistic method that distributes products directly from a supplier or factory [12]. The cross-dock method ensures timely transportation and better coordination between supply and demand [13]. In recent years, one of the issues facing the supply chain was clean production and environmental protection [5]. One of the critical issues of clean output is using supply chains that pay attention to environmental issues in all supply chain processes [14,15].

In this research, we investigate the operational cross-docking decisions that concern the optimization of short-term decisions such as CO_2 emission and energy consumption that are directly related to the transit of products from inbound trucks to outbound trucks. We focus on the optimization of the truck to door assignment problem, which is one of the key issues in cross-docking. It attempts to find the optimal assignment for each incoming inbound or outbound truck to the appropriate inbound or outbound dock according to the cross-dock characteristics. When the problem of truck-to-door assigning is mentioned, two different questions should be answered: at which door and in which order the trucks should be docked. The first question was answered by many researchers, but the second question receives less attention and this research tries to investigate the answer to this question. The cross-docking system targets to reduce the cost of inventory holding and minimize the delivery time from suppliers to retailers in the supply chain. However, this paper addresses the cross-dock selection and optimization problem, which was first considered with the vehicle routing problem (VRP) by Maknoon and Laporte (2017), to make optimal decisions on route construction and freight consolidation cross-docks. In other words, selecting cross-docks as freight consolidation points leads to gaining economies of scale. The products are processed and transported to the customers by at least one cross-dock in the CDS problem. It has a vast application in production and retail companies and logistic service providers that handle various shipments on large networks.

Dondo and Cerda (2014) took a different approach to applying the time window so that a variable was used to determine the time window interval [16]. If there is a time window, this variable will be greater than zero for error; otherwise, it will be zero [13]. Mohtashami et al. (2015), in a study to provide and optimize a multi-objective mathematical model in the cross-dock and the entire supply chain, assumed that the supplier can also send the product directly to the customer and send it to the cross-dock [15]. On the other hand, multiple objectives, such as minimizing the total transport time of trucks, total cost, number of shipments, and model solving with NSGA-II and MOPSO with unsuccessful sorting and comparing the results of the above two algorithms, were included in their research. Mohtashami (2015) presented a new dynamic genetic algorithm based on the vehicle scheduling method in the cross-dock system to reduce the operation time. This algorithm assumed the existence of a temporary warehouse for receiving and sending goods, frequent entry and exit of trucks, and two types of chromosomes determined for incoming and outgoing trucks, providing a shorter solution for the operating time [17]. Ponboon et al., 2016, evaluated the cost structure and the impact of parameters on location issues and time windows in the distribution network. Nine scenarios of the Osaka distribution network of freight cargo with different warehouse locations were tested with two criteria, warehouse size and vehicle size. Therefore, using a large warehouse with a large vehicle will reduce costs. Warehouse features, vehicles, and shipping information were also discussed in detail [18]. Gomes et al. (2018) conducted bibliometric research in warehouse management systems from 2006 to 2016 [19].

Birim (2016) investigated the vehicle routing problem with cross-dock, considering heterogeneous vehicles with variable capacity and seeking to minimize the objective function of the total cost of transportation. The problem was solved with a simulated annealing algorithm, and the best answer is compared [20]. Yin and Chuang (2016) emphasized that we must focus on the green supply chain for sustainable development in the supply chain and pay attention to the friendly environment and business values. This paper deals with the minimum cost of green vehicle routing in transporting final products from suppliers to customers through a cross-dock with limited carbon dioxide emissions. Load management for long distances with the lowest cost and carbon dioxide emissions determine high fuel efficiency with the tabu search algorithm [21]. Wisittipanich and Hengmeechai (2017) said that one of the fundamental operational management problems is scheduling trucks in assigning them to cross-dock doors and arranging all domestic and foreign trucks for loading and unloading. This paper presents a mathematical model of integer hybrid programming for allocating and setting trucks in a multi-door cross-docking system. The purpose of this model is to minimize the total operation time. Then, a modified particle swarm algorithm is proposed and optimized with special unique designs, coding, and decoding to solve the problem of truck scheduling in a cross-docking system [16]. Zulaga et al. (2017) showed that this type of dock has results such as reducing cost time and improving management information in reverse processes. Sensitivity analysis of this model helps companies enhance their competitive position by providing flexibility concerning products, reducing the likelihood of product returns from the secondary market, and combining product returns and cross-dock costs relative to traditional warehousing [14]. Mohtashami and Najafabadi (2014) examined the reduction of operating time in the entire supply chain in their article. In the proposed model, the incoming trucks move directly towards the customers or the cross-dock in the first stage after loading the products from the suppliers. The products are unloaded in the cross-dock and loaded in the output trucks in the second stage. Then, the products are transferred to the customers [22].

Table 1 summarize the literature regarding optimizing multi cross-docking systems (MCDS) and compare the existing problems with the problem proposed in this paper.

Table 1. Summary of literature regarding MCDS and its different variants.

References	**Multiple Cross-Docks**	**Type of Vehicle**		**Time Windows**	**Capacity in Crossdocks**		**Multiple Objectives**	**Solution Method**		
		Homogeneous	**Heterogeneous**		**Limited**	**Unlimited**		**Exact**	**Heuristic**	**Metaheuristic**
[23]	*		*			*			*	
[24]	*	*			*			*	*	
[25]	*		*			*		*		
[26]		*		*		*			*	
[27]	*	*			*			*		*
[28]	*		*	*		*		*		
[29]		*				*			*	
[30]		*				*	*		*	
[31]		*				*		*		
[32]	*		*	*		*		*		
[9]	*		*	*	*			*		*
[33]	*		*	*		*	*	*		
[34]			*	*	*					*
[11]	*	*		*	*		*			*
[35]		*		*	*		*			*
Current research	*		*	*		*	*	*		*

Since cross-docks and related transportations have significant roles in increasing the efficiency of large-scale supply chain distribution networks, considering the other real-world essential factors such as reliability and pollution is attractive in designing or redesigning this network. Reliability maximization leads to customer satisfaction and earning a lot of market shares, whereas pollution minimization creates eco-friendly industries.

The research conducted was generally related to cross-dock and focused on optimizing the timing of trucks and the cost of transportation inside the dock. In this research, by considering several cross-docks and concentrating on green management with a norm time window approach for customers, issues related to out-of-cross-dock and the entire supply chain operations are also addressed. This study focuses on optimizing the cost of transporting trucks by locating the cross-dock, the sequence of transporting trucks, and the amount of carbon dioxide emissions in the environment simultaneously throughout the supply chain. In summary, the contributions of this research are described as follows.

- Developing a novel mathematical model to integrate CDS results in a comprehensive problem with a great application in industries;
- Making the problem closer to the real-world conditions by considering the effects of GHG emission in the transportation process and calculating energy consumption dependent on traffic time;
- Investigating the total cost minimization related to transportation, minimizing truck transportation sequences and carbon emissions through cross-docking simultaneously;
- Designing high-quality algorithms, including NSGA-II and MOPSO, efficiently to solve the large-sized problem.

2. Problem Statements

In this paper, three objective functions, including minimizing the total cost of operation, truck transport sequence, and carbon dioxide emissions, are examined. The proposed model also examines the relationship between (cross-dock–suppliers), (cross-dock–customers), (suppliers–customers), (suppliers–suppliers), (customers–customers), and (cross-dock–cross-dock) in terms of fuzzy scheduling with a norm time window approach for truck transport and reducing greenhouse gas emissions. The assumptions of the model are:

- All incoming and outgoing trucks are available in zero time;
- Homogeneous vehicles with different capacities;
- Considering the period of customer service of norm type (having the earliest service start time and the latest service start time);
- Existence of several cross-docks so that suppliers choose one of the docks to send the goods considering the minimum cost;
- Ability to connect suppliers, docks, and customers with each other;
- The type and number of products supplied by suppliers, as well as the type and number of customer demand, are clear and constant;
- In the transport sequence, a truck can load products from more than one supplier and unload products from more than one customer;
- One or more suppliers may meet a customer's requirements;
- The type and quantity of products transported by incoming trucks must be equal to the demands of customers.

To better understand and explain the research problem, the flow of operations in the supply chain is drawn in Figure 1. This figure considers several cross-dock distribution and transportation operation flows and several customers and suppliers. Vehicles with different capacities are also used to transport products in the supply chain.

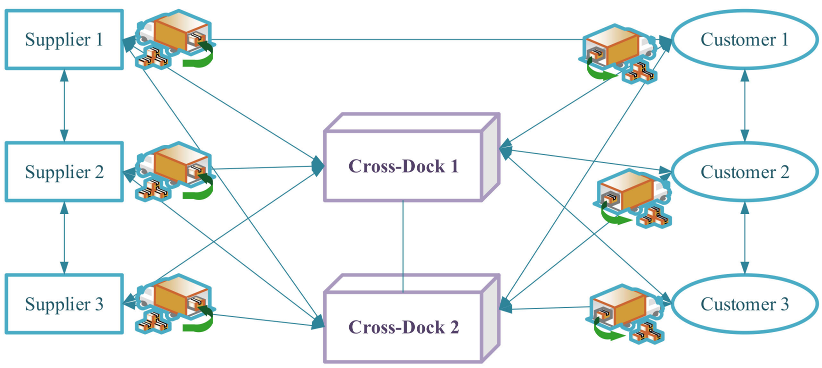

Figure 1. Operation process in the cross-dock.

After loading the products from the suppliers, the incoming trucks move directly to the customers, move to another supplier, move to one of the cross-docks, and unload the products in the cross-dock. They are then loaded onto output trucks and delivered to customers [36]. It should also be noted that a truck can load products from more than one supplier, and a truck can move to more than one customer and unload products between them. Three trucks with different capacities are used to transport the products.

2.1. Mathematical Model

The main variables and important parameters in the model and algorithms are presented in Table 2:

Table 2. Sets, parameters, and decision variables of the model.

Sets	
S	Set of suppliers (1, . . . ,S)
C	Set of customers (1, . . . ,C)
k_{in}	Set of incoming trucks (Receiving) (1, . . . ,K)
k_{out}	Set of outgoing trucks (Sending) (1, . . . ,K′)
G	Set of products type (order) (1, . . . ,G)
Cd	Set of Cross-docks (1, . . . ,Z)
Parameters	
T	Transportation time
$\mathbf{A^{i}_{scG}}$	Time of entry or exit of incoming truck (i) from supplier (s), while loading the order of type (G) and moving towards the customer (c)
$\mathbf{A^{j}_{cdcG}}$	When the incoming truck (j) enters to the customer (c) from the cross-dock (cd) while loading the order of type (G)
$\mathbf{T^{i}_{scG}}$	Transportation time of incoming truck (i) from supplier (s) to customer (c) while loading the order of type (G)
$\mathbf{T^{j}_{cdcG}}$	Transport time of outgoing truck (j) from cross-dock (cd) to customer (c), while loading the order of type (G)
$\mathbf{UL^{i}_{scG}}$	Offloading time of each product type (G) from input truck (i) to customer (c)
$\mathbf{UL^{i}_{scG}}$	Offloading time of each product type (G) from output truck (j) to customer (c)
$\mathbf{D_{cG}}$	Customer order (c) of type (G) goods
$\mathbf{P_{Gs}}$	Production rate of product with type (G)
$\mathbf{w_{G}}$	Weight of product with type (G)
$\mathbf{d_{i}}$	Demand of customer (i)
$[\mathbf{e_i}, \mathbf{E_i}, \mathbf{L_i}, \mathbf{l_i}]$	Time window interval of customer (i)
$\gamma_i = \mathbf{E_i} - \mathbf{A^{i}_{scG}}$	Early arrival time of incoming or outgoing truck in the time window
$\theta_i = \mathbf{A^{i}_{scG}} - \mathbf{L_i}$	Incoming or outgoing truck late arrival time
$PE_{k_{out}}$	Penalty for delay or early arrival of a vehicle exiting the cross-dock (i) for the customer (c)
$PE_{k_{in}}$	Penalty for delay or early arrival of the incoming vehicle from the supplier (i) to the customer (i)
$\mathbf{w_{i}}$	Vehicle waiting time at the location of customer (c)
$\mathbf{o_{i}}$	Cost of reopening the cross-dock (i)
$\mathbf{Q^{i}}$	Capacity of incoming truck (i)
$\mathbf{Q^{j}}$	Capacity of output truck (i)
$\mathbf{d_{scd}}$	Distance between supplier (s) and cross-dock (cd)
$\mathbf{d_{sc}}$	Distance between supplier (s) and customer (c)

Table 2. *Cont.*

Sets	
$\mathbf{d_{cdc}}$	Distance between cross-dock (cd) and customer (c)
$\mathbf{C}_{k_{in}}$	The fuel conversion rate of the unloaded incoming truck to carbon dioxide
$\mathbf{C}^{/}_{k_{in}}$	The difference between the conversion rate of the fuel of an incoming truck with a load of one unit of product or more with the same truck without a load of carbon dioxide
$\mathbf{C}^{/}_{k_{out}}$	The difference between the conversion rate of the fuel of an outgoing truck with a load of one unit of product or more with the same truck without a load of carbon dioxide
$\mathbf{C}_{k_{out}}$	Conversion rate of unloaded truck fuel into CO_2
$\mathbf{g^{i}_{sc}}$	Incoming truck fuel consumption rate for travel between supplier (s) and customer (c) without load
$\mathbf{g^{i}_{scd}}$	The fuel consumption rate of incoming trucks for travel between supplier (s) and cross-dock (cd) without load
$\mathbf{g^{i}_{s_m s_n}}$	Incoming truck fuel consumption rate for travel between the source supplier (s) and the destination supplier without load
$\mathbf{g^{i}_{cd_m cd_n}}$	Incoming truck fuel consumption rate for travel between the cross-dock (cd) of origin and the cross-dock of destination without cargo
$\mathbf{g^{j}_{c_m c_n}}$	Outgoing truck fuel consumption rate for travel between origin and destination customer (c) without load
$\mathbf{g^{j}_{cdc}}$	Outgoing truck fuel consumption rate for travel between cross-dock (cd) and customer (c) without load
$\mathbf{Q}_{k_{in}}$	The difference in the fuel consumption rate of an incoming truck traveling with one product or more with the same unladen truck
$\mathbf{Q}_{k_{out}}$	The difference in the fuel consumption rate of an outgoing truck traveling with one product or more with the same unladen truck
$\mathbf{f_{scdG}}$	The number of product types (G) the truck carries between supplier (s) and cross-dock (cd)
$\mathbf{f_{scG}}$	The number of product types (G) the truck carries between supplier (s) and the customer (c)
$\mathbf{f_{cdcG}}$	The number of product types (G) the truck carries between cross-dock (cd) and customer (c)
$\mathbf{f_{s_m s_n G}}$	The number of product types (G) the truck carries between source supplier and destination supplier
$\mathbf{f_{c_m c_n G}}$	The number of product types (G) the truck carries between source customer and destination customer
$\mathbf{f_{cd_m cd_n G}}$	The number of product types (G) the truck carries between source cross-dock and destination cross-dock
E	Big number
Variables	
$\mathbf{q_{Gs}}$	The number of product types (G) that are loaded from the supplier (s) into the input truck (i)
$\mathbf{q_{Gcd}}$	The number of product types (G) that are loaded from the cross-dock (cd) into the output truck (j)
$\mathbf{q_{Gc}}$	The number of product types (G) that are unloaded from the incoming truck (i) at the customer's location
$\mathbf{q_{Gs_n}}$	The number of product types (G) that are unloaded from the input truck (i) at the destination supplier s_n
$\mathbf{q_{Gcd_n}}$	The number of product types (G) that are unloaded from the output truck (j) at the destination cd_n
$\mathbf{X^{i}_{cdsG}}$	$\begin{cases} 1 & \textit{If incoming truck } \mathrm{i} \textit{ moves from cross} - \textit{dock } \mathrm{cd} \textit{ to supplier } \mathrm{S} \textit{ while loading the order of type } \mathrm{G} \\ 0 & \textit{otherwise} \end{cases}$
$\mathbf{X^{i}_{s_m s_n G}}$	$\begin{cases} 1 & \textit{If incoming truck } \mathrm{i} \textit{ moves from supplier } \mathrm{Sm} \textit{ to destination Supplier } \mathrm{Sn} \textit{ while loading the order of type } \mathrm{G} \\ 0 & \textit{otherwise} \end{cases}$
$\mathbf{X^{i}_{scdG}}$	$\begin{cases} 1 & \textit{If incoming truck } \mathrm{i} \textit{ moves from supplier } \mathrm{S} \textit{ to cross} - \textit{dock } \mathrm{cd} \textit{ while loading order of type } \mathrm{G} \\ 0 & \textit{otherwise} \end{cases}$
$\mathbf{X^{i}_{scG}}$	$\begin{cases} 1 & \textit{If incoming truck } \mathrm{i} \textit{ moves from supplier } \mathrm{S} \textit{ to customer } \mathrm{C} \textit{ while loading the order of type } \mathrm{G} \\ 0 & \textit{otherwise} \end{cases}$

Table 2. *Cont.*

Sets	
$\mathbf{X^{i}_{c_m c_n G}}$	$\begin{cases} 1 & \text{If incoming truck i moves from customer Cm to destination customer Cn while loading the order of type G} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{j}_{cdcG}}$	$\begin{cases} 1 & \text{If outgoing truck j moves from cross-dock cd to customer C while loading the order of type G} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{i}_{ccdG}}$	$\begin{cases} 1 & \text{If incoming truck i moves from customer C to cross-dock cd while offloading type G order} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{j}_{ccdG}}$	$\begin{cases} 1 & \text{If incoming truck j moves from customer C to cross-dock cd while offloading type G order} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{i}_{cd_m cd_n G}}$	$\begin{cases} 1 & \text{If incoming truck i moves from source cross-dock cdm to destination cross-dock cdn while loading type G order} \\ 0 & \text{otherwise} \end{cases}$
F_{cdc}	$\begin{cases} 1 & \text{if customer demand i is met from cross-dock i} \\ 0 & \text{otherwise} \end{cases}$
y_i	$\begin{cases} 1 & \text{if cross-dock i is selected} \\ 0 & \text{otherwise} \end{cases}$
P_{ij}	$\begin{cases} 1 & \text{if input truck i precedes to input truck j} \\ 0 & \text{otherwise} \end{cases}$
$q_{g_1 g_2}$	$\begin{cases} 1 & \text{if input truck } g_1 \text{ precedes to input truck } g_2 \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{i}_{scd}}$	$\begin{cases} 1 & \text{If incoming truck i moves from supplier S to cross-dock cd while offloading type G order} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{i}_{sc}}$	$\begin{cases} 1 & \text{If incoming truck i moves from supplier S to customer C while offloading type G order} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{j}_{cdc}}$	$\begin{cases} 1 & \text{If outgoing truck j moves from cross-dock cd to customer C while offloading type G order} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{i}_{c_m c_n}}$	$\begin{cases} 1 & \text{If incoming truck i moves from source customer Cm to destination customer Cn while offloading type G order} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{i}_{s_m s_n G}}$	$\begin{cases} 1 & \text{If incoming truck i moves from source supplier Sm to destination Supplier Sn while loading type G order} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{i}_{s_m s_n}}$	$\begin{cases} 1 & \text{If incoming truck i moves from source supplier Sm to destination Supplier Sn while offloading type G order} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{X^{i}_{cd_m cd_n}}$	$\begin{cases} 1 & \text{If incoming truck i moves from source cross-dock cdm to destination cross-dock cdn while offloading type G order} \\ 0 & \text{otherwise} \end{cases}$

The first objective function minimizes the total cost of operations within the supply chain (including transportation, fuel, delay, shortage, holding, and opening costs). The second objective function minimizes the truck transport sequence between suppliers, cross-docks, and customers, and the third objective function minimizes CO_2 emissions from trucks throughout the system.

$$
\begin{aligned}
\text{Min } Z_1 \quad = & \Bigg[\Bigg(\sum_{i\in K_{in}=1}^{K} \sum_{i\in s=1}^{S} \sum_{i\in cd=1}^{Z} \sum_{i\in G=1}^{G} (C^{i}_{scdG} + C^{i}_{sGcd}) * X^{i}_{scdG} \Bigg) \\
& + \Bigg(\sum_{i\in K_{in}=1}^{K} \sum_{i\in s=1}^{S} \sum_{i\in c=1}^{C} \sum_{i\in G=1}^{G} C^{i}_{scG} * X^{i}_{scG} \Bigg) \\
& + \Bigg(\sum_{i\in k_{in}=1}^{K} \sum_{s=1}^{S} \sum_{i\in G}^{G} C^{i}_{s_m s_n G} * X^{i}_{s_m s_n G} \Bigg) \\
& + \Bigg(\sum_{i\in K_{in}=1}^{K} \sum_{i\in s=1}^{S} \sum_{i\in cd=1}^{Z} \sum_{i\in G=1}^{G} (C^{i}_{cdsG} * X^{i}_{cdsG} \Bigg) \\
& + \Bigg(\sum_{i\in G=1}^{G} \sum_{i\in c=1}^{C} \sum_{i\in k_{in}}^{K} C^{i}_{c_m c_n G} * X^{i}_{c_m c_n G} \Bigg) \\
& + \Bigg(\sum_{j\in k_{out}=1}^{K} \sum_{i\in cd=1}^{Z} \sum_{i\in c=1}^{C} \sum_{i\in G=1}^{G} (C^{j}_{cdcG} * X^{j}_{cdcG} \Bigg) \\
& \Bigg(\sum_{j\in K_{in}=1}^{K} \sum_{i\in\ i\in cd=1}^{Z} \sum_{i\in G=1}^{G} C^{i}_{cd_m cd_n G} * X^{i}_{cd_m cd_n G} \Bigg) \\
& + \Bigg(\sum_{j\in K_{out}=1}^{K} \sum_{i\in\ c=1}^{C} \sum_{i\in G=1}^{G} C^{j}_{c_m c_n G} * X^{j}_{c_m c_n G} \Bigg) \\
& + \Bigg(\sum_{j\in K_{out}=1}^{K} \sum_{i\in c=1}^{C} \sum_{i\in cd=1}^{Z} \sum_{i\in G=1}^{G} c^{j}_{ccdG} * X^{j}_{CCdG} \Bigg) \\
& + \Bigg(\sum_{j\in K_{in}=1}^{K} \sum_{i\in c=1}^{C} \sum_{i\in s=1}^{S} \sum_{i\in G=1}^{G} c^{i}_{csG} * X^{i}_{CsG} \Bigg) \\
& + (\sum_{i\in cd=1}^{Z} O_i y_i + \sum_{i\in s=1}^{S} \sum_{i\in I=1}^{I} \sum_{i\in G=1}^{G} F_i * X^{i}_{scdG}) + [\sum_{i\in K_{in}=1}^{K} \sum_{i\in s=1}^{S} \sum_{i\in c=1}^{C} \sum_{i\in G=1}^{G} PE_{k_{in}} * (\gamma_i + \theta_i) * X^{i}_{scG} \Bigg] \\
& + [\sum_{j\in K_{out}=1}^{K} \sum_{i\in I=1}^{I} \sum_{i\in c=1}^{C} \sum_{i\in G=1}^{G} PE_{k_{OUT}} * (\gamma_i + \theta_i) * X^{j}_{cdcG}]
\end{aligned}
\tag{1}
$$

$$
\text{Min } Z_2 = \text{F} = \sum_{i\in s=1}^{S} \sum_{i\in c=1}^{C} \sum_{i\in cd}^{Z} \sum_{i\in k_{in}=1}^{K} \sum_{j\in k_{out}}^{K} \sum_{i\in G=1}^{G} X^{i}_{scG} + X^{i}_{scdG} + X^{i}_{s_m s_n G} + X^{i}_{csG} + X^{i}_{cdsG} + X^{j}_{cdcG} + X^{j}_{c_m c_n G} + X^{j}_{ccdG}
\tag{2}
$$

$$
\begin{aligned}
\text{Min } Z_3 \quad = & \Bigg[[\Bigg(\sum_{i\in s=1}^{S} \sum_{i\in cd=1}^{Z} \sum_{i\in k_{in}=1}^{K} C_{k_{in}} * g^{i}_{scd} * d_{scd} * X^{i}_{scd} \Bigg) \\
& + (\sum_{i\in s=1}^{S} \sum_{i\in s=1}^{S} \sum_{i\in G=1}^{G} \sum_{i\epsilon k_{in}}^{K} C^{/}_{k_{in}} * Q_{k_{in}} * f_{scdG} * W_G * d_{SCdG} * X^{i}_{scdG})] + [(\sum_{i\in k_{in}=1}^{K} \sum_{i\in s=1}^{S} \sum_{i\epsilon c=1}^{C} C_{k_{in}} * g^{i}_{sc} * d_{sc} * X^{i}_{sc}) \\
& + (\sum_{i\in s=1}^{S} \sum_{i\in c=1}^{C} \sum_{i\in G=1}^{G} \sum_{i\epsilon k_{in}}^{K} C^{/}_{k_{in}} * Q_{k_{in}} * f_{scG} * W_G * d_{SCG} * X^{i}_{scG})] + [(\sum_{i\in k_{in}=1}^{K} \sum_{i\in s=1}^{S} C_{k_{in}} * g^{i}_{s_m s_n} * d_{s_m s_n} * X^{i}_{s_m s_n}) \\
& + (\sum_{i\in k_{in}=1}^{K} \sum_{i\in s=1}^{S} \sum_{i\in G=1}^{G} C^{/}_{k_{in}} * Q_{k_{in}} * f_{s_m s_n G} * W_G * d_{s_m s_n G} * X^{i}_{s_m s_n G})] + [(\sum_{i\in k_{in}=1}^{K} \sum_{i\in cd=1}^{Z} C_{k_{in}} * g^{i}_{cd_m cd_n} * d_{cd_m cd_n} * X^{i}_{cd_m cd_n}) \\
& + (\sum_{i\in k_{in}=1}^{K} \sum_{i\in cd=1}^{Z} \sum_{i\in G=1}^{G} C^{/}_{k_{in}} * Q_{k_{in}} * f_{cd_m cd_n G} * W_G * d_{cd_m cd_n G}) * X^{i}_{cd_m cd_n G}] + [(\sum_{i\in cd=1}^{Z} \sum_{i\in c=1}^{C} \sum_{i\in k_{out}=1}^{K} C_{k_{out}} * g^{j}_{cdc} * d_{cdc} * X^{j}_{cdc}) \\
& + (\sum_{i\in cd=1}^{Z} \sum_{i\in c=1}^{C} \sum_{i\in G=1}^{G} \sum_{i\epsilon k_{out}}^{K} C^{/}_{k_{out}} * Q_{k_{out}} * f_{cdcG} * W_G * d_{CdcG} * X^{j}_{cdcG})] + [(\sum_{i\in k_{out}=1}^{K} \sum_{i\in c=1}^{C} C_{k_{out}} * g^{j}_{c_m c_n} * d_{c_m c_n} * X^{j}_{c_m c_n}) \\
& + (\sum_{i\in k_{out}=1}^{K} \sum_{i\in c=1}^{C} \sum_{i\in G=1}^{G} C^{/}_{k_{out}} * Q_{k_{OUT}} * f_{c_m c_n G} * W_G * d_{c_m c_n G} * X^{j}_{c_m c_n G})] \Bigg]
\end{aligned}
\tag{3}
$$

S.t:

$$\sum_{i\in k_{in}=1}^{K}\sum_{i\in c=1}^{C}\sum_{i\in s=1}^{S}\sum_{i\in G=1}^{G} q_{Gs} * X^{i}_{scG} = D_{cG} \tag{4}$$

$$\sum_{i\in cd=1}^{Z}\sum_{j\in k_{out}=1}^{K}\sum_{i\in C=1}^{C}\sum_{i\in G=1}^{G} q_{Gcd} \underset{cdcG}{\overset{j}{*X}} + \sum_{i\in K_{in}=1}^{K}\sum_{i\in s=1}^{S}\sum_{i\in c=1}^{C}\sum_{i\in G=1}^{G} q_{Gs} * X^{i}_{scG} = P_{Gs} \tag{5}$$

$$\sum_{i\in K_{in}=1}^{K}\sum_{i\in s=1}^{S}\sum_{i\in c=1}^{C}\sum_{i\in G=1}^{G} X^{i}_{scG} \geq 1 \tag{6}$$

$$X^{i}_{cdsG} = \sum_{i\in K_{in}=1}^{K}\sum_{i\in s=1}^{S}\sum_{i\in G=1}^{G} x^{i}_{s_m s_n G} + \sum_{i\in K_{in}=1}^{K}\sum_{i\in s=1}^{S}\sum_{i\in cd=1}^{Z}\sum_{i\in G=1}^{G} X^{i}_{scdG} + \sum_{i\in K_{in}=1}^{K}\sum_{i\in s=1}^{S}\sum_{i\in c=1}^{C}\sum_{i\in G=1}^{G} X^{i}_{scG} \tag{7}$$

$$X^{i}_{s_m s_n G} = \sum_{i\in K_{in}=1}^{K}\sum_{i\in s=1}^{S}\sum_{i\in G=1}^{G} x^{i}_{s_n s_m G} + \sum_{i\in K_{in}=1}^{K}\sum_{i\in s=1}^{S}\sum_{i\in cd=1}^{Z}\sum_{i\in G=1}^{G} X^{i}_{s_n cdG} + \sum_{i\in K_{in}=1}^{K}\sum_{i\in s=1}^{S}\sum_{i\in c=1}^{C}\sum_{i\in G=1}^{G} X^{i}_{scG} \tag{8}$$

$$X^{j}_{cdcG} = \sum_{j\in k_{out}=1}^{K}\sum_{i\in c=1}^{C}\sum_{i\in G=1}^{G} X^{j}_{c_m c_n G} + \sum_{i\in k_{out}}^{K}\sum_{i\in c=1}^{C}\sum_{i\in cd=1}^{Z}\sum_{i\in G=1}^{G} X^{j}_{ccdG} \tag{9}$$

$$X^{i}_{c_m c_n G} = \sum_{i\in K_{in}=1}^{K}\sum_{i\in c=1}^{C}\sum_{i\in G=1}^{G} X^{i}_{c_n c_m G} + \sum_{i\in K_{in}=1}^{K}\sum_{i\in c=1}^{C}\sum_{i\in cd=1}^{Z}\sum_{i=G=1}^{G} X^{i}_{ccdG} \tag{10}$$

$$X^{j}_{c_m c_n G} = \sum_{j\in k_{out=1}}^{K}\sum_{i\in c=1}^{C}\sum_{i\in G=1}^{G} X^{j}_{c_n c_m G} + \sum_{j\in k_{out}=1}^{K}\sum_{i\in c=1}^{C}\sum_{i\in cd=1}^{Z}\sum_{i\in G=1}^{G} X^{j}_{ccdG} + \sum_{j\in k_{out}=1}^{K}\sum_{i\in c=1}^{C}\sum_{i\in cd=1}^{Z}\sum_{i\in G=1}^{G} X^{j}_{cd_m cd_n G} \tag{11}$$

$$X^{i}_{scG} = \sum_{i\in K_{in}=1}^{K}\sum_{i\in c=1}^{C}\sum_{i\in G=1}^{G} X^{i}_{c_m c_n G} + \sum_{i\in K_{in}=1}^{K}\sum_{i\in c=1}^{C}\sum_{i\in cd=1}^{Z}\sum_{i\in G=1}^{G} X^{i}_{ccdG} \tag{12}$$

$$\sum_{i\in K_{in}=1}^{K}\sum_{i\in s=1}^{S}\sum_{i\in cd=1}^{Z}\sum_{i\in G}^{G} q_{Gs} * X^{i}_{scdG} = \sum_{i\in K_{in}=1}^{K}\sum_{i\in c=1}^{C}\sum_{i\in cd=1}^{Z}\sum_{i\in G=1}^{G} q_{Gcd} * X^{j}_{cdcG} \tag{13}$$

$$\sum_{i\in s=1}^{S}\sum_{i\in cd=1}^{Z}\sum_{i\in G=1}^{G} X^{i}_{scdG} * q_{Gs} \leq Q^{i} \tag{14}$$

$$\sum_{i\in c=1}^{C}\sum_{i\in cd=1}^{Z}\sum_{i\in G=1}^{G} X^{j}_{cdcG} * q_{Gcd} \leq Q^{j} \tag{15}$$

$$\sum_{i\in s=1}^{S}\sum_{i\in C=1}^{C}\sum_{i\in G=1}^{G} \left(A^{i}_{scG} + T^{i}_{scG} + UL^{i}_{scG} + w_i\right) * X^{i}_{scG} \leq A^{i}_{csG} \tag{16}$$

$$\sum_{i\in cd=1}^{Z}\sum_{i\in C=1}^{C}\sum_{i\in G=1}^{G} \left(A^{j}_{cdcG} + T^{j}_{cdcG} + UL^{j}_{cdcG} + w_i\right) * X^{j}_{cdcG} \leq A^{i}_{csG} \tag{17}$$

$$\sum_{j\in K_{out}=1}^{K}\sum_{i\in c=1}^{C}\sum_{i\in cd=1}^{Z}\sum_{i\in G=1}^{G} X^{j}_{cdcG} + X^{j}_{CCdG} \leq 1 + F_{cdc} \tag{18}$$

$$\sum_{i\in k_{in=1}}^{K}\sum_{i\in s=1}^{S}\sum_{i\in cd=1}^{Z}\sum_{i\in G}^{G} X^{i}_{scdG} \leq y_i \tag{19}$$

$$\gamma_i \geq E_i - A^{i}_{scG} \tag{20}$$

$$\theta_i \geq A^{i}_{scG} - L_i \tag{21}$$

$$\left(X^{i}_{cdsG}, X^{i}_{s_m s_n G}, X^{i}_{scdG}, X^{i}_{scG}, X^{i}_{c_m c_n G}, X^{j}_{cdcG}, X^{j}_{c_m c_n G}, X^{i}_{ccdG}, X^{j}_{ccdG}, X^{j}_{cd_m cd_n G}, y_i\right) \in \{0, 1\}, \tag{22}$$

$$\left(X^{i}_{cds}, X^{i}_{s_m s_n}, X^{i}_{scd}, X^{i}_{sc}, X^{i}_{c_m c_n}, X^{j}_{cdc}, X^{j}_{c_m c_n}, X^{i}_{ccd}, X^{j}_{ccd}, X^{j}_{cd_m cd_n}, F_{cdc}\right) \in \{0,\ 1\} \tag{23}$$

$$I = 1,2,3,\ldots,I,\ c = 1,2,3,\ldots,C\ ,\ G = 1,2,3,\ldots,G\, j = 1,\ 2,\ 3,\ \ldots,\ J \tag{24}$$

$$\text{All variables } \geq 0,\ \forall\, i, s, c \textit{ and } h \tag{25}$$

2.2. Description of Objective Functions and Constraints

The first objective function minimizes the cost of transporting vehicles throughout the system. The second objective function minimizes the transport sequence of trucks throughout the system. The objective is utilized to sequence inbound and outbound trucks to minimize the makespan. The third objective function minimizes the amount of carbon dioxide emitted from vehicles throughout the system. Each route between origin and destination has two sections. In the first part, the truck is unloaded, and in the second part, the truck is loaded.

Constraint (1) ensures that the total number of units of type G order product loaded from the supplier (s) in the truck (i) is precisely equal to the number of units of type (h) order product that the customer (c) needs. Constraint (2) ensures that the total number of order types (h) that are loaded from the supplier (s) in the entry truck (i) and move to the customer (c), as well as the total number of order types (h) that are loaded from the cross-docks (cd) in the output truck (j), are loaded and transferred toward the customer (c) and are exactly equal to the sum of the number of h-type order products which are produced by the supplier (s). Constraint (3) ensures that at least one of the incoming trucks (i) loads the supply (s) of type G product and transports all products to the customer (c) at once. Constraint (4) ensures that if the incoming truck (i) moves towards the supplier (s), it selects three modes to exit the supplier. The first case is that the truck moves towards another supplier, the second one moves towards one of the cross-docks (cd), and the last one moves towards the customer (c). Constraint (5) ensures that if the incoming truck (i) moves from the source supplier (Sm) to the destination supplier (Sn), it selects one of three modes to exit the supplier (Sn). The first case is that the truck is moving towards another supplier, the second one is moving towards one of the cross-docks (cd), and the third one is moving towards the customer (c). Constraint (6) ensures that if the outgoing truck (j) moves towards the customer (c), it selects one of two modes to exit the customer (c). The first case is that the truck moves towards another customer and the second moves towards one of the cross-docks (cd). Constraint (7) ensures that if the incoming truck (i) moves from the source customer (cm) to the destination customer (cn), it selects one of two modes to exit the destination customer (cn). The first case is that the truck moves towards another customer and the second moves towards one of the cross-docks (cd). Constraint (8) is similar to constraint (7), and the difference between these two constraints is that in constraint (8), the outgoing truck (j) is considered in the constraint, and the third case, the outgoing truck crosses from a cross-dock to another cross-dock. Constraint (9) ensures that if the incoming truck (i) moves from the supplier (s) to customer (c), it selects one of two modes to exit the customer (c). The first case is that the truck is moving towards another customer, and the second case is that the truck is moving towards the cross-dock (cd). Constraint (10) ensures that the number of units of product that the incoming truck (i) transports from the supplier (s) to one of the cross-docks (cd) and unloads at the cross-dock is precisely equal to the number of units of products in the outgoing truck (j) loaded for the customer (c). Constraints (11) and (12) relate to the capacity of incoming and outgoing trucks and ensure that trucks do not load products beyond their capacity. Constraint (13) ensures that the total time of arrival of the vehicle to the customer, the time elapsed between the manufacturer and the customer, the time of unloading the cargo at the customer's place, and the waiting time of the vehicle should not exceed the specified time of departure for each customer. Constraint (14) guarantees that the total time of arrival of the vehicle to the customer, the time elapsed between the cross-dock and the customer, the time of unloading the cargo at the customer's place, and the waiting time of the vehicle should not exceed the specified time of departure of the vehicle from the customer's location. Constraint (15) ensures that a customer is assigned to a cross-dock if a truck connects the two. Constraint (16) ensures that inbound truck (i) from the supplier (i) only moves to a cross-dock. Constraints (17) and (18) are soft to calculate the number of early and late trucks to provide customer service in the norm time window.

3. Solution Methodology

In a small example, the designed mathematical model was first solved in GAMZ software by the ϵ-constraint method [37]. Then, with the development of the example, due to the large volume of model variables over time, it was not possible to solve with GAMZ software. Therefore, it was solved using NSGA-II MOPSO meta-heuristic algorithms.

3.1. Genetic Algorithm

Charles Darwin's theory, proposed in 1859, occupied a special place in optimization problems [38]. This theory is based on the best evolution, and it can be considered a starting point for evolutionary calculations. Genetic algorithms are random search algorithms whose idea is derived from nature. Genetic algorithms for classical optimization have successfully solved linear, convex, and similar problems, but genetic algorithms are much more efficient for solving discrete and nonlinear problems [39]. In nature, better generations emerge from a combination of better chromosomes. In the meantime, there are sometimes mutations in the chromosomes that may improve the next generation. Genetic algorithms also use this idea to solve problems [40]. Figure 2 show the schematic procedure of the NSGA-II algorithm, and then the details of the proposed genetic algorithm are described.

Step 1: Initialize a parent population P_k, where $k = 0$ with size N

Step 1.1: Create a representation for each chromosome from Uniform (0,1)
Step 1.2: Return restoration variables from generated chromosomes
Step 1.3: Check the feasibility related to restoration variables
Step 1.4: Call the mathematical model to obtain supply flow variables

Step 2: Generate an offspring population Q_k with size N from a parent population P_k

Step 2.1: Use a binary selection operator based on the sorting procedure
Step 2.2: Use a crossover operator (Alpha vector) with a crossover probability (Cp)
Step 2.3: Use a mutation operator (swap, slide, flip) with a mutation probability (Mp)
Step 2.4: Use an immigration operator with an immigration probability (Ip)
Step 2.5: Check the feasibility of the model

Step 3: Mate parent and offspring populations to create $R_k = P_k \cup Q_k$ with size $2N$

Step 4: Sort R_k to form non-dominated fronts

Step 4.1: Start with $h = 1$ (first front). For each chromosome i, identify the number of solutions that dominate i (ni), and create the set of solutions that i dominates (Si)
Step 4.2: Place each unplaced solution i with $ni = 0$ in the h^{th} front (F_h)
Step 4.3: For each solution i in F_h, find each chromosome j in Si and reduce nj by one
Step 4.4: If all solutions are placed in a front, stop. Otherwise, update $h \leftarrow h+1$ and go to Step 4.2

Step 5: Calculate a crowding distance

Step 5.1: For each objective m, sort all solutions in a particular front h (F_h) in a non-increasing order of the m^{th} objective function value
Step 5.2: Calculate the crowding distance of each chromosome i with respect to objective m (CD_{im})
Step 5.3: Compute the total crowding disatnce of each chromosome i

Step 6: Update a next parent population by selecting N best solutions from $2N$ solutions in R_k to form $P_{(k+1)}$ based on sorted fronts and crowding distance for the last front

Step 7: Update a new iteration ($k \leftarrow k+1$) and go to Step 2. When the stoping criteria is reached (k_{max}), stop the algorithm

Figure 2. A schematic procedure of the NSGA-II algorithm.

3.1.1. Chromosome Structure

Due to the complexity of the problem and the model, the chromosome consists of three components. These three components include the truck's movement between the origins and destinations and the loading modes of various products. First, how to produce the problem chromosome is described, and then for each example of the problem, in three levels: small, medium, and large, according to the number of trucks, manufacturers, cross-docks, customers, determination and capacity of trucks, time loading, unloading and handling at the cross-dock, distances between origins and destinations, and customer demand, the amount of production of producers are produced as a function of uniform distribution.

Table 3 shows the truck movement modes between the sources. Trucks identify products and loading locations before starting from the cross-dock. In Table 3, for example, four primary sources are considered: two suppliers and two cross-docks. According to Table 3, if the truck moves to the supplier or the cross-dock, it is assigned the number one; otherwise, it is given the number zero. The second column of Table 3, which contains the numbers (0, 0, 0, 1), states that the truck moves only towards supplier 1 and loads the products of supplier 1. Briefly in the table below, S1: supplier 1, S2: supplier 2, CD1: cross-dock 1, and CD2: cross-dock 2.

Table 3. Truck motion matrix at the origins.

Source	**S1**	**S2**	**CD1**	**CD2**	**S1S2**	**S1CD1**	**S1CD2**	**S2CD1**	**S2CD2**	**CD1CD2**	**S1S2CD1**	**S1S2CD2**	**S1S2CD1CD2**
CD1	1	1	0	1	1	0	1	0	1	0	1	1	1
S1	0	0	0	0	0	0	0	1	1	1	0	0	0
CD2	0	0	1	0	1	1	0	1	0	0	1	1	1
S2	0	0	0	0	0	1	1	0	0	1	1	1	1

Table 4 show the trucks' destinations after passing through the sources. For example, in this table, four main destinations are considered, including two customers and two cross-docks, and trucks can transport different modes of products to their destinations. According to Table 4, if the truck moves to the customer or the intersection dock, it is assigned the number one; otherwise, it is given the number zero. The second column of Table 4, which contains the numbers (0, 0, 0, 1), states that the truck moves only to customer one and unloads the customer's products. Briefly in Table 4, C1: customer 1 and: C2: customer 2 and CD1: cross-dock 1 and CD2: cross-dock 2.

Table 4. Truck movement matrix at destinations.

Destination	**C1**	**C2**	**CD1**	**CD2**	**C1C2**	**C1CD1**	**C1CD2**	**C2CD1**	**C2CD2**	**C1C2CD1**	**C1C2CD2**	**C1C2CD1CD2**
CD1	1	1	0	0	1	1	1	0	1	1	1	1
C1	0	0	0	0	0	0	0	1	1	0	0	0
CD2	0	0	1	0	1	1	0	1	0	1	1	1
C2	0	0	0	1	0	0	1	0	0	0	0	0

For example, four different types of products are used. These four types of products are delivered to customers by production suppliers and trucks. Table 5 show the different modes of loading products from suppliers in trucks. A truck can load other methods of products in its truck. According to Table 5, if the truck loads one of the product ordering modes, it will be number one; otherwise, it will be zero. The second column of Table 5, which contains the numbers (0, 0, 0, 1), states that the truck only loads product A.

Table 5. Matrix of product ordering modes.

Ordering Mode	A	B	C	D	AB	AC	AD	BC	BD	CD	ABC	ABD	ACD	BCD	ABCD
A	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1
B	0	1	0	0	1	0	0	1	1	0	1	1	0	1	1
C	0	0	1	0	0	1	0	1	0	1	1	0	1	1	1
D	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1

3.1.2. The Structure of the Cross Method

In the cross method, the production of new children is from the elite parents of the current generation. New children should inherit the main characteristics of parents and be more privileged than them. How children are produced depends on the structure of the chromosome. Using the tournament method as a selection mechanism, two chromosomes are selected as parents. The single-point cross is then performed separately for each part of the chromosome using row (horizontal) and columnar (vertical), randomly selected and described as follows. Table 6 show two samples of the parent chromosome to produce new offspring.

Table 6. Two hypothetical parent chromosomes.

First Parent												
1	1	0	1	1	0	1	0	1	0	1	1	1
0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	1	0	1	1	0	1	0	0	1	1	1
0	0	0	0	0	1	1	0	0	1	1	1	1
Second Parent												
0	0	0	1	0	0	0	1	0	0	1	1	1
0	1	0	0	0	0	0	0	1	1	0	0	0
1	0	0	0	1	1	0	1	1	0	1	0	1
0	0	1	0	1	1	1	0	0	1	1	1	1

A—Array cross operator: In this case, we produce a random number in the range of the number of random devices. The generated random number indicates the array location of the cross in the parent chromosomes. Then, we replace the corresponding genes of the parents from the desired array, and new offspring are produced. For example, array 19 is selected at the cross of the array site 19 in the parent chromosomes, and then the corresponding genes of the parents are shifted from the array site 19 times in Table 7, and new offspring are produced.

B—Horizontal cross operator: In this case, we generate a random number in the range of the number of rows in the matrix at random. The generated random number indicates the location of the cross in the parent chromosomes. The selected line then replaces the corresponding genes of the parents, and new offspring are produced. For example, if the second row is selected, the junction of the second row is selected on the parent chromosomes, and then the corresponding parent genes are switched from the second row, as shown in Table 8, and new offspring are produced.

Table 7. Children of the cross-array operator.

The First Child												
0	0	0	1	0	0	0	1	1	0	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0
1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	1	0	0	1	1	0	0	1	1	1	1
The Second Child												
1	1	0	1	1	0	0	1	0	0	1	1	1
0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	1	1	0	1	0	1
0	0	0	0	1	1	1	0	0	1	1	1	1

Table 8. Children of linear (horizontal) cross operator.

The First Child												
0	0	0	1	0	0	0	1	0	0	1	1	1
0	1	0	0	0	0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	1	0	0	1	1	1
0	0	0	0	0	1	1	0	0	1	1	1	1
The Second Child												
1	1	0	1	1	0	1	0	1	0	1	1	1
0	0	0	0	0	0	0	1	1	1	0	0	0
1	0	0	0	1	1	0	1	1	0	1	0	1
0	0	1	0	1	1	1	0	0	1	1	1	1

C—Vertical cross operator: In this case, we generate a random number in the range of matrix columns. The generated random number indicates the location of the cross in the parent chromosomes. The selected column then replaces the corresponding genes of the parents, and new offspring are produced. For example, the fourth column is selected, the intersection is performed from the location of the fourth column in the parent chromosomes, and then the corresponding parent genes are switched from the location of the fourth column according to Table 9, and new offspring are produced.

Table 9. Children of the linear (vertical) cross operator.

The First Child												
1	1	0	1	1	0	1	0	1	0	1	1	1
0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	1	0	1	1	0	1	0	0	1	1	1
0	0	0	0	0	1	1	0	0	1	1	1	1
The Second Child												
0	0	0	1	1	0	1	0	1	0	1	1	1
0	0	0	0	0	0	0	1	1	1	0	0	0
1	0	0	0	1	1	0	1	0	0	1	1	1
0	0	1	0	0	1	1	0	0	1	1	1	1

3.1.3. Structure of the Mutation Method

The mutation operator is mainly used to bring about transformation diversity and prevent divergence in the population. Due to the chromosome structure, the mutation operator, similar to the crossover operator, is performed as one of the array methods: row (horizontal) and columnar (vertical), which are randomly selected by the algorithm.

A—Array mutation operator: In this case, we produce a random number in the range of random arrays. The generated random number indicates the location of the mutation on the parent chromosome. Then, if the number in the array location is zero, it becomes one, and if it is one, it becomes zero. For example, array 19 is selected in parent one and the mutation is performed from array location 19 in the parent chromosomes. Then the corresponding parent gene from array location 19 is replaced by Table 10, and a new chromosome is produced.

Table 10. New chromosome derived from the array mutation operator.

1	1	0	1	1	0	1	0	1	0	1	1	1
0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	1	0	0	1	0	1	0	0	1	1	1
0	0	0	0	0	1	1	0	0	1	1	1	1

B—Horizontal mutation operator: In this case, we generate a random number in the range of the number of rows in the matrix. The generated random number indicates the location of the mutation on the parent chromosome. The corresponding parent genes are then selected from the row location, which has values of zero and one, and we move the values of the two, and a new child is produced. For example, the second row is selected in the first parent and the mutation is performed from the second-row location in the parent chromosome. Then the corresponding parent genes are moved from the second-row location in Table 11, and a new chromosome is produced.

Table 11. New chromosome derived from the horizontal mutation operator.

1	1	0	1	1	0	1	0	1	0	1	1	1
1	1	1	1	1	1	1	0	0	0	1	1	1
0	0	1	0	1	1	0	1	0	0	1	1	1
0	0	0	0	0	1	1	0	0	1	1	1	1

C—Vertical mutation operator: In this case, we generate a random number in the range of the number of columns in the matrix. The generated random number indicates the location of the mutation on the parent chromosome. The corresponding parent genes are then selected from the column location, which has values of zero and one, and we move the values of the two, and a new child is produced. For example, the seventh column is selected in the first parent and the mutation is performed from the seventh-column location in the parent chromosome. Then the corresponding parent genes are moved from the seventh-column location in Table 12, and a new chromosome is produced.

Table 12. New chromosome derived from the vertical mutation operator.

1	1	0	1	1	0	0	0	1	0	1	1	1
0	0	0	0	0	0	1	1	1	1	0	0	0
0	0	1	0	1	1	1	1	0	0	1	1	1
0	0	0	0	0	1	0	0	0	1	1	1	1

3.2. Particle Swarm Optimization Algorithm—PSO

Kennedy and Eberhart (1995) presented the PSO algorithm in two papers for optimization problems whose continuous nature indicates the answers [41]. The population of answers is called a *swarm*. Each answer is like a *bird* in a group of birds and is called a *particle*. It is similar to the chromosome in the genetic algorithm. All particles have a *fitness value* calculated using the fitness function, and the particle fitness function must be optimized. The velocity vector of that particle determines the direction of motion of a particle. The PSO algorithm starts with random solutions (particles) and then seeks the optimal solution by synchronizing the particles in each iteration. If the decision variables, also their particles, are zero and one, each particle's velocity and position vectors in each iteration of the algorithm are calculated according to Equations (26)–(29).

$$\mathrm{V_{it}} = \mathrm{w.V_{it-1}} + \mathrm{c_1.r_1.(pBest_i - x_{it})} + \mathrm{c_2.r_2.(nBest_i - x_{it})} \tag{26}$$

$$-\mathrm{V_{max}} \leq \mathrm{V_{it}} \leq \mathrm{V_{max}} \tag{27}$$

$$\mathrm{s_i} = 1/1 + \mathrm{e}^{\mathrm{V_{it}}} \tag{28}$$

$$\mathrm{x_{it}} = \begin{cases} 1 \ \rho \leq \mathrm{s_i} \\ 0 \ \text{otherwise} \end{cases} \tag{29}$$

According to the relation of 19, new velocity vectors of each particle based on the previous velocity of the particle itself ($\mathrm{V_{it-1}}$), the best position the particle has ever reached ($\mathrm{pBest_i}$) and the position of the best particle in the neighborhood of the particles obtained so far ($\mathrm{nBest_i}$), are calculated. If the neighborhood of each particle contains all the particles in the group, then $\mathrm{nBest_i}$ indicates the position of the best particle in the group, denoted by gBest. r_1 and r_2 are two random numbers with a uniform distribution between [0, 1] generated independently of each other. c_1 and c_2 are referred to as learning coefficients, they control the effect of pBest and nBest during the search process. w represents the weight coefficient of inertia. The value limits the particle velocity vector ($\mathrm{V_{max}}$). $\mathrm{V_{max}}$ is a constraint that controls the global search capability of a particle group. Using the relation of 20 vectors, the velocity of each particle is converted to the change probability vector. In the above relation, s_i indicates the probability that x_{it} is equal to 1. Then, using the relation of 22 vectors, the position of each particle is updated. In the above relation, ρ is random with a uniform distribution between zero and one. The pseudo-code of group PSO is shown in Table 13.

Table 13. PSO pseudo-code.

```
For each particle
Initialize particle
End For
Do
For each particle
Calculate fitness value of the particle fp
/*updating particle's best fitness value so far)*/
If fp is better than pBest
set current value as the new pBest
End For
/*updating population's best fitness value so far)*/
Set gBest to the best fitness value of all particles
For each particle
Calculate particle velocity according to equation
Update particle position according to equation
End For While maximum iterations OR minimum error criteria is not attained
```

3.3. Validation of the Designed Model

Although many articles studied the issue of truck scheduling and transportation time minimization at the intersection dock, the issue of cost minimization (transportation, loading, unloading, and relocation) throughout the supply chain, considering the relationships and assumptions, was not previously studied. This article represents a new beginning for future work in this field. Therefore, since it is impossible to compare the results with other existing articles, a numerical example is presented in the next section, and the answers to the problem are compared using genetic algorithms and particle swarm.

Method of Generating Random Problems

Sample production problems are randomly generated in small, medium, and large groups. From each dimension of 7 sample problems, 21 sample problems are described according to Table 14 of the production. Then, the computational results will be presented, and finally, by repeating the algorithm implementations and using the uniform distribution function, the problem parameters are generated, and new numbers are generated in each sample. Additionally, the sensitivity analysis of the results is analyzed using the number of trucks, and in the end, the two algorithms are compared using different performance criteria. The solution of the model was obtained using MATLAB (R2019a) software. The problem was that the trucks would move from suppliers to customers or cross-dock and unload the loaded products after loading the products. To better understand how the two algorithms work, the problem-solving details of Example 6 are shown in a small size, which is almost the most straightforward example problem.

Table 14. Fixed parameters of the problems.

Parameters	**Problem 1**							**Problem 2**							**Problem 3**						
	1	**2**	**3**	**4**	**5**	**6**	**7**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number of incoming trucks	2	4	4	3	4	3	3	3	4	2	4	4	3	3	12	11	13	14	11	12	11
Number of outgoing trucks	2	4	3	4	4	3	2	2	3	3	2	3	2	2	11	12	11	12	11	10	12
Number of suppliers	4	3	4	4	5	3	2	10	9	9	8	7	8	11	18	17	17	19	18	16	20
Number of cross-dock	4	3	3	7	4	3	3	8	9	9	7	8	6	6	12	10	11	12	10	10	14
Product types	10	6	8	10	9	10	11	12	12	14	12	11	13	15	20	16	18	20	20	20	25
Number of customer	5	3	4	8	4	3	3	11	12	10	9	10	11	15	22	20	23	21	20	18	20

Several parameter values are generated using the uniform distribution function in this problem. Parameters that are generated randomly include the capacity of the trucks, the amount of customer demand, the amount of production of the manufacturers, the number of products, the fuel consumption rate, and the distance traveled by the trucks between the origins and the destinations. Sample parameter values are created and given in the relevant tables for each problem. Table 15 list the minimum and maximum values that the parameters can be assigned.

Table 15. Random parameters of the problem.

Customer demand	U(0,30)	U(0,30)	U(0,30)
Product supply rate	U(0,30)	U(0,30)	U(0,30)
Product weight	U(0,10)	U(0,10)	U(0,10)
Incoming truck capacity	U(0,10) × 1000	U(0,10) × 1000	U(0,10) × 1000
Outgoing trucks capacity	U(0,10) × 1000	U(0,10) × 1000	U(0,10) × 1000
Origin between destination distance	U(1,100)	U(1,100)	U(1,100)
Number of product type G	U(1,10)	U(1,10)	U(1,10)
Fuel consumption rate	U(1,20)	U(1,20)	U(1,20)

Two meta-heuristic algorithms for solving multi-objective problems, including the NSGA-II faulty sorting genetic algorithm and the MOPSO multi-objective particle swarm algorithm, were used to solve the proposed model. The Taguchi method was used to adjust the parameters of the algorithms. The advantage of the Taguchi method over other experimental design methods is that in addition to cost, obtaining optimal levels of parameters can be achieved in less time. One of the most important steps in this method is to select an orthogonal array that estimates the effects of factors on the mean response and variation. In this paper, the most suitable three-level test design is identified, and according to the Taguchi standard orthogonal arrays, the L9 array is selected as the appropriate test design to parameterize the proposed algorithms. A statistical measure of performance is considered the S/N ($signal\ to\ noise$) ratio to adjust the optimal parameters, which includes the average and changes. This ratio is desirable. The considered response variable is the average of the four standard indices MID (mean ideal distance), MD (maximum spread or diversity), spacing, and NPS (number of pareto solution) for multi-objective meta-heuristic algorithms. Since this response variable is less, its corresponding S/N ratio is considered as Equation (30). The proposed meta-heuristic algorithms were implemented for each Taguchi experiment, and then the S/N ratios were calculated using Minitab 18 software. The optimal values of the parameters of each algorithm are shown in Table 16.

$$S/NRatio = -10\log(\frac{sum(y^2)}{n}) \tag{30}$$

Table 16. Optimal values of parameters in algorithms.

Algorithm	Parameters	Parameter Domin	Amounts
NSGA-II	Iteration	100–300	300
	Population size	50–100	100
	Intersection rate	0.6–0.8	0.8
	Mutation rate	0.1–0.2	0.2
MOPSO	Iteration	100–300	300
	Population size	50–100	100
	Cognitive constants, C1	1–3	3
	Social constant, C2	1–2	2

4. Results

Sample problem 6 was solved at the small size level with the fixed parameters listed in Table 4 and the random parameters listed in Table 6 with the four algorithms NSGA-II and MOPSO. Based on this, the sequence of truck transportation operations in sample case 6, for all four algorithms, is presented as the output analysis of the variables according to the following tables. Table 17 show the transport sequences of the first truck in sample problem 6 by solving the NSGA-II algorithm, and Table 18 show the transport sequences of the first truck in sample problem 6 using the MOPSO solution method.

Table 17. Truck transport sequence 1 with NSGA-II algorithm for sample problem 1.

Origin	Destination	Product Type	Number of Products	Shipping Sequence
Supplier 1	Cross-dock 1	E	7	1
Supplier 2	Cross-dock 1	F	15	
Supplier 3	Cross-dock 3	G	6	2
Supplier 3	Customer 1	D	14	3
Supplier 3	Customer 2	D	14	
Supplier 2	Customer 3	D	13	
Supplier 1	Customer 3	E	17	
Supplier 2	Customer 1	F	8	
Supplier 2	Customer 2	F	16	
Supplier 2	Customer 3	F	18	

Table 17. *Cont.*

Origin	Destination	Product Type	Number of Products	Shipping Sequence
Supplier 2	Customer 2	G	7	
Supplier 3	Supplier 1	E	7	
Supplier 2	Supplier 1	F	15	
Supplier 3	Supplier 1	F	16	4
Cross-dock 3	Cross-dock 1	D	14	5
Cross-dock 2	Cross-dock 1	F	16	
Customer 3	Customer 1	D	14	
Customer 3	Customer 2	E	8	
Customer 3	Customer 2	F	18	
Customer 3	Customer 2	G	13	6
Cross-dock 1	Customer 1	B	9	
Cross-dock 1	Customer 3	D	14	
Cross-dock 1	Customer 1	G	10	
Cross-dock 1	Customer 2	G	7	
Cross-dock 2	Customer 3	C	11	7
Cross-dock 3	Customer 1	E	17	

Table 18. Truck transport sequence 1 with MOPSO algorithm for sample problem 1 (small level).

Origin	Destination	Product Type	Number of Products	Shipping Sequence
Supplier 2	Cross-dock 2	D	8	
Supplier 2	Cross-dock 2	E	8	
Supplier 2	Cross-dock 2	H	11	1
Supplier 2	Customer 1	D	14	
Supplier 2	Customer 2	D	13	
Supplier 2	Customer 1	E	17	
Supplier 2	Customer 2	E	4	2
Supplier 2	Customer 3	E	8	
Supplier 1	Customer 3	D	14	
Supplier 2	Supplier 1	D	8	
Supplier 2	Supplier 1	H	11	3
Cross-dock 2	Cross-dock 1	D	12	4
Customer 3	Customer 2	E	8	
Customer 1	Customer 2	H	4	5
Cross-dock 2	Customer 3	C	17	
Cross-dock 2	Customer 1	E	17	
Cross-dock 2	Customer 2	E	4	
Cross-dock 2	Customer 3	G	7	
Cross-dock 1	Customer 3	E	17	6
Cross-dock 3	Customer 2	E	8	

4.1. Comparison of NSGA-II and MOPSO Algorithms

Tables 19 and 20 show the values related to the criteria of several answers, time to solve problems, distance from the ideal point, distance, and maximum coverage to compare NSGA-II and MOPSO algorithms.

Execution time—One of the crucial criteria for measuring the performance quality of an algorithm is its execution time, which in some articles is also referred to as execution speed. This criterion becomes essential when the dimensions and complexity of the problem increase. The execution time of these four methods in the three levels of small, medium, and large sample problems are given in Tables 19 and 20 and Figure 3. In general, the NSGA-II algorithm is more acceptable in terms of average runtime at both medium and large sample problem levels.

Table 19. Criteria values of the NSGA-II algorithm.

Example	Size	NSGA-II				
		NPS	Time (S)	MID	DM	Spacing
1	Small	100	878.22	1.0918	50964	0.8515
2	Small	97	526.19	1.0601	3351	0.9425
3	Small	100	898.23	1.0361	5745.4	0.9863
4	Small	100	1204.84	1.1609	5082.3	0.8294
5	Small	100	911.26	1.0537	5281.4	0.9158
6	Small	99	392.67	1.0077	3667.4	0.9317
7	Small	99	711.97	1.1094	4616	1.0001
	Mean	99.28	798.054	1.0742	4691.4	0.9226
1	Middle	100	2011.51	1.0466	9406.5	0.8803
2	Middle	100	2213.24	1.0494	9415	0.9962
3	Middle	100	3415.81	1.0453	11,761	1.0038
4	Middle	100	4313.86	1.043	11,291	1.0075
5	Middle	100	4201.49	1.0371	12,580	1.0718
6	Middle	100	3919.35	1.0394	13,449	0.9942
7	Middle	99	5077.08	1.0556	12,997	0.963
	Mean	99.85	3593.19	1.045	11,557.07	0.9881
1	Large	100	19,202.23	1.0627	19,654	1.1429
2	Large	99	21,020.54	1.0748	21,012	1.1436
3	Large	100	210,695.14	1.0452	20,654.12	1.1259
4	Large	100	230,458.87	1.0872	23,012.45	1.1248
5	Large	100	26,748.65	1.0925	24,896.87	1.0258
6	Large	99	25,874.96	1.0745	23,968.97	1.0387
7	Large	100	27,987.56	1.0998	25,984.23	1.0587
	Mean	99.714	22,715.28	1.0766	22,624.79	1.0941

Table 20. Criteria values of the MOPSO algorithm.

Example	Size	MOPSO				
		NPS	Time (S)	MID	DM	Spacing
1	Small	100	817.45	1.1166	2708.6	0.8495
2	Small	77	473.64	1.1234	1365.3	0.803
3	Small	99	831.9	1.0548	2972.3	0.9092
4	Small	94	1161.45	1.2043	2897.9	0.7604
5	Small	95	853.09	1.0943	3108.5	0.8189
6	Small	98	381.79	1.0829	2430	0.907
7	Small	95	650.25	1.1492	2324.6	0.7806
	Mean	94	738.51	1.1179	2453.88	0.8326
1	Middle	100	2044.69	1.0468	6090.3	0.9312
2	Middle	92	2320.03	1.0455	7155.5	1.0476
3	Middle	100	3451.65	1.0547	6141	0.8661
4	Middle	98	4364.47	1.049	7536.4	1.0063
5	Middle	98	4113.73	1.0404	7265.8	0.9996
6	Middle	100	3996.17	1.0364	8293.4	0.9286
7	Middle	96	5258.46	1.0625	6351.8	0.8655
	Mean	97.714	3649.88	1.0479	6976.3	0.9493
1	Large	100	19,795.71	1.0877	10395	0.8874
2	Large	100	22,101.36	1.0689	11,256	0.8658
3	Large	100	20,985.87	1.0589	10,365.85	0.8953
4	Large	100	21,895.3	1.0489	12,365.35	0.9587
5	Large	96	27,014.32	1.0845	12,985.36	0.8596
6	Large	100	26,579.89	1.0895	12,645.84	0.9741
7	Large	93	28,012.56	1.0114	13,586.96	0.8254
	Mean	98.42	23,769.27	1.06	11,942.91	0.8951

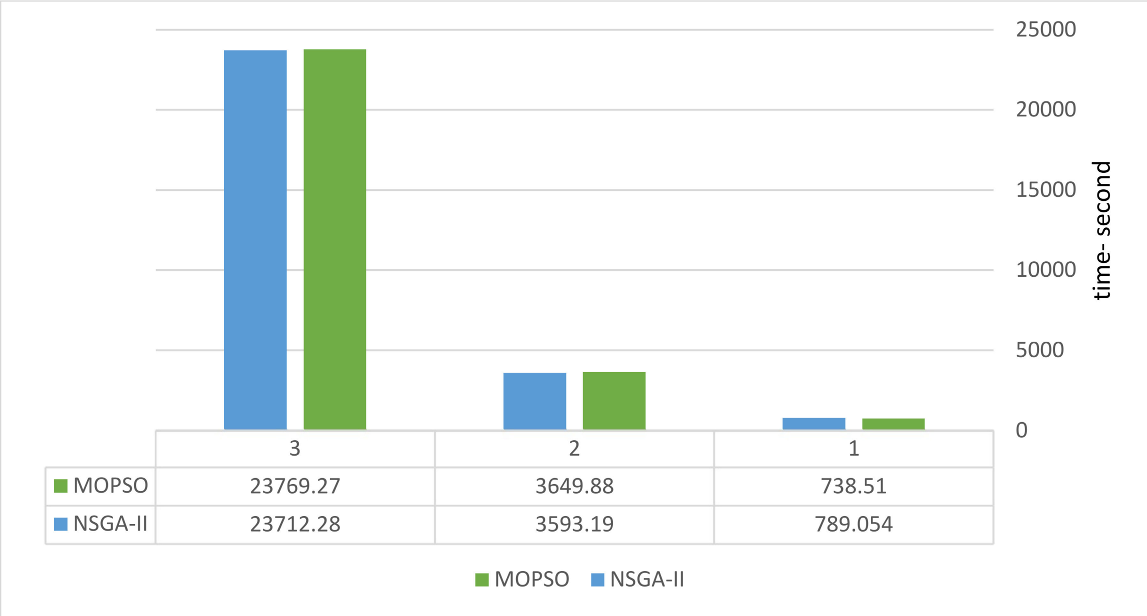

Figure 3. Comparison between the average execution time of NSGA-II and MOPSO.

4.1.1. Distance from the Ideal Point

The MID index is used to calculate the mean distance of Pareto solutions from the origin's so-called ideal point. Considering the MID, it is clear that the lower the value of this index, the higher the efficiency of this algorithm. The ideal point distance criterion, which measures the optimal close Pareto proximity measure, was calculated for 21 sample problems for the NSGA-II and MOPSO algorithms. In these 21 problems, the average values of the NSGA-II algorithm in small and medium problems were less than for the MOPSO algorithm. The DM index shows the variety of Pareto solutions. The higher the DM value is, the better the algorithm performs.

4.1.2. Spacing

The spacing criterion for each sample problem was calculated for the NSGA-II MOPSO algorithms, and the comparison of the mean of these two values for the sample problems was performed at three levels to determine the most desirable algorithm. The spacing criterion in the MOPSO algorithm was smaller than the NSGA-II algorithm and had better performance.

4.1.3. The Most Expansion

The maximum expansion criterion for the 21 sample problems was calculated for the NSGA-II and MOPSO algorithms. The criterion measures the maximum expansion density of Pareto archive solutions, and a simple method was used to calculate it, i.e., without considering the intermediate solutions, only the boundary solutions were used to plot the extent of Pareto archive solutions. In this criterion, the average of the calculations shows that the NSGA-II algorithm offers more comprehensive answers than the MOPSO algorithm.

5. Conclusions and Suggestions

The problem studied in this article was used to provide a model for the optimal sequence of trucks and the cost of operations within the supply chain concerning three objectives (minimizing the cost of transportation, minimizing the sequence of transporting trucks, and minimizing carbon dioxide emissions) [42–45]. The mathematical model approach is complicated due to the high number of variables and limitations related to the number of trucks sending and receiving and the number of products required to solve the problem. When increasing these variables, the dimensions of the problem and the solution time increase exponentially [46,47]. Therefore, two meta-heuristic algorithms NSGA-II and MOPSO, were used to solve the model. The results of the calculations were compared using four criteria. On average, the NSGA-II algorithm is more acceptable in runtime for 21 sample problems at the three small, medium, and large levels. In terms of distance from the ideal point, the NSGA-II algorithm is better. In the spacing criterion, which measures the density of Pareto archive solutions, the MOPSO algorithm scores better than the NSGA-II algorithm. The NSGA-II algorithm offers broader answers than the MOPSO algorithm at the maximum expansion criterion. Because research in this area is in its infancy, there is much opportunity for future research. Several areas in which research can be expanded are listed below.

Generalization of the proposed mathematical model for modelling in the presence of cross-dock systems with many entrances and exit doors.

It was assumed that temporary storage has unlimited capacity, while temporary storage is usually limited in practice. This real-world assumption can be added to the model.

Generalization of the proposed mathematical model by considering the multi-period mode.

Generalization of the proposed mathematical model of the problem by considering uncertain parameters (for example, considering fuzzy demand for customers) given the complex nature of docking systems.

Author Contributions: Conceptualization, I.M.-K. and N.S.; methodology, S.G. and N.S.; software, F.Y.; validation, I.M.-K.; formal analysis, S.G. and F.Y.; investigation, I.M.-K. and R.Č.; resources, S.G. and R.Č.; writing—original draft preparation, S.G. and N.S.; writing—review and editing, I.M.-K.; visualization, S.G. and R.Č.; supervision, I.M.-K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data of this study are available from the authors upon request.

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

References

1. Rijal, A.; Bijvank, M.; de Koster, R. Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors. *Eur. J. Oper. Res.* **2019**, *278*, 752–771. [CrossRef]
2. Serrano, C.; Delorme, X.; Dolgui, A. Scheduling of truck arrivals, truck departures and shop-floor operation in a cross-dock platform, based on trucks loading plans. *Int. J. Prod. Econ.* **2017**, *194*, 102–112. [CrossRef]
3. Seyedi, I.; Hamedi, M.; Tavakkoli-Moghadaam, R. Developing a mathematical model for a multi-door cross-dock scheduling problem with human factors: A modified imperialist competitive algorithm. *J. Ind. Eng. Manag. Stud.* **2021**, *8*, 180–201.
4. Motaghedi-Larijani, A.; Aminnayeri, M. Optimizing the admission time of outbound trucks entering a cross-dock with uniform arrival time by considering a queuing model. *Eng. Optim.* **2017**, *49*, 466–480. [CrossRef]
5. Gelareh, S.; Glover, F.; Guemri, O.; Hanafi, S.; Nduwayo, P.; Todosijević, R. A comparative study of formulations for a cross-dock door assignment problem. *Omega* **2020**, *91*, 102015. [CrossRef]
6. Xi, X.; Changchun, L.; Yuan, W.; Hay, L.L. Two-stage conflict robust optimization models for cross-dock truck scheduling problem under uncertainty. *Transp. Res. Part E Logist. Transp. Rev.* **2020**, *144*, 102123. [CrossRef]
7. Sahebi, I.G.; Mosayebi, A.; Masoomi, B.; Marandi, F. Modeling the enablers for blockchain technology adoption in renewable energy supply chain. *Technol. Soc.* **2022**, 101871. [CrossRef]
8. Nassief, W. Cross-dock Door Assignments: Models, Algorithms and Extensions. Ph.D. Thesis, Concordia University, Montréal, QC, Canada, 2017. Available online: https://spectrum.library.concordia.ca/id/eprint/982540/1/Nassief_PhD_S2017.pdf (accessed on 19 November 2021).
9. Nasiri, M.M.; Rahbari, A.; Werner, F.; Karimi, R. Incorporating supplier selection and order allocation into the vehicle routing and multi-cross-dock scheduling problem. *Int. J. Prod. Res.* **2018**, *56*, 6527–6552. [CrossRef]
10. Chargui, T.; Bekrar, A.; Reghioui, M.; Trentesaux, D. Simulation for Pi-hub cross-docking robustness. In *Service Orientation in Holonic and Multi-Agent Manufacturing*; Springer: Berlin, Germany, 2018; pp. 317–328.
11. Tirkolaee, E.B.; Goli, A.; Faridnia, A.; Soltani, M.; Weber, G.-W. Multi-objective optimization for the reliable pollution-routing problem with cross-dock selection using Pareto-based algorithms. *J. Clean. Prod.* **2020**, *276*, 122927. [CrossRef]

12. Wang, H.; Alidaee, B. The multi-floor cross-dock door assignment problem: Rising challenges for the new trend in logistics industry. *Transp. Res. Part E Logist. Transp. Rev.* **2019**, *132*, 30–47. [CrossRef]
13. Dondo, R.; Cerdá, J. The heterogeneous vehicle routing and truck scheduling problem in a multi-door cross-dock system. *Comput. Chem. Eng.* **2015**, *76*, 42–62. [CrossRef]
14. Zuluaga, J.P.S.; Thiell, M.; Perales, R.C. Reverse cross-docking. *Omega* **2017**, *66*, 48–57. [CrossRef]
15. Mohtashami, A.; Tavana, M.; Santos-Arteaga, F.J.; Fallahian-Najafabadi, A. A novel multi-objective meta-heuristic model for solving cross-docking scheduling problems. *Appl. Soft Comput.* **2015**, *31*, 30–47. [CrossRef]
16. Wisittipanich, W.; Hengmeechai, P. Truck scheduling in multi-door cross docking terminal by modified particle swarm optimization. *Comput. Ind. Eng.* **2017**, *113*, 793–802. [CrossRef]
17. Mohtashami, A. A novel dynamic genetic algorithm-based method for vehicle scheduling in cross docking systems with frequent unloading operation. *Comput. Ind. Eng.* **2015**, *90*, 221–240. [CrossRef]
18. Ponboon, S.; Qureshi, A.G.; Taniguchi, E. Evaluation of cost structure and impact of parameters in location-routing problem with time windows. *Transp. Res. Procedia* **2016**, *12*, 213–226. [CrossRef]
19. Gomes, C.F.S.; Ribeiro, P.C.C.; de Matos Freire, K.A. Bibliometric research in Warehouse Management System from 2006 to 2016. In Proceedings of the World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, FL, USA, 8 July 2018; Volume 22, pp. 200–204.
20. Birim, Ş. Vehicle routing problem with cross docking: A simulated annealing approach. *Procedia Social Behav. Sci.* **2016**, *235*, 149–158. [CrossRef]
21. Yin, P.-Y.; Chuang, Y.-L. Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking. *Appl. Math. Model.* **2016**, *40*, 9302–9315. [CrossRef]
22. Mohtashami, A.; Fallahian-Najafabadi, A. Scheduling trucks transportation in supply chain regarding cross docking using meta-heuristic algorithms. *Ind. Manag. Stud.* **2014**, *11*, 55–84.
23. Sung, C.S.; Song, S.H. Integrated service network design for a cross-docking supply chain network. *J. Oper. Res. Soc.* **2003**, *54*, 1283–1295. [CrossRef]
24. Chen, P.; Guo, Y.; Lim, A.; Rodrigues, B. Multiple crossdocks with inventory and time windows. *Comput. Oper. Res.* **2006**, *33*, 43–63. [CrossRef]
25. Sung, C.S.; Yang, W. An exact algorithm for a cross-docking supply chain network design problem. *J. Oper. Res. Soc.* **2008**, *59*, 119–136. [CrossRef]
26. Wen, M.; Larsen, J.; Clausen, J.; Cordeau, J.-F.; Laporte, G. Vehicle routing with cross-docking. *J. Oper. Res. Soc.* **2009**, *60*, 1708–1718. [CrossRef]
27. Musa, R.; Arnaout, J.-P.; Jung, H. Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. *Comput. Ind. Eng.* **2010**, *59*, 85–92. [CrossRef]
28. Dondo, R.; Méndez, C.A.; Cerdá, J. The multi-echelon vehicle routing problem with cross docking in supply chain management. *Comput. Chem. Eng.* **2011**, *35*, 3002–3024. [CrossRef]
29. Santos, F.A.; Mateus, G.R.; Da Cunha, A.S. The pickup and delivery problem with cross-docking. *Comput. Oper. Res.* **2013**, *40*, 1085–1093. [CrossRef]
30. Morais, V.W.C.; Mateus, G.R.; Noronha, T.F. Iterated local search heuristics for the vehicle routing problem with cross-docking. *Expert Syst. Appl.* **2014**, *41*, 7495–7506. [CrossRef]
31. Vincent, F.Y.; Jewpanya, P.; Redi, A.A.N.P. Open vehicle routing problem with cross-docking. *Comput. Ind. Eng.* **2016**, *94*, 6–17.
32. Wang, J.; Jagannathan, A.K.R.; Zuo, X.; Murray, C.C. Two-layer simulated annealing and tabu search heuristics for a vehicle routing problem with cross docks and split deliveries. *Comput. Ind. Eng.* **2017**, *112*, 84–98. [CrossRef]
33. Rahbari, A.; Nasiri, M.M.; Werner, F.; Musavi, M.; Jolai, F. The vehicle routing and scheduling problem with cross-docking for perishable products under uncertainty: Two robust bi-objective models. *Appl. Math. Model.* **2019**, *70*, 605–625. [CrossRef]
34. Baniamerian, A.; Bashiri, M.; Tavakkoli-Moghaddam, R. Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking. *Appl. Soft Comput.* **2019**, *75*, 441–460. [CrossRef]
35. Zhou, B.; Zong, S. Adaptive memory red deer algorithm for cross-dock truck scheduling with products time window. *Eng. Comput.* **2021**, *38*, 3254–3289. [CrossRef]
36. Liao, T.W. Integrated Outbound Vehicle Routing and Scheduling Problem at a Multi-Door Cross-Dock Terminal. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 5599–5612. [CrossRef]
37. Safari, H.; Ajali, M.; Ghasemiyan Sahebi, I. Determining the strategic position of an educational institution in the organizational life cycle with fuzzy approach (Case Study: Social Sciences Faculty of Khalij Fars University). *Mod. Res. Decis. Mak.* **2016**, *1*, 117–138.
38. Khalili-Damghani, K.; Tavana, M.; Santos-Arteaga, F.J.; Ghanbarzad-Dashti, M. A customized genetic algorithm for solving multi-period cross-dock truck scheduling problems. *Measurement* **2017**, *108*, 101–118. [CrossRef]
39. Corsten, H.; Becker, F.; Salewski, H. Integrating truck and workforce scheduling in a cross-dock: Analysis of different workforce coordination policies. *J. Bus. Econ.* **2020**, *90*, 207–237. [CrossRef]
40. Maknoon, Y.; Laporte, G. Vehicle routing with cross-dock selection. *Comput. Oper. Res.* **2017**, *77*, 254–266. [CrossRef]
41. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.

42. Sahebi, I.; Masoomi, B.; Ghorbani, S.; Uslu, T. Scenario-based designing of closed-loop supply chain with uncertainty in returned products. *Decis. Sci. Lett.* **2019**, *8*, 505–518. [CrossRef]
43. Kusolpuchong, S.; Chusap, K.; Alhawari, O.; Suer, G. A Genetic Algorithm Approach for Multi Objective Cross Dock Scheduling in Supply Chains. *Procedia Manuf.* **2019**, *39*, 1139–1148. [CrossRef]
44. Arab, A.; Sahebi, I.G.; Modarresi, M.; Ajalli, M. A Grey DEMATEL approach for ranking the KSFs of environmental management system implementation (ISO 14001). *Calitatea* **2017**, *18*, 115.
45. Arab, A.; Sahebi, I.G.; Alavi, S.A. Assessing the key success factors of knowledge management adoption in supply chain. *Int. J. Acad. Res. Bus. Soc. Sci.* **2017**, *7*, 2222–6990. [CrossRef]
46. Sayed, S.I.; Contreras, I.; Diaz, J.A.; Luna, D.E. Integrated cross-dock door assignment and truck scheduling with handling times. *Top* **2020**, *28*, 705–727. [CrossRef]
47. Heidari, F.; Zegordi, S.H.; Tavakkoli-Moghaddam, R. Modeling truck scheduling problem at a cross-dock facility through a bi-objective bi-level optimization approach. *J. Intell. Manuf.* **2018**, *29*, 1155–1170. [CrossRef]