

Article

A Hybrid and Self-Adaptive Differential Evolution Algorithm for the Multi-Depot Vehicle Routing Problem in Egg Distribution

Karn Moonsri ¹, Kanchana Sethanan ^{1,*}, Kongkidakhon Worasan ² and Krisanarach Nitisiri ¹

¹ Research Unit on System Modelling for Industry, Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand; hotphai@gmail.com (K.M.); krisni@kku.ac.th (K.N.)

² Faculty of Business Administration and Accountancy, Khon Kaen University, Khon Kaen 40002, Thailand; kongwo@kku.ac.th

* Correspondence: skanch@kku.ac.th

Abstract: This paper presents the Hybrid and Self-Adaptive Differential Evolution algorithms (HSADE) to solve an egg distribution problem in Thailand. We introduce and formalize a model for a multi-product, multi-depot vehicle routing problem with a time window, a heterogeneous fleet and inventory restrictions. The goal of the problem is to minimize the total cost. The multiple products comprise customers' demands with different egg sizes. This paper presents a Mixed Integer Linear Programming (MILP) model, an initial solution-based constructive heuristic, a new self-adaptive mutation strategy, and a neighborhood search structure with the probability to improve DE. The two measurements of criteria are the heuristic performance (HP) compared with the solution obtained by MILP and the relative improvement (RI) of the solution compared with Thailand's current egg distribution practice. The computational results show that the performance of HSADE is better than the current practice, and HSADE can provide on average a 14.13% improvement in total cost. Additionally, our proposed algorithm can be applied to similar agriculture logistics in Thailand and worldwide.

Keywords: egg distribution; multi-depot; multi-product; differential evolution algorithms



Citation: Moonsri, K.; Sethanan, K.; Worasan, K.; Nitisiri, K. A Hybrid and Self-Adaptive Differential Evolution Algorithm for the Multi-Depot Vehicle Routing Problem in Egg Distribution. *Appl. Sci.* **2022**, *12*, 35. <https://doi.org/10.3390/app12010035>

Academic Editor: Giancarlo Mauri

Received: 2 November 2021

Accepted: 17 December 2021

Published: 21 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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

Thailand is the second economy of Southeast Asia and one of the world's leading producers of agriculture products. According to the statistics of Thai exports in 2020, the Thai agricultural export growth decreased by 9.31%. The export of agricultural production consists of rice 22.24%, cassava 21.47%, sugar 29.51% and others 26.78% [1]. Thailand has expanded exports to China by 2.86%, with the top three products being frozen chicken, fresh fruit and frozen shrimp. The frozen chicken had the highest exports. China has encountered a shortage of pork because of African swine fever, which caused the high price of pork in China's market. Chinese people have turned to consume chicken instead of pork. Therefore, China imports more chicken products from Thailand [2]. In the past, the government and private sectors have worked continually to co-develop Thailand's logistics system. In addition, the Thai business sector has increased the awareness and importance of the development of logistics management systems within the organization. In 2020, Thailand increased logistics costs per GDP, with the freight forwarding agency increasing by 6.7% and the warehouse storage by 5.7% [3].

The poultry industry of Thailand is the most promising agri-food segment, especially the egg industry. The logistics of the egg industry can be separated into three parts, which are inbound logistics, internal logistics and outbound logistics (Figure 1). First, the inbound logistics of egg production starts with purchasing chicks from a hatchery and transporting them to pullet-raising farms. Second, the internal logistics of egg production is when chicks

turn into a hen at about 17 weeks old; the hens must be moved to a laying unit (hen houses) for harvesting eggs until the age of 75 weeks. Eggs must be sorted by size at the egg processing center, then sent to the egg packaging process. Once the hens cannot perform as layers (spent hens), they will be moved to be slaughtered at the farm's slaughterhouse. Third, outbound logistics of production is where companies collect eggs (finished goods), store the eggs for safe inventory, and distribute products (eggs and chicken) to customers or end consumers. In the logistics of the egg industry, egg distribution planning is considered a crucial problem because egg distribution is a main cause of increasing total cost in the egg industry. Nowadays, the case study company has insufficient suitable decision tools.

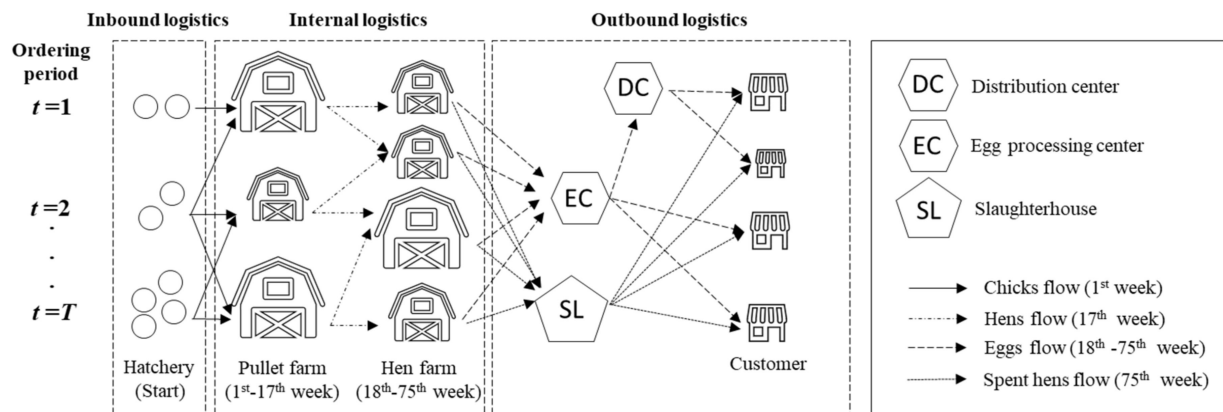


Figure 1. Flowchart of hen and egg production in the case study company.

At present, the case study company has no decision tools for minimizing total cost in outbound logistics. The egg distribution problem is highly complex, primarily because of the need to respect labor costs of a driver, variety of vehicle capacity (volume), duration time of route restriction, the time window of the customer, various demands of egg sizes for each customer and inventory restrictions. Moreover, the characteristic of the problem can be classified as a multi-depot problem since eggs can be delivered from more than one distribution center, and the numbers of available vehicles are different at each depot. Therefore, if the outbound logistics of the egg industry cannot be managed effectively, there will be an increase in transportation distance, transportation time and the number of transport vehicles which directly increases the total cost.

This paper focuses on the outbound logistics activity of the case study. The problem is formulated as a Multi-Depot, Multi-Product Heterogeneous Fleet Vehicle Routing Problem with Time Window and Inventory restriction (MDPVRPTWI) in the various practical operations of the real-world industry. The objective is to minimize total costs consisting of transportation costs, labor costs of drivers and holding costs. To solve this problem, the customer's distance from each depot, numbers of vehicles used and holding cost at each depot are considered for modeling and solving to obtain optimal solutions. We developed a Mixed Integer Linear Programming (MILP) model for a small-sized problem. This problem is considered an NP-hard (non-deterministic polynomial-time hardness) problem that cannot be solved by MILP optimally in a small computational time. We developed a hybrid and new self-adaptive process for the Differential Evolution (DE) algorithm to solve a large-sized problem.

The remainder of this paper is organized as follows. Section 2 is the literature review. An overview of the egg distribution problem is presented in Section 3. Section 4 introduces the proposed solution strategy. The experimental results are given in Section 5. Finally, Section 6 is the conclusion of the work of this paper.

2. Literature Review

A brief review of multi-depot, heterogeneous fleet, time window, multi-product, and inventory restriction and solution is presented in this section.

Pick-up and delivery problems (PDPs) are concerned with a type of vehicle routing problem (VRP) in which goods or persons come from different origins and go to different destinations [4]. PDP issues occur in various contexts, including logistics, ambulatory care and robotics [5,6]. Vehicles available to pick up and deliver cargo can have heterogeneous capacities. Each vehicle's load must not exceed its capability. However, depending on the problem, additional assumptions, and restrictions on vehicle paths, such as time windows or repeated visits at certain vertices, can be presented. Wang et al. [7] proposed an exact method and a constructive heuristic approach from the min-max split delivery multi-depot vehicle routing problem with a minimum service time requirement. In this problem, each customer requires a specified amount of service time. They generated instance ranges from 10 to 500 and 3 to 20 for the customer and depot, respectively, to investigate the constructive heuristic. They tested the heuristic on three sets of data with 43, 21 and 258 instances. The heuristic result on the first and second sets of data was 2% and 1.5%. However, the third set of heuristics is more substantial when the average number of customers per route is small and there is a medium level of average customer service time. Kachitvichyanukul et al. [8] studied a multi-depot vehicle routing problem with multiple pick-up and delivery requests. They developed a solution approach based on Particle Swarm Optimization (PSO) to create a new particle for an assignment using a vehicle that created a decoding method to find the best solution. They used 56 sample problems with the best-known solution and divided the instance problems into 13, 17 and 26 for clustered location, random location, mixed cluster and random locations. All the instance problems were compared with their previous work and the best-known solutions. The conclusion showed that the new decoding method for PSO provided better solutions for their problems than their previous work. Mancini [9] developed an exact method, initial solution and Adaptive Large Neighborhood Search (ALNS) based metaheuristic on the multi-depot multi-period vehicle routing problem with a heterogeneous fleet for minimizing the total cost. They proposed 1 to 9 instances and a number of customers ranging from 50 to 75 customers. They compared the LANS algorithm results with the Large Neighborhood Search (LNS) algorithm. LANS showed higher-quality performance over LNS that averaged a computational time of 400 s and obtained an averaged percentage improvement of 55% to the constructive heuristic.

Most authors have developed a mathematical model to solve multi-product vehicle routing problems and planning problems in transport-related industries, which are difficult to solve in a reasonable amount of time for medium or large-sized problems [10] and for the multi-depot vehicle routing problem [11,12]. In 2014, Salhi et al. [13] proposed a mixed vehicle routing problem on the multi-depot routing with a heterogeneous vehicle and a mixed-integer linear model for solving the problem. The multi-product VRP can be better illustrated in Figure 2.

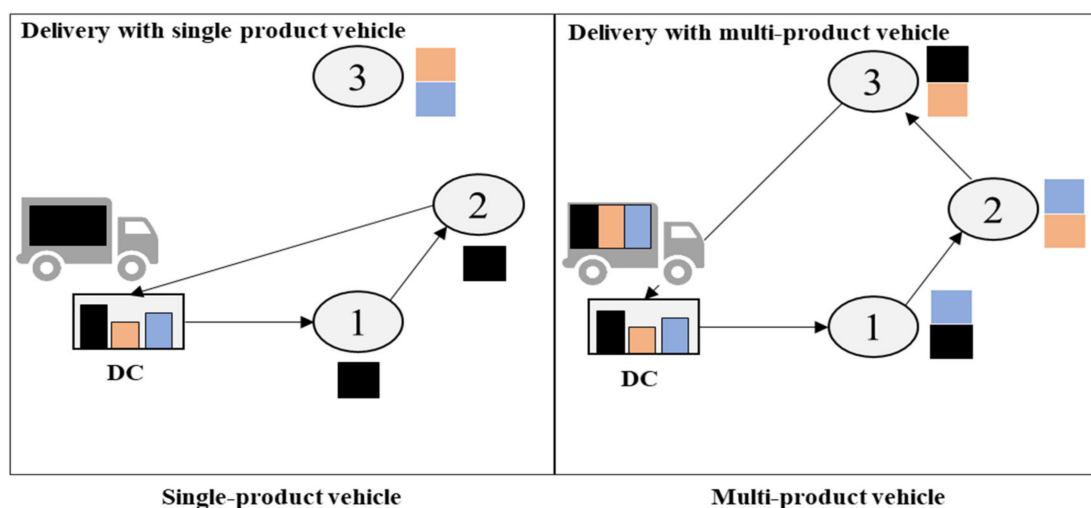


Figure 2. The delivery route for single-product vehicle and multi-product vehicle.

In the last decade, inventory restrictions have also been extensively studied to solve the problem. For example, Chen et al. [14] proposed an improved variable neighborhood search (VNS) algorithm with a heuristic to solve VRP with an unpaired pick-up, delivery and split loads. They considered a constraint about the demand of customers for each various product and inventory restriction. The inventory levels and quantity of demanded products were described that should be suitably allocated to minimize total cost. Qiu et al. [15] studied multi-product multi-vehicle production routing problems (MMPRPs) with fixed production startup cost, setup cost of each problem, and inventory restriction. Their objective was to minimize total cost, including the operational cost of the production, inventory cost and transportation cost. Bouanane et al. [16] proposed a hybrid genetic algorithm that incorporates three different procedures to solve the multi-depot vehicle routing problem: simultaneous delivery and pick-up, and inventory restrictions. Table 1 summarizes the studies about the VRP in the literature.

Table 1. Summary of literature for the VRP.

Paper	Constraints							Solution Method		
	Multi-Depot	Time-Window	Heterogeneous Fleet	Multi-Product	Periodic	Pick Up & Delivery	Inventory	Exact	Heuristic	Metaheuristic
Landrieu et al. [5]		✓				✓			✓	✓
Ai and Kachitvichyanukul [6]						✓		✓	✓	✓
Liu et al. [11]	✓					✓		✓	✓	✓
Hasani-Goodarzi and Tavakkoli-Moghaddam [10]	✓									✓
Battarra et al. [4]						✓		✓		
Salhi et al. [13]	✓		✓					✓		✓
Chen et al. [14]				✓			✓	✓	✓	✓
Kachitvichyanukul et al. [8]	✓		✓			✓		✓		✓
Wang et al. [7]	✓							✓	✓	✓
Mancini [9]	✓		✓					✓		✓
Lalla-Ruiz et al. [12]	✓							✓		✓
Qiu et al. [15]	✓			✓	✓		✓	✓		✓
Bouanane et al. [16]	✓					✓	✓			✓
This research	✓	✓	✓	✓			✓	✓		✓

In the scholarly literature, logistics for egg industry applications have received much attention. Most of the egg industry research has been focused on value chain optimization and the allocating pullets problems, such as transportation supply for smooth operations. We refer readers to Seydim and Dawson [17], Boonmee et al. [18], Dechampai et al. [19] and Hisasaga et al. [20]. In previous work in the same company case study, Boonmee et al. [18] studied the inbound logistics of egg production for allocating pullets to hen houses to minimize the total cost. The total cost included the cost of farm utilization, transportation cost from pullet farms to hen farms, and loss from mixing hens at different ages in the same hen houses. They used traditional Growing Neural Gas (GNG) and Hybrid GNG to solve their problem. Additionally, Dechampai et al. [19] presented a vehicle routing problem with mixed pick-up and delivery services (PDP). They developed two heuristics to minimize the total cost of fuel and vehicle utilization in the internal logistics of egg production to minimize the distance from pullet farm to hen farm and the vehicle rental cost. The first heuristic used the Differential Evolution (DE) algorithm for clustering the pullet houses. The second heuristic used multifactor-based evolving Self-Organizing Maps (SOM) with the DE algorithm for minimizing the transfer from pullet farms to the hen farms.

Various approaches have been used to solve the problem. DE has been established as a new evolutionary strategy during the last decade. The DE algorithm was first implemented by Storn and Price [21] and was effectively implemented to optimize algorithms. One of the most efficient evolutionary algorithms is DE, which is a population-based metaheuristic. DE has been used successfully in various areas, including output scheduling [22] and transportation [23]. For the VRP scheduling problem, Kunnapapdeelert and Kachitvichyanukul [24] created four new DE-based algorithms for solving the multi-depot vehicle routing problem. Two of the improved DE algorithms are focused on subgrouping, and the other two are based on strategy swapping. In the same year, Akararungruangkul and Kaewman [25] proposed DE to solve the location routing problem (SLRP) when the

objective function is fuel consumption. Fan et al. [26] presented a DE with knowledge-based control parameters and strategy adaptation, combining three well-known test suites. Later, Zhang et al. [27] proposed the parallel DE to minimize transport costs and penalties due to logistics service providers' delays. Marinaki et al. [28] developed a distance-related DE to maximize the total price collected. Recently, Sethanan and Jamrus [29] proposed a hybrid DE to solve the multi-trip vehicle routing problem with backhauls and heterogeneous fleets in the beverage logistics industry. They demonstrated the benefit of using self-adaptive strategies because the user does not have to change the control parameters. The total computational effort was able to converge faster and significantly reduced into efficient solutions. As a result, this paper proposes a new Hybrid and Self-Adaptive Differential Evolution algorithm (HSADE) that combines their respective benefits, increasing the robustness to solve the problem. We suggest the HSADE that includes Hybrid and Self-Adaptive modification of the DE algorithm, which can also include self-adaptive mutation strategies, control parameters for crossover rate probability and hybrid neighborhood search structures using the probability exchange equation. We focus on addressing egg distribution planning transportation by improving the efficient solution in terms of the total cost, including the transportation cost, labor cost and holding cost. There is no work in the literature to the best of our knowledge interested in the interaction between these constraints: multi-depot, heterogeneous fleet, time window, multi-product and inventory restriction.

3. Overview of the Egg Distribution Problem

3.1. Problem Overview

The egg distribution problem consists of multi-depot, multi-product, heterologous fleet vehicle routing problems, time window and inventory restriction. The problem includes multi-depots; each vehicle must start its route from these depots and return to the same depot. Each depot has different maximum quantities of egg sizes from other depots, and the cargo space of each depot is also different. That means an egg shortage can occur at each depot. The fleet of the vehicle is heterologous, which means that the capacity of all vehicles is different. The vehicle must return to the depot within the duration time of the route. The limit of working hours of drivers is considered. Each depot has a different maximum number of vehicles. Split delivery is not considered, which means that customers must be visited only once. The Time window and service time of customers must be respected. The demand of each customer has at least one egg size and must be less than or equal to the total capacity of the vehicle, so customers must receive different egg sizes from a single-vehicle; all of the different egg sizes can also be supplied by each depot.

3.2. Assumptions

The presented mathematical model is based on the following assumptions:

- Each customer must be serviced by a single-vehicle;
- Each customer has a demand for multi-products;
- A vehicle must service each customer under the time window of the customer;
- Each vehicle must start and return from its route to the same depot;
- A vehicle must return to the depot before the duration time of the route;
- A vehicle can deliver multi-products;
- Each depot has a different maximum number of vehicles;
- Each depot has different maximum quantities of each product;
- Demands of all customers must be less or equal to the supplies of all depots;
- The demands of all customers must be satisfied;
- The fleet of vehicles is heterogeneous.

This section proposed a mixed-integer linear programming model with the objective of minimizing the total cost consisting of transportation cost, labor cost and holding cost. The model is presented below with an explanation of each assumption and each constraint.

3.3. Indices, Parameters and Decision Variables

This section defines the indices, parameters and variables used in the proposed mixed-integer linear programming model. The variables and parameters for the model are summarized in Table 2.

Table 2. Indices Parameters and Decision variable of egg distribution problem.

Symbol	Description
Indices	
i, j	Indices of all customers and depots; $i, j = 1, 2, \dots, N + M$
k	Index of all vehicles; $k = 1, 2, \dots, K$
p	Index of all egg sizes; $p = 1, 2, \dots, P$
Parameters	
N	Number of customers
M	Number of depots
K	Number of vehicles
P	Number of products
C_{ij}	Transportation cost between customer i and customer j
D_{ij}	Distance between customer i and customer j
t_{ij}	Travel time between customer i and customer j
f_k	Labor cost of vehicles k
Q_{kp}	Maximum capacity of vehicle k for egg size p
v_{ip}	Inventory of egg size p at depot i
H_{ip}	Holding cost of egg size p at depot i
d_{ip}	Demand of customer i for egg size p
L	Positive Large number
D'	Maximum distance a vehicle can travel
K'_i	Maximum number of vehicles departing from depot i
s_i	Service time of the delivery at customer i
e_i	Earliest arrival/delivery time at customer i
l_i	Latest time at customer i
T_k	Maximum duration time of vehicle k
I_i	Maximum inventory at depot i
Decision variable	
x_{ijk}	$\begin{cases} 1 & \text{if vehicle } k \text{ travels from customer } i \text{ to customer } j \\ 0 & \text{otherwise} \end{cases}$
u_k	$\begin{cases} 1 & \text{if vehicle } k \text{ is used} \\ 0 & \text{otherwise} \end{cases}$
g_i	$\begin{cases} 1 & \text{if depot } i \text{ is operated} \\ 0 & \text{otherwise} \end{cases}$
y_{ipk}	$\begin{cases} 1 & \text{if the demand of customer } i \text{ for egg size } p \text{ is fulfilled by vehicle } k \\ 0 & \text{otherwise} \end{cases}$
z_{ij}	$\begin{cases} 1 & \text{if the customer } j \text{ is allocated to depot } i \\ 0 & \text{otherwise} \end{cases}$
a_i	Arrival time of the delivery vehicle at customer i
w_i	Waiting time of the delivery vehicle at customer i
A_{ik}	A non-negative time that indicates when vehicle k departs from customer i
T'_{ik}	Variable used for elimination of sub-tours.

Objective function

$$\begin{aligned} \min \quad & \sum_{i=1}^{N+M} \sum_{j=1}^{N+M} \sum_{k=1}^K C_{ij} x_{ijk} + \sum_{k=1}^K f_k u_k \\ & + \left(\sum_{i=1}^{N+M} \sum_{p=1}^P (v_{ip} H_{ip} g_i) - \sum_{i=1}^{N+M} \sum_{j=1}^{N+M} \sum_{k=1}^K (z_{ij} d_{jk} H_{ip}) \right) \end{aligned} \quad (1)$$

Subject to

$$\sum_{i=N+1}^{N+M} \sum_{j=1}^N x_{ijk} \leq u_k \quad \forall k = 1, \dots, K \quad (2)$$

$$\sum_{k=1}^K \sum_{i=1}^{N+M} x_{ijk} = 1 \quad \forall j = 1, \dots, N \quad (3)$$

$$\sum_{k=1}^K y_{ipk} = 1 \quad \forall i = 1, \dots, N, \forall p = 1, \dots, P \quad (4)$$

$$\sum_{p=1}^P y_{ipk} \leq L \sum_{j=1}^{N+M} x_{jik} \quad \forall i = 1, \dots, N, \forall k = 1, \dots, K \quad (5)$$

$$\sum_{j=1}^{N+M} x_{jik} \leq \sum_{p=1}^P y_{ipk} \quad \forall i = 1, \dots, N, \forall k = 1, \dots, K \quad (6)$$

$$\sum_{i=1}^{N+M} x_{ijk} \leq 1 \quad \forall j = 1, \dots, N, \forall k = 1, \dots, K \quad (7)$$

$$\sum_{i=1}^{N+M} x_{ihk} - \sum_{j=1}^{N+M} x_{hjk} = 0 \quad \forall h = 1, \dots, N+M, \forall k = 1, \dots, K \quad (8)$$

$$\sum_{i=1}^N x_{ijk} \leq 1 \quad \forall k = 1, \dots, K, \forall j = 1, \dots, N+M \quad (9)$$

$$\sum_{i=1}^N y_{ipk} d_{ip} \leq Q_{kp} \quad \forall k = 1, \dots, K, \forall p = 1, \dots, P \quad (10)$$

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} D_{ij} x_{ijk} \leq D' \quad \forall k = 1, \dots, K \quad (11)$$

$$\sum_{j=1}^N \sum_{k=1}^K x_{ijk} \leq K'_i \quad \forall i = N+1, \dots, M \quad (12)$$

$$\sum_{i=N+1}^{N+M} \sum_{j=1}^N x_{ijk} = u_k \quad \forall k = 1, \dots, K \quad (13)$$

$$\sum_{i=1}^N \sum_{j=N+1}^{N+M} x_{ijk} = u_k \quad \forall k = 1, \dots, K \quad (14)$$

$$T''_{ik} - T''_{jk} + (N+M)x_{ijk} \leq N+M-1 \quad \forall i = 1, \dots, N+M, \forall j = 1, \dots, N, \forall k = 1, \dots, K \quad (15)$$

$$T''_{ik} \geq 0 \quad \forall i = 1, \dots, N+M, \forall k = 1, \dots, K \quad (16)$$

$$a_j \geq a_i + w_i + s_i + C_{ij} + L \sum_{k=1}^K (x_{ijk} - 1) \quad \forall i = N+1, \dots, N+M, \forall j = 1, \dots, N, \quad (17)$$

$$e_i \leq a_i + w_i \leq l_i \quad \forall i = 1, \dots, N \quad (18)$$

$$a_i + w_i + s_i + t_{ij} + L(x_{ijk} - 1) \leq A_{ik} \quad \forall i = 1, \dots, N, \forall j = 1, \dots, N+M, \forall k = 1, \dots, K \quad (19)$$

$$A_{ik} \leq T_k \quad \forall j = 1, \dots, N+M, \forall k = 1, \dots, K \quad (20)$$

$$\sum_{j=1}^N z_{ij} d_{jp} \leq v_{ip} g_i \quad \forall i = N+1, \dots, N+M \quad (21)$$

$$\sum_{j=1}^N \sum_{p=1}^P z_{ij} d_{jp} \leq I_i \quad \forall i = N+1, \dots, N+M, \quad \forall p = 1, \dots, P \quad (22)$$

$$-z_{ij} + \sum_{h=1}^{N+M} (x_{ihk} + x_{hjk}) \leq 1 \quad \forall i = N+1, \dots, N+M, \forall j = 1, \dots, N \quad (23)$$

$$\forall k = 1, \dots, K$$

$$\sum_{i=N+1}^{N+M} z_{ij} \leq 1 \quad \forall j = 1, \dots, N \quad (24)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i = 1, \dots, N+M, \forall j = 1, \dots, N+M, \forall k = 1, \dots, K \quad (25)$$

$$y_{ipk} \in \{0, 1\} \quad \forall i = 1, \dots, N+M, \forall p = 1, \dots, P, \forall k = 1, \dots, K \quad (26)$$

$$u_k \in \{0, 1\} \quad \forall k = 1, \dots, K \quad (27)$$

$$z_{ij} \in \{0, 1\} \quad \forall i = N+1, \dots, N+M, \forall j = 1, \dots, N \quad (28)$$

$$g_i \in \{0, 1\} \quad \forall i = 1, \dots, N+M. \quad (29)$$

The objective Equation (1) minimizes the total cost, which includes the transportation cost, labor cost and holding cost. Equations (2) and (3) guarantee each customer is delivered to by one vehicle. Equation (4) guarantees a single vehicle must deliver each customer's demand for given eggs. Equations (5) and (6) ensure the demand of customer i for egg size p can be fulfilled by vehicle k only when this vehicle delivers to customer i . Flow conservation constraints are guaranteed in Equations (7) and (9). The limit of the capacity of vehicle constraint is in (10). Equation (11) limits the maximum distance of the vehicle. Equation (12) is related to the limits of the maximum number of vehicles departing from depot i . Equations (13) and (14) check vehicle availability. Sub-tour elimination constraints are in Equations (15) and (16). Equations (17) and (18) specify the time window at each customer, and the service time is the time interval between deliveries. Equations (19) and (20) define the duration time of the route. Equation (21) is the operation cost when each depot is assigned. Equation (22) ensures the inventory constraints for the depot are given. Equations (23) and (24) specifies that a customer can be assigned to a depot only if there is a route from that depot going through that customer. Equations (25)–(29) are the basic restrictions on the decision and binary variables.

4. Solution Strategies

The MILP developed in Section 3 can be solved optimally for small-sized instances. We developed a Hybrid and Self-Adaptive Differential Evolution algorithm (HSADE) to solve our proposed model and compared performance solutions through numerical experiments in handling large-sized instances. The traditional Differential Evolution algorithm (DE) initializes a set of the dimension of vectors and obtains the best dimensions iteratively via the DE mechanisms (i.e., mutation, recombination). In solution strategies, HSADE combines the constructive heuristic for the initial solution, the self-adaptive mutation strategies, and the neighborhood search by probability (NSP).

In Sections 4.1–4.4 we divided this section to explain the frameworks for the proposed HSADE; first, the initialization representation is described, then the procedure of the self-adaptive mutation strategies and the proposed HSADE are presented and, finally, the fitness value and the selection operation of HSADE are described. The Framework of the HSADE algorithm is shown in Figure 3

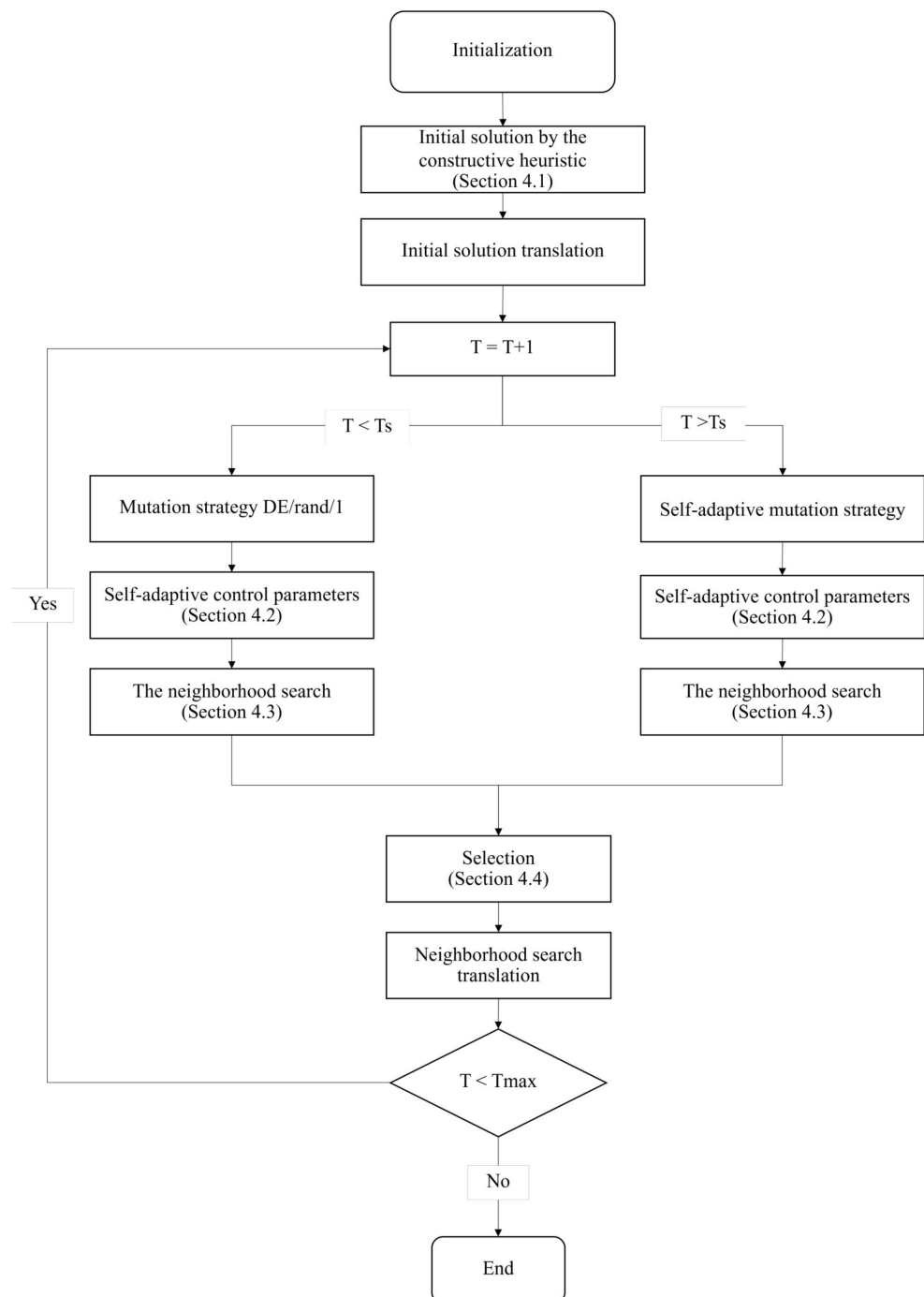


Figure 3. Framework of the HSADE algorithm.

4.1. Initialization of HSADE

In HSADE, the dimensions in each vector are identified by a limited number of the vehicles on each depot id and a customer id sequence in which the customer's id is sorted in increasing order. A random real number in the range (0–1) is generated for each corresponding id (Figure 4). The vectors are split into two sub-vector arrays of length $|D| + |N|$ since there are $|D|$ depots and $|N|$ customers, for example, two depots and ten customers, denoted as vector array Tg . The first array is the ratio number of the route for each depot id in $Tg(1, |D|)$, while the second array is the customer's id and the random read number in $Tg(1, |N|)$ (Figure 4a)

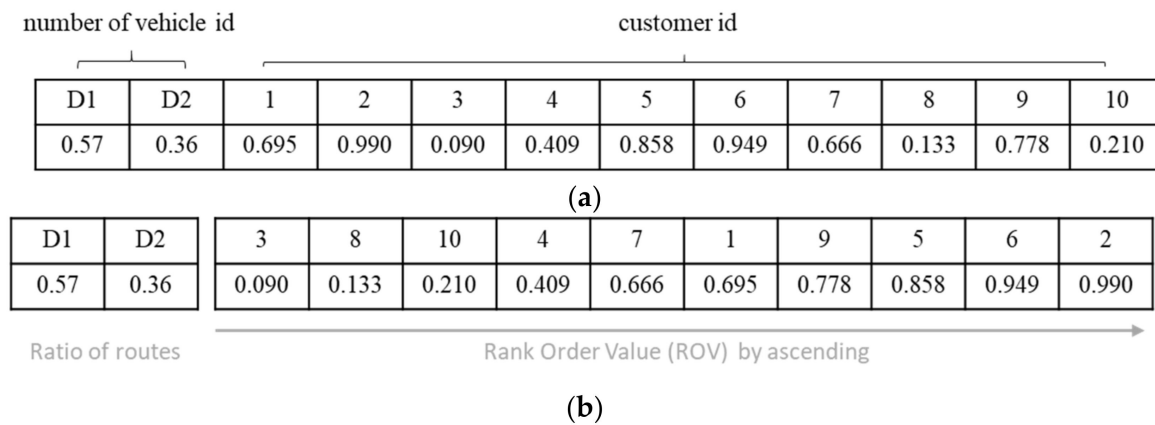


Figure 4. The vector initialization. (a) HSADE vector; (b) Customer sequence and depot.

For example, see the first array in Figure 4a, where the limited number of the vehicles from each depot equals three and four. The dimensions with the random number are {0.57, 0.36}. Depot 1 has the probability of selecting a vehicle equal to $1/3$, which means the range [0,0.33], [0.33,0.66], and [0.66,1.00] for vehicles number 1, 2 and 3, respectively. Depot 1, as the random number is 0.57, must have at least two vehicles. Similarly, Depot 2 has the random number 0.36 so must assign at least two vehicles. (see Figure 4b).

In the second array, the customer sequence can be obtained with ascending Rank Order Value (ROV) decoding from a randomly real number. Consequently, the customer sequence is depicted as {3,8,10,4,7,1,9,5,6,2} Figure 4b.

4.1.1. Initial Solution of HSADE

This section represents generating a feasible initial solution by the constructive heuristic. There are two parts to generate an initial solution. The first is to cluster customers to each of the depots $|D|$. The clustered customers are assigned to the nearest depot because the objective function is to minimize the total cost. For example, there are two depots, D1 and D2, and $|N|$ customers. The distances between customers and the depot are $d_{i,D1}$ and $d_{i,D2}$. Each customer must be assigned to a single depot. The selection, if $d_{i,D1}$ is less than $d_{i,D2}$, then assigns customer i to depot 1; if $d_{i,D1}$ is greater than $d_{i,D2}$, then assign customer i to depot 2; if $d_{i,D1}$ is equal to $d_{i,D2}$, assign a depot arbitrarily.

The second is to find the route to each customer for each depot by the Nearest Neighbor Heuristic (NNH) [30]. The NNH starts with the nearest customer of each depot, then finds the shortest distance connecting the previous customer until all customers are selected.

Referring to Section 4.1, the encoding of the initial solution with initialization of HSADE, Figure 5a represents the customer sequences {4,2,6,5}, and the customer sequences {10,7,9,1,3,8} are clustered by D1 and D2, respectively. Both sequencings are generated by NNH.

4.1.2. Initial Solution Translation of HSADE

The purpose of the initial solution translation is to convert the initial solution back to the current vector. As explained, a vector is encoded as a vector array, which consists of the customer's id (denoted by CT-10) and a random real number (Figure 5a) (denoted by RN-10). Referring to the Section 4.1.1 (Figure 5a), the constructive heuristic obtained a customer sequence {4,2,6,5,10,7,9,1,3,8} (denoted by FT-10); the dimension sequence CT-10 is replaced into FT-10. The initial solution translation is to change the RN-10 to the new position dimension that corresponds with a customer sequence (1, $|N|$).

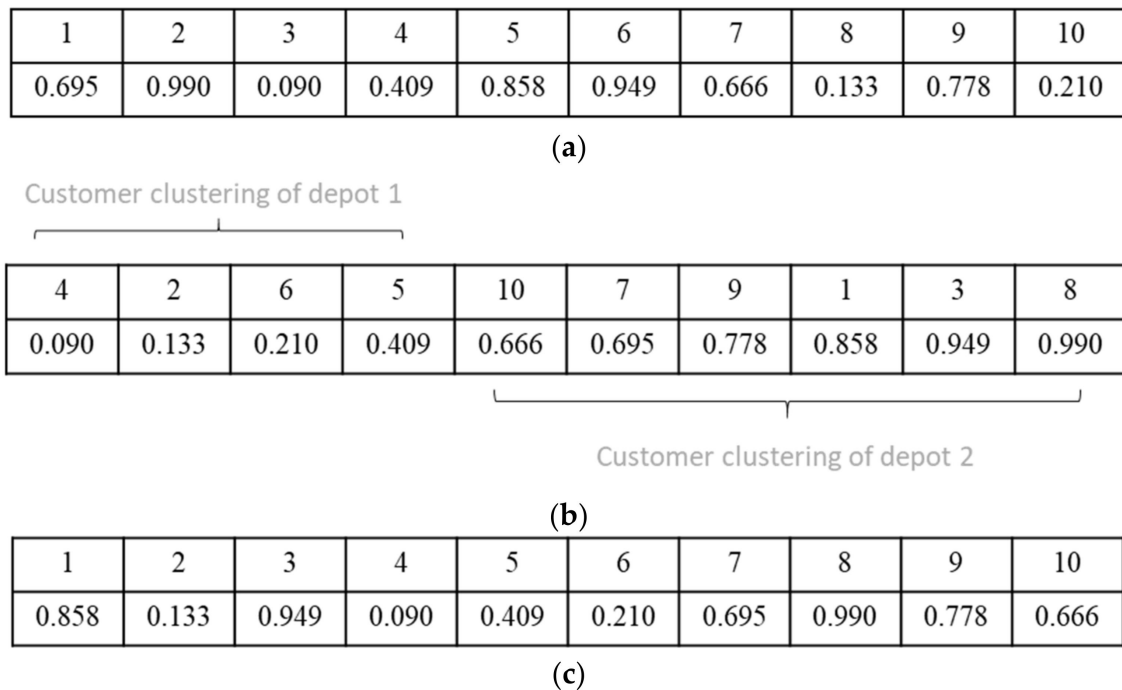


Figure 5. Initial solution translation of HSADE. (a) Current vector (before the initial solution translation); (b) Initial solution by constructive heuristic; (c) Current vector (after the initial solution translation).

For example, Figure 5b obtains a customer sequence CT-10, the dimension sequence corresponds to {0.090, 0.133, 0.210, 0.409, 0.666, 0.695, 0.778, 0.858, 0.949, 0.990} (denoted by HCT-10) and when decoded the vector can return to the initial solution of a customer sequence FT-10. Thus, the translation obtains the new dimension changed from RN-10 to HCT-10 (Figure 5c). In the vector iteration, the initial solution dimension is the role to guide for updating the new generation of target vectors.

4.2. Self-Adaptive Differential Evolution Algorithm with Mutation Strategies Adaptation

This section represents the new self-adaptive DE algorithm with mutation strategies adaptation. The mutation strategy is used to generate a mutated vector (v_i^{t+1}), where the performance of DE is significantly influenced by the selection of mutation strategies [31].

The famous mutation equation strategies are listed as follows:

$$\text{DE/rand/1} : v_i^{t+1} = x_{r1}^t + F(x_{r2}^t - x_{r3}^t) \quad (30)$$

$$\text{DE/rand/2} : v_i^{t+1} = x_{r1}^t + F(x_{r2}^t - x_{r3}^t) + F(x_{r4}^t - x_{r5}^t) \quad (31)$$

$$\text{DE/rand/current-to-best/1} : v_i^{t+1} = x_i^t + F(x_{best}^t - x_i^t) + F(x_{r1}^t - x_{r2}^t) \quad (32)$$

$$\text{DE/rand/current-to-best/2} : v_i^{t+1} = x_i^t + F(x_{best}^t - x_i^t) + F(x_{r1}^t - x_{r2}^t + x_{r3}^t - x_{r4}^t) \quad (33)$$

$$\text{DE/rand/rand-to-best/1} : v_i^{t+1} = x_{r1}^t + F(x_{best}^t - x_i^t) + F(x_{r1}^t - x_{r2}^t), \quad (34)$$

where r_1, \dots, r_5 are the random vectors within the range of the number of the population $[1, NP]$, and each r must be different (i.e., $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5$). The mechanism of self-adaptive mutation strategies is introduced below.

4.2.1. Self-Adaptive Mutation Strategies

The self-adapting mutation procedure can be described as follows:

(1) The number of vectors of each mutation strategy for adapting mutation strategies (called the Weak strategy) is computed as follows:

$$v_{i,Str_{name}}^t = v_{weak_i,Str_{name}}^t \text{ if } \Delta\beta_i^t \geq \Delta\gamma_i^t, \quad (35)$$

where Equation (35) denotes $\Delta\beta_i^t = |f(u_{i,Str_{name}}^t) - best(f(u^t))|$, which is the difference between the objective function value of each vector and the best objective function value, and $\Delta\gamma_i^t = |avg(f(u^t)) - f(u_{i,Str_{name}}^t)|$ is the difference between the average objective function value and the objective function value of each vector. It is how to select the weak strategies out of all vectors.

(2) the number of weak strategy updates for the new strategy is as follows:

$$v_{i,Str_{name}}^{t+1} = v_{weak_i,Str_{name}}^t = \begin{cases} v_{i,bestStr_{name}}^t, & \text{if } Str_{name} \neq bestStr_{name} \\ v_{i,randomStr_{name}}^t, & \text{if } Str_{name} = bestStr_{name} \end{cases}. \quad (36)$$

The Equation (36) is to eliminate the weak mutation strategies from the individual vector. For example, the best strategy is strategy 1. If vector i is the weak vector that is formed from another mutation strategy except strategy 1, it must be updated to use strategy 1. In other words, vector i is formed from the best mutation strategy (strategy 1). It has been updated from random strategies.

4.2.2. Self-Adaptive Control Parameters

In order to increase the performance of the Crossover Rate (CR), this section represents the self-adaptive control parameters $CR \in [0, 1]$. The crossover mechanism is to generate a trial vector by replacing the dimension of the target vector. The following equation creates the trial vector:

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1}, & \text{if } rand_j \leq CR \text{ or } j = randn(i) \\ x_{i,j}^t, & \text{if otherwise} \end{cases}. \quad (37)$$

This section has used Equation (38) for adapting the crossover probability.

$$CR = CR_{min} + t * \frac{CR_{max} - CR_{min}}{Max(t)} \quad (38)$$

where CR_{min} , CR_{max} , t , $Max(t)$ are the minimum crossover rate probability, maximum crossover rate probability, computation time and limited computation time, respectively. Due to it being in the early stages of evolution, the trial vectors are identical to the target vectors. As the iteration increases, the similarity to the target vectors decreases, and the trial vectors become more identical to the mutated vectors. By increasing the CR, the capacity for local search can be improved.

4.3. The Neighborhood Search by Probability

This section aims to inter-change the clustering of customers in each depot. The neighborhood search by probability (called NSP) is used to exchange the customers based on a probability calculated by Equation (39):

$$p(C_i, D_n) = \frac{\max\{\overline{d(C_i, D)} - d(C_i, D_n), 0\}}{\sum_{n=1}^N \max\{\overline{d(C_i, D)} - d(C_i, D_n), 0\}}, \quad (39)$$

where $d(C_i, D_n)$ indicates the distance customer i to depot n , and $\overline{d(C_i, D)}$ indicates the average distance between customer i and all depots.

For example, the distance between customer 1 and depots 1, 2 and 3 are 7, 4 and 5, respectively. The average distance of customer 1 and all depots is 5.34. The probability that customer 1 is assigned to depots 1, 2 and 3 equals 0.00%, 80.00% and 20.00%, respectively. Although depot 2 is closest to customer 1, there is a chance to assign customer 1 to depot 2.

After calculating the probability, the next step is to randomize the number of customers to inter-change nodes (k), where $k \in [2, \text{round up}(N * 0.2)]$ that mean k should be the random number between 2 to round up 20% of the number of customers.

The neighborhood search strategies, a kind of changing sequence, is developed to four neighborhood search structures and described for $k = 2$ as follows:

- Inter-exchange by probability, randomize k customer nodes from each depot in a clustered customer sequence, as demonstrated in Figure 6a. The neighborhood structure is defined as Nh1;
- Inter-insertion by probability, randomize k customer nodes from one depot to another depot in a clustered customer sequence, as demonstrated in Figure 6b. The neighborhood structure is defined as Nh2;
- Intra-exchange without probability is similar to inter-exchange but change k customer nodes only for the same depot. The neighborhood structure is defined as Nh3;
- Intra-insertion without probability is similar to inter-insertion but insert k customer nodes only for the same depot. The neighborhood structure is defined as Nh4.

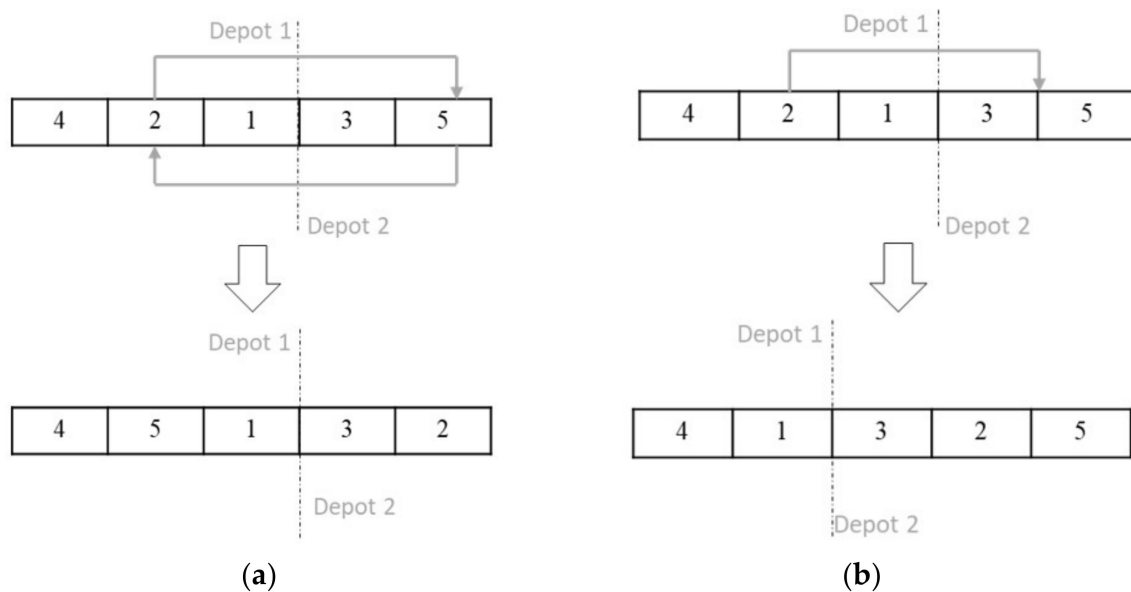


Figure 6. Methods of neighborhood search. (a) Inter-exchange; (b) Inter-insertion structure.

The sequencing of the structure neighborhood is selected by two random operations between inter-depot and intra-depot neighborhood search structures. For example, iteration 1 used sequencing of structure neighborhood {Nh1,Nh4}, but iteration 2 used {Nh2,Nh3}. The idea of the NPS is to increase the search to prevent moving into a trap solution.

Neighborhood Search Translation

Similarly to Section 4.1.2, the neighborhood search translation is used to return the position sequence with the new changed customer sequence. For example, a customer sequence before entering the NSP is {6, 3, 7, 8, 5, 1, 2, 4, 9, 10} denoted by NS10 (Figure 7), the dimension position corresponding to the customer sequence is {0.090, 0.133, 0.210, 0.409, 0.666, 0.695, 0.778, 0.858, 0.949, 0.990} (denoted by RS10). The NSP that changes NS10 to a new customer sequence is {8, 3, 7, 6, 10, 2, 1, 4, 9, 5} (denoted by NSP10) (Figure 7b) with the corresponding dimension position RS10.

Depot 1					Depot 2				
6	3	7	8	5	1	2	4	9	10
0.090	0.133	0.210	0.409	0.666	0.695	0.778	0.858	0.949	0.990

(a)

Depot 1					Depot 2				
8	3	7	6	10	2	1	4	9	5
0.090	0.133	0.210	0.409	0.666	0.695	0.778	0.858	0.949	0.990

(b)

1	2	3	4	5	6	7	8	9	10
0.778	0.695	0.133	0.858	0.990	0.409	0.210	0.090	0.949	0.666

(c)

Figure 7. Neighborhood search translation. (a) Before NSP; (b) After NSP; (c) Translated NSP.

Neighborhood search translation returns NSP10 to the initial customer id $\{1, \dots, |N|\}$, and corresponding dimension position RS10 reorders to original dimension position (Figure 7). Therefore, the Neighborhood search translation changes as shown in Figure 7a to Figure 7c. The random dimension position must be translated to change to a new dimension position before the new iteration is performed in the current iteration.

4.4. The Fitness Value and Selection Operation

The selection mechanism of DE differs from the other evolutionary algorithms. The fitness value of the trial vector $u_{i,j}^t$ is compared with that of its target vector $x_{i,j}^t$ in each generation; the one with a better fitness value (in this case, lower total cost) will propagate the population of the next generation. The population of the next generation $x_{i,j}^{t+1}$ is selected as follows:

$$x_{i,j}^{t+1} = \begin{cases} v_{i,j}^t, & \text{if } f(v_{i,j}^t) \leq f(x_{i,j}^t) \\ x_{i,j}^t, & \text{otherwise.} \end{cases} \quad (40)$$

5. Numerical Experiments

In this section, we report on numerical experiments using a PC with an Intel Core i7, 2.20 GHz CPU and 16 GB DDR4 RAM to demonstrate the performance of HSADE. Our computational experiments were performed on three-size instances: small-sized, medium-sized, and large-sized instances. Specifically, the small-sized instance is compared with HSADE for heuristic performance (HP), calculated from Equation (41). The MILP was implemented by LINGO v.17 (LINDO Systems, Inc. 1415 North Dayton Street, Chicago, IL 60642). The HSADE algorithm was developed through the MATLAB Version 9.4.0.813654 (R2018a) software (John N. Little, Cleve Moler, Steven Bangert, Natick, Massachusetts, United States). The relative improvement (RI) is determined using Equation (42) to measure the percentage improvement between current practices and the HSADE algorithm.

$$HP = \left(\frac{Optimal}{Sol_{name}} \right) \times 100 \quad (41)$$

$$RI = \left(\frac{Sol_{CT} - Sol_{name}}{Sol_{CT}} \right) \times 100, \quad (42)$$

where

HP = the heuristic performance (%).

$Optimal$ = the optimal solution obtained from the mathematical model.

RI = the relative improvement (%) between current practices and meta-heuristics.

Sol_{CT} = the solution obtained from current practices.

Sol_{name} = the solution obtained from the meta-heuristics.

The numerical experiments were tested with 18 randomly generated instances and 1 case study. The factors of all instances and a case study are shown in Table 3.

Table 3. The factors of data generated for each problem instance.

Size Instances	Instance Name	No. Depot	No. Customer	Size of Demand (Egg Trays)	No. Vehicles of Each Depot	No. Product Types	Capacity Size of Vehicles S/L
Small	S1	2	12	U (10,60)	4	2	60/100
	S2	2	12	U (10,60)	4	2	60/100
	S3	3	15	U (10,60)	4	2	60/100
	S4	3	15	U (10,60)	4	3	60/100
	S5	3	20	U (10,60)	4	3	60/100
Medium	M1	4	50	U (20,100)	8	4	80/120
	M2	4	50	U (20,100)	8	4	80/120
	M3	4	50	U (20,100)	10	4	80/120
	M4	4	75	U (20,100)	10	4	80/120
	M5	4	75	U (20,100)	15	5	80/120
	M6	4	100	U (20,100)	15	5	80/120
	M7	4	100	U (20,100)	15	5	80/120
Large	L1	6	160	U (40,150)	40	6	150/200
	L2	6	160	U (40,150)	45	6	150/200
	L3	6	240	U (40,150)	50	6	150/200
	L4	6	240	U (40,150)	55	6	150/200
	L5	6	360	U (40,150)	60	6	150/200
	L6	6	360	U (40,150)	65	6	150/200
	CS1	8	500	U (30,300)	250	6	300/400

In Table 3, the first column shows the size instances, including small (instances S1–S5), medium (instances M1–M7), large size instances (instances L1–L6), and the case study (instance CS1). The second column shows instance names, and the third to sixth columns show the characteristics of the problems in which the number of depots, the number of customers, size of demand (egg trays) (egg tray carries 36 eggs), the number of vehicles of each depot, number of product types, and capacity size of the vehicles. Note that the size of demand is randomly generated using a uniform distribution U (min, max). The vehicles' capacity size, including two fleet sizes, is the small fleet size (S) or large fleet size (L). From Table 3, the number of depots is divided into three groups, which are small, medium, and large-scale test instances. There are two to three, four and six for small, medium, and large-sized instances, respectively. Sizes of demand (egg trays) for small, medium, and large-sized test instances are between 10 to 60 egg trays, 20 to 100 egg trays, and 40 to 150 egg trays. The number of customers for small, medium, and large-sized test instances is 12–20, 50–100, and 160–360 nodes, respectively. The number of vehicles in each depot for small, medium, and large-sized test instances are 4, 8–15, 40–65 vehicles, respectively. The number of product types for small, medium, and large-sized test instances are 2–4, 4–5, 6 product types, respectively. The capacity sizes of vehicles for small, medium, and large-sized test instances are 60/100, 80/120, and 150/200. Moreover, presently, the case study company has the customers' orders size of demand ranging from 30 to 300 for 500

customers per day. The case study company has 250 vehicles per eight depots with capacity sizes of vehicles 300/400 to transport the products. The number of product types is six.

We set the computation time as the stopping criteria of our proposed method. The limited computation time of LINGO is set to 30 min, 1440 min and 4320 min for small, medium and large-sized instances, respectively. Our proposed method used a limited computation time of 5 min, 15 min and 30 min for small, medium and large-sized instances, respectively. The traditional DE, DE combined initial solution (IDE), and IDE combined self-adaptive (ISDE) were selected for comparison with the HSADE algorithm. The parameter values used in our implementation include scaling factor (F) = 2, $C_{min} = 0.5$, $C_{max} = 1$ [32] and the population is set to 25. The DE and IDE used the DE/rand/1 (Equation (30)) mutation strategy. The self-adaptive mutation strategy is set as $T_s = 0.2 \times T_{max}$ for using the single mutation strategy DE/rand/1 when $T < T_s$. If $T \geq T_s$, then the self-adaptive mutation strategy is used to change the mutation strategies of the vectors. The current practice in egg distribution is based on a greedy algorithm.

The computational results are shown in Table 4, and the statistical test has been performed using the Wilcoxon signed-rank test shown in Table 5. The HSADE yields a better solution than all other proposed algorithms, including the current practice, DE, IDE and ISDE. The statistical test indicates that the solutions obtained from our proposed algorithm were significantly different at a p -value ≤ 0.05 compared to those obtained from the LINGO for 19 instances (Table 5, second row). However, our proposed solution for small-sized instances was not significantly different compared to the LINGO. In Table 5, the fifth column for the HSADE shows that the solutions were significantly different, compared with all algorithms.

Table 4. Computational results showing the total cost.

Instance	LINGO	Current Practice	The Proposed Method (Best)			
	Total Cost		DE	IDE	ISDE	HSADE
S1	1836	1863	1836	1836	1836	1836
S2	1880	1899	1880	1880	1880	1880
S3	1750	1820	1750	1750	1750	1750
S4	2056	2120	2056	2056	2056	2056
S5	2344	2511	2344	2344	2344	2344
M1	2740	3191	2972	2906	2939	2867
M2	5652	6628	6142	6177	6006	5897
M3	6233	7726	6976	6746	6677	6582
M4	6657	8318	7253	7078	7111	7001
M5	8315	9973	9473	9134	8970	8773
M6	15,001	17,961	16,909	16,445	16,054	15,742
M7	19,692	23,972	22,014	21,491	21,149	20,805
L1	76,850	112,064	84,311	82,263	81,618	80,645
L2	124,300	171,747	139,627	134,522	133,432	130,321
L3	161,918	213,755	182,669	175,969	173,796	171,020
L4	190,615	277,488	209,356	205,258	204,531	200,409
L5	240,685	324,516	275,091	259,837	259,487	252,696
L6	281,520	367,720	318,981	305,452	303,086	297,469
CS1	400,924	539,238	444,532	435,715	426,236	421,168

Table 5. Statistical test results of the total cost obtained from Table 4.

Method	Current Practice	DE	IDE	ISDE	HSADE
LINGO	0.0001	0.001	0.001	0.001	0.001
DE			0.001	0.001	0.001
IDE				0.002	0.001
ISDE					0.001

Additionally, the heuristic performance (HP) percentage was measured using the solution obtained from all compared algorithms by comparing it with the optimal solution obtained from the LINGO. The results show that the HSADE algorithm provided 100% HP for the small-sized instance. Based on the medium-sized instance, the range of HP of the HSADE algorithm is 94.66% to 95.85% (with an average of 95.14%), compared to the lower bound solution. Similarly to the medium-sized instances, the large-sized instances demonstrate a range of HP of 94.64% to 95.38% (with an average of 95.06%). The HP of the HSADE algorithm compared with the case study was 95.20%. The average HPs of current practice, DE, IDE, ISDE and HSADE algorithms are 82.92%, 92.48%, 94.40%, 95.11% and 96.40%, respectively (Table 6, Figure 8).

Table 6. The heuristic performance (HP) results for testing instances.

Instance	Current Practice	Heuristic Performance (HP) %			
		DE	IDE	ISDE	HSADE
S1	98.56	100.00	100.00	100.00	100.00
S2	99.00	100.00	100.00	100.00	100.00
S3	96.16	100.00	100.00	100.00	100.00
S4	96.99	100.00	100.00	100.00	100.00
S5	93.35	100.00	100.00	100.00	100.00
M1	85.87	92.20	94.29	93.23	95.58
M2	85.28	92.03	91.51	94.11	95.85
M3	80.68	89.35	92.40	93.36	94.70
M4	80.04	91.79	94.06	93.62	95.09
M5	83.38	87.78	91.04	92.70	94.78
M6	83.52	88.72	91.22	93.45	95.30
M7	82.15	89.46	91.63	93.12	94.66
L1	68.58	91.16	93.42	94.16	95.30
L2	72.38	89.03	92.41	93.16	95.38
L3	75.75	88.65	92.02	93.17	94.68
L4	68.70	91.05	92.87	93.20	95.12
L5	74.17	87.50	92.63	92.76	95.25
L6	76.56	88.26	92.17	92.89	94.64
CS1	74.36	90.20	92.02	94.07	95.20
Average	82.92	92.48	94.40	95.11	96.40

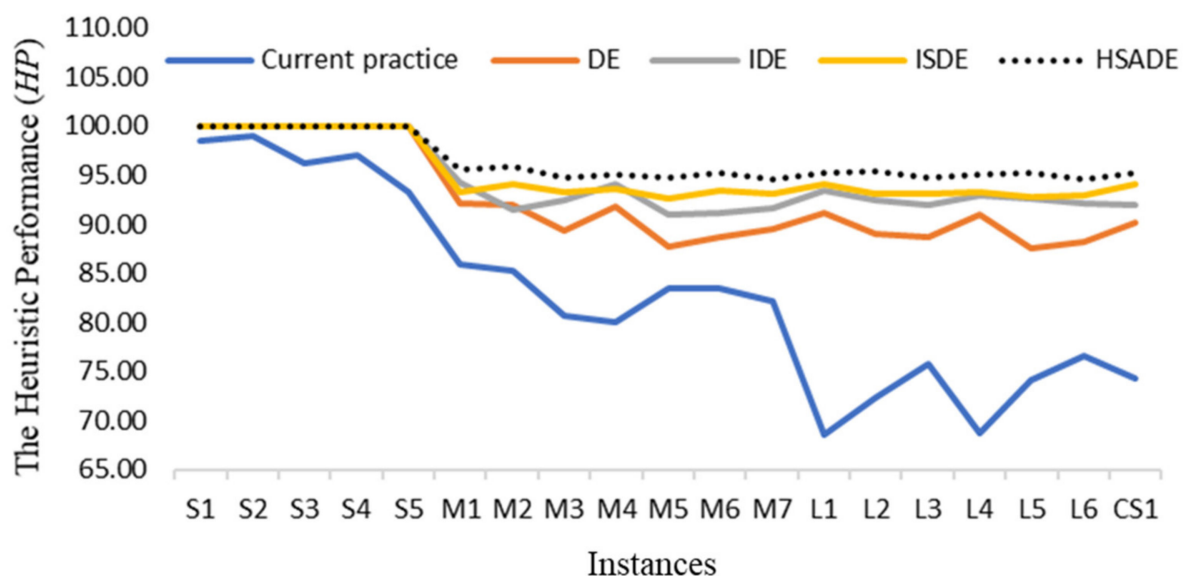


Figure 8. The heuristic performance (HP) results for testing instances.

The relative improvement (*RI*) results are shown in Table 7. The total costs of the solutions obtained from the current practice were compared with the DE, IDE, ISDE and HSADE algorithms. The results show that the HSADE algorithm outperforms all other algorithms (Figure 9). The average *RI* of DE, IDE, ISDE and HSADE algorithms is 10.55%, 12.35%, 12.99% and 14.13%, respectively; that means that the HSADE algorithm combined with the initial solution, self-adaptive mutation formula and hybrid neighborhood strategies search is more effective than DE, IDE and ISDE.

Table 7. The relative improvement (*RI*) results for testing instances.

Instance	Relative Improvement (<i>RI</i>) %			
	DE	IDE	ISDE	HSADE
S1	1.45	1.45	1.45	1.45
S2	1.01	1.01	1.01	1.01
S3	3.85	3.85	3.85	3.85
S4	3.02	3.02	3.02	3.02
S5	6.66	6.66	6.66	6.66
M1	6.87	8.94	7.90	10.16
M2	7.34	6.81	9.39	11.03
M3	9.71	12.69	13.58	14.81
M4	12.81	14.91	14.52	15.84
M5	5.02	8.42	10.06	12.04
M6	5.86	8.45	10.62	12.36
M7	8.17	10.35	11.78	13.22
L1	24.77	26.60	27.17	28.04
L2	18.71	21.68	22.31	24.13
L3	14.55	17.68	18.70	20.00
L4	24.56	26.03	26.30	27.78
L5	15.24	19.94	20.04	22.14
L6	13.26	16.94	17.58	19.11
CS1	17.57	19.20	20.96	21.90
Average	10.55	12.35	12.99	14.13

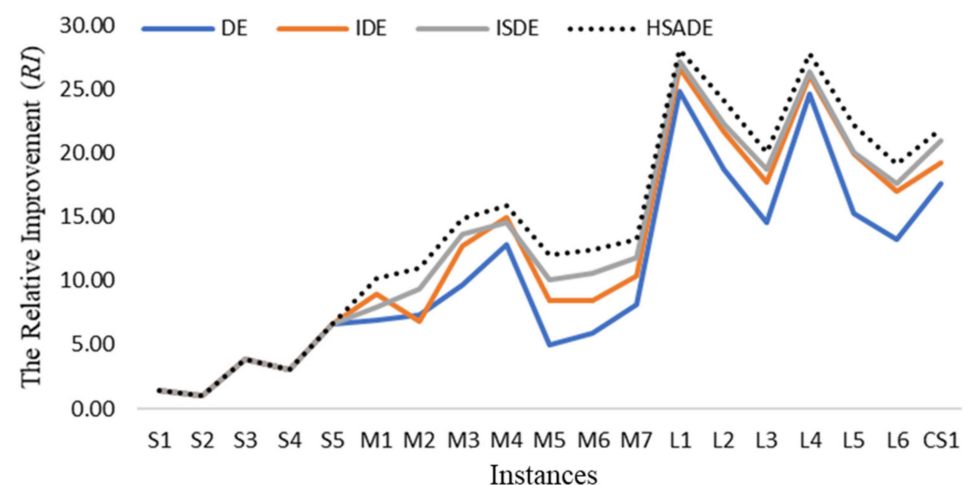


Figure 9. The relative improvement (*RI*) results for testing instances.

6. Conclusions

This paper proposes the HSADE algorithm to solve the realistic egg distribution problem in Thailand. The problem has dealt with the multi-product (egg sizes), multi-depot vehicle routing problem, with heterogeneous fleets, time window constraints and inventory restrictions to minimize the total cost. The MILP model of egg distribution in Thailand has been formalized. The differential evolution algorithm is selected as the core mechanic to develop the best algorithm to solve the large-sized test instance and the case study. The

constructive heuristics adopt the initial solution to guide the solution through a customer clustering method. The hybrid and self-adaptive differential evolution were developed. The hybrid with neighborhood structure, including k -random inter-exchange and k -random insertion by probability, k -random intra-exchange, and k -random intra-insertion, were implemented to avoid trapping in a local optimum. The self-adaptive mutation strategy was developed with a new self-adaptive algorithm to change the mutation formula. The optimal solution for a small-sized instance and the lower bound solution for medium and large-sized problems are the criterion to measure the Heuristic Performance (*HP*) with our proposed algorithm. The current practice of egg distribution in Thailand has been developed to measure relative improvement (*RI*) to demonstrate the ability to improve our proposed algorithm from the current practice. For the computation result of our HSADE algorithm applied to egg distribution, LINGO solves the MILP model with a prespecified time limit for 19 instances. The test results show that our proposed algorithm can provide the optimal solution; the HSADE algorithm significantly outperforms LINGO both in solution quality and CPU time. The heuristic performance (*HP*) showed that the HSADE algorithm obtained a near-optimal solution ranging from 94.64% to 95.85% and the current practice obtained a solution ranging from 68.58% to 85.87% for medium and large-sized instances. The average heuristic performance (*HP*) of HSADE is 96.40% for all instances. Additionally, the relative improvement (*RI*) has been compared to the total cost of the solution obtained from the current practice with DE, IDE, ISDE, and HSADE algorithms. Those algorithms had average relative improvements equal to 10.55%, 12.35%, 12.99%, and 14.13%, respectively.

The statistical results obtained using the Wilcoxon signed-rank test showed that compared the total cost of LINGO and our proposed algorithm. As a result, our proposed algorithm obtains solutions that are not significantly different from the optimal solution for the small-sized instances. Additionally, the statistical test for large-sized instances showed that the proposed HSADE algorithm was significantly better than the DE, IDE, and ISDE algorithms, respectively. Furthermore, the solutions of IDE were significantly different from the solutions of ISDE which DE combined initial solution based on nearest neighbor heuristic (IDE) and IDE combined self-adaptive strategy (ISDE). This implies that the self-adaptive mutation strategy did significantly improve the IDE solution. Therefore, the HSADE algorithm in this paper will assist the egg distribution industry in reducing total transportation costs.

The HSADE algorithm had solutions that yield better results than each other algorithm for all test instances because the initial solution, mutation strategy adaptation, and the neighborhood search structure can enhance the exploitation capability of DE. In the larger-size instances and the case study, the HSADE algorithm obtained the solution near the lower bound solution from LINGO. Additionally, the HSADE algorithm took only 30 min instead of 4320 min. Therefore, within the time constraints of the egg industry in Thailand, Thai entrepreneurs can apply the HSADE algorithm to improve egg distribution planning transportation. The mathematical and proposed algorithms in this paper should prove beneficial to similar agriculture logistics in Thailand and worldwide.

Our future work will consider a variant of the egg distribution problem in Thailand that considers horizontal planning, such as a multiple-period, periodic, multiple-time window, mixed-time window, and so forth. Additionally, although our proposed mathematical model is efficient, research to determine the solutions to the problem should be carried out by using different mathematical models to compare the strengths of various approaches in solving the problems of this nature.

Author Contributions: Conceptualization, K.M. and K.S.; methodology, K.M.; software, K.M.; validation, K.M., K.W. and K.N.; formal analysis, K.M.; data curation, K.M.; writing—original draft preparation, K.M. and K.W.; writing—review and editing, K.M., K.S., K.W. and K.N.; supervision, K.M. and K.S.; project administration, K.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received Faculty of Engineering, Khon Kaen University grant numbers SMI.KKU 64006, Thailand.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: This work is supported by the Research Unit on System Modeling for Industry (Grant No. SMI.KKU 64006), Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University, Thailand.

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

Abbreviations

N	Number of customers;
D	Number of depots;
NNH	Nearest neighbor heuristic;
Nh	Neighborhood search strategies;
T	Computation time;
Ts	Computation time set for self-adaptive mutation strategy;
U	Uniform distribution;
HP	Heuristic performance;
RI	Relative improvement.

References

- Office of Agricultural Economics. The Proportion of Exports of Agricultural Products and Food. Available online: https://public.flourish.studio/visualisation/3023843/?utm_source=showcase&utm_campaign=visualisation/3023843 (accessed on 15 April 2021).
- National Bureau of Agricultural Commodity and Food Standards. Trade Trends in Agricultural Products in 2020. Available online: <https://bit.ly/3yL6VD4> (accessed on 20 April 2021).
- Office of the National Economic and Social Development Council. *Thailand's Logistics Report 2019*. Available online: https://www.nesdc.go.th/ewt_dl_link.php?nid=10791 (accessed on 23 September 2021).
- Battarra, M.; Cordeau, J.F.; Iori, M. Chapter 6: Pickup-and-Delivery Problems for Goods Transportation. In *Vehicle Routing: Problems, Methods, and Applications*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2014; pp. 161–191. [CrossRef]
- Landrieu, A.; Mati, Y.; Binder, Z. A tabu search heuristic for the single vehicle pickup and delivery problem with time windows. *J. Intell. Manuf.* **2001**, *12*, 497–508. [CrossRef]
- Ai, T.J.; Kachitvichyanukul, V. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Comput. Oper. Res.* **2009**, *36*, 1693–1702. [CrossRef]
- Wang, X.; Golden, B.; Wasil, E.; Zhang, R. The min–max split delivery multi-depot vehicle routing problem with minimum service time requirement. *Comput. Oper. Res.* **2016**, *71*, 110–126. [CrossRef]
- Kachitvichyanukul, V.; Sombuntham, P.; Kunnappadeelert, S. Two solution representations for solving multi-depot vehicle routing problem with multiple pickup and delivery requests via PSO. *Comput. Ind. Eng.* **2015**, *89*, 125–136. [CrossRef]
- Mancini, S. A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: Formulation and adaptive large neighborhood search based matheuristic. *Transp. Res. Part C Emerg. Technol.* **2016**, *70*, 100–112. [CrossRef]
- Hasani-Goodarzi, A.; Tavakkoli-Moghaddam, R. Capacitated vehicle routing problem for multi-product cross-docking with split deliveries and pickups. *Procedia Soc. Behav. Sci.* **2012**, *62*, 1360–1365. [CrossRef]
- Liu, R.; Jiang, Z.; Fung, R.Y.; Chen, F.; Liu, X. Two-phase heuristic algorithms for full truckloads multi-depot capacitated vehicle routing problem in carrier collaboration. *Comput. Oper. Res.* **2010**, *37*, 950–959. [CrossRef]
- Lalla-Ruiz, E.; Expósito-Izquierdo, C.; Taheripour, S.; Voß, S. An improved formulation for the multi-depot open vehicle routing problem. *OR Spectrum* **2016**, *38*, 175–187. [CrossRef]
- Salhi, S.; Imran, A.; Wassan, N.A. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Comput. Oper. Res.* **2014**, *52*, 315–325. [CrossRef]
- Chen, Q.; Li, K.; Liu, Z. Model and algorithm for an unpaired pickup and delivery vehicle routing problem with split loads. *Transp. Res. Part E Logist. Transp. Rev.* **2014**, *69*, 218–235. [CrossRef]
- Qiu, Y.; Wang, L.; Xu, X.; Fang, X.; Pardalos, P.M. Formulations and branch-and-cut algorithms for multi-product multi-vehicle production routing problems with startup cost. *Expert Syst. Appl.* **2018**, *98*, 1–10. [CrossRef]

16. Bouanane, K.; Benadada, Y.; Bencheikh, G. Multi-depots vehicle routing problem with simultaneous delivery and pickup and inventory restrictions: Formulation and resolution. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 110–120. [[CrossRef](#)]
17. Seydim, A.C.; Dawson, P.L. Packaging effects on shell egg breakage rates during simulated transportation. *Poult. Sci.* **1999**, *78*, 148–151. [[CrossRef](#)]
18. Boonmee, A.; Sethanan, K.; Arnonkijpanich, B.; Theerakulpisut, S. Minimizing the total cost of hen allocation to poultry farms using hybrid growing neural gas approach. *Comput. Electron. Agric.* **2015**, *110*, 27–35. [[CrossRef](#)]
19. Dechampa, D.; Tanwanichkul, L.; Sethanan, K.; Pitakaso, R. A differential evolution algorithm for the capacitated VRP with flexibility of mixing pickup and delivery services and the maximum duration of a route in poultry industry. *J. Intell. Manuf.* **2017**, *28*, 1357–1376. [[CrossRef](#)]
20. Hisasaga, C.; Griffin, S.E.; Tarrant, K.J. Survey of egg quality in commercially available table eggs. *Poult. Sci.* **2020**, *99*, 7202–7206. [[CrossRef](#)]
21. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
22. Pitakaso, R.; Sethanan, K. Modified differential evolution algorithm for simple assembly line balancing with a limit on the number of machine types. *Eng. Optim.* **2016**, *48*, 253–271. [[CrossRef](#)]
23. Supattananon, N.; Akararungruangkul, R. Modified Differential Evolution Algorithm for a Transportation Software Application. *J. Open Innov. Technol. Mark. Complex.* **2019**, *5*, 84. [[CrossRef](#)]
24. Kunnapapdeelert, S.; Kachitvichyanukul, V. New enhanced differential evolution algorithms for solving multi-depot vehicle routing problem with multiple pickup and delivery requests. *Int. J. Serv. Oper. Manag.* **2018**, *31*, 370–395. [[CrossRef](#)]
25. Akararungruangkul, R.; Kaewman, S. Modified differential evolution algorithm solving the special case of location routing problem. *Math. Comput. Appl.* **2018**, *23*, 34. [[CrossRef](#)]
26. Fan, Q.; Wang, W.; Yan, X. Differential evolution algorithm with strategy adaptation and knowledge-based control parameters. *Artif. Intell. Rev.* **2019**, *51*, 219–253. [[CrossRef](#)]
27. Zhang, M.; Pratap, S.; Zhao, Z.; Prajapati, D.; Huang, G.Q. Forward and reverse logistics vehicle routing problems with time horizons in B2C e-commerce logistics. *Int. J. Prod. Res.* **2020**, *59*, 6291–6310. [[CrossRef](#)]
28. Marinaki, M.; Marinakis, Y.; Matsatsinis, N. A Novel Solution Encoding in the Differential Evolution Algorithm for Optimizing Tourist Trip Design Problems. In *Learning and Intelligent Optimization; Proceedings of the 13th International Conference, LION 13, Chania, Crete, Greece, 27–31 May 2019; Revised Selected Papers*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 253–267. [[CrossRef](#)]
29. Sethanan, K.; Jamrus, T. Hybrid differential evolution algorithm and genetic operator for multi-trip vehicle routing problem with backhauls and heterogeneous fleet in the beverage logistics industry. *Comput. Ind. Eng.* **2020**, *146*, 106571. [[CrossRef](#)]
30. Reinelt, G. *The Traveling Salesman: Computational Solutions for TSP Applications*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 840.
31. Fan, Q.; Zhang, Y. Self-adaptive differential evolution algorithm with crossover strategies adaptation and its application in parameter estimation. *Chemom. Intell. Lab. Syst.* **2016**, *151*, 164–171. [[CrossRef](#)]
32. Erbao, C.; Mingyong, L. A hybrid differential evolution algorithm to vehicle routing problem with fuzzy demands. *J. Comput. Appl. Math.* **2009**, *231*, 302–310. [[CrossRef](#)]