Article

Performance Comparison between Particle Swarm Optimization and Differential Evolution Algorithms for Postman Delivery Routing Problem

Warisa Wisittipanich ¹ , Khamphe Phoungthong ² , Chanin Srisuwannapa ³, Adirek Baisukhan ³ and Nuttachat Wisittipanit ^{3,4,*}

- ¹ Center of Healthcare Engineering System, Department of Industrial Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand; warisa@eng.cmu.ac.th
- ² Environmental Assessment and Technology for Hazardous Waste Management Research Center, Faculty of Environmental Management, Prince of Songkla University, Songkhla 90112, Thailand; khamphe.p@psu.ac.th
- ³ Department of Materials Engineering, School of Science, Mae Fah Luang University, Chiang Rai 57100, Thailand; chanin.sri@mfu.ac.th (C.S.); adirek.bai@mfu.ac.th (A.B.)
- ⁴ Center of Innovative Materials for Sustainability (iMatS), Mae Fah Luang University, Chiang Rai 57100, Thailand
- * Correspondence: nuttachat.wis@mfu.ac.th

Citation: Wisittipanich, W.; Phoungthong, K.; Srisuwannapa, C.; Baisukhan, A.; Wisittipanit, N. Performance Comparison between Particle Swarm Optimization and Differential Evolution Algorithms for Postman Delivery Routing Problem. *Appl. Sci.* **2021**, *11*, 2703. <https://doi.org/10.3390/app11062703>

Academic Editor: Farouk Yalaoui

Received: 18 February 2021

Accepted: 14 March 2021

Published: 17 March 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/>).

Abstract: Generally, transportation costs account for approximately half of the total operation expenses of a logistics firm. Therefore, any effort to optimize the planning of vehicle routing would be substantially beneficial to the company. This study focuses on a postman delivery routing problem of the Chiang Rai post office, located in the Chiang Rai province of Thailand. In this study, two metaheuristic methods—particle swarm optimization (PSO) and differential evolution (DE)—were applied with particular solution representation to find delivery routings with minimum travel distances. The performances of PSO and DE were compared along with those from current practices. The results showed that PSO and DE clearly outperformed the actual routing of the current practices in all the operational days examined. Moreover, DE performances were notably superior to those of PSO.

Keywords: postman delivery; vehicle routing problem; particle swarm optimization algorithm; differential evolution algorithm

1. Introduction

The topic of global warming has been heavily addressed in recent decades due to increasing greenhouse gases [1]. Unsurprisingly, the transportation industry globally accounts for a large portion (mainly CO₂ emission), amounting to approximately 20% [2]. Since our economics and life wellness deeply depend on the industry, decent mitigation efforts are needed to enhance its operation efficiency. For logistics organizations, with transportation costs being their largest expenditure, an improved operation not only reduce the environmentally harmful gas, but also reduces the operational costs such as fuel, manpower and vehicle depreciation, significantly augmenting profits in the process [3]. As a result, a large logistics firm that handles thousands of transport vehicles per operational day should absolutely adopt an effective transportation management and scheduling system. The post office is one of the logistics establishments that has to deal with such management, involving area assignments for delivery vehicles and the type and number of vehicles, as well as delivery time windows. Some even engage in detailed routing of each vehicle, trying to determine the most optimized ways to deliver parcels to customers [4]. Such a challenge is essentially called the vehicle routing problem (VRP). However, most post offices do not feel the need to find solutions to the VRP and simply rely on the experience of their delivery drivers, which might be because of the high costs and complications

involved in deploying such systems. Nevertheless, relying on skills and experience alone might not be suitable for a problem with exceptionally high complexities.

The VRP is one of the optimization problems that seeks optimized routing for vehicles that traverse either to deliver or pickup goods. An approach to finding a solution to the VRP might be to determine a set of routes having a minimized total distance traveled by all the vehicles involved [5]. Other VRP variations include enforcing the total delivery time to be within a certain time window [6], limiting vehicle capacity [7], having stochastic customer demands [8] or allowing customers to select their delivery options [9]. Generally, the problem sizes (such as number of vehicles and customer nodes) of the VRP that can be solved for exact optimized solutions by mathematical methods are limited [10], due to the inherent complexity of the VRP. However, actual VRP applications usually have large problem sizes, containing approximately hundreds or thousands of vehicles and customers (e.g., a beverage firm that has to deliver millions of bottles per day to a large number of depots and small businesses). Therefore, solving for optimized solutions by mathematical procedures alone is impractical for most applications. As a result, heuristic and metaheuristic optimization algorithms are usually selected as a pragmatic method to solve the VRP, since they can obtain feasible solutions relatively fast and the quality of those solutions is decent enough for real uses [11].

This study investigated the postman operations of the Chiang Rai post office, which is a logistics establishment that does not apply any VRP optimization techniques, and its delivery drivers have to rely on their own expertise and plans in order to traverse between customer locations. The routing of two delivery vehicles was tracked, in which they dispatched parcels to customers in their responsible areas for 50 operational days. The routing information consisted of traveled paths, distance covered and the actual geographic coordinates of customer locations which the vehicles visited each day. Then, we performed the routing optimization procedures based on those customer coordinates, using the metaheuristics algorithms for all the operational days of those delivery vehicles.

Since the delivery routes depended on the skills and experience in the area of their drivers, those routes might not be well optimized. Moreover, if new drivers were to replace the previous ones, they would have to learn anew and gain enough experience to better traverse the area. Thus, the efficiency of the delivery process was not consistent. As such, this research paper was focused on providing a smart routing decision, solving for VRP solutions such that the delivery performance could improve remarkably and be more consistent. Two metaheuristic algorithms were applied—particle swarm optimization (PSO) and differential evolution (DE)—and integrated with local search techniques. The algorithms sought to discover the most optimized routing those two delivery vehicles could take so that the total travel distance could be minimized. The solutions from the two algorithms, including those from current practices, had their performances compared in terms of the total distance covered.

The rest of the article is organized as follows. Section 2 provides literature reviews related to the VRP and the algorithms used in the study. The detailed dataset, mathematical model, algorithm frameworks and solution representation are given in Section 3. Finally, the experimental results and conclusion are provided in Sections 4 and 5, respectively.

2. Literature Review

The VRP, a nondeterministic polynomial time hardness (NP-hardness) problem [12], refers to an optimization problem related to routing management, first devised by Dantzig and Ramser [5] in 1959. It started as a simple problem of determining the optimal routes for a fleet of vehicles dispatching from a single terminal to multiple variants, including ones that limit vehicle capacity and delivery time, have multiple terminals or allow the customer to select a delivery option. Obtaining the exact optimal solution for the VRP is possible if the customer nodes are not too large for a mathematical solver. However, real VRP applications tend to have a substantial number of customer nodes. Therefore, heuristic methods are often chosen to solve the VRP [13].

There are several studies that focused on the VRP of post office deliveries with different strategies, objectives and constraints. Meira et al. [14] conducted an experiment to solve the multi-objective VRP with only one constraint—the route distance—using a large number of geographical delivery locations (~30,000 points) in Autur Nogueira, Brazil. Dumez et al. [9] tried to solve the VRP for a post office in which each customer could select a choice of vehicle routing problem delivery options (VRPDOs) with varying places and times, such as receiving parcels at home in the evening or at work during office hours. In addition, the VRP with a time window (VRPTW) was investigated by Niroomand et al. [15], who developed an optimization model for a network of post offices with aims to minimize the total operational cost within time constraints (i.e., office working hours).

Various algorithms were employed to solve the VRP and its variants. Most are meta-heuristic, including differential evolution (DE) and particle swarm optimization (PSO). Other metaheuristic algorithms such as the large neighborhood search (LNS) [16] and hybrid intelligent algorithms [17] were also adopted for the VRP. DE is a population-based optimization method that is similar to other evolutionary algorithms in terms of conventional approaches (e.g., mutation and cross-over operations). However, it utilizes a less stochastic and greedier solving strategy [18]. DE has been successfully applied to many VRP studies. Kromer et al. [19] employed DE to find solutions to the stochastic VRP with simultaneous pickup and delivery by using actual bus service data in Anbessa, Ethiopia. Their objective was to minimize both the total travel distance and number of vehicles, and the algorithm yielded superior performance compared with that of a traditional savings algorithm. Sethanan [20] utilized DE (hybridized with a genetic operator) for a multi-trip VRP with backhauls and a heterogeneous fleet (MTVRPB). They devised an optimization method specifically for a company that needed to retrieve glass bottles (soft drinks) back from clients after delivery with a single objective (i.e., minimizing the total distance traveled). Erbao et al. [18] used both DE and genetic algorithms to find an optimal solution to the VRP with simultaneous delivery and pick-up and time windows (VRP-SDPTW). They built a mathematical model using mixed integer programming for their proposed VRP variant and yielded satisfying optimized results. Xing et al. [21] applied a hybrid discrete differential evolution (HDDE) to the split delivery VRP (SDVRP). The authors focused on vehicle routing management to deliver large supplies of hospital items split between multiple unmanned transport vehicles—a suitable solution in the pandemic era—with the objective of minimizing the total distance traveled, and the proposed HDDE algorithm produced superior performance compared with the forest-based tabu search (FBTS) and genetic algorithm (GA) in many problem instances. There are also studies that applied DE and local search to solve the VRP. For instance, Song and Don [22] proposed a DE algorithm integrated with local search to solve the capacitated VRP. They employed two strategies for the local search—two-random swap and two-opt testing of the combined method with the benchmark instances—and demonstrated that such a hybrid method performed better than other existing DE algorithms. Jun and Jian [23] modified DE further by using greedy subtour crossover in the mutation operation and modified ordered crossover in the crossover operation. In addition, local search with only the two-opt strategy was used in the decoding process.

In addition to DE, PSO was also used frequently in the VRP field. Moghaddam et al. [24] developed an advanced PSO to solve a VRP variant where the demands of customers were uncertain and demand distribution was undetermined, a special form of capacitated VRP (CVRP). They devised a new decoding method for the PSO and were able to obtain superior solutions compared with those from well-known studies using the same test instances. Another variant of the capacitated VRP, called the capacitated location-routing problem (CLRP), was also studied by Peng et al. [25], in which PSO was combined with clustering and local search techniques to solve the problem. Marinakis et al. [26] proposed the use of multi-adaptive PSO (MAPSO) with three different strategies to solve the VRPTW, where each strategy had a certain role in achieving optimal solutions faster, yielding promising results compared with those of other PSO variants. Chen and Shi [27] used

hybrid PSO (HPSO) combined with simulated annealing (SA) to determine solutions for the multi-compartment VRP with a time window (MCVRPTW), where each vehicle possessed multiple compartments, each one containing goods with the same characteristics. They tested the algorithm with benchmark instances, and the results showed that HPSO had better performance than normal PSO (PSO without SA). Chen et al. [28] performed two types of PSO, namely discrete PSO (DPSO) and neural-like DPSO (NDPSO), on the periodic vehicle routing problem (PVRP). The PVRP deals with the tasks of finding an ideal customer service mode and determining optimal vehicle routing in accordance with the found mode. Sedighizadeh and Mazaheripour [29] combined PSO and the artificial bee colony (ABC) algorithm to solve the VRP, which converged faster than individual algorithms (PSO and ABC) and also yielded better solutions. Lastly, Yanwei et al. [30] employed PSO with self-adaptive inertia weight to solve the open VRP with time-dependent travel time (OVRPTD), a special VRP variant where vehicles do not have to return to the depot after finishing a delivery job and travel time is time-dependent, producing better solutions compared to those of PSO and the nearest-neighboring search (NNS) algorithm.

3. Materials and Methods

Details on the post office operation and how to obtain the necessary dataset and prepare the inputs for the study are presented in this section. Additionally, the mathematical model, algorithmic procedures of PSO and DE and process workflow for the postman delivery problem are included here.

3.1. Dataset

The dataset used in this study was obtained from the delivery operations belonging to the Chiang Rai post office in the Chiang Rai province of Thailand. The operations mainly involved postal delivery service to residential and commercial locations. The Chiang Rai post office holds delivery duties for customers in the area of the Viang and Rob-Viang districts, which are clustered into six delivery zones. Detailed data of the delivery operation were collected for 50 days, tracking two delivery vehicles equipped with Global Positioning System (GPS)-enabled android devices with location-tracking applications installed. The average number of customer locations to be delivered to per day by both vehicles was 97.22. Those two vehicles were motorcycles, each with large parcel pouches strapped on both sides, and both had duties to deliver parcels to customers in specific areas under the Chiang Rai post office's responsibility.

The GPS devices recorded routing data, including all customer locations and where the vehicles visited, in geographic coordinate form. Then, for each day of delivery operations, the total travel distance of both vehicles was extracted. Moreover, the stored latitude and longitude coordinates, which corresponded to the post office and customer locations, were used to construct a distance matrix of $n \times n$ dimensions, where n was the number of all location coordinates. Each cell of the distance matrix represented the smallest distance between a pair of location coordinates, and this distance value was determined by a python script using the `googlemaps` module from the Google Maps Application Programming Interface (API) [31]. The obtained matrix might not have been the symmetric type, since the shortest route from one coordinate to another may not have been the same the other way around, taking into account that one-way traffic could be enforced in some routes. In total, 50 matrices were constructed, each one representing one day of delivery operations. These matrices were essential inputs to the optimization algorithms, acting as a lookup distance table.

3.2. Mathematical Model

As mentioned earlier, this study addresses the delivery routing problem of the Chiang Rai post office. The problem is classified as a single-depot VRP, in which all vehicles that depart from the post office must return to the post office. For each operational day, a vehicle must decide on the delivery route for a set of different customers. All customers must be

visited within an operational day with no specific time window. The traffic conditions were not considered in this case, due to the fact that there was little traffic in the investigated area. Thus, the delivery time and cost were primarily based on the total travel distance, and the objective of this problem was to find optimal routes with the lowest total travel distance. In addition, the capacity of the parcel pouches was assumed to be infinite, since the postman used two large pouches that could always contain all parcels, and there has been no case where the capacity was insufficient.

In this study, the postman delivery routing problem is represented as an integer programming (IP) model. The parameters and decision variables used in formulating the model are defined as follows:

Indices

- i, j : Post office and customer locations ($i, j = 1$ for the post office location; $i, j = 2, 3, \dots, n$ for customer locations).
 k : Vehicle ($k = 1, 2, \dots, m$).

Parameters

- $d_{i,j}$: Distance from i to j
 U_i, U_j : Constant number used to avoid a subtour.
 N : Number of locations.
 M : Number of vehicles.

Decision Variable

- $x_{i,j,k}$: If vehicle k travels from i to j or 0 otherwise.

The minimization of the total travel distance is expressed as

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^m d_{ij} * x_{ijk} \quad (1)$$

The constraints are as follows:

$$\sum_{i=1}^n \sum_{k=1}^m x_{ijk} = 1 \quad \forall j; j = \{2, 3, \dots, n\} \quad (2)$$

$$\sum_{j=1}^n \sum_{k=1}^m x_{ijk} = 1 \quad \forall i; i = \{2, 3, \dots, n\} \quad (3)$$

$$\sum_{i=1}^n x_{ipk} - \sum_{j=1}^n x_{pjk} = 0 \quad \forall_p \forall_k; p = \{1, 2, \dots, n\}, k = \{1, 2, \dots, m\} \quad (4)$$

$$\sum_{i=2}^n x_{i1k} \leq 1 \quad \forall_k; k = \{1, 2, \dots, m\} \quad (5)$$

$$\sum_{j=2}^n x_{1jk} \leq 1 \quad \forall_k; k = \{1, 2, \dots, m\} \quad (6)$$

$$U_i - U_j + N \sum_{k=1}^m x_{ijk} \leq N - 1 \quad \forall_i \forall_j; i, j = \{2, 3, \dots, n\} \text{ and } i \neq j \quad (7)$$

$$x_{ijk} \in \{0, 1\} \quad \forall_i \forall_j \forall_k; i, j = \{1, 2, \dots, n\}, k = \{1, 2, \dots, m\} \quad (8)$$

The objective of the model was to minimize the total travel distance as shown in Equation (1). Equations (2) and (3) ensured that each customer was visited once and by only one vehicle. The constraint in Equation (4) guaranteed the route continuity of all vehicles. Equations (5) and (6) ensured that each vehicle could take only one route, and it needed to depart from and return to the post office. Equation (7) was used to eliminate subtours. Equation (8) states that a decision variable is a binary number.

3.3. Application of PSO and DE to the Postman Delivery Routing Problem

As mentioned earlier, the postman delivery routing problem is NP-hard. Thus, solution methods based on metaheuristic approaches are more preferable than the traditional exact methods for finding solutions in real-world practices. This section presents the application of two effective metaheuristic approaches—particle swarm optimization (PSO) and differential evolution (DE)—to solve the postman delivery routing problem. A solution representation with an encoding and decoding scheme is proposed to transform a metaheuristic solution into a practical solution. The procedures of these algorithms are described below.

3.3.1. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO), proposed by Kennedy and Eberhart in 1995, is a population-based random search technique that imitates the behavior of fish schooling or birds flocking. In PSO, a solution is represented as a particle, and the population of solutions is called a swarm of NP particles. Each particle consists of two main attributes: The position and the velocity. The evolutionary concept of PSO is that each particle learns from the cognitive knowledge of its experiences (personal best, or $pbest$) and the social knowledge of the swarm (global best, or $gbest$) to guide itself to the better position. A particle moves to a new position using the updated velocity, which is adjusted according to the experiences of a particle. In particular, the updated velocity of a particle depends on three main control parameters: The inertia weight, personal best coefficient (cp) and global best coefficient (cg). Once a new position is obtained, the best position of each particle and the best position of the swarm are then updated as needed. These processes are repeated until a stopping criterion is met. Figure 1 shows the evolutionary procedure of the PSO.

Figure 1. Evolutionary procedure of the particle swarm optimization (PSO) algorithm.

3.3.2. Differential Evolution (DE)

Differential evolution (DE), proposed by Storn and Price in 1995, is one of the evolutionary algorithms (EAs) used for global optimization over a continuous search space. Similar to other population-based random search techniques, DE starts by randomly generating an initial population of NV vectors. A solution in DE is represented by the D -dimensional vector, and each value in the D -dimensional space is represented as a real number. The crucial idea of DE is its mechanism for generating trial vectors by mutation and crossover operations with the use of a couple control variables: The scale factor (F)

and the crossover rate (Cr). In this study, the classic but efficient mutation operation of DE/rand/1 was applied to generate a mutant vector by adding a weighted difference (F) between two randomly selected vectors to the third randomly selected vector. DE is able to generate better diverse solutions, since the best solution in the population does not exert any influence on the other solutions in the population. In addition, the mutant vector is always a solution that is not from the original population; therefore, the crossover operation in DE is always between a solution from the population and a newly generated one [32]. Then, the selection operation or replacement of an individual vector occurs only if the trial vector outperforms its corresponding current vector. As shown in Figure 2, the evolution procedure of a DE population continues through repeated cycles of three main operations—mutation, crossover and selection—until a stopping criterion is met.

Figure 2. Evolutionary procedure of the differential evolution (DE) algorithm.

3.3.3. Solution Representation

In order to apply PSO and DE for solving a combinatorial problem like the postman delivery routing problem, solution representation or solution mapping was required to transform the real number of the D -dimensional space into practical solutions for vehicle routing. Figure 3 presents the procedure of solution representation with the encoding and decoding scheme.

The encoding process aimed to represent a solution as a string of dimensions. In this study, the postman delivery routing problem with $n - 1$ customers (noting that n is the number of locations, including the post office) and m vehicles was encoded as a string of D dimensions. The number of dimensions (D) was set to be equal to the total number of customers, and the dimension index stood for the customer index. Each value in a dimension was initially filled with a uniform random number in the range $[0, 1]$. Next, the process of decoding was divided into two stages: (1) the assignment of customers to vehicles and (2) the sequencing or routing for each vehicle. In the assignment stage,

the range (max–min) of the dimension value was split by the number of vehicles as in Equation (9):

$$\text{Split} = (\text{Max} - \text{Min}) / m \quad (9)$$

where m is the number of vehicles, Min is the minimum value in the dimensions and Max is the maximum value in the dimensions.

Figure 3. Solution representation procedures.

According to Equation (9), the range of the dimension value was equally partitioned into m subsections corresponding to m vehicles. The span of each subsection is determined in Figure 4. It is noted that a subsection index denoted a vehicle index.

Figure 4. The span of m subsections.

Then, a customer was assigned to a vehicle, considering its customer dimension value with the corresponding subsection. For example, if the dimension value of customer 1 fell into Section 2, that customer was assigned to vehicle 2 correspondingly. This process continued until all customers were considered. Once the assignment of customers to vehicles was decided, a heuristic called the nearest-neighbor search (NNS) was applied

to determine the delivery routing of each vehicle. Thus, for each vehicle, the next visited customer was the one with the nearest distance to the current location. The completed route of a vehicle was obtained when all customers assigned to that vehicle were considered. It is noted that all vehicles needed to depart from and return to the post office.

To illustrate the process of solution representation, let us consider an example of the postman delivery routing problem with six customers and two vehicles. A distance matrix between the post office and customers is shown in Figure 5. The sample matrix in this example is symmetric; however, a matrix constructed from the actual operation might be asymmetric, depending on the locations where the delivery process takes place.

Distance	Post office	1	2	3	4	5	6
Post office	-	193	71	79	165	43	54
1	193	-	198	44	168	173	112
2	71	198	-	66	80	67	123
3	79	44	66	-	176	21	78
4	165	168	80	176	-	112	52
5	43	173	67	21	112	-	79
6	54	112	123	78	52	79	-

Figure 5. A distance matrix for an example of the postman delivery routing problem with six customers and two vehicles.

In the encoding process, the number of dimensions was set to six, which was equal to the number of customers. Each value in a dimension was initially generated with a uniform random number between 0 and 1, as shown in Figure 6.

dimension	1	2	3	4	5	6
	0.51	0.25	0.72	0.43	0.87	0.34

Figure 6. Encoding example of the postman delivery routing problem with six customers and two vehicles.

Figure 7 illustrates the decoding procedure for obtaining the routing of a vehicle. First, according to Equation (9), the range of the dimension value is divided by the number of vehicles, which is $(0.87 - 0.25)/2 = 0.31$. In this example, two subsections are obtained according to the number of vehicles. Then, the condition of the customer assignment is set. The span of sections 1 and 2 are $(0.25, 0.56)$ and $(0.56, 0.87)$, respectively. When the customer dimension value falls into the first subsection, that customer is assigned to vehicle 1; otherwise, they are assigned to vehicle 2. Thus, customers 1, 2, 4 and 6 are assigned to vehicle 1, and customers 3 and 5 are assigned to vehicle 2. After that, the routing of each vehicle is determined using NNS method. According to the distance matrix in Figure 5, vehicle 1 starts from the post office, then goes to customer 6, customer 4, customer 2 and customer 1 before returning to the post office. Vehicle 2 starts from the post office, then goes to customer 5 and customer 3 before returning to the post office. The total distance of this particular example is $557 + 143 = 720$ units.

Figure 7. Decoding procedures to transform a D -dimensional solution into a vehicle routing solution.

4. Experimental Results

Information for the experiment, including the algorithm parameters for PSO and DE and the computational results, are shown in this section. The results emphasize the performance comparison among PSO, DE and current practices. In addition, box plots comparing the results of DE and PSO in some instances are provided.

4.1. Parameter Setting of Metaheuristics

Typically, the performance of metaheuristics is influenced by the searching and solution mapping mechanisms, while the setting of parameters decides how well the solutions converge. For a fair comparison between PSO and DE, the number of function evaluations for both algorithms was equally set to 2500, with population sizes of 50 and 50 iterations. Tables 1 and 2 show the parameter settings of PSO and DE, respectively. It is noted that these settings were empirically determined from preliminary experiments.

Table 1. The setting of PSO parameters.

Parameter	Parameter Value
Swarm Size	50
Iterations	50
Inertia Weight	0.9
Personal Best Coefficient (c_p)	0.5
Global Best Coefficient (c_g)	0.3

Table 2. The setting of DE parameters.

Parameter	Parameter Value
Population Size	50
Iterations	50
Scale Factor (F)	Random between 1.5 and 2.5
Crossover rate (Cr)	Linearly Increasing from 0.2 to 0.8
Crossover Type	Binomial Crossover

4.2. Computational Results

In this study, the metaheuristics were implemented using the Python programming language [33] under PyCharm IDE [34] version 2020.3.1 with Windows 10 as the operating system. The programs ran on an Intel® Core™ i5-4690 CPU 3.50 GHz platform with 32 GB of RAM. The performances of the algorithms were evaluated using the dataset derived from a real case study of postman delivery in association with the Chiang Rai post office for a period of 50 days. Table 3 shows the VRP solutions (total travel distance) obtained from the proposed metaheuristics, which include the best, average and standard deviation of the total distance from 10 replicated runs for each instance. The average time duration to run one optimization cycle for each instance is also reported.

Table 3. Experimental results of total distance.

Instance	Current Practice	PSO			Time (s)	DE			Time (s)
		Best	Average	SD		Best	Average	SD	
1	33.64	21.38	22.09	0.49	190.56	17.64	17.98	0.34	185.12
2	29.08	18.85	19.52	0.44	125.89	15.61	16.39	0.35	135.13
3	30.07	18.97	20.83	0.72	170.04	15.06	15.81	0.42	175.41
4	36.91	20.95	22	0.56	228.81	16.19	16.44	0.2	237.17
5	34.26	19.59	20.23	0.44	177.36	14.93	15.56	0.53	175.24
6	33.77	19.17	20.3	0.65	200.5	15.26	16.16	0.5	209.27
7	37.93	21.25	22.7	0.54	232.88	17.24	18.79	0.64	250.97
8	32.72	20.48	21.57	0.59	182.46	16.71	17.63	0.41	208.32
9	27.65	16.74	17.68	0.6	159.66	12.72	13.33	0.39	164.96
10	32.37	20.97	21.69	0.33	192.24	17.34	17.88	0.46	198.62
11	37.29	21.24	22.33	0.48	226.31	16.21	16.99	0.51	234
12	38.23	21.43	22.14	0.42	214.3	17.24	17.83	0.4	221.5
13	30.50	19.99	20.57	0.3	181.79	16.19	16.75	0.34	187.85
14	32.79	18.98	19.98	0.52	187.21	14.93	15.28	0.35	193.23
15	34.78	21.11	21.94	0.56	236.78	16.37	17.3	0.56	244.41
16	32.83	20.41	21.02	0.41	182.01	16.68	16.98	0.42	184.99
17	31.49	20.99	21.63	0.38	202.57	16.39	16.79	0.47	218.21
18	32.70	20.84	21.44	0.4	217.46	16.04	16.96	0.46	214.46
19	35.23	19.26	20.16	0.5	202.23	15.32	15.78	0.25	216.41
20	36.13	23.08	23.81	0.32	229.84	18.17	18.84	0.39	240.47
21	27.44	17.89	19.05	0.56	147.86	15.11	16.04	0.42	142.17
22	30.19	18.21	19.38	0.54	160.94	15.45	16.33	0.59	155.08
23	28.68	18.26	19.85	0.82	133.99	16.37	16.97	0.34	149.74
24	33.77	20.07	20.86	0.43	210.24	15.6	16.26	0.42	218.33
25	34.83	20.93	21.52	0.3	224.59	16.56	17.06	0.36	232.34
26	35.37	19.93	21.02	0.67	217.01	15.91	16.23	0.25	224.51
27	33.39	20.88	21.4	0.33	177.07	15.17	15.91	0.57	183.18
28	38.75	20.96	21.55	0.44	199.51	16.32	16.88	0.41	206.43
29	35.04	20.46	21.64	0.59	211.91	16.47	17.16	0.53	219.39
30	30.44	18.84	19.48	0.43	170.01	14.26	15.21	0.4	176.16
31	33.03	18.04	19.39	0.48	143.94	15.94	16.45	0.4	148.96
32	30.78	19.44	20.07	0.5	158.49	14.74	15.38	0.43	150.77
33	37.21	20.29	21.64	0.59	173.11	18.3	18.61	0.2	196.03
34	31.78	19.03	20.38	0.64	171.02	16.03	16.77	0.37	186.87

Table 3. Cont.

Instance	Current Practice	PSO			Time (s)	DE			Time (s)
		Best	Average	SD		Best	Average	SD	
35	35.12	20.7	22.09	0.7	166.5	17.49	17.89	0.37	174.06
36	34.74	20.72	21.66	0.55	180.39	16.98	17.91	0.38	185.7
37	30.19	19.34	20.57	0.48	164.51	14.54	16.33	0.95	164.86
38	31.82	18.49	19.36	0.46	155.19	15.56	16.55	0.55	162.95
39	31.39	17.82	18.84	0.48	152.59	14.74	15.28	0.4	155.57
40	29.36	18.68	19.59	0.41	148.9	14.19	14.91	0.47	147.4
41	31.88	18.53	19.21	0.4	157.6	14.53	15.13	0.39	156.06
42	32.20	19.32	20.33	0.44	200.58	15.15	16.23	0.53	203.92
43	30.42	18.77	19.46	0.29	132.47	15.16	15.86	0.4	124.59
44	33.57	18.64	19.31	0.43	163.96	15.72	15.95	0.16	172.84
45	30.40	17.99	18.65	0.42	129.07	15.66	16	0.3	137.2
46	38.38	21.45	22.05	0.41	220.58	16.74	17.48	0.53	216.86
47	28.97	17.31	19.16	0.67	157.94	15.34	15.64	0.29	153.27
48	30.18	19.46	20.37	0.46	190.09	15.69	16.34	0.4	202.63
49	33.70	20.81	21.44	0.27	177.48	16.57	16.95	0.44	201.89
50	31.68	18.19	18.85	0.38	172.02	14.32	14.85	0.52	163.4

To further analyze the effectiveness of the metaheuristics for solving the postman delivery routing problem, the percentage deviation (*PD*) between solutions obtained from the metaheuristics and those from the current practices were calculated using Equation (10). The results for the *PD* are shown in Table 4:

$$PD = ((Sol_{current_practice} - Sol_{heu}) / Sol_{current_practice}) \times 100, \quad (10)$$

where $Sol_{current_practice}$ is the solution obtained from the current practice and Sol_{heu} is the solution obtained from a metaheuristic.

Table 4. Experimental results of the percentage deviation (*PD*).

Instance	Current Practice	PSO		DE	
		Best	<i>PD</i>	Best	<i>PD</i>
1	33.64	21.38	36.44	17.64	47.56
2	29.08	18.85	35.18	15.61	46.32
3	30.07	18.97	36.91	15.06	49.92
4	36.91	20.95	43.24	16.19	56.14
5	34.26	19.59	42.82	14.93	56.42
6	33.77	19.17	43.23	15.26	54.81
7	37.93	21.25	43.98	17.24	54.55
8	32.72	20.48	37.41	16.71	48.93
9	27.65	16.74	39.46	12.72	54
10	32.37	20.97	35.22	17.34	46.43
11	37.29	21.24	43.04	16.21	56.53
12	38.23	21.43	43.94	17.24	54.9
13	30.5	19.99	34.46	16.19	46.92
14	32.79	18.98	42.12	14.93	54.47
15	34.78	21.11	39.3	16.37	52.93
16	32.83	20.41	37.83	16.68	49.19
17	31.49	20.99	33.34	16.39	47.95
18	32.70	20.84	36.27	16.04	50.95
19	35.23	19.26	45.33	15.32	56.51
20	36.13	23.08	36.12	18.17	49.71
21	27.44	17.89	34.8	15.11	44.93
22	30.19	18.21	39.68	15.45	48.82
23	28.68	18.26	36.33	16.37	42.92

Table 4. Cont.

Instance	Current Practice	PSO		DE	
		Best	PD	Best	PD
24	33.77	20.07	40.57	15.6	53.81
25	34.83	20.93	39.91	16.56	52.45
26	35.37	19.93	43.65	15.91	55.02
27	33.39	20.88	37.47	15.17	54.57
28	38.75	20.96	45.91	16.32	57.88
29	35.04	20.46	41.61	16.47	53
30	30.44	18.84	38.11	14.26	53.15
31	33.03	18.04	45.38	15.94	51.74
32	30.78	19.44	36.84	14.74	52.11
33	37.21	20.29	45.47	18.3	50.82
34	31.78	19.03	40.12	16.03	49.56
35	35.12	20.7	41.06	17.49	50.2
36	34.74	20.72	40.36	16.98	51.12
37	30.19	19.34	35.94	14.54	51.84
38	31.82	18.49	41.89	15.56	51.1
39	31.39	17.82	43.23	14.74	53.04
40	29.36	18.68	36.38	14.19	51.67
41	31.88	18.53	41.88	14.53	54.42
42	32.20	19.32	40	15.15	52.95
43	30.42	18.77	38.3	15.16	50.16
44	33.57	18.64	44.47	15.72	53.17
45	30.40	17.99	40.82	15.66	48.49
46	38.38	21.45	44.11	16.74	56.38
47	28.97	17.31	40.25	15.34	47.05
48	30.18	19.46	35.52	15.69	48.01
49	33.70	20.81	38.25	16.57	50.83
50	31.68	18.19	42.58	14.32	54.8
Average			39.93		51.62

According to the results in Table 3, it is obvious that both the PSO and DE algorithms performed very well in generating solutions with less distance, compared with those obtained from the current practices for all instances, with fast computing times. Both algorithms were able to provide robust solutions with small standard deviations. However, it was observed that DE clearly outperformed PSO, since DE was able to generate the better solution quality for the best and average travel distances in all instances. In Table 4, DE shows its superiority to PSO for all instances by providing higher *PD* values from the current practice solutions. The percentage deviation (*PD*) of the distance ranged from 33.34 to 45.91 with an average of 39.93 for the PSO algorithm and from 42.92 to 57.88 with an average of 51.62 for the DE algorithm.

Moreover, the performance differences between PSO and DE are visually demonstrated in Figure 8, showing box plots of the total travel distance from 10 replicated runs of the PSO and DE algorithms, which used solutions from instances 1, 20, 40 and 50. According to the results in Figure 8, it is clearly seen that the solutions obtained from DE were significantly better than those obtained by PSO.

Lastly, the behavior of the algorithms was investigated in terms of time and space complexity (i.e., how the algorithm runtime and space requirements scaled when the inputs grew). Specifically, the plots of the algorithm runtimes and memory usages versus the number of visited locations (customers plus the post office) were constructed as shown in Figures 9 and 10. The number of locations ranged from 10 to 100, with the distance matrix randomly generated specifically for the complexity examination of those algorithms.

Figure 8. Box plot comparison of the total travel distance (km) from 10 replicated runs between the PSO and DE algorithms (instances 1, 20, 40 and 50).

Time Complexity of DE vs PSO Algorithms

Figure 9. Plot between the algorithm runtimes (DE and PSO) and the visited locations. The quadratic regression fits and their functions are included.

Space Complexity of DE vs PSO Algorithms

Figure 10. Plot between the algorithm memory usage (DE and PSO) and the visited locations. The quadratic regression fits and their functions are included.

According to Figures 9 and 10, the quadratic regression was used to fit both the time and space complexity plots for DE and PSO. The coefficients of determination (R^2) of the selected regression model in the time complexity plot were 99.43% for DE and 99.62% for PSO, and those in the space complexity plot were 99.96% for DE and 99.97% for PSO. Therefore, the plots clearly demonstrate that the time and space complexity of both algorithms were classified as having an order of n^2 complexity, or $O(n^2)$, since both the runtime and space requirements had quadratic relationships with the input sizes.

5. Discussion and Conclusions

This research employed metaheuristic algorithms, namely the PSO and DE algorithms, to solve the single-objective VRP for optimized routing solutions. The objective was to determine a solution with the lowest travel distance. The study used actual operational data from two delivery vehicles of the Chiang Rai post office over a period of 50 days. The data included geographic coordinates of the customer locations where those vehicles delivered parcels, which were then transformed into distance matrices for use by the optimization algorithms. For each delivery day, the algorithms tried to search and assign new routes to those two vehicles such that the total travel distance was minimized. The routing performances from the PSO and DE algorithms, as well as those from actual operations, were compared. The experimental results showed that both metaheuristic algorithms were far superior to the real delivery operations of the post office in all 50 days of the experiment. However, DE clearly outperformed PSO by providing outstanding solution quality for all instances. This result was mainly due to the different evolution mechanisms of DE and PSO. A PSO swarm has a high tendency to cluster rapidly and becomes stagnant, since the movement of a swarm is highly influenced by the global best solution. In addition, the diversification of DE is better than PSO, because the best solution in the population does not exert any influence on other solutions in the population. A new solution in DE would be selected only if it had better fitness, and thus the average fitness of the population would be equal or better from iteration to iteration. It is important to note that the results obtained from this study were based on the delivery constraints of the Chiang Rai post office. However, for the operations of other post offices, other constraints such as vehicle capacity, a customer's time window and the delivery time horizon could be taken into consideration. In such cases, the problem must be remodeled, and new solutions will be obtained under those circumstances.

Another essential factor was the duration for running one optimization cycle in order to obtain the optimized solution. For both the DE and PSO algorithms, the average duration was approximately six minutes. Combining the decent results and acceptable runtime duration, the methods used in this study have the potential to be employed in real post office operations. Additionally, even though there were two vehicles in this study, the proposed algorithms can also be applied to large cities in which the post office operates several delivery vehicles. In essence, the methods proposed in this study could drastically help reduce the delivery cost and time for postman delivery operations. Additionally, the vehicle routing optimization process (with about 100 customer locations) only takes minutes rather than hours, which is practical in the actual post office work environment where all parcels have to be processed and dispatched quickly.

There are some challenges to be considered if these methods are to be used in actual delivery operations. First, the post office operators need to determine a quick approach to convert customer addresses into geographic coordinates, which should be done systematically and automatically. Second, the duration to run an optimization cycle is proportional to the number of addresses and delivery vehicles. Therefore, the optimization parameters such as the population sizes and iterations should be well adjusted according to those numbers. Thus, the operators could utilize machine learning, artificial intelligence and image processing methods to quickly transform those customer addresses into definite GPS locations. They should also give enough lead time to account for the unexpected increase in processing time before all deliveries can start. In addition, the proposed PSO and DE use

deterministic data, in which the customers and distance matrices are known in advance, and the routing is determined using these data. When some data are changed during delivery, the only way to adapt to such change is to rerun the algorithms. Thus, another important challenge is to develop an adaptive algorithm to respond to changes effectively.

In future studies, other metaheuristic algorithms, such as the adaptive large neighborhood search and genetic programming algorithms, are planned to be tested for their performance with datasets obtained from larger delivery operations, including not only postman delivery problems but also problems from other industries.

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

Funding: This research was funded by Mae Fah Luang University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data used in this study are available upon reasonable request to the corresponding author.

Acknowledgments: The authors are grateful to the Chiang Rai post office. The authors would like to acknowledge Mae Fah Luang University and Chiang Mai University for a partial financial sponsorship and their facility support in this research.

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

References

- Case, M.J.; Johnson, G.G.; Bartowitz, K.J.; Hudiburg, T.W. Forests of the future: Climate change impacts and implications for carbon storage in the Pacific Northwest, USA. *For. Ecol. Manag.* **2021**, *482*, 118886. [CrossRef]
- Our World in Data. Available online: <http://ourworldindata.org/co2-emissions-from-transport> (accessed on 21 January 2021).
- Skerlic, S.; Muha, R. The importance of systems for controlling logistics costs in the supply chain: A case study from the Slovenian automotive industry. *Promet Traffic Traffico.* **2016**, *26*, 299–310. [CrossRef]
- United States Postal Service, Office of Inspector General. Available online: <http://uspsoidg.gov/document/postal-service-dynamic-route-optimization-and-cost-savings-initiative> (accessed on 21 January 2021).
- Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [CrossRef]
- Guzairov, M.B.; Yusupova, N.I.; Smetanina, O.N.; Rassadnikova, E.Y. Models and algorithms for the vehicle routing problem with time windows and other conditions. In Proceedings of the 2016 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE), Novosibirsk, Russia, 3–6 October 2016; pp. 412–416.
- Wang, Z.; Ren, X.; Ji, Z.; Huang, W.; Wu, T. A novel bio-heuristic computing algorithm to solve the capacitated vehicle routing problem based on Adelman-Lipton model. *BioSystems* **2019**, *184*, 103997. [CrossRef] [PubMed]
- Niu, Y.; Zhang, Y.; Cao, Z.; Gao, K.; Xiao, J.; Song, W.; Zhang, F. MIMO: A membrane-inspired multi-objective algorithm for green vehicle routing problem with stochastic demands. *Swarm Evol. Comput.* **2021**, *60*, 100767. [CrossRef]
- Dumez, D.; Lehoude, F.; Peton, O. A large neighborhood search approach to the vehicle routing problem with delivery options. *Transp. Res. B* **2021**, *104*, 103–132. [CrossRef]
- Baldacci, R.; Mingozzi, A. Lower bounds and an exact method for the capacitated vehicle routing problem. In Proceedings of the 2006 International Conference on Service System and Service Management, Troyes, France, 25–27 October 2006; pp. 1536–1540.
- Ropke, S. Heuristic and Exact algorithms for Vehicle Routing Problems. Ph.D. Thesis, University of Copenhagen, København, Denmark, 2006.
- Ladner, R.E. On the structure of polynomial time reducibility. *J. ACM* **1975**, *22*, 151–171. [CrossRef]
- Tordecilla, R.D.; Copado-Mendez, P.J.; Panadero, J.; Quintero-Araujo, C.L.; Montoya-Torres, J.R.; Juan, A.A. Combining heuristics with simulation and fuzzy logic to solve a flexible-size location routing problem under uncertainty. *Algorithms* **2021**, *14*, 45. [CrossRef]
- Meira, L.A.A.; Martins, P.S.; Menzori, M.; Zeni, G.A. Multi-objective vehicle routing problem applied to large scale office. *arXiv* **2018**, arXiv:1801.00712.

15. Niroomand, I.; Khataie, A.H.; Galankashi, M.R. Vehicle routing with time window for regional network services—Practical modelling approach. In Proceedings of the 2014 IEEE International Conference on Industrial Engineering and Engineering Management, Bandar Sunway, Malaysia, 9–12 December 2014. [CrossRef]
16. Nasri, M.; Hafidi, I.; Metrane, A. Multithreading parallel robust approach for the VRPTW with uncertain service and travel times. *Symmetry* **2021**, *13*, 36. [CrossRef]
17. Zhou, Z.; Ha, M.; Hu, H.; Ma, H. Half open multi-depot heterogeneous vehicle routing problem for hazardous materials transportation. *Sustainability* **2021**, *13*, 1262. [CrossRef]
18. Erbao, C.; Mingyong, L.; Kai, N. A Differential Evolution & Genetic Algorithm for Vehicle Routing Problem with Simultaneous Delivery and Pick-up and Time Windows. In Proceedings of the 17th World Congress the International Federation of Automatic Control, Seoul, Korea, 6–11 July 2008.
19. Kromer, P.; Abraham, A.; Snasel, V.; Berhan, E.; Kitaw, D. On the differential evolution for vehicle routing problem. In Proceedings of the 2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR), Hanoi, Vietnam, 15–18 December 2013; pp. 384–389.
20. 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*. [CrossRef]
21. Xing, L.N.; Liu, Y.; Li, H.; Wu, C.; Lin, W.C.; Song, W. A hybrid discrete differential evolution algorithm to solve the split delivery vehicle routing problem. *IEEE Access* **2020**, *8*, 207962–207972. [CrossRef]
22. Song, L.; Dong, Y. An improved differential evolution algorithm with local search for capacitated vehicle routing problem. In Proceedings of the 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), Xiamen, China, 29–31 March 2008.
23. Lei, J.; Li, J. Solving capacitated vehicle routing problems by modified differential evolution. In Proceedings of the 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010), Wuhan, China, 6–7 March 2010; pp. 513–516.
24. Moghaddam, B.; Ruiz, R.; Sadjadi, S.J. Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Comput. Ind. Eng.* **2012**, *62*, 306–317. [CrossRef]
25. Peng, Z.; Manier, H.; Manier, M.A. Particle swarm optimization for capacitated location-routing problem. *Comput. Ind. Eng.* **2020**, *146*. [CrossRef]
26. Marinakis, Y.; Marinaki, M.; Migdalas, A. A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows. *Inf. Sci.* **2019**, *481*, 311–329. [CrossRef]
27. Chen, J.; Shi, J. A multi-compartment vehicle routing problem with time windows for urban distribution—A comparison study on particle swarm optimization algorithms. *Comput. Ind. Eng.* **2019**, *133*, 95–106. [CrossRef]
28. Chen, R.M.; Shen, Y.M.; Hong, W.Z. Neural-like encoding particle swarm optimization for periodic vehicle routing problems. *Expert Syst. Appl.* **2019**, *138*, 112833. [CrossRef]
29. Sedighzadeh, D.; Mazaheripour, H. Optimization of multi objective vehicle routing problem using a new hybrid algorithm based on particle swarm optimization and artificial bee colony algorithm considering Precedence constraints. *Alex. Eng. J.* **2018**, *57*, 2225–2239. [CrossRef]
30. Yanwei, Z.; Bin, W.; Wanliang, W.; Jingling, Z. Particle swarm optimization for open vehicle routing problem with time dependent travel time. In Proceedings of the 17th World Congress the International Federation of Automatic Control, Seoul, Korea, 6–11 July 2008.
31. Google Maps Platform, Geo-Location APIs. Available online: <http://cloud.google.com/maps-platform> (accessed on 15 May 2019).
32. Kachitvichyanukul, V. Comparison of three evolutionary algorithms: GA, PSO, and DE. *Ind. Eng. Manag. Syst.* **2012**, *11*, 215–223. [CrossRef]
33. Rossum, G.V. Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI). Amsterdam, May 1995. Available online: <https://www.python.org> (accessed on 9 April 2019).
34. PyCharm, the Python IDE for Professional Developers. Available online: <http://jetbrains.com/pycharm> (accessed on 5 November 2020).