Article

Modified Differential Evolution Algorithm Solving the Special Case of Location Routing Problem

Raknoi Akararungruangkul [1,*] and Sasitorn Kaewman [2]

[1] Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University,
    Khon Kaen 40000, Thailand
[2] Faculty of Informatics, Mahasarakham University, Maha Sarakham 44000, Thailand; sasitorn.k@msu.ac.th
* Correspondence: raknoi55@gmail.com

Abstract: This research article aims to solve the special case of the location routing problem (SLRP) when the objective function is the fuel consumption. The fuel consumption depends on the distance of travel and the condition of the road. The condition of the road causes the vehicle to use a different speed, which affects fuel usage. This turns the original LRP into a more difficult problem. Moreover, the volume of the goods that are produced in each node could be more or less than the capacity of the vehicle, and as the case study requires the transportation of latex, which is a sensitive good and needs to be carried within a reasonable time so that it does not form solid before being used in the latex process, the maximum time that the latex can be in the truck is limited. All of these attributes are added into the LRP and make it a special case of LRP: a so-called SLRP (a special case of location routing problem). The differential evolution algorithms (DE) are proposed to solve the SLRP. We modified two points in the original DE, which are that (1) the mutation formula is introduced and (2) the new rule of a local search is presented. We call this the modified differential evolution algorithm (MDE). From the computational result, we can see that MDE generates a 13.82% better solution than that of the original version of DE in solving the test instances.

Keywords: location routing problem; rubber; modify differential evolution algorithm; vehicle routing problem

1. Introduction

More than 14% of the Thai GDP comes from the transportation and logistics sector. More than 60% of the transportation cost incurred in Thailand is by one actor: the agricultural supply chain. This is because Thailand is known as an agricultural country. The major agricultural industry is the rubber industry. More than 80% of rubber is produced in the southern part of Thailand, with the remaining part of rubber production mainly in the northeastern part of Thailand. The rubber industry in the South has a long history, and much suitable infrastructure has been developed. Aside from the northeastern part of Thailand, the infrastructure and knowledge, including technology, knowledge, instruments and so forth, are not yet sufficient for the high growth of the rubber industry. The transportation costs in the rubber industry are not paid by the government or the person who takes care of this issue; farmers still deliver latex themselves to sell at the rubber collecting points. This activity generates a very high transportation cost for the whole country. Our research team is engaged in a project to design the latex collection system. This system comprises of finding the location of the collection points and the transportation route of the vehicles. This problem is actually similar to the vehicle routing problem which was first proposed by Dantzig and Ramser [1] and has been proven to be an NP-hard problem [2]. Apart from the vehicle routing problem (VRP), this problem must also

decide the collection point of the latex; thus, the VRP turns out to be a location routing problem (LRP). Location routing [3].

The proposed problem is not the general LRP that we found in the literature but has a few characteristics that make it a special case of LRP. These characteristics are as follows:

(1) The volume of latex which is available in the rubber field can be more or less than the capacity of the vehicles;

(2) The potential location and the rubber fields have different attributes such as different road conditions which can affect the speed used and the fuel consumption rate of the truck.

(3) The maximum duration or distances used for each vehicle are limited due to the latex's fast transformation to solid;

(4) We focus the objective function on minimizing the total fuel consumption instead of the total distance as we often see in the literature.

These four special attributes make this problem a special case of the LRP problem (SLRP). The SLRP has never been found in the literature due to its 4 special attributes explained above. This problem can be found in many real-world applications. Nakorn Panom Province is one of the cities that has the fastest growing rate of development in the agricultural industry. The rubber cultural area has increased by more than 150% in the last 5 years in Nakorn Panom Province. We will take Nakorn Panom Province as the case study. The method which is presented here is customized to the rubber field industry, and it is also applicable to other industries where it has the same attributes as mentioned above.

In this study, the effective modified differential evolution algorithm will be presented to solve the proposed problem. The article is organized as follows; Section 2 is the literature review; Section 3 is the problem definition, and current practice procedures will be presented; Section 4 presents the proposed heuristics; and Section 5 gives the conclusions and outlook of the article.

2. Literature Review

Dantzig and Ramser [1] introduced the vehicle routing problem (VRP) and Lenstra and Rinnooy [2] proved that it is an NP-hard problem. Various types of VRP have been presented in many research articles. Braekers and Ramaekers [4] have given an overview of the scenario and the problem's physical characteristics, extending the basic uncapacitated VRP. Besides the various types of VRP problems, there are also various types of methods for solving the VRP. The metaheuristic method is one of the solution methods that is very popular for solving VRP as it is fast and effective. The solution approaches presented in many articles are simulated annealing [5].

The location routing problem (LRP) is one of many various types of VRP. This problem has two decision variables which need to be solved. These two decision variables are (1) the suitable location to be used and (2) the routing of the field or the clients that are assigned to that location [6,7]. The optimal value of the suitable location can be determined before or during the assignment of the clients to the locations.

There are many researchers that have presented a methodology to solve the LRP. Most of the algorithms proposed in the literature are two-phase heuristics, which are (1) cluster first, (2) route second or (1) route first, (2) cluster second [8]. From the literature, both choices perform well in many test problems; thus, selecting one will not affect the solution quality. In our research, the proposed heuristics employ the idea of using two-phase heuristics. We select cluster first, route second as the starting point of our algorithm.

There are many researchers who have focused on developing metaheuristics to solve the problem such as tabu search [9] and the simulated annealing (SA) technique [10,11].

In this research, the differential evolution algorithm (DE) is presented as it is a fast and effective algorithm.

The differential evolution (DE) algorithm is a branch of the evolutionary algorithm developed by Rainer Storn and Kenneth Price [12] for optimization problems over continuous optimization. Thereafter, DE has been successfully used in combinatorial optimization such as in assembly line balancing [13,14], the location-allocation problem [15], machine layout [16,17], and the manufacturing problem [18].

The DE has been applied to VRP problem [19–25]. The design of DE to solve VRP problem has both the traditional [20] way or the slight modification of some mechanism to get a more effective algorithm. Key successes of DE are the good designing of vectors to represent the problem, the encoding and decoding methods, the setting of good predefined parameters in the DE mechanism and the effectiveness of the mutation, recombination, and selection procedure [23–25]. Moreover, with DE, it is also easy to add some procedure to increase its performance such as add local search [19], adjust decoding method [21], and self-adaptive some parameters [23]. Though there are many articles proposing DE to solve VRP, we cannot find an effective DE to solve the LRP [22]. So it is our contribution to present the effective DE to solve the special case of LRP. When all key successes have been introduced to DE, DE will be very effective compared with other metaheuristics proposed in the literature. Moreover, DE has the advantage that it is fast and efficient; thus, in this article, we will present the modified version of it to solve the problem.

This article is organized as follows: in the next section, the problem definition and the current practice heuristics is presented. Section 3 presents the proposed heuristics. Section 4 proposes the computational results and Section 5 is the conclusion of the article.

3. The Problem Definition and the Current Practice Heuristics

3.1. The Problem Definition

The case study is composed of 110 rubber fields and 30 potential areas that can be set as the collecting points. All connections between the location and fields have different road conditions, with different speed limits and road condition factors. The example of the case study is shown in Figure 1.
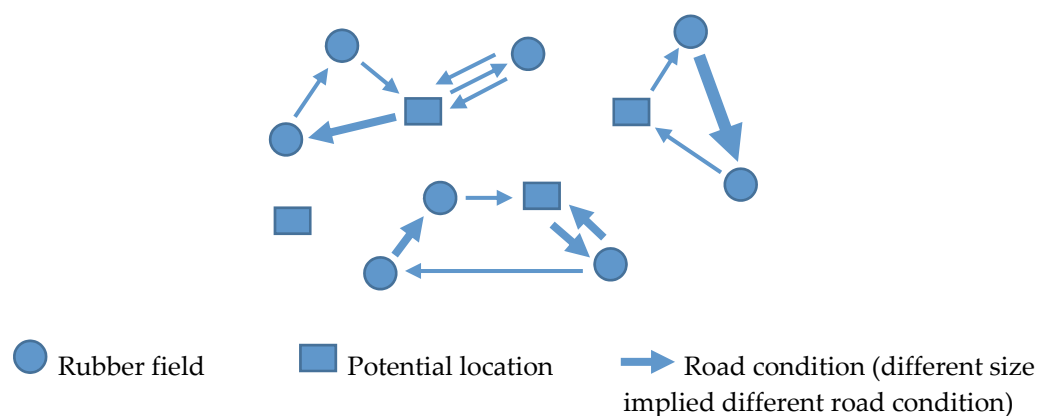

● Rubber field　　■ Potential location　　➤ Road condition (different size implied different road condition)

Figure 1. The problem definition.

From Figure 1, we can see that one field can be visited by direct shipping (which can be visited more than once by direct shipping) and routing or even combined direct shipping and routing. This makes the problem much harder than the normal LRP. The different size of the arrows used implies different road conditions and speed limits that are permissible to use, and the speed used on the road affects the fuel consumption of the vehicle. The objective function which is used here is the total fuel consumption, and the conditions that need to be satisfied are as follows:

(1) The speed limit of the road need to be controlled;
(2) All fields must be visited at least one time;
(3) The maximum duration of traveling has to be under a limit;
(4) The total working hours of the truck needs to be controlled.

The objective function and the constraints addressed above make the LRP harder to solve. The current practice method which is used in the Nakorn Panom Province will be explained in the next section.

The types of road are set to seven types, distinguished by using the average driving speed and the fuel consumption rate, as shown in Table 1. The consumption rate is calculated from https://en.wikipedia.org/wiki/Fuel_economy_in_automobiles.

Table 1. The road type and its consumption rate.

Road Type	Ave. Speed	Litre/km
S	30	0.118
T	40	0.107
V	50	0.112
W	60	0.090
X	70	0.098
Y	80	0.098
Z	90	0.102

Ave. Speed is the average speed of the vehicle that can be driven on the particular road types. In the case study, the historical data of the average speed used for that road is collected for two weeks. After that, the average speed is converted into road type and the fuel used using Table 1. The consumption rate is assigned to the road, then multiplied by the distance of the connection road. The average fuel used on that road will be obtained. For example, if the connection between fields 1 and 2 has a distance of 50 km and is a type C road. Thus, the fuel needed to travel on this road is $0.112 \times 50 = 5.6$ L. This mechanism will be applied to all connections. Finally, the metric of the fuel consumption of the road will be obtained.

3.2. The Current Practice Procedure

Previously, the practice procedure has not taken into account the fuel consumption rate and the speed limit. Thus, the current practice procedure will be decided based on the information given in the distance matric. The example used to explain the current procedure and the proposed heuristics composed of 5 potential locations (A, B, C, D, E) and 6 rubber fields. The traveling time of the locations to fields and fields to field is given in Tables 2 and 3 respectively. The road type of all locations and field connections is shown in Tables 6 and 7. The maximum traveling time per round of the 12-ton truck is 60 min and when the truck needs to travel more than one round, it cannot travel for more than 100 min. The procedure practice procedure can be explained as follows:

(1) Select the current location to be used. Start by using the potential location that has the highest S_k when S_k is calculated by Formula (1):

$$S_k = \frac{C_k}{F_k} \tag{1}$$

where C_k is the capacity of location k and F_k is the operating cost per day of location k. For example, if a particular location has a capacity of 80 tons per day and an operating cost of 1500 baht per day, the S_k is 0.533. If we have five locations (A–E) with a capacity of 30, 20, 50, 40, and 50 tons and an operation cost of 1200, 1100, 1400, 1000, and 1200, then we get the S_k of each location as 0.025, 0.018, 0.035, 0.04, and 0.041, respectively. Location E will be selected to be the location first, then step 2 will be executed.

(2)　Assign the field to the current location that is in use. This step simply assigns the fields according to the closest distance to the location that is in use. For example, if we have six fields, we have five locations; if we have six fields to travel, the traveling time between fields is calculated using the average speed of the road. The traveling times from locations to the fields are calculated and shown in Table 2, while the traveling time between the fields is shown in Table 3. Each field has a volume of latex of 5, 20, 40, 10, 5, and 13 tons, respectively. From step 1, location E is opened and it has a 50-ton capacity; thus, the closest fields to it that have a cumulative volume of latex of fewer than 50 tons are field 1 (5), 4 (10), 6 (13), and 5 (5). After all possible fields are assigned to location E and location E is full, then the next location will be opened (redo step 1). The second highest S_k is location D, which has a capacity of 40 tons. The remaining field will be assigned to location D. The closest field to D is field 2, which has a latex volume of 20, and we cannot assign field 3 to location D due to location D not having enough capacity. Step 1 will be performed again and we will open location C which has a capacity of 50. Field 3 will be assigned to this location. To conclude steps 1 and 2, the result of the assignment of the fields to the locations is shown in Table 4.

Table 2. The traveling times of the locations to the fields.

-	1	2	3	4	5	6
A	12	12	14	19	12	17
B	19	6	10	21	20	21
C	21	19	19	17	7	18
D	17	10	15	27	5	19
E	13	22	20	14	19	16

Table 3. The raveling times between fields.

-	1	2	3	4	5	6
1	0	11	15	14	6	9
2	11	0	13	9	6	16
3	15	13	0	27	11	15
4	14	9	27	0	16	25
5	6	6	11	16	0	14
6	9	16	15	25	14	0

Table 4. The result of steps 1 and 2 in the current practice procedure.

Location	Field	Full Cap.	Used Cap.
E	1, 4, 6, 5	50	33
D	2	40	20
C	3	50	40

(3)　Route the fields in the opened locations using the nearest neighbor heuristics. Please note that a vehicle can travel more than one route, as long as it has enough time to travel in one day. If the truck has a capacity of 12 tons, the time limit per round is 60 min (including the loading-in and out time of (0.5 + 0.5 = 1) minutes per ton), and the maximum time per day that the truck can be in operation is 100 min. The loading time of latex to the collecting point is 1 min per ton.

Routing location E: A truck that has a capacity of 12 tons has to travel less than 60 min per round and the cumulative time used per day of that truck must not exceed 100 min. This truck will be sent to pick up the latex. The route is formed based on the nearest neighbor heuristics (the shortest route is preferred). Please note that the traveling time is different from the distance. Truck 1 will go out to pick up 5 tons from field 1 and 7 tons from field 4; then, it comes back to location E.

This route carries 12 tons and uses 41 min of traveling time and 12 min of loading time, thus the total time used in the first route is 53 min. Truck 1 still has a total time of less than 100 min, and so truck 1 will then go out to pick up 3 tons from field 4 and 5 tons from field 5. This route uses 57 min and carries 8 tons. The first truck has a total time of 98 min. This means that the second truck needs to start working. Truck 2 will be sent to field 6 two times due to field 6 having a latex volume of 13 tons, which cannot be carried back to E in only one route. The result of the current practice for the example is shown in Table 5.

From Table 5, we can see that the total time used is 438 min. This has a loading time of 93 min; thus, the traveling time total is 345 min.

The type of road for each connection is shown in Table 6 (location–field) and Table 7 (field–field).

Table 5. The result of the routing of the example.

Loc.	Truck	Route	T.U.	V.L.
	1	E-1-4-E	41	12
	1	E-4-5-E	57	8
E	2	E-6-E	44	12
	2	E-6-E	44	1
D	1	D-2-D	32	12
	1	D-2-D	28	8
	1	C-3-C	50	12
	1	C-3-C	50	12
C	2	C-3-C	50	12
	2	C-3-C	42	4
		Total	438	93

Remark: Loc. is location, T.U. is time used in the route, V.L. is volume carried in that route.

Table 6. The road type from location to the field.

-	1	2	3	4	5	6
A	S	T	V	W	S	V
B	V	T	S	V	W	V
C	W	X	V	V	W	V
D	T	S	X	T	S	X
E	X	X	T	Y	Y	V

Table 7. The road type from field to field.

-	1	2	3	4	5	6
1	-	S	W	V	T	W
2	S	-	S	X	T	S
3	W	S	-	V	X	W
4	V	X	V	-	X	S
5	T	T	X	X	-	X
6	W	S	W	S	X	-

The traveling distance of each connection is shown in Tables 8 and 9.

Table 8. The distance (km) from locations to fields.

-	1	2	3	4	5	6
A	6	8	11.7	19	6	14.2
B	15.8	4	5	17.5	20	17.5
C	21	22.2	15.8	14.2	8.17	15
D	11.3	5	17.5	18	2.5	22.2
E	15.2	25.7	13.3	18.7	25.3	13.3

Table 9. The distance (km) from field to field.

	1	2	3	4	5	6
1	0	5.5	15	11.7	4	9
2	5.5	0	6.5	10.5	4	8
3	15	6.5	0	22.5	12.8	15
4	11.7	10.5	22.5	0	18.7	12.5
5	4	4	12.8	18.7	0	16.3
6	9	8	15	12.5	16.3	0

From the distances given in Tables 8 and 9, when we combine these distances with the road types of each connection given in Tables 6 and 7 and use the fuel consumption rate shown in Table 1, the fuel use in the traveled connections in Table 5 can be calculated. For example, for location E, route 1 of truck 1, the route is E-1-4-E, which has three connections. These are E-1, 1-4, and 4-E, which have distances of 15.2 km, 11.7 km, and 18.7 km, respectively. These connections have fuel consumptions of 0.098(X), 0.112(V), and 0.098(Y) liters per kilometer, respectively. Therefore, the fuel used in these three connections total 1.4896, 1.3104, and 1.8326 L, respectively. In total, in location E, route 1, truck 1 consumes 4.6324 L. The same mechanism will be performed with all trucks, routes, and locations. The result of fuel consumption using the current practice procedure is shown in Table 10.

Table 10. The fuel consumption of all routes.

Loc.	Truck	Route	Fuel Consumption (L)
E	1	E-1-4-E	4.6324
	1	E-4-5-E	6.1446
	2	E-6-E	2.9792
	2	E-6-E	2.9792
D	1	D-2-D	1.18
	1	D-2-D	1.18
C	1	C-3-C	3.5392
	1	C-3-C	3.5392
	2	C-3-C	3.5392
	2	C-3-C	3.5392
Total			33.2522

From Table 10, the total fuel consumption for all locations is 33.2522 L per day.

4. The Proposed Heuristic

The proposed heuristic is designed to solve the problem. Many metaheuristics are available in the literature. The differential evolution algorithm (DE) is selected to solve this problem because it is a fast and effective heuristic. Generally, DE is composed of four steps:

(1)   Randomly generate the initial vector or solution;
(2)   Use the mutation process;
(3)   Use the recombination process;
(4)   Use the selection process.

Steps (2) to (4) will be iteratively executed. Details of each step are explained in Section 4.1.

4.1. Randomly Generated Vector or Solution

The solution can be obtained by generating a set of vectors. Each vector is composed of D positions, where D is the number of fields. The number of the population is 5 (or NP = 5). Table 11 shows an example of the values in each position of five vectors which are randomly generated in the first iteration.

When the vectors are generated, obtaining the solution of the problem, the decoding method needs to be executed. There are five steps of decoding (transferring) the vector into the problem's solution, which are explained in Sections 4.1.1–4.1.4.

Table 11. The example of five random vectors.

Farmer		1	2	3	4	5	6
Vector	Volume	5	20	40	10	5	13
NP1		0.39	0.10	0.42	0.35	0.31	0.89
NP2		0.48	0.33	0.81	0.26	0.86	0.64
NP3		0.41	0.93	0.77	0.02	0.37	0.18
NP4		0.01	0.57	0.62	0.17	0.53	0.42
NP5		0.98	0.14	0.64	0.76	0.12	0.17

Figure 1: Assigning random code numbers to each vector in each position.

4.1.1. Finding the Order of the Fields

In this step, we need to sort the value of the position of each vector to get the order of the fields. The position's value is sorted according to the increasing order. For example, for vector 4, the original value of positions 1, 2, 3, 4, 5, and 6 are 0.01, 0.57, 0.62, 0.17, 0.53, and 0.42. The order of the value in each position according to increasing order is 0.01, 0.17, 0.42, 0.53, 0.57, and 0.62. This order of value generates an order of position for vector 4 of 1, 4, 6, 5, 2, and 3.

4.1.2. Randomly Select the Location

The method of selecting the opened location is explained in the following example. The probability of selecting the location can be calculated using Formula (2):

$$p_k = \frac{S_k}{\sum_{k=1}^{K} S_k} \tag{2}$$

where p_k is the probability of location k and S_k is the heuristics information as explained in Section 3.2.

From the example, the probabilities p_k of locations A, B, C, D, and E are 0.16, 0.11, 0.22, 0.25, and 0.26, respectively. In the next step, the cumulative probability of each location is 0.16, 0.27, 0.49, 0.74, and 1. If the random number is 0.57, then location D is selected to be the first opened location. We can apply this mechanism to get the order of the location. The order of the locations in this step is D, E, A, B, and C; then, the next step, the decoding method, will be executed.

4.1.3. Assigning the Field to the Location According to Their Place in the Order

For example, the field order is 1, 4, 6, 5, 2, and 3 and the location order is D, A, E, B, and C. We start to assign the field to the location by adding the field in the very first order until this location is full, and then we move forward to the next possible location and continue doing this until all fields are assigned to exactly one location.

In this example, field 1 will be assigned to location D, and the capacity of D is updated whenever an assignment has been done. The location will close when it loads a volume of latex in the field reaching the full capacity level. The result of the assignment is shown in Table 12.

Table 12. The result of the decoding method of the differential evolution (DE) algorithm.

Location	Field	Cap	Load
D	1, 4, 6, 5	40	33
A	2	30	20
E	3	50	40

From Table 12, locations D, A, and E will be in use while the remaining locations will be closed.

4.1.4. Routing All Fields in the Locations

In this step, we construct the route of the fields that are assigned to the locations according to the capacity of the truck per round, the longest distance or traveling time per round and the total distance or time that the truck can travel per day. The result of Section 4.1.1 is the same as in Table 10 because the routing phase (Section 4.1.1) is the same as the routing phase of the current practice procedure and the order of the fields and locations are the same as in the current practice example.

4.2. Mutation Process

4.2.1. The Original Mutation Process

In this step, after we get the initial solution, we apply the mutation process formulas. In the proposed heuristic, the mutation formula is Formula (3):

$$V_{i,j,G} = X_{best,j,G} + F\left(X_{r_1,j,G} - X_{r_2,j,G}\right) \tag{3}$$

where $F = 1.5$, $X_{i,j,G}$ is a randomly selected target vector and $X_{best,j,G}$ is the vector that has the best solution compared among all target vectors.

4.2.2. The Modified Mutation Process

Originally, all vectors were random; we take them from the current iteration target vectors, and the best solution is the set of best vectors obtained from the best among all target vectors. The modified version of the mutation process is one in which we add one set of the best vectors. The number of the best vectors in this set is equal to the number of the population in the normal DE method. For example, if there are 10 target vectors (NP = 10), there will be 10 vectors in the best vectors set as well. The set of best vectors is obtained by collecting the best vectors that are found during the simulation. These vectors are not the same as the normal target vectors due to the collection of all good vectors, even if they come from the same vector number. The target vector of vector 1 at the current iteration is the best vector that obtains the best solution for vector 1. The second best for vector 1 is the vector from the last iteration if the current iteration found a new solution. This vector will forget the last best solution whenever it finds a better solution, even if the last best solution is better than that of the other vector in the set of target vectors. In the modified version of the mutation process, the second-best vector of a vector can be kept in the best vectors set if it is better than that of the other

target vectors. The mutant vector used in the modified mutation process is generated from both the normal target vector and the set of best vectors, as shown in Figure 2.

From Figure 2, the mutant vector of vector 2 is generated from the best vector (assuming that vector 1 is the best vector) and the two random vectors are vector 3 (from the set of target vectors) and 5 (from the set of best vectors) and the result is given in Figure 2. The switching between using the set of best vectors or the set of target vectors can be executed by randomly choosing the using probability. Each vector is randomly chosen before the mutant process is executed. The probability function of the use of the set of best vectors is shown in Formula (4):

$$C = 1 - exp^{-[\,(It - \frac{MaxIt}{2})^2\,]} \tag{4}$$

If the random number is less than or equal to *C*, then the random vector used in the mutant process is drawn from the set of best vectors instead of the set of target vectors.

The updating of the set of best vectors can be done by getting rid of the worst vector in the set and putting the new best vector in if the new best vector is better than at least one vector in the set of best vectors. This is done whenever a new best vector (better than at least the worst vector in the set of best vectors) is found. The result of the mutant process is mutant vector $V_{i,j,G}$, where i is the number of the vector, *j* is the position number, and *G* is the current iteration.
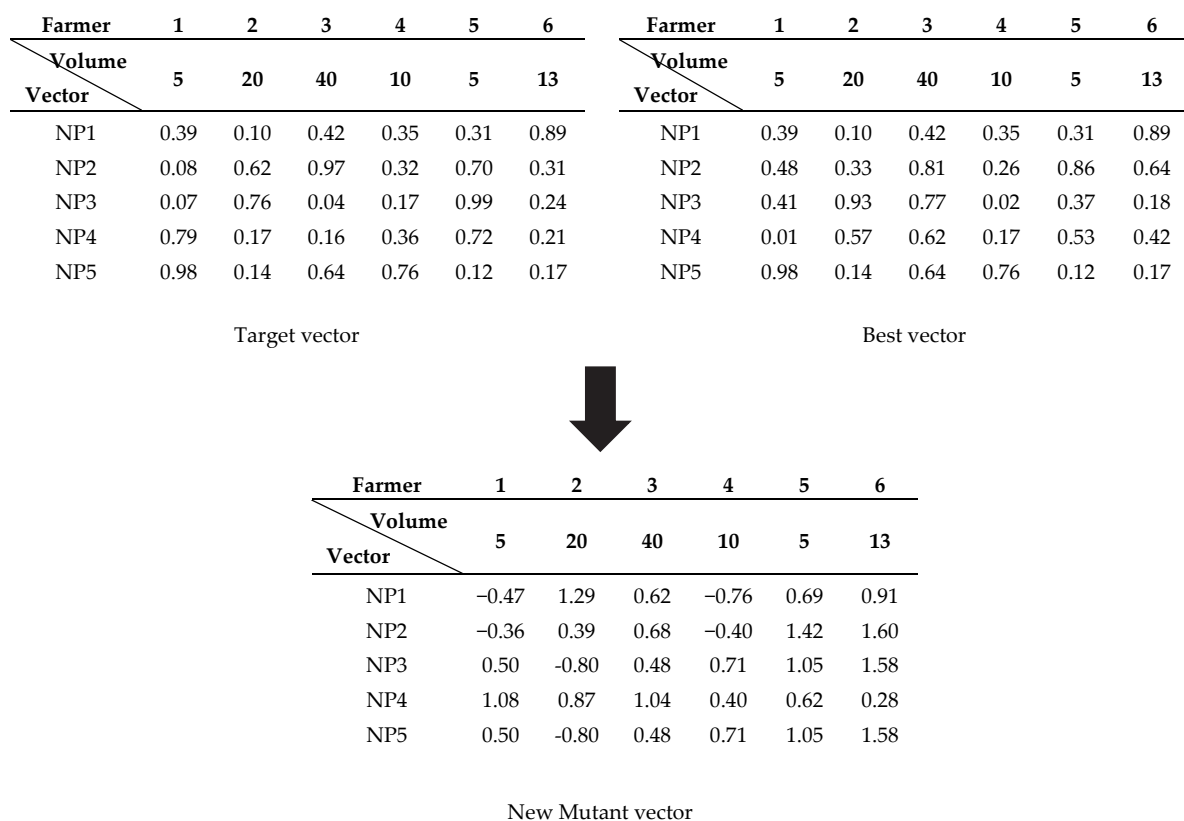
Farmer	1	2	3	4	5	6
Volume Vector	5	20	40	10	5	13
NP1	0.39	0.10	0.42	0.35	0.31	0.89
NP2	0.08	0.62	0.97	0.32	0.70	0.31
NP3	0.07	0.76	0.04	0.17	0.99	0.24
NP4	0.79	0.17	0.16	0.36	0.72	0.21
NP5	0.98	0.14	0.64	0.76	0.12	0.17

Target vector

Farmer	1	2	3	4	5	6
Volume Vector	5	20	40	10	5	13
NP1	0.39	0.10	0.42	0.35	0.31	0.89
NP2	0.48	0.33	0.81	0.26	0.86	0.64
NP3	0.41	0.93	0.77	0.02	0.37	0.18
NP4	0.01	0.57	0.62	0.17	0.53	0.42
NP5	0.98	0.14	0.64	0.76	0.12	0.17

Best vector

Farmer	1	2	3	4	5	6
Volume Vector	5	20	40	10	5	13
NP1	−0.47	1.29	0.62	−0.76	0.69	0.91
NP2	−0.36	0.39	0.68	−0.40	1.42	1.60
NP3	0.50	-0.80	0.48	0.71	1.05	1.58
NP4	1.08	0.87	1.04	0.40	0.62	0.28
NP5	0.50	-0.80	0.48	0.71	1.05	1.58

New Mutant vector

Figure 2. The modified mutant process.

4.3. Recombination Process

The recombination process can be executed by randomly assigning one number for each position in a vector; if this random number is less than or equal to the predefined parameter *CR*, then the value in position *j* of vector *i* is taken from the value of position *j* in vector *i* of the mutant vector. On the

other hand, if the random number is higher than that of *CR*, the value in that position is taken from the target vector. The formula of the original version of DE is shown in Formula (5):

$$U_{i,j,G} = \begin{cases} V_{i,j,G} \ if \ rand_{i,j} \leq CR \\ X_{i,j,G} \ if \ rand_{i,j} > CR \end{cases} \tag{5}$$

When it is executed, the set of trial vectors ($U_{i,j,G}$) will be obtained.

In this article, a new recombination formula is presented in Formula (6).

$$U_{i,j,G} = \begin{cases} V_{i,j,G} \ if \ rand_{i,j} \leq CR_1 \\ X_{i,j,G} \ if \ CR_1 \ < rand_{i,j} \leq CR_2 \\ B_{i,j,G} \ \geq CR_2 \end{cases} \tag{6}$$

Predefined parameters CR_1 and CR_2 have to be set first and lie between 0 and 1. If the random number $rand_{i,j}$ is less than CR_1, then the value in position j of vector i is drawn from the mutant vector. If it is greater than CR_1 but less than CR_2, the value in position j, vector i will be taken from the target vector. Finally, if the random number is higher than CR_2, that position will take the value from the set of best vectors ($B_{i,j,G}$). After the trial vector is obtained, the local search will be applied for all vectors.

4.4. The Local Search

The local search that we use in the proposed heuristic is the SWAP algorithm. The swap algorithm will be executed in the order of the fields and the locations generated in the decoding method. For example, in Figure 3, the current interchange of the fields is order 1 and 2, which is now fields 5 and 4. The exchange is performed and obtains the order of fields as 4, 5, 1, 2, 6, and 3, respectively. The new order will use the decoding method using the location order E, A, D, C, and B, which is the exchange order between the locations B and C. The order of location has to exchange all positions, such as E exchanging with A, D, C, and B. After the exchange, the new order will be used to decode with the current order of the fields being 4, 5, 1, 2, 6, and 3.

After all possible exchanges of location are performed with the current order of fields, then the new order of the fields is generated by exchanging the next position in the field order.
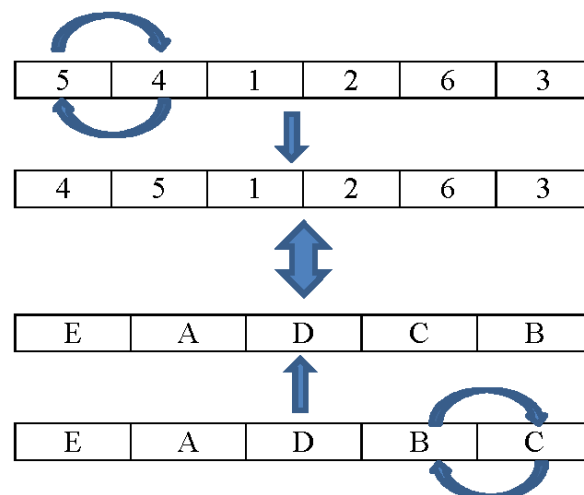

Figure 3. An example of the local search.

4.5. Selection Process

The selection process is according to Equation (7).

$$X_{i,j,G+1} = \begin{cases} U_{i,j,G} \; if \; f(U_{i,j,G}) \leq f(X_{i,j,G}) \\ X_{i,j,G} \quad otherwise \end{cases} \tag{7}$$

When the selection process is finished, we will get the target vectors of the next generation (iteration).

The procedure of the proposed heuristics is shown in Figure 4.

Set NP, CR, CR1, CR2, F, D (size of vector)
Generate Initial Solution
Begin
For G = 1 to G$_{max}$ when G = iterations and G$_{max}$ = Maximum iteration
  For N = 1 to NP
    Random Generate Random Number r$_1$, r$_2$, r$_3$: r$_1$, r$_2$, r$_3$ $\in$[1..NP], r$_1 \neq$ r$_2 \neq$ r$_3 \neq$ N
    Produce Mutant Vector N (Mutation Process)
    For i = 1..D
      $V_{i,j,G} = X_{best,j,G} + F(X_{r_1,j,G} - X_{r_2,j,G})$
    Next Produce Trial Vector N (Recombination Process)

    $U_{i,j,G} = \begin{cases} V_{i,j,G} \; if \; rand_{i,j} \leq CR \\ X_{i,j,G} \; if \; rand_{i,j} > CR \end{cases}$ (optional)

    $U_{i,j,G} = \begin{cases} V_{i,j,G} \; if \; rand_{i,j} \leq CR_1 \\ X_{i,j,G} \; if \; CR_1 \; < rand_{i,j} \leq CR_2 \\ B_{i,j,G} \; \geq CR_2 \end{cases}$ (optional)

    Next Produce New Target Vector (Selection Process)
      Apply Decoding Method (4.1.1–4.1.4)
      Apply SWAP
      Selection Process
      Collect Best Vector $B_{i,j,G}$
End

Figure 4. The pseudo code of the modified DE.

5. Computational Results

The computational results are obtained by testing the DE (original version), the modified DE (MDE), and the current practice procedure with 10 test instances and one case study. The test instances are named N-1 to N-10 and the case study is named Case. The detail of the number of locations (#location) and the number of fields (#field) of each test instance is shown in Table 13.

Table 13. Parameter setting of test instances.

Test Instance	#Location	#Field
N-1	10	15
N-2	10	15
N-3	10	25
N-4	20	30
N-5	20	40
N-6	20	60
N-7	30	80
N-8	30	90
N-9	30	90
N-10	30	100
Case	30	110

CR is set to 0.6, F is set to 2 [13], CR_1 is set to 0.6, and CR_2 is set to 0.8. The simulation has been executed five times, and the best solutions among all five results are drawn to be representative of the algorithm. The simulation was performed in a Computer notebook Core™ i5-2467M CPU 1.6 GHz.

The stopping criterion which is used in the first test is the runtime limitation, which is set to 10 min. The computational result of the 11 test instances is shown in Table 14.

Table 14. The comparison of the current practice and the modified DE (MDE) results.

Instance	Fuel Used (L)		
	Current Practice	**DE**	**MDE**
N-1	20.3	18.47	16.2
N-2	21.4	19.2	16.1
N-3	39.4	37.1	32.1
N-4	45.8	46.3	40.2
N-5	55.4	55.6	49.2
N-6	70.3	69.4	60.1
N-7	80.4	82.5	71.3
N-8	90.4	94.3	80.2
N-9	95.4	94.2	80.5
N-10	102.1	109.4	92.4
CASE	112.3	110.1	95.4

From Table 14, DE and MDE outperform the current practice procedure. To prove that they are outperforming it statistically, the result of the statistical test using the Wilcoxon signed-rank test with a 95% confident interval is shown in Table 15.

Table 15. The statistical result of the proposed heuristics.

-	Current Practice	DE	MDE
Current Practice	-	>	>
DE	<	-	<
MDE	<	<	-

From Table 15, DE and MDE generate a significantly better solution than that of the current practice. The solution obtained from MDE is statistically better than that of DE.

Now, the performance of DE and MDE will be tested to draw the contribution of the proposed heuristics. The proposed heuristics are split into 4 sub-heuristics, which are DE (original DE), MDE-1 (DE + new mutation Formula (6)), MDE-2 (DE + local search), and MDE-3 (DE + new mutation

Formula (6) + local search). The stopping criterion in this test is the runtime limitation, which is set to 10 min. The results of all proposed heuristics are shown in Table 16.

Table 16. The results of the proposed heuristics.

-	Fuel Used (L)			
Instance	**DE**	**MDE-1**	**MDE-2**	**MDE-3**
N-1	18.47	17.82	17.98	16.2
N-2	19.2	18.2	18.03	16.1
N-3	37.1	35.3	35.63	32.1
N-4	46.3	45.02	44.5	40.2
N-5	55.6	54.61	52.3	49.2
N-6	69.4	67.31	66.5	60.1
N-7	82.5	80.57	79.14	71.3
N-8	94.3	88.9	84.3	80.2
N-9	94.2	92.3	89.5	80.5
N-10	109.4	104.2	102.56	92.4
CASE	110.1	108.76	106.85	95.4

The significance test has been executed for all results using the Wilcoxon signed-rank test with a 95% confident interval as shown in Table 17.

Table 17. The statistic test for the proposed heuristics.

-	**DE**	**MDE-1**	**MDE-2**	**MDE-3**
DE	-	>	>	>
MDE-1	-	-	>	>
MDE-2	-	-	-	>
MDE-3	-	-	-	-

From Table 17, all MDE algorithms outperform the original version of DE, and MDE-3 outperforms MDE-1 and MDE-2. This means that both the local search and the new mutation formula should be combined to get a better solution. The local search alone can improve the solution better than that of DE using the new mutation formula. The new mutation formula works well in the DE due to it statistically improving the solution from the original DE. The percentage difference of each proposed heuristic to the best solution (MDE-3) is shown in Table 18.

Table 18. The percentage difference between the proposed heuristics to the best-proposed heuristics (MDE-3).

-	Fuel Used (L)			
Instance	**DE**	**MDE-1**	**MDE-2**	**MDE-3**
N-1	14.012	10.000	10.988	0.000
N-2	19.255	13.043	11.988	0.000
N-3	15.576	9.969	10.997	0.000
N-4	15.174	11.990	10.697	0.000
N-5	13.008	10.996	6.301	0.000
N-6	15.474	11.997	10.649	0.000
N-7	15.708	13.001	10.996	0.000
N-8	17.581	10.848	5.112	0.000
N-9	17.019	14.658	11.180	0.000
N-10	18.398	12.771	10.996	0.000
CASE	15.409	14.004	12.002	0.000
Ave.	16.056	12.116	10.173	0.000

From Table 18, we can see that the best heuristic (MDE-3) has a percentage difference of fuel usage less than that of MDE-2, MDE-1, and DE of 10.173%, 12.116%, and 16.056%, respectively.

6. Conclusions

This research presents solution approaches to solving a special case of the location routing problem (SLRP). The SLRP could have a volume of latex more or less than the capacity of the vehicle. The objective function of SLRP is to minimize the fuel usage, and the fuel usage depends on the road condition and the distance of the road.

We developed a differential evolution algorithm to solve the problem. The new mutation formula is presented in the article. The new rule of applying SWAP is presented and is used as the local search in the proposed heuristics.

The heuristics proposed in the article are split into four sub-heuristics, which are DE (original DE), MDE-1 (DE + new mutation Formula (6)), MDE-2 (DE + local search), MDE-3 (DE + new mutation Formula (6) + local search). The best heuristic is MDE-3, which has a 16.056% difference from the original version of DE and is 10.173 and 12.116 better than MDE-2 and MDE-1, respectively.

From the computational result, we can see that using the new mutation formula and the local search is beneficial to the original DE. The new mutation formula and the local search is designed based on the idea of increasing the intensification of DE. This makes MDE-1, MDE-2, and MDE-3 outperform the original DE.

Author Contributions: S.K. collected the important data and designed the algorithm; R.A. designed and performed the experiment and analyzed the result.

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

References

1. Dantzig, G.; Ramser, J. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [CrossRef]
2. Lenstra, J.K.; Rinnooy Kan, A.H.G. Complexity of vehicle routing and scheduling problems. *Networks* **1981**, *11*, 221–227. [CrossRef]
3. Watson-Gandy, C.; Dohrn, P. Depot location with van salesmen—A practical approach. *Omega* **1973**, *1*, 321–329. [CrossRef]
4. Braekers, K.; Ramaekers, K.; Van Nieuwenhuyse, I. The vehicle routing problem: State of the art classification and review. *Comput. Ind. Eng.* **2015**, *99*, 300–313. [CrossRef]
5. Hasanpour, H.A.; Mosadeghkhah, M.; Tavakoli Moghadam, R. Solving a Stochastic Multi-Depot Multi-Objective Vehicle Routing Problem by A Simulated Annealing. *J. Ind. Eng.* **2009**, *43*, 25–36.
6. Min, H. Consolidation terminal location-allocation and consolidated routing problems. *J. Bus. Logist.* **1996**, *17*, 235–263.
7. Nagy, G.; Salhi, S. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *Eur. J. Oper. Res.* **2005**, *162*, 126–141. [CrossRef]
8. Barreto, S.; Ferriera, C.; Paixao, J.; Santos, B.S. Using clustering analysis in a capacitated location-routing problem. *Eur. J. Oper. Res.* **2007**, *179*, 968–977. [CrossRef]
9. Tuzun, D.; Burke, L.I. A two-phase tabu search approach to the location routing problem. *Eur. J. Oper. Res.* **1999**, *116*, 87–99. [CrossRef]
10. Lin, C.K.Y.; Chow, C.K.; Chen, A. A location-routing-loading problem for bill delivery services. *Comput. Ind. Eng.* **2002**, *43*, 5–25. [CrossRef]
11. Lin, C.K.Y.; Chow, C.K.; Chen, A. Multi-objective metaheuristics for a location-routing problem with multiple use of vehicles on real data and simulated data. *Eur. J. Oper. Res.* **2006**, *175*, 1833–1849. [CrossRef]
12. 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]

13. Pitakaso, R. Differential evolution algorithm for simple assembly line balancing type 1 (SALBP-1). *J. Ind. Prod. Eng.* **2015**, *32*, 104–114. [CrossRef]
14. 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]
15. Thongdee, T.; Pitakaso, R. Differential Evolution Algorithms Solving a Multi-Objective, Source and Stage Location-Allocation Problem. *Ind. Eng. Manag. Syst.* **2015**, *14*, 11–21. [CrossRef]
16. Storn, R.; Price, K. *Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*; Technical Report TR-95-012; ICSI: New Delhi, India, 1995.
17. Nearchou, A.C. Meta-heuristics from nature for the loop layout design problem. *Int. J. Prod. Econ.* **2006**, *101*, 312–328. [CrossRef]
18. Lampinen, J.; Zelinka, I. Mechanical engineering design optimization by differential evolution. In *New Ideas in Optimization*; Corne, D., Dorigo, M., Glover, F., Eds.; McGraw-Hill: London, UK, 1999; pp. 127–146.
19. Boon, E.T.; Ponnambalam, S.G.; Kanagaraj, G. Differential evolution algorithm with local search for capacitated vehicle routing problem. *Int. J. Bio-Inspired Comp.* **2015**, *7*, 321–342.
20. Huan, X.; Jiechang, W. Differential Evolution Algorithm for the Optimization of the Vehicle Routing Problem in Logistics. In Proceedings of the Eighth International Conference on Computational Intelligence and Security, Guangzhou, China, 17–18 November 2012; pp. 48–51.
21. Mingyong, L.; Erbao, C. An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. *Eng. Appl. Artif. Intell.* **2010**, *23*, 188–195. [CrossRef]
22. Drexl, M.; Schneider, M. A survey of variants and extensions of the location-routing problem. *Eur. J. Oper. Res.* **2015**, *241*, 283–308. [CrossRef]
23. Zhi-Zhong, L.; Yong, W.; Shengxiang, Y.; Ke, T. An Adaptive Framework to Tune the Coordinate Systems in Nature-Inspired Optimization Algorithms. *IEEE Trans. Cybern.* **2018**, *99*, 1–14.
24. Yong, W.; Zhi-Zhong, L.; Jianbin, L.; Han-Xiong, L.; Jiahai, W. On the selection of solutions for mutation in differential evolution. *Front. Comp. Sci.* **2018**, *12*, 297–315.
25. Yong, W.; Zhi-Zhong, L.; Jianbin, L.; Han-Xiong, L.; Gary, G.Y. Utilizing cumulative population distribution information in differential evolution. *Appl. Soft Comput.* **2016**, *48*, 329–346.