Local Search Algorithms for Vehicle Routing Problems of a Chain of Convenience Stores

Yuwadee Prakaiphetkul and Pisut Pongchairerks
Faculty of Engineering, Thai-Nichi Institute of Technology, Bangkok, Thailand
Email: {pr.yuwadee_st, pisut}@tni.ac.th

***Abstract*—This paper considers a generalized vehicle routing problem in a real application of a chain of convenience stores. The problem is to find the best truck routes, based on bi-criteria, for distributing a variety of products from a distribution center to convenience stores by using non-identical trucks. The distances between two places are asymmetric. This paper proposes multiple local search algorithms. Some of them use single neighborhood structure and some use double neighborhood structures. The aim of this paper is to find the local search algorithm with the specific neighborhood structure that performs very well for the generalized vehicle routing problem proposed in this paper.**

***Index Terms*—bi-criteria optimization problem, local search algorithm, neighborhood structure, vehicle routing problem**

I. Introduction

Vehicle routing problem is one of the most popular problems in the field of logistics and supply chain management. This is because this problem is often found in industries and services and it is also the heart of the way to minimize cost as well as maximize customer satisfaction. The classical vehicle routing problem, called VRP, is very simple to understand but very hard to solve to optimality. VRP starts with one depot and a fixed number of customers. Each customer is visited once by exactly one vehicle. The problem is to find the shortest vehicle routes used by identical vehicles to deliver the identical products to the customers. The loading capacity of each vehicle is limited and predefined; the demands of each customer are also predefined. Usually, VRP is very practical in many situations. However, each enterprise has its own managerial way to deliver the materials, products, people, and so on; thus, each enterprise may have its own VRP which differs from the model of the classical problem.

This paper presents a generalized vehicle routing problem simulated based on the real application of a chain of convenience stores. In this problem, the products have to be supplied from a distribution center to a number of convenience stores. An interesting point of the problem is that the distribution center does not have its own trucks, so that the distribution center has to employ a subcontractor, which has a number of trucks, to deliver the products for it. Thus, the distribution center attempts to minimize the cost of employment while the subcontractor attempts to minimize the total distances. Beyond that, the problem presented in this paper is from the real problem so that its model is more complicated than the classical problem, such as asymmetric distances, non-identical trucks, non-identical products, etc.

To solve the generalized vehicle routing problem given above, this paper proposes nine local search algorithms developed based on the well-known neighborhood structures such as the swap structure, the insert structure, the reverse structure. These nine local search algorithms consist of three single-phase local search algorithms using single neighborhood structure and six double-phase local search algorithms using double neighborhood structures.

II. Literature Review

As presented in Section I, the classical vehicle routing problem or VRP expressed in 1959 by [1] may not be practical for some enterprise's real applications. Hence, the researchers have been developed many variants of VRP to cope with the real problems which have specific conditions. Below shows some well-known variants of VRP:

1) Vehicle routing problem with delivery and pick-up, or VRPPD, is the VRP in that a number of products have to be moved from the pick-up locations to other delivery locations, see [2].
2) Vehicle routing problem with time windows, or VRPTW, is the VRP whose customers have time windows within the visits must be done, see [3].
3) Vehicle routing problem with split deliveries, or VRPSD, is the VRP where each customer can be served by more than one vehicle, see [4].
4) Vehicle routing problem with time windows and split deliveries, also known as VRPTWSD, is the VRPTW where each customer can be served by more than one vehicle, as shown in [5].
5) Open vehicle routing problem, also called OVRP, is the VRP where vehicles may not return to the depot, as shown in [6].

Note that there are many other variants of VRP given in literature. In the last fifty years, researchers have attempted to solve VRP by using many different methods which can be classified into three categories:

1) Exact algorithms which guarantee finding the optimal solution such as branch-and-bound algorithm

Manuscript received May 20, 2014; revised November 4, 2014.

doi: 10.12720/jiii.3.3.266-273

presented by [7] and branch-and-cut algorithm presented by [8].

2) Problem-based heuristics which is the simple heuristics, some of them can compute by hand easily such as nearest neighbor algorithm shown in [9], and saving algorithm presented by [10].
3) Meta-heuristics which use a higher-level procedure to find generate or select a lower-level procedure that may return a good solution such as tabu search presented by [11] and genetic algorithm presented by [12].

The local search algorithms proposed in this paper are developed based by the swap, insert, and reverse structures, as expressed in [13]. The concept of double-phase local search algorithms proposed in this paper is similar to the concept of variable neighborhood search algorithm [14] in the way to improve the search performance using multiple neighborhood structures. However, the algorithms proposed in this paper are simpler.

III. Problem Description

This paper proposes a generalized VRP simulated by the real application model from a chain of convenience stores managed by a company. This problem states with one distribution center and a fixed number of convenience stores. The company needs to distribute a variety of products from the distribution center to the convenience stores. The company however does not have its own trucks; therefore, it has to pay a lump-sum payment per each trip of a truck for a subcontractor to delivery the products. The subcontractor is the owner of all trucks which may have the different maximum loading capacities based on their sizes. The loading capacity is measured in units of cubes. Each truck must start its trip from the distribution center and end its trip at the distribution center; each truck can be used for more than one trip. Although there are many kinds of products, the products can be packed altogether into a package in units of cubes. Thus, the demands of each convenience store are also measured in units of cubes. The distances from the distribution center to each convenience store in forward direction and in backward direction may be different. As well, the distances between two convenience stores in the opposite directions may be different.

More about this proposed problem, the lump-sum amount per one trip of a truck paid from the company to the subcontractor does not depend on the distance, but depend on these two criteria:

1) Area zone of the convenience stores served by the truck in each trip
2) Truck size

The calculation of the lump-sum amount per trip just mentioned is expressed by these following case examples, using the data from the appendices. For example, if a big-size truck is assigned for the stores 31, 125 and 134 all located in the area's zone A, the company has to pay for 652 Bath for this trip. If another trip of a truck has to serve the zone A's stores 145, 190 and 245, the company will pay equally 652 Bath without considering about the distances. However, if a small-size truck is assigned to serve the zone A's stores, the company has to pay only 291 Bath. In cases that the convenience stores from two different area zones will be served by one trip of a truck. The company does not need to pay the sum of the two lump-sum amounts; the company just needs to pay the highest lump-sum amount for this trip. For example, if a big-size truck has been assigned to deliver the products for Stores 31, 125 and 134 in zone A and also for Store 54 in zone B, then, as the service charge rate for zone A is 652 Bath and that for zone B is 843 Bath, the company has to pay only 843 Bath for this trip, not 1,495 Bath.

The problem objective is to find the truck routes consuming the lowest total lump-sum amounts, and using the shortest total transportation distances as well. Thus, this problem is a bi-criteria optimization problem because some truck routes consuming the low total lump-sum amounts may use the high total transportation distances, and vice versa.

IV. Mapping Procedure

For the problem instances of *n* convenience stores, indexing from 1 to *n*, and *m* trucks, indexing from 1 to *m*, the priority of selecting the truck to distribute the products for each trip is given as follows. The trucks that have never been used for any trip will have higher priority for selecting to use than the trucks that have already been used for one trip, the trucks that have already been used for one trip will have higher priority for selecting to use than the trucks that have already been used for two trips, and so on. For two trucks that have been equally used in the number of trips, the bigger-size truck, i.e. the truck that has the higher maximum loading capacity, has the higher priority than the smaller-size truck. For two same-size trucks that have been equally used in the number of trips, the truck of the lower index number has the higher priority.

Now, the mapping procedure is given as follows. For the problem instances of *n* convenience stores and *m* trucks, the set of truck routes, as a problem solution, is represented by a permutation of *n* integer numbers starting from 1 to *n*. Each integer number appears in the permutation exactly once. The interpretation of the permutation is given as follows. Each integer number appears in the permutation represents the index number of each convenience store, i.e. '1' represents store 1. The order of appearance of each store's index number in the permutation represents the priority of the store for the order of delivery. The leftmost store's index number represents the store which has the highest priority to select for delivery; the second leftmost store's index number represents the store which has the second highest priority to select for delivery; and so on. For example, the permutation (1, 3, 2, 5, 4) means than the convenience stores can be arranged from the highest priority to the lowest priority as follows: store 1, store 3, store 2, store 5 and then store 4. This mapping procedure can be found in [15].

The process of generating the truck routes based on the predefined truck priorities and store priorities is given below. Note that every truck must start its trip at the distribution store only.

1) Assign the highest-priority truck to use in the current single round-trip.
2) The current single round-trip can be constructed as follows.
   2.1) Start the round-trip of the truck selected in step 1 from the distribution center.
   2.2) Select the highest-priority convenience store from all unvisited stores that has the demands (in cubes) less than the remaining loading capacity of the truck (in cubes); assign the truck to serve this just selected convenience store and the remaining loading capacity of this truck is then reduced by the demands of the just visited store.
   2.3) Repeat from step 2.1 until there are no any unvisited convenience store from all n stores that has the demands less than the remaining loading capacity of the truck. Then let the truck go back to the distribution center. This round-trip is now finished.
3) Repeat from step 1 for the next trip until the products have been delivered to all convenience stores successfully.

V. Proposed Algorithms

This paper proposes multiple local search algorithms based on different neighborhood structures, i.e. swap, insert and reverse. To generate a neighborhood permutation S_1 of a permutation S in the swap structure, the algorithm will randomly select two integer numbers from S and then swap the positions of these two integer numbers in the permutation. Let S_1 = swap(S) present that S_1 is a neighborhood permutation of S in the swap structure.

In the insert structure, to generate a neighborhood permutation S_1 of a permutation S, the algorithm will randomly select two integer numbers from S and then move the second selected integer number to the position in front of the first selected integer number. Let S_1 = insert(S) present that S_1 is a neighborhood permutation of S in the insert structure.

In the reverse structure, to generate a neighborhood permutation S_1 of a permutation S, the algorithm will randomly select two integer numbers from S and then all integer numbers located between these two integer numbers including themselves will be relocated in the reverse positions. That is, the leftmost number will be the rightmost number; the second leftmost number will be the second rightmost number; and so on. Let S_1 = reverse(S) present that S_1 is a neighborhood permutation of S in the reverse structure.

Let the local search algorithm using the swap structure, the local search algorithm using the insert structure, and the local search algorithm using the reverse structure are called the S algorithm, the I algorithm, and the R algorithm, respectively. Algorithm 1 given below presents the steps of the S algorithm. T is the maximum number of neighbors of which the algorithm can be used for each solution.

Algorithm 1, as the steps of the S algorithm, is given below.

1) Generate the initial permutation S_0 randomly.
2) Let t = 0.
3) Generate a neighborhood permutation S_1 = swap(S_0). If the truck routes given from S_1 returns the lower total lump-sum amounts as well as the lower total transportation distances than the truck routes given from S_0, then assign S_0 = S_1 and repeat from step 2; otherwise, t = t + 1 and repeat from step 3 until t = T then stop the algorithm. The S_0 is the best solution found by the algorithm after it is stopped.

The steps of the I algorithm are easily developed by replacing S_1 = swap(S_0) in step (3) of Algorithm 1 by S_1 = insert(S_0). The steps of the R algorithm are developed by replacing S_1 = swap(S_0) in step (3) of Algorithm 1 by S_1 = reverse(S_0).

This paper then develops the six double-phase local search algorithms. The S-I algorithm uses the swap structure in phase 1 and then the insert structure in phase 2. The S-R algorithm uses the swap structure and then the reverse structure, the I-S algorithm uses the insert structure and then the swap structure, the I-R algorithm uses the insert structure and then the reverse structure, the R-S algorithm uses the reverse structure and then the swap structure, and the R-I algorithm uses the reverse structure and then the insert structure. Algorithm 2 presents the steps of the S-I algorithm.

Algorithm 2, as the steps of S-I algorithm, is given below.

1) Generate the initial permutation S_0 randomly.
2) Let t = 0.
3) Generate a neighborhood permutation S_1 = swap(S_0). If the truck routes given from S_1 uses lower total lump-sum amounts and lower total transportation distances than the truck routes given from S_0 and , then S_0 = S_1 and repeat from step (2); otherwise, t = t + 1 and repeat from step (3) until t = T_1 then go to step (4).
4) Let t = 0.
5) Generate a neighborhood permutation S_1 = insert(S_0). If the truck routes given from S_1 uses lower total lump-sum amounts and lower total transportation distances than the truck routes given from S_0 and , then S_0 = S_1 and repeat from step 4; otherwise, t = t + 1 and repeat from step 5 until t = T_2 then step the algorithm. The S_0 is the best solution found by the algorithm after it is stopped.

The steps of the other double-phase local search algorithms can be developed by changing some parts of Algorithm 2 as follows.

1) The S-R algorithm is developed by replacing S_1 = insert(S_0) with S_1 = reverse(S_0) in step 5.
2) The I-S algorithm is developed by replacing S_1 = swap(S_0) with S_1 = insert(S_0) in step 3 and also replacing S_1 = insert(S_0) with S_1 = swap(S_0) in step 5.
3) The I-R algorithm is developed by replacing S_1 = swap(S_0) with S_1 = insert(S_0) in step 3 and also replacing S_1 = insert(S_0) with reverse(S_0) in step 5.

4) The R-S algorithm is developed by replacing S_1 = swap(S_0) by reverse(S_0) in step 3 and replacing S_1 = insert(S_0) by swap(S_0) in step 5.

The R-I algorithm is developed by replacing S_1 = swap(S_0) by S_1 = reverse(S_0) in step 3.

VI. Experiments and Results

This paper generates the three problem instances based on the real data. However, to protect the company's confidential data, all the distances, the lump-sum amounts paid for the service zones and the store's demands of the company's data are multiplied by confidential numbers and are then used as the data for the problem instances in this paper. These three problem instances use the same data, excepting the data of demands. The high-season problem instance, the medium-season problem instance and the low-season problem instances are generated by using the data collected in the high season, the medium season and the low season, respectively. For all three problem instances, the subcontractor has 145 trucks, 65 trucks of them are the small-size trucks which have the maximum loading capacity of 5.5 cubes, and 80 trucks of them are the big-size trucks which have the maximum loading capacity of 10 cubes; the company has one distribution center and 896 convenience stores. The data about distances (in km), convenience store demands (in cubes), area zones and their involved lump-sum amounts (in Bath) are given in appendixes. The convenience stores with demands of 0 cubes means that they have no demands at the day of collected data, but the truck is still required to visit there to submit some small documents or letters.

To compare the performance of the proposed algorithms on the problem, all algorithms are coded from C# program under Windows and they are executed by a 1.75 GHz AMD E2-2000 APU processor. Each algorithm will be repeated thrice with different initial solutions on each problem instances. For the single-phase local search algorithms, the maximum number of neighbors of each solution $T = n(n - 1) \div 2$. This value is used because it is the number of all outcomes of selecting two integer numbers from n integer numbers in the permutation S. For the double-phase local search algorithms, the maximum number of neighbors of each solution for the first phase $T_1 = n(n - 1) \div 4$ and the maximum number of neighbors of each solution for the second phase $T_2 = n(n - 1) \div 4$. This value makes $T_1 + T_2 = T$ and results in the fair comparisons in the computational times.

The problem proposed in this paper has bi-criteria, i.e. the total lump-sum amounts of which the company has to pay the subcontractor and the total distances of which the subcontractor's truck is used. In order to combine the two criteria into a single criterion, this paper transforms the total distances into the total gasoline fees by multiplying the gasoline fee per kilometer, i.e. 0.75 Bath; then, the new single criterion is the summation of the lump-sum amounts paid by the company and the total gasoline fees paid by the subcontractor. Note that the value of 0.75 is from the real gasoline fee per kilometer multiplied by the same confidential number used for the lump-sum amounts of the service zones.

A. *Results from High-volume Instance*

Table I shows the total lump-sum amounts (in Baht) of the truck routes given by the three runs of all algorithms on the high-volume instance. In the table, Best means the total lump-sum amounts of the best solution given by each algorithm. Based on the results shown in the table, The S algorithm performs best in the comparison, and the R-S algorithm is the second best algorithm in the comparison.

TABLE I. Lump-sum Amounts on High-volume Instance

Algorithm	1	2	3	Best
S	**230,765**	**231,557**	**230,118**	**230,118**
I	246,953	247,880	247,260	246,953
R	245,388	246,005	246,794	245,388
S-I	234,853	235,073	234,482	234,482
S-R	233,612	234,894	233,428	233,428
I-S	234,774	234,203	235,402	234,203
I-R	247,661	246,730	245,183	245,183
R-S	**233,137**	**234,453**	**232,662**	**232,662**
R-I	244,525	244,664	246,118	244,525

Later on, Table II shows the total distances (in km) consumed by the truck routes on the high-volume instance. In this table, Best is the total distances of the best solution found by each algorithm. The S algorithm performs best in the comparison, and the I-S algorithm is the second best algorithm in the comparison.

TABLE II. Distances on High-volume Instance

Algorithm	1	2	3	Best
S	**216,504**	**215,070**	**211,212**	**211,212**
I	231,217	227,124	227,803	227,124
R	227,397	229,914	230,338	227,397
S-I	220,013	218,410	219,099	218,410
S-R	219,878	216,933	218,056	216,933
I-S	**212,678**	**213,746**	**222,156**	**212,678**
I-R	228,895	227,004	225,379	225,379
R-S	214,340	221,400	225,076	214,340
R-I	228,940	230,537	224,521	224,521

TABLE III. Sums of Costs on High-volume Instance

Algorithm	1	2	3	Best
S	**393,143**	**392,859**	**388,527**	**388,527**
I	420,366	418,223	418,112	418,112
R	415,936	418,441	419,548	415,936
S-I	399,863	398,880	398,806	398,806
S-R	398,520	397,594	396,970	396,970
I-S	394,282	394,512	402,018	394,282
I-R	419,332	416,983	414,217	414,217
R-S	**393,892**	**400,503**	**401,469**	**393,892**
R-I	416,230	417,566	414,508	414,508

Table III presents the summations of the total lump-sum amounts paid by the distribution center and the total gasoline fees from the truck routes paid by the subcontractor which are generated by the nine proposed algorithms.

In Table III, the S algorithm performs best in the comparison, and the R-S algorithm is the second best algorithm in the comparison. Although I-S performs very well in the total distances, it is the fourth in the lump-sum amounts; thus, the R-S algorithm wins the I-S algorithm in the combining criterion.

B. Results from Medium-volume Instance

Table IV presents the total lump-sum amounts (in Baht) of the truck routes on the medium-volume instance. Based on the results shown in the table, again, the S algorithm performs best in the comparison, and the R-S algorithm is the second best algorithm in the comparison.

TABLE IV. LUMP-SUM AMOUNTS ON MEDIUM-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**204,576**	**204,368**	**204,313**	**204,313**
I	221,683	222,546	221,840	221,840
R	219,670	219,925	220,164	219,670
S-I	208,464	210,020	210,009	208,464
S-R	207,358	206,863	209,732	206,863
I-S	209,823	208,489	208,989	208,489
I-R	221,699	221,061	222,125	221,061
R-S	**206,535**	**208,101**	**206,651**	**206,535**
R-I	216,506	222,035	219,287	216,506

Table V then shows the total distances (in km) used by the truck routes on the medium instance. The S algorithm is the highest performance algorithm and the S-R algorithm is the second best algorithm in the comparison.

TABLE V. DISTANCES ON MEDIUM-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**187,899**	**179,425**	**184,074**	**179,425**
I	214,702	218,096	212,337	212,337
R	218,710	213,421	214,055	213,421
S-I	192,560	193,491	194,351	192,560
S-R	**188,945**	**186,490**	**192,254**	**186,490**
I-S	195,874	196,673	193,540	193,540
I-R	216,083	216,709	215,754	215,754
R-S	193,799	197,177	191,139	191,139
R-I	214,160	218,976	216,735	214,160

TABLE VI. SUMS OF COSTS ON MEDIUM-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**345,500**	**338,937**	**342,369**	**338,937**
I	382,889	386,118	381,093	381,093
R	383,702	379,991	380,706	379,991
S-I	352,884	355,139	355,772	352,884
S-R	**349,067**	**346,731**	**353,922**	**346,731**
I-S	356,729	355,994	354,143	354,143
I-R	383,761	383,592	383,941	383,592
R-S	351,885	355,983	350,005	350,005
R-I	377,125	386,267	381,838	377,125

Table VI shows the summations of the total lump-sum amounts and the total gasoline fees from the truck routes generated by the algorithms. In this table, the S algorithm performs best in the comparison, and the S-R algorithm is the second best algorithm in the comparison. Although the R-S algorithm performs slightly better than the S-R algorithm in the criterion of the total lump-sum amounts, the S-R algorithm is better than the S-R algorithm in the combining criterion because the S-R algorithm performs much better than the R-S algorithm in the criterion of the total distances.

C. Results from Low-volume Instance

Table VII provides the total lump-sum amounts (in Baht) of the truck routes given on the low-volume instance. Based on the results shown in the table, The S algorithm performs best and the R-S algorithm is the second in the comparison.

TABLE VII. LUMP-SUM AMOUNTS ON LOW-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**166,747**	**167,617**	**166,214**	**166,214**
I	180,383	180,517	180,128	180,128
R	176,589	178,095	178,801	176,589
S-I	169,910	171,739	170,840	169,910
S-R	169,236	171,460	169,450	169,236
I-S	170,362	169,838	169,395	169,395
I-R	180,834	179,832	179,424	179,424
R-S	**169,361**	**170,387**	**169,020**	**169,020**
R-I	178,763	177,701	177,631	177,613

Table VIII shows the total distances (in km) consumed by the truck routes on the low-volume instance. The S algorithm is the best and the S-R algorithm is the second best in the comparison.

TABLE VIII. DISTANCES ON LOW-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**208,396**	**207,785**	**208,884**	**207,785**
I	219,358	224,519	216,592	216,592
R	218,722	221,368	222,294	218,722
S-I	213,576	214,637	213,314	213,314
S-R	**209,384**	**209,755**	**215,310**	**209,384**
I-S	215,933	216,233	215,736	215,736
I-R	221,406	220,742	223,159	220,742
R-S	215,781	216,260	214,416	214,416
R-I	220,996	219,762	220,481	219,762

TABLE IX. SUMS OF COSTS ON LOW-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**323,044**	**323,455**	**322,877**	**322,877**
I	344,902	348,906	342,572	342,572
R	340,630	344,121	345,522	340,630
S-I	330,091	332,717	330,826	330,091
S-R	**326,274**	**328,776**	**330,932**	**326,274**
I-S	332,312	332,012	331,197	331,197
I-R	346,888	345,389	346,793	345,389
R-S	331,197	332,581	329,831	329,831
R-I	344,510	342,522	342,974	342,522

Table IX shows the summations of the total lump-sum amounts and the total gasoline fees from the truck routes generated by the algorithms. In this table, the S algorithm performs best, and the S-R algorithm is the second best algorithm in the comparison. Again, although the R-S algorithm performs slightly better than the S-R algorithm in the criterion of the total lump-sum amounts, the S-R algorithm is better than the S-R algorithm in the combining criterion because the S-R algorithm performs much better than the R-S algorithm in the criterion of the total distances.

Based on the three problem instances' results in Tables I, IV and VII, the S algorithm is the best algorithm and the R-S algorithm is the second best algorithm in the comparison of the total lump-sum amounts spent by the company. This paper then do the hypothesis test of H0: the population mean of the differences between the best solution values of the S algorithm and the best solution values of the R-S algorithm is zero versus H1: the population mean of the differences is less than zero for the significance level of 0.10 by using Minitab. Hereafter, in all hypothesis tests on this paper, a best solution value means a best solution value taken from any three solutions. The Minitab program returns the *p*-value of 0.002; thus it concludes that the mean of the best solution values of the S algorithm is better than the mean of the best solution values of the R-S algorithm by the significance level of 0.10 in the criterion of the total lump-sum amounts.

In the comparison of the total distances of the truck routes spent by the subcontractor on the three problem instances based on data from Table II, V and VIII, the S algorithm is the best algorithm and the S-R algorithm is the second best algorithm in the comparison. This paper then do the hypothesis test of H0: the population mean of the differences between the best solution values of the S algorithm and the best solution values of the S-R algorithm is zero versus H1: the population mean of the differences is less than zero for the significance level of 0.10 by using Minitab. The Minitab program returns the *p*-value of 0.05; thus it concludes that the mean of the best solution values of the S algorithm is better than the mean of the best solution values of the S-R algorithm by the significance level of 0.10 in the criterion of the total distances.

Based on the three problem instances' results in Tables III, VI and IX, the S algorithm is the best algorithm and the S-R algorithm is the second best algorithm in the criterion of the sum of the total lump-sum amounts and the total gasoline fees. This paper then do the hypothesis test of H0: the population mean of the differences between the best solution values of the S algorithm and the best solution values of the S-R algorithm is zero versus H1: the population mean of the differences is less than zero for the significance level of 0.10 by using Minitab. The Minitab program returns the *p*-value of 0.027; thus it concludes that the mean of the best solution values of the S algorithm is better than the mean of the best solution values of the R-S algorithm by the significance level of 0.10 in the combining criterion.

All proposed algorithms perform equally in terms of computational time consumption. Average computational time per run of each algorithm is given as follows: S algorithm uses 23 minutes, I algorithm uses 26 minutes, S-I algorithm uses 22 minutes, S-R algorithm uses 24 minutes, I-S algorithm uses 24 minutes, I-R algorithm uses 26 minutes, R-S algorithm uses 24 minutes and R-I algorithm uses 26 minutes.

Based on the experiment results, the S algorithm performs best both in the total lump-sum amounts paid by the company and the total distances consumed by the subcontractor. The R-S algorithm performs as the second-best in the objective function of total lump-sum amounts and the S-R algorithm performs as the second-best in the objective function of total distances.

VII. Additional Experiments

Based on the findings from Section VI, the S algorithm outperforms other algorithms when the single-phase algorithms use $T = n(n-1) \div 2$ and the double-phase algorithms use $T_1 = T_2 = n(n-1) \div 4$. However, it is possible that the results may change if T increases because the S algorithm may get struck in the local optimum when T is increased. And it will open the change to the double-phase local search algorithms, e.g. the S-R algorithm and the R-S algorithm, to win the competition.

Thus, this section will run each of the S algorithm, S-R algorithm and R-S algorithm thrice on the three problem instances by using $T = n(n-1)$ for the S algorithm, and $T_1 = T_2 = n(n-1) \div 2$ for the S-R algorithm and the R-S algorithm. This section selects only the S-R and R-S algorithms to compare to the S algorithm because they perform well in all criteria. The results of total lump-sum amounts on the three instances are given in Tables X, XI and XII.

TABLE X. Lump-sum Amounts on High-volume Instance

Algorithm	1	2	3	Best
S	**226,406**	**226,856**	**226,694**	**226,406**
S-R	230,623	228,345	231,033	228,345
R-S	230,726	230,164	227,982	227,982

TABLE XI. Lump-sum Amounts on Medium-volume Instance

Algorithm	1	2	3	Best
S	**202,387**	**201,685**	**200,425**	**200,425**
S-R	205,569	203,976	204,381	203,976
R-S	203,616	203,176	203,400	203,176

TABLE XII. Lump-sum Amounts on Low-volume Instance

Algorithm	1	2	3	Best
S	**162,546**	**166,556**	**163,713**	**162,546**
S-R	166,556	166,048	167,426	166,048
R-S	163,713	166,181	166,292	163,713

As shown in Tables X, XI and XII, the sample mean of the best solution values of S algorithm is the lowest, the R-S algorithm is the second best algorithm, and the S-R algorithm is worst in the competition. Then, this paper do the hypothesis test of H0: the population mean of the differences between the best solution values of the S algorithm and the best solution values of the R-S algorithm is zero versus H1: the population mean of the differences between the best solution values of the S algorithm and the best solution values of the R-S algorithm is less than zero, using the significance level of 0.1. The Minitab program returns the *p*-value of 0.031; thus, it concludes that the population mean of the best solution values of the S algorithm is better than the population mean of the best solution values of the R-S algorithm by the significance level of 0.10 in the criterion of the total lump-sum amounts.

Later on, the results of total distances generated by the three algorithms on the three instances are presented in Tables XIII, XIV and XV, respectively. Based on the data from these tables, the sample mean of the best solution values of the S algorithm is the lowest, the sample mean of the best solution values of the S-R algorithm is the second lowest, and the sample mean of the best solution values of the R-S algorithm is worst in the criterion of the total distances used by the trucks.

TABLE XIII. DISTANCES ON HIGH-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**210,573**	**211,354**	**209,202**	**209,202**
S-R	216,352	213,533	214,104	213,533
R-S	215,040	217,315	212,610	212,610

TABLE XIV. DISTANCES ON MEDIUM-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**178,907**	**179,728**	**176,985**	**176,985**
S-R	188,105	180,149	182,010	180,149
R-S	183,847	189,307	190,813	183,847

TABLE XV. DISTANCES ON LOW-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**205,882**	**205,983**	**200,760**	**200,760**
S-R	211,485	207,176	210,577	207,176
R-S	208,732	210,100	206,292	206,292

This paper do the hypothesis test of H0: the population mean of the differences between the best solution values of the S algorithm and the best solution values of the S-R algorithm is zero versus H1: the population mean of the differences between the best solution values of the S algorithm and the best solution values of the S-R algorithm is less than zero, using the significance level of 0.1. The Minitab program returns the *p*-value of 0.02; thus, it concludes that the mean of the best solution values of the S algorithm is lesser than the mean of the best solution values generated by the S-R algorithm by the significance level of 0.10 in the criterion of the total distances.

Based on the results received, the S algorithm is the highest performance algorithm in both criteria. This result of competition is different from the prediction mentioned earlier; it is expected that one of the S-R and R-S algorithms will win the S algorithm when increasing the value of T. The results from this paper show that the swap structure dominates over other structures for the proposed problem. However, the second best algorithms in the lump-sum amounts and in the total distances are different. The R-S algorithm is the second in the lump-sum amounts while the S-R algorithm is the second in the total distances. Thus, which one is the real second best algorithm is judged by using the combining criterion.

Tables XVI, XVII, and XVIII show the summations of the total lump-sum amounts and the total gasoline fees from the truck routes generated by the algorithms on the three problem instances.

TABLE XVI. SUMMATION OF COSTS ON HIGH-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**384,335**	**385,371**	**383,595**	**383,595**
S-R	392,886	388,495	391,611	388,495
R-S	392,006	393,151	387,439	387,439

TABLE XVII. SUMMATION OF COSTS ON MEDIUM-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**336,567**	**336,481**	**333,164**	**333,164**
S-R	346,648	339,088	340,888	339,088
R-S	341,501	345,156	346,510	341,501

TABLE XVIII. SUMMATION OF COSTS ON LOW-VOLUME INSTANCE

Algorithm	1	2	3	Best
S	**316,957**	**318242**	**315,440**	**315,440**
S-R	325,170	321,430	325,358	321,430
R-S	320,262	323,756	321,011	320,262

The results from the tables show that the sample mean of the best solution values of the S algorithm is lowest, that of the S-R algorithm is second lowest. For the hypothesis test of H0: the population mean of the differences between the best solution values of the S algorithm and that of the S-R algorithm is zero versus H1: the population mean of the differences between the best solution values of the S algorithm and the best solution values of the S-R algorithm is less than zero using the significance level of 0.1, the *p*-value is 0.002. Thus, it concludes that the population mean of the best solution values of the S algorithm is less than that of the S-R algorithm with the significance level of 0.1. This shows that the S algorithm is the highest performing algorithm for all criteria.

In the comparison between the S-R algorithm and the R-S algorithm, the hypothesis test of H0: the population mean of the differences between the best solution values of the S-R algorithm and the best solution values of the R-S algorithm is zero versus H1: the population mean of the differences between the best solution values of the S-R algorithm and the best solution values of the R-S algorithm is less than zero using the significance level of 0.1 returns the *p*-value of 0.481. Thus, it concludes that there is not different between the population mean of the best solution values of the S-R algorithm and that of the R-S algorithm with the significance level of 0.1.

VIII. CONCLUSION

This paper introduces the generalized VRP with the bi-criteria. The VRP aims to deliver products from a distribution center to the convenience stores. However, the distribution center itself has no the trucks; thus, it uses the subcontractor to deliver its products. The distribution center has to pay the subcontractor trip-by-trip based on the service area's zone traveled by a truck. This problem considers the criterion of the total lump-sum amount paid by the distribution center to the subcontractor and the criterion of the total distances used by the trucks of the subcontractor. To cope with the problem, this paper introduces the nine algorithms, i.e. the S algorithm, I algorithm, R algorithm, S-I algorithm, S-R algorithm, I-S algorithm, I-R algorithm, R-S algorithm, and R-I algorithm. These proposed algorithms are tested on the three problem instances, i.e. the high-volume instance, medium-volume instance and low-volume instance. The results of the experiments shows the S algorithm outperforms on all criteria.

APPENDIXES

The data of the problem instances used in this paper are shown in the three appendices, namely Appendixes A–C, which can be accessed online via the following link: https://drive.google.com/file/d/0B2XqS3TSsvP7UWg4dU51dEJOMFk/edit?usp=sharing. Appendix A provides the details of the distances between the distribution center and each convenience store, and the distances between two convenience stores. Appendix B provides the details of the demands of the convenience stores for the three problem instance. Appendix C provides the details of the area zones and the corresponding lump-sum amounts for the big-size trucks and the small-size trucks.

ACKNOWLEDGMENT

We would like to give special thanks to Mr. Adisak Anusornsrirung for his technical support of debugging the computer programs of the algorithms.

REFERENCES

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80-91, 1959.

[2] K. Ganesh and T. T. Narendran, "CLOVES: A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up," *European Journal of Operational Research*, vol. 178, no. 3, pp. 699-717, 2002.

[3] P. Badeau, F. Guertin, M. Gendreau, J. Potvin, and E. Taillard, "A parallel tabu search heuristic for the vehicle routing problem with time windows," *Transportation Research Part C*, vol. 5, no. 2, pp. 109-122, 1997.

[4] D. Gulczynski, B. Golden, and E. Wasil, "The split delivery vehicle routing problem with minimum delivery amounts," *Transportation Research Part E*, vol. 46, no. 5, pp. 612-626, 2010.

[5] S. C. Ho and D. Haugland, "A tabu search heuristic for the vehicle routing problem with time windows and split deliveries," *Computers & Operations Research*, vol. 31, no. 12, pp. 1947-1964, 2004.

[6] F. Li, B. Golden, and E. Wasil, "The open vehicle routing problem: algorithms, large-scale test problems, and computational results," *Computers & Operations Research*, vol. 34, no. 10, pp. 2918-2930, 2007.

[7] G. H. Dastghaibifard, E. Ansari, S. M. Sheykhalishahi, A. Bavandpouri, and E. Ashoor, "A parallel branch and bound algorithm for vehicle routing problem," in *Proc. International Multi Conference of Engineers and Computer Scientists*, 2008.

[8] J. F. Bard, L. Huang, M. Dror, and P. Jaillet, "A branch and cut algorithm for the VRP with satellite facilities," *IIE Transactions*, pp. 821-834, 1998.

[9] P. C. Pop, C. P. Sitar, I. Zelina, V. Lupse, and C. Chira, "Heuristic algorithms for solving the generalized vehicle routing problem," *International Journal of Computers, Communications & Control*, vol. 4, no. 1, pp. 158-165, 2011.

[10] T. Pichpibul and R. Kawtummachai, "New enhancement for Clarke-Wright savings algorithm to optimize the capacitated vehicle routing problem," *European Journal of Scientific Research*, vol. 78, no. 1, pp. 119-134, 2012.

[11] H. Jia, Y. Li, B. Dong, and H. Ya, "An improved tabu search approach to vehicle routing problem," *Procedia–Social and Behavioral Sciences*, vol. 96, pp. 1208-1217, 2013.

[12] B. M. Baker and M. A. Ayechew, "A genetic algorithm for the vehicle routing problem," *Computers & Operations Research*, vol. 30, no. 5, pp. 787-800, 2003.

[13] E. Aarts and J. K. Lenstra, *Local Search in Combinatorial Optimization*, New Jersey: Princeton University Press, 2003.

[14] K. Fleszar, I. H. Osman, and K. S. Hindi, "A variable neighborhood search algorithm for the open vehicle routing problem," *European Journal of Operational Research*, vol. 195, no. 3, pp. 803-809, 2009.

[15] W. Y. Szeto, Y. Wu, and S. C. Ho, "An artificial bee colony algorithm for the capacitated vehicle routing problem," *European Journal of Operational Research*, vol. 215, no. 1, pp. 126-135, 2011.

Yuwadee Prakaiphetkul is a master's degree student of Master Program in Engineering Technology. B.Sc. in Applied Statistics from King Mongkut's Institute of Technology Ladkrabang,, Bangkok, Thailand. She works as MIS Logistics officer of convenience store business and worked as process engineer at Furukawa FITEL. Her thesis's title is to develop the local search algorithms by using multiple neighborhood structures for a generalized vehicle routing problem in real world situations.

Pisut Pongchairerks is an Assistant Professor in Production Engineering Program, Faculty of Engineering, Thai-Nichi Institute of Technology. He holds B.Eng. and M.Eng. in Industrial Engineering from Kasetsart University and D.Eng. in Industrial Engineering & Management from Asian Institute of Technology. His research interests focus on Operations Research, Production Planning and Control, Scheduling and Sequencing, Meta-heuristic algorithms for Optimization Problems, Logistics and Supply Chain Management, and Quality Control.