Jurnal
Teknologi

Full Paper

A VARIABLE NEIGHBORHOOD SEARCH FOR THE HETEROGENEOUS FIXED FLEET VEHICLE ROUTING PROBLEM

Article history
Received
13 November 2015
Received in revised form
18 April 2016
Accepted
15 August 2016

Arif Imran[a,d], Martino Luis[b]*, Liane Okdinawati[c]

[a]Department of Industrial Engineering, Institut Teknologi Nasional, Bandung 40124, Indonesia
[b]Othman Yeop Abdullah Graduate School of Business, Universiti Utara Malaysia, 06010 UUM Sintok, Kedah Darul Aman, Malaysia
[c]Business Logistics Department, Politeknik Pos Indonesia, Bandung, 40151, Indonesia
[d]School of Industrial Engineering, Telkom University, Bandung 40257, Indonesia

*Corresponding author
martino@uum.edu.my

Abstract

The heterogeneous fixed fleet vehicle routing problem (HFFVRP) is investigated using the variable neighborhood search (VNS). The initial solution is generated using the Sweep algorithm and the 2-opt procedure and then the customers are allocated to the smallest vehicle first by considering vehicle occupancy level. The proposed VNS algorithm uses several neighborhoods and a number of local search methods which are adapted for this problem. The local searches are implemented within a multi-level framework. The performance of the proposed algorithm is then tested using data set taken from literature and the experiments show competitive results with less computing time. Future research directions are also highlighted.

Keywords: Heuristic, routing, heterogeneous, multi-level, variable neighborhood search

1.0 INTRODUCTION

The heterogeneous fleet vehicle routing problem (HFVRP) is a variant of the vehicle routing problem (VRP). The vehicles used have different characteristics, such as age and capacity. Here we have a certain number of vehicle types (K), each of which has a vehicle capacity Q_k, a fixed cost F_k and a running cost R_k , say k=1,..., K. As in the VRP, each vehicle starts and finishes at the same depot, the capacity of a vehicle and the maximum length of the route cannot be exceeded, and each customer must be served by one vehicle only. Unlike the classical VRP, the aim is not only to consider the routing of the vehicles, but also the composition of the vehicle fleet.

In the HFFVRP we also have to consider the limited number of vehicles available. It makes the HFFVRP more complex and harder to solve than the HFVRP. Though the HFFVRP is one of the most practical variant in distribution management, as in practice no company has an unlimited number of vehicles, it appears to suffer from the lack of investigation. One of the reasons is as stated by Taillard [1], the HFFVRP can be more difficult to solve than the HFVRP due to the restriction on the number of vehicles per type used.

The remainder of the paper is organized as follows. In the next section, the literature review is provided. Section 3 presents mathematical model of the heterogeneous fixed fleet vehicle routing problem. In Section 4, the implementation of the VNS is described. The following section thereafter discusses the details of the proposed VNS algorithm. In Section 6, the computational results and discussion are given. Finally, the last section presents our conclusions and highlights some future research avenues.

2.0 LITERATURE REVIEW

Taillard [1] was among the first who addressed the HFFVRP although this idea was already mentioned by Salhi *et al*. [2]. Taillard [1] presented a heuristic using a column generation method for solving the HFVRP and the HFFVRP. The method uses tabu search to generate a set of solutions and then solves a set partitioning problem to select the best combination of routes.

Tarantilis *et al.* [3] proposed a variant of the threshold accepting (TA) heuristic known as the list based threshold accepting (LBTA). A list of threshold values which is obtained from the second step of their algorithm is created and used as a basis for accepting the new solution in the next searching process. If the value of the new threshold is smaller than the value of the threshold in the list, then the threshold value of the new solution is inserted into the list and the old threshold value is potted from the list. The process is continued until there is no new threshold value which is smaller or equal to the one in the list.

Another variant of the threshold accepting heuristic, the backtracking adaptive threshold accepting (BATA), is also put forward by Tarantilis *et al*. [4]. BATA algorithm differs from the generic threshold accepting meta-heuristic as the threshold value in BATA algorithm is more flexible than the used in the classical TA. Li *et al*. [5] adapted a record-to-record travel algorithm originally proposed by Dueck [6] which is a deterministic variant of the simulated annealing. They also solved large-scale generated test problems. Euchi and Chabchoub [7] designed tabu search with adaptive memory to deal with HFFVRP. Brandao [8] proposed tabu search algorithm that equipped with additional features such as strategic oscillation, shaking and frequency-based memory. The tabu search algorithm produces excellent results when compared to the existing results.

2.1 Applications of the HFFVRP

Tarantilis and Kiranoudis [9] presented a threshold accepting approach to address a problem arising from the distribution of fresh milk for a densely populated area in Athens, Greece. The company has to determine the optimal fleet composition and their associated set of feasible routes with a minimal cost. This approach yields considerable improvements in the operational performance of the company. Tarantilis and Kiranoudis [10] proposed a metaheuristic to solve two case studies from a dairy and a construction company. The metaheuristic is based on the Bone Route. Application of the HFFVRP can be found in residential and commercial waste collection, where it is common for fleets to have different capacity vehicles (see for example, Levy [11]). Levy [11] also mentioned the use of the HFFVRP in distribution of the single copy newspaper that has a varied fleet.

3.0 MATHEMATICAL MODEL OF THE HETEROGENEOUS FIXED FLEET VEHICLE ROUTING PROBLEM

The following mathematical formulation for HFFVRP is extended from the one of Salhi and Rand [12] which is given for the heterogeneous fleet vehicle routing problem by introducing the fixed number of each type of vehicle.

Minimize: $\sum_{k=1}^{K} F_k \sum_{j=1}^{N} X_{0jk} + \sum_{k=1}^{K}\sum_{i=0}^{N}\sum_{j=0}^{N} C_{ij} . X_{ijk}$ (1)

Subject to:

$$\sum_{k=0}^{K} F_k \sum_{j=0}^{N} X_{ijk} = 1,\ j = 1,\dots,N \quad (2)$$

$$\sum_{i=0}^{N} X_{ijk} - \sum_{l=0}^{N} X_{jlk} = 0$$

$$j = 0,\dots,N, \quad k = 1,\dots,K, \quad (3)$$

$$\sum_{i=0}^{N} Y_{ij} - \sum_{l=0}^{N} Y_{jl} = q_j,\ j = 1,\dots,N, \quad (4)$$

$$\sum_{i=1}^{K} Y_{i0} = 0 \quad (5)$$

$$\sum_{j=1}^{N} Y_{oj} = \sum_{j=1}^{N} q_i, \quad (6)$$

$$Y_{ij} \le \sum_{k=1}^{K} X_{ijk}. Q_k,\ i \ne j = 0,\dots,N, \quad (7)$$

$$R_{ij} \le \sum_{k=1}^{K} T_k . X_{ijk}, i \ne j = 0,\dots,N, \quad (8)$$

$$R_{0j} = \sum_{k=1}^{K} X_{0jk} . T_k - \sum_{k=1}^{K} X_{0jk}. t_{0j},\ j = 1,\dots,N \quad (9)$$

$$\sum_{i=0}^{N} R_{ip} - \sum_{j=0}^{N} R_{pj} = \sum_{k=1}^{K}\sum_{j=0}^{N} X_{pjk} . t_{pj},\ p = 1,\dots,N \quad (10)$$

$$\sum_{j=1}^{N} X_{ojk} \le N_k,\ k = 1,\dots,K, \quad (11)$$

$$X_{ijk} \in \{0,1\},\ Y_{ij} \ge 0,\ Y_{ii} = 0,\ R_{ij} \ge 0,$$

$$i = 0,\dots,N,\ j = 0,\dots,N,\ k = 1,\dots,K, \quad (12)$$

Here *N* is the number of customers, '*0*' is the depot and *K* the number of vehicle types. Vehicle fixed cost, capacity, and the maximum time allowed for a vehicle of type *k* are denoted by F_k, Q_k and T_k respectively. q_j is the j^{th} customer demand, t_{ij} and C_{ij} denote the time and the running cost to travel link *ij* (link between customer *i* and customer *j*).

Decision variables:
$X_{ijk} = 1$ if a vehicle of type *k* travels link *ij*, 0 otherwise.
Y_{ij} is a continuous variable which denotes the amount of goods carried between customers *i* and *j*.
R_{ij} is also a continuous variable and denotes the spare distance (time) a vehicle has after covering link *ij*.

Note: $\sum_{j=1}^{N} X_{0jk}$ represents the number of vehicles of type *k* used in the fleet.

The objective function (1) consists of the sum of the total fixed cost (first term) and the total running cost (second term). Constraints (2) show that each customer is visited only once and Constraints (3) ensure that each vehicle enters and leaves each customer location. Constraints (4) denote that the difference between the total amount of goods in a vehicle arriving at a customer and leaving such a customer is exactly that customer demand. Equation (5) shows that nothing is returned to the depot (empty vehicles) and Equation (6) denotes that the total quantity leaving the depot is equal to the total customer demand. Constraints (7) ensure that goods travel from *i* to *j* only when there is a vehicle travelling from *i* to *j*. This also shows that the total load on link *ij* should not exceed the capacity of vehicle assigned to that trip (including links of the form 0*j*). In other terms, constraints (4)-(7) represent capacity restriction and sub tour elimination constraints. Constraints (8) denote that there is a spare time for travelling along link *ij*. Constraints (9) guarantee that the spare time (distance) after covering links leaving the depot does not exceed the maximum time minus the time required to travel to that customer. Equation (10) means that every time a vehicle travels between two customers, its spare time is reduced by the distance of that link. Constraints (12) represent the limited number of vehicles of each type.

4.0 ADAPTATION OF THE VARIABLE NEIGHBORHOOD SEARCH

VNS was officially proposed by Brimberg and Mladenović [13] and Hansen and Mladenović [14] for tackling facility location problems. VNS has been widely implemented to solve various combinatorial and global optimization problems including location problems, vehicle routing problems, travelling salesman problems, scheduling, and partition problems, among others. The works of [15, 16, and 17] provide comprehensive reviews on variants and successful applications of VNS. The main reasoning of this metaheuristic is based on the idea of a systematic change of neighborhoods within a local search method.

4.1 The Proposed VNS Algorithm

In this study, the basic VNS algorithm is adapted to solve the HFFVRP. The basic VNS algorithm is enhanced by the use of additional features which include adopting a set of local search procedures and using Multi-Level heuristic of Salhi and Sari [18] to manage the local searches. The proposed VNS algorithm is described below.

Step 1: _Initialization_. Define a set of neighborhood structures N_k, for $k = 1,\ldots,k_{\max}$ and a set of local searches R_l, for $l = 1,\ldots,l_{\max}$ .Generate an initial solution x and set $x_{best} = x$.

Step 2: Set $k \leftarrow 1$

Step 3: _Repeat_ the following steps until $k = k_{\max}$:

- **(a)** _Shaking._ Generate a point x' at random from the k^{th} neighborhood of x $(x' \in N_k(x))$
- **(b)** _Local search:_ Apply a multi-level approach to find the best neighbour x''.
- **(c)** _Move or not._ If the local optimum x'' is better than the incumbent x, set $x \leftarrow x''$ and go to (2); otherwise set $k \leftarrow k+1$.

4.2 An Overview of The Proposed Algorithm

An initial solution x is first generated and it is used as the initial global best, x_{best}. We have a set of neighborhood structures N_k, $(k = 1,\ldots, k_{\max})$ and a set of refinement procedures which will be described later. The search begins by generating a random feasible solution x' from $N_1(x)$, which is taken as the temporary solution. x' is then improved by the set of local searches (refinement procedures) which are implemented within a Multi-Level framework. If the solution obtained by the Multi-level approach, x'', is better than the incumbent best solution x, then $x = x''$ and the search reverts back to N_1. However, if x'' is found to be worse or the same as x, we generate x' from the next neighborhood say $N_k(x)$ and apply the multi-level approach again. The process is repeated until the search reaches $N_{k_{\max}}$.

5.0 EXPLANATION OF THE MAIN STEPS

This section focuses on the procedures used within the steps of the algorithm.

5.1 Initial Solution (Step 1)

The initial solution is obtained in three steps; (a) construct a giant tour using the sweep algorithm of Gillett and Miller [19], (b) improve this tour using the 2-opt of Lin [20], and (c) construct the cost network, starting from the smallest vehicles. To avoid using the largest distance between two successive customers in a given route, we use similar procedure applied in Imran *et al*. [21], the starting points, in the construction of the cost network, are used as those that generate the largest distances between two successive customers (i.e. gaps) in the giant tour.

5.2 Neighborhood Structures (Step 3a)

Four neighborhoods, which are briefly described in this subsection, are used in this study (i.e. $k_{\max} = 4$). These include the 1-1 interchange (swap), the 2-0 shift, the 2-1 interchange, and the perturbation. The order of the neighborhoods is as follows; the 1-1 interchange is used as N_1, the 2-0 shift as N_2, the 2-1 interchange as N_3, and the perturbation as N_4.

Unlike in the HFVRP, for the HFFVRP we cannot remove the vehicle if neighborhoods such as 2-0 shift, 2-1 interchange and perturbation produce an empty route as the vehicle number is fixed. This is also applied for the local search (1-0 insertion inter route) used in our algorithm (Step 3b).

The 1-1 interchange (the swap procedure)

This neighborhood is aimed at generating a feasible solution by swapping a pair of customers from two routes. This procedure starts by taking a random customer from a randomly chosen route and tries to swap it systematically with other customers by taking into consideration all other routes. This procedure is repeated until a feasible move is found.

The 2-0 shift

In the 2-0 shift, two consecutive random customers from a randomly chosen route are selected. These two customers are considered together for possible insertion in other routes in a systematic manner. This procedure is reiterated until a feasible move is found.

The 2-1 interchange

This type of insertion attempts to shift two consecutive random customers from a randomly chosen route to another route selected systematically while getting one customer from the receiver route until a feasible move is obtained.

A perturbation mechanism

This scheme was initially developed by Salhi and Rand [22] for the VRP by considering three routes simultaneously. Here, it starts by taking a random customer from a randomly chosen route and tries to relocate that customer into another route without considering capacity and time constraints in the receiver route. A customer from the receiver route is then shifted to the third route if both capacity and time constraints for the second and the third route are not violated.

5.3 Local Search (Step 3b)

Six refinement procedures are adopted to make up our local search. The order of the refinement procedures is as follows: the 1-insertion intra-route as the first refinement procedure R_1, the 2-insertion intra-route as R_2, the swap intra-route as R_3, 2-opt intra-route as R_4, 1-insertion inter-route as R_5, and finally the 2-opt inter-route as R_6. This is similar with the order used in Imran *et al.* [21] but in this study we have to consider fixed vehicle number.

The process starts by generating a random feasible solution x' from N_1, which is used as the temporary solution. The multi-level approach then starts by finding the best solution x'' using R_1. If x'' is better than x', then $x' = x''$ and the search returns to R_1, otherwise the next refinement procedure is applied. This process is repeated until R_4 cannot produce a better solution.

The 1-insertion procedures (inter-route and intra-route)

Two types of the 1-insertion procedures are used. The first is the 1-insertion intra-route and the second is the 1-insertion inter-route. In the 1-insertion intra-route we remove a customer from its position in a route and try to insert it elsewhere within that route in order to have a better solution. Meanwhile, in the 1-insertion inter-route, each customer from a route is shifted from its position and tried to be inserted elsewhere into another route. If this shifting does not violate any constraints and improves the solution, the selected customer is then permanently removed.

The 2-insertion (intra-route)

The 2-insertion intra-route allows us to remove two consecutive customers and insert them elsewhere within a route to produce a cheaper route.

The 2-opt (inter-route and intra-route)

The 2-opt intra-route, usually refers to as the 2-opt (see [20]), is a simple and an effective improvement procedure that works by removing two non-adjacent arcs and adding two new arcs while maintaining the tour structure. A given exchange is accepted if the resulting total cost is lower than the previous total cost. The exchange process is continued until no further improvement can be found. The 2-opt inter-route is similar to the 2-opt intra-route except that it considers two routes where each of the two arcs belongs to a different route and reverses directions of the corresponding affected path of each route.

The swap (intra-route)

The swap intra-route is aimed at reducing the total cost of a route by swapping positions of a pair of customers within the route.

6.0 RESULTS AND DISCUSSION

The proposed VNS algorithm was programmed in C++ and performed on a PC with an Intel Core i3 CPU @ 2.10GHz processor, 4.00 GB of RAM. To evaluate the performance of the proposed VNS algorithm, benchmark HFFVRP instances of Taillard [1] were used (see Table 1). Taillard [1] adapted 8 problems of Golden *et al*. [23] that were originally used for the heterogeneous fleet VRP. For the HFFVRP instances the variable cost per unit distance for each type of vehicle and the number of vehicles of each type are added to the Golden *et al.* [23] instances.

The problems are using the numbering scheme of Golden *et al.* [23], such as 13, 14..., 20. The computational results and CPU time are given in Table 2 and Table 3 respectively. Table 2 displays the solutions obtained by published methods and VNS. The best results are given in **bold**. Number of best solution (NB) for each method is given in the last row of Table 2. For each of the 8 instances, the deviation (in %) is computed as follows: (cost (heuristic) - cost (best)) / cost(best)) x 100. The average deviation (AVD) is then computed over all instances in the data set. The solutions of the VNS algorithm are then compared to the results of Taillard [1], Tarantilis *et al*. [3, 4], Li *et al*. [5] and Brandao [8]. It can be seen in Table 2 that the proposed VNS algorithm produces good results. It produced one best solution, similar with Tarantilis *et al.* [4]. Taillard [1] and Tarantilis *et al.* [3] did not produce best solution, Li *et al.* [5] and Brandao [8] obtained 7 best solutions. In term of AVD the VNS algorithm is only better than the one of Taillard [1].

Each method is executed using different machine, Taillard [1] used Sun Sparc workstation, 50 MHz, Tarantilis *et al*. [3] used Pentium III, 550 MHz, 128MB of RAM, Tarantilis *et al.* [4] used Pentium II, 400 MHz, 128MB of RAM, Li *et al.* [5] used Athlon, 1 GHz, 256MB of RAM and Brandao [8] used Intel Pentium M at 1.4GHz, 512MB of RAM. Though it is difficult to compare the CPU time under different machines, from Table 3, it can be noted that the proposed VNS algorithm consumes small CPU time.

Table1 Vehicle specifications of Taillard's [1] data set

No	*N*	Q_A	a_A	*nA*	Q_B	a_B	*nB*	Q_C	a_C	*nC*	Q_D	a_D	*nD*	Q_E	a_E	*nE*	Q_F	a_F	*nF*
13	50	20	1.0	4	30	1.1	2	40	1.2	4	120	1.7	4	120	2.5	2	200	3.2	1
14	50	120	1.0	4	160	1.1	2	300	1.4	1									
15	50	50	1.0	4	100	1.6	3	160	2.0	2									
16	50	40	1.0	2	80	1.6	4	140	2.1	3									
17	75	50	1.0	4	120	1.2	4	200	1.5	2	350	1.8	1						
18	75	20	1.0	4	50	1.3	4	100	1.9	2	150	2.4	2	250	2.9	1	400	3.2	1
19	100	100	1.0	4	200	1.4	3	300	1.7	3									
20	100	60	1.0	6	140	1.7	4	200	2.0	3									

N: number of customers; Q_t: capacity of vehicle type *t* (*t* = A, B, C, D, E, F); a_t : variable cost per unit distance of vehicle type *t*; number of vehicle type*t*

Table 2 Comparison with other methods

No	Size	Best Sol.	Taillard [1]	Tarantilis *et al.* [3]	Tarantilis *et al.* [4]	Li et al. [5]	Brandao [8]	The Proposed VNS Algorithm
13	50	**1517.84**	1536.55	1519.96	**1517.84**	**1517.84**	**1517.84**	1551.98
14	50	**607.53**	623.05	612.51	611.39	**607.53**	**607.53**	615.64
15	50	**1015.29**	1022.05	1017.94	**1015.29**	**1015.29**	**1015.29**	**1015.29**
16	50	**1144.94**	1159.14	1148.19	1145.52	**1144.94**	**1144.94**	1146.38
17	75	**1061,96**	1095.01	1071.67	1071.01	**1061.96**	**1061.96**	1078.05
18	75	**1823.58**	1894.73	1852.13	1846.35	**1823.58**	**1823.58**	1846.75
19	100	**1117.51**	1156.93	1125.64	1123.83	1120.34	1120.33	1124.58
20	100	**1534.17**	1592.16	1558.56	1556.35	**1534.17**	**1534.17**	1563.46
AVD			2.50	0.79	0.57	0.03	0.03	0.87
NB			0	0	1	7	7	1

Table 3 CPU time (in second)

No	Size	Taillard [1]	Tarantilis *et al*. [3]	Tarantilis *et al*. [4]	Li *et al.* [5]	Brandao [8]	The Proposed VNS Algorithm
13	50	473	110	843	258	56	10
14	50	575	51	387	141	55	7
15	50	335	94	368	166	59	6
16	50	350	11	341	188	94	6
17	75	2245	221	363	216	206	13
18	75	2876	310	971	366	198	12
19	100	5833	309	428	404	243	20
20	100	3402	675	1156	447	302	19

7.0 CONCLUSION

In this study, we put forward the VNS-based algorithm to solve the HFFVRP. The initial solution is obtained in three steps; (a) construct a giant tour using the sweep algorithm of Gillett and Miller [19], (b) improve this tour using the 2-opt of Lin [20], and (c) construct the cost network, starting from the smallest vehicles. The algorithm is equipped with four neighbourhoods and six refinement procedures to make up our local search. A multi-level based approach acts as the local search engine. Encouraging results are discovered when the VNS algorithm is tested on the existing data set from the literature. In term of CPU time, the VNS algorithm consumes less CPU time than other methods, though it is difficult to compare the CPU time as other results were obtained by using different machines.

For the future research, in order to improve the current solutions, applying different neighborhood, local search and introducing a diversification procedure into the VNS-based algorithm can be considered. The proposed algorithm can be developed for solving the multi-depot fixed fleet vehicle routing problem, a variant of that is harder to solve than the HFFVRP and can be found in daily practice in logistics field.

References

[1] Taillard, E.D. 1999. A Heuristic Column Generation Method for the Heterogeneous Fleet VRP. *Recherche Operationnelle*. 33: 1-14.

[2] Salhi, S., Sari, M., Sadi, D., and Touati, N. 1992. Adaptation of Some Vehicle Fleet Mix Heuristic. *Omega*. 20: 653-660.

[3] Tarantilis, C. D., Kiranoudis, C. T., and Vassiliadis, V. S. 2003. A List Based Threshold Accepting Metaheuristic for the Heterogeneous Fixed Fleet Vehicle Routing Problem. *Journal of the Operational Research Society*. 54: 65-71.

[4] Tarantilis, C. D., Kiranoudis, C. T., and Vassiliadis, V. S. 2004. A Threshold Accepting Metaheuristic for the Heterogeneous Fixed Fleet Vehicle Routing Problem. *European Journal of Operational Research*. 152: 148-158.

[5] Li, F.Y., Golden, B., and Wasil, E. 2007. A Record-to-Record Travel Algorithm for Solving the Heterogeneous Fleet Vehicle Routing Problem. *Computers & Operations Research*. 34: 2734-2742.

[6] Dueck, G. 1993. New Optimization Heuristics: The Great Deluge Algorithm and Record to Record Travel. *Journal of Computation Physics*. 104: 86-92.

[7] Euchi, J. and Chabchoub, H. 2010. A Hybrid Tabu Search to Solve the Heterogeneous Fixed Fleet Vehicle Routing Problem. *Logistics Research*. 2: 3-11.

[8] Brandao, J. 2011. A Tabu Search Algorithm for the Heterogeneous Fixed Fleet Vehicle Routing Problem. *Computers & Operations Research*. 38: 140-151.

[9] Tarantilis, C. D. and Kiranoudis, C. T. 2002. Distribution of Fresh Meat. *Journal of Food Engineering* 51: 85-91.

[10] Tarantilis, C. D. and Kiranoudis, C. T. 2002. BoneRoute: An Adaptive Memory-Based Method for Effective Fleet Management. *Annals of Operations Research.* 115: 227-241.

[11] Levy, L. 2005. *Private Communication.* RouteSmart Technologies, Inc.

[12] Salhi, S. and Rand, G. K. 1993. Incorporating Vehicle Routing into the Vehicle Fleet Composition Problem. *European Journal of Operational Research.* 66: 313-360.

[13] Brimberg, J. and Mladenović, N. 1996. Variable Neighbourhood Algorithm for Solving the Continuous Location-Allocation Problem. *Studies in Locational Analysis*. 10: 1-12.

[14] Hansen P. and Mladenović, N. 1997. Variable Neighbourhood Search for the *p*-Median. *Location Science*. 5: 207-225.

[15] Mladenović, N., and Hansen, P. 1997. Variable Neighbourhood Search. *Computers & Operations Research*. 24: 1097-1100.

[16] Hansen, P. and Mladenović, N. 2001. Variable Neighborhood Search: Principles and Applications. *European Journal of Operational Research*. 130: 449-467.

[17] Hansen, P., Mladenović, N., and Moreno Perez, J. A. 2010. Variable Neighborhood Search: Methods and Applications. *Annals of OR*. 175: 367-407.

[18] Salhi, S., and Sari, M. 1997. A Multi-Level Composite Heuristic for the Multi-Depot Vehicle Fleet Mix Problem. *European Journal of Operational Research*. 103: 95-112.

[19] Gillett, B. E., and Miller, L. R. 1974. A Heuristic Algorithm for the Vehicle Dispatch Problem. *Operations Research*. 22: 340-344.

[20] Lin, S. 1965. Computers Solutions of the Traveling Salesman Problem. *Bell System Technical Journal*. 44: 2245-2269.

[21] Imran, A., Salhi, S., and Wassan, N. A. 2009. A Variable Neighborhood-Based Heuristic for the Heterogeneous Fleet Vehicle Routing Problem. *European Journal of Operational Research*. 197: 509-518.

[22] Salhi, S., and Rand, G.K. 1987. Improvements to Vehicle Routing Heuristics. *Journal of the Operational Research Society*. 38: 293-295.

[23] Golden, B., Assad, A., Levy, L., and Gheysens, F. G. 1984. The Fleet Size and Mix Vehicle Routing Problem. *Computers and Operations Research*. 11: 49-66.