http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (23): 7137-7142, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.7137.7142

Study on Multi Depot Heterogeneous Vehicle Routing Problem with an Improved Variable Neighborhood Search Algorithm

[1]Yuexiang Yang, [1,2]Yingcheng Xu and [3]Li Wang
[1]Quality Management Branch, China National Institute of Standardization, Beijing, 100191, China
[2]Department of Industrial Engineering, Tsinghua University, Beijing 100084, China
[3]School of Economics and Management, Beihang University, Beijing 100191, China

Abstract: This study deals with the Multi Depot Heterogeneous Vehicle Routing Problem with Time Windows (MDHVRPTW) which is complex and still not resolved well. The objective is to determine the best fleet composition as well as the set of routes that minimize the total cost with known demands. To solve the problem, mathematical model of the MDHVRPTW is constructed and an improved variable neighborhood search algorithm is proposed. In the algorithm, the hybrid operators of insert and exchange are used to achieve the shaking process and the later optimization process is presented to improve the solution space, the best-improvement strategy is adopted which make the algorithm can achieve a better balance in the solution quality and running time. The idea of simulated annealing is introduced to take control of the acceptance of new solutions. The developed algorithm was tested in benchmark instances. The results obtained are quite competitive with those found in the literature and new improved solutions are reported. And finally the proposed model and algorithm is applied to the large water project in China to solve the allocation of vehicles and routes, it demonstrates that the systematic method is effective and feasible.

Key words: Multi depot, heterogeneous, vehicle routing problem, variable neighborhood search

INTRODUCTION

The multi depot heterogeneous vehicle routing problem with time windows (MDHVRPTW) is a variant of the Vehicle Routing Problem (VRP) where the vehicles do not necessary have the same capacity and they belong to different the distribution centers or depots. Therefore, the MDHVRPTW involves designing a set of vehicle routes, each starting and ending at the depot, for a heterogeneous fleet of vehicles which services a set of customers with known demands. Each customer is visited exactly once and the total demand of a route does not exceed the capacity of the vehicle type assigned to it. The routing cost of a vehicle is the sum of its fixed cost and a variable cost incurred proportionately to the travel distance. The objective is to minimize the total of such routing costs. The number of available vehicles of each type is assumed to be unlimited. Variable Neighborhood Search (VNS) was initially proposed by Hansen *et al.* (2001) for solving combinatorial and global optimization problems. The main reasoning of this metaheuristic is based on the idea of a systematic change of neighborhoods within a local search method. In the past few decades, many scholars used VNS to solve the different VRP, such as DVRP (Gendreau *et al.*, 2006), HFVRP (Imran *et al.*, 2009) and made some achievements. At present, VNS has become a hot topic and it is also applied to solve other problems, these can be found in the literatures (Hansen *et al.*, 2007; Adibi *et al.*, 2010). Samanta and Jha (2011) analyzed the underlying complexities of MDPVRPTW and presented a heuristic approach to solve the problem, in this algorithm, two modification operators namely, crossover and mutation are designed specially to solve the MDPVRPTW. Subramanian *et al.* (2012) proposed hybrid algorithm which was composed by an Iterated Local Search (ILS) based heuristic and a Set Partitioning (SP) formulation to solve the Heterogeneous Fleet Vehicle Routing Problem. Salhi *et al.* (2013) dealt with the fleet size and mix vehicle routing problem with backhauls (FSMVRPB) based on the ILP formulation.

Current VRP optimization models are useful for a variety of practical applications. However, due to the complexity of the problem, the current solving quality and efficiency for the large-scale problem of MDHVRPTW are far from the practical requirements. This study presents an improved variable neighborhood search algorithm to solve MDHVRPTW; it integrates local search operator,

Corresponding Author: Yingcheng Xu, Quality Management Branch, China National Institute of Standardization, Beijing, 100191, China

optimization process and the simulated annealing algorithm into the VNS algorithm framework. Through the comparison with other algorithms, it shows the proposed algorithm gets the better solution.

PROBLEM DESCRIPTIONS

Problem definition: The number of customers is denoted by n and the number of depots is denoted by m. Thus, the problem is defined on a complete graph G = (V, E), where $V = \{v_1,\dots v_n, v_{n+1},\dots,v_{m+1}\}$ is the vertex set and $E = \{(v_i, v_j: v_i, v_j \in V, i \neq j)\}$ is the arc set. The customer set $C = \{v_1,\dots,v_n\}$ represents n customers while vertices set $D = \{v_{n+1},\dots,v_{n+m}\}$ corresponds to m depots. Each vertex $v_i \in V$ has several non-negative weights associated with it, namely, a demand q_i, a service time s_i, as well as an earliest e_i and latest l_i possible start time for the service which define a time window $[e_i, l_i]$. For the depots these time windows correspond to the opening hours. Furthermore, the depot vertices v_{n+1} to v_{n+m} feature no demands and service times, i.e., $q_i = s_i = 0; \forall i \in \{1,\dots, m\}$. Associated to C^l_{ijk} is the transportation cost from customer ito customer j for k vehicles of l type. The delivery vehicles set $T = \{T_{11},\dots,T_{lk},\dots,T_{LK}\}$ corresponds to the set of k vehicles for l type. Each l type vehicle k has associated a non-negative capacity D^l_k and non-negative maximum route duration D^l_k. w^l_k is the capacity of k vehicles for l type. d_{ij} is the linear distance from customer i to customer j. R_m is the capacity of m depots. ρ_i is penalty cost for unit-time violations of the specified time window for customer node i, ρ_k is penalty cost for unit-time violations of the maximum working time for vehicle k, Δa_i represents ith-time window violation due to early service, Δb_i ith-time window violation due to late service, ΔB_i corresponds to working time violation for vehicle k. A_i is vehicle arrival time at node i.

A feasible solution to the MDHVRPTW problem must satisfy the following constraints:

- The distribution of vehicles over the depots is fixed a priori and given as the input data
- Each vehicle starts and ends at its home depot
- Each customer is served by one and only one vehicle
- The total load and duration of vehicle k does not exceed D_k and T_k respectively
- The service at each customer i begins within the associated time window $[e_i,l_i]$ and each vehicle route starts and ends within the time window of its depot
- The goal is to minimize the total transportation cost of by all vehicles

Problem mathematical formulation

Objective function: The problem objective 1 aims to minimize the overall service expenses, including traveling distance and time costs, waiting and service time costs and penalty costs:

$$\min \sum_{i\in D\cup C}\sum_{j\in D\cup C}\sum_{k\in T} C^l_{ijk}X^l_{ijk}d_{ij} + \rho_k \Delta B_k + \sum_{i\in C}\rho_i(\Delta a_i + \Delta b_i) \quad (1)$$

Problem constraints:

- **Assignment of nodes to vehicles:** Equation 2 states that every customer node must be serviced by a single vehicle:

$$\sum_{l,k\in T}\sum_{i\in D\cup C} X^l_{ijk} = 1, \quad j\in C \quad (2)$$

- **Capacity constraints:** Constraint 3 states that the overall load to deliver to customer sites serviced by a used vehicle v should never exceed its cargo-capacity w^l_k. The distribution center (depot) has also the capacity constraint shown Eq. 4:

$$\sum_{i\in C}\sum_{j\in D\cup C} q_i X^l_{ijk} \le w^l_k, \quad l,k\in T \quad (3)$$

$$\sum_{i\in C} q_i Z_{ij} - \sum_{m\in D} R_m \le 0 \quad (4)$$

- **Assignment of vehicles to nodes:** Constraint 5 ensures that the vehicle can only reach a customer node for one time:

$$\sum_{i\in D\cup C} X^l_{ijk} = Y^l_{ik}, \quad j\in D\cup C, \quad l,k\in T \quad (5)$$

Constraints 6 states that certain vehicles can only depart from a customer node for one time:

$$\sum_{j\in D\cup C} X^l_{ijk} = Y^l_{jk}, \quad i\in D\cup C, \quad l,k\in T \quad (6)$$

- **Assignment of vehicles to depots:** Equation 7 ensures that each vehicle only belongs to a distribution center (depot):

$$\sum_{m\in D}\sum_{i\in C} X^l_{mik} \le 1, \quad l,k\in T \quad (7)$$

- **Relationship between the routes and depots:** Constraint 8 states any routes contain only a distribution center (depot):

$$S_{ik} - S_{jk} + nX^l_{ijk} \le n-1, i, j\in C, l, k\in T \quad (8)$$

- **Overall traveling time for vehicle k:** Constraint 9 states each route does not exceed the maximum mileage of the vehicle:

$$\sum_{l,k\in T}\sum_{i\in D\bigcup C} X^l_{ijk} d_{ij} \le D^l_k, \quad j\in C \tag{9}$$

Time constraint violations due to early/late services at customer sites.

$$\Delta a_i \ge e_i - A_i, \quad \forall i\in C \tag{10}$$

$$\Delta b_i \ge A_i - l_i, \quad \forall i\in C \tag{11}$$

- **Other constraints:**

$$X^l_{ijk} = 0,1 \quad i,j\in D\bigcup C, \quad l,k\in T \tag{12}$$

$$Y^l_{ik} = 0,1, \quad i\in C, \quad l,k\in T \tag{13}$$

$$Z_{ij} = 0,1, \quad i\in C, \quad j\in D \tag{14}$$

AN IMPROVED VARIABLE NEIGHBORHOOD SEARCH ALGORITH

Initial solution: Using variable neighborhood search algorithm, it first needs to build one or more initial feasible solution, an initial feasible solution mainly completes two tasks: Customer allocation and path planning.

To assume h_{ij} represents the distance of from distribution center i to the customer j, the distance set is $H_{ij} = \{h_{ij} | i = 1, 2,... m ; j = 1, 2,..., n\}$. The minimum value is min H_{ij} in the distance set, the second minimum is second min H_{ij}. Sum_k means the total number of customers that vehicles k delivers the goods, the set $U_k = \{y_{ik} | i = 0, 1,..., Sum_k\}$ corresponds to the vehicle k to serve the customers, y_{ik} presents the vehicle k as a transportation for customer i, y_{0k} means the distribution center (depot) is the initial point of the vehicle k.

The initial solution obtained by the above method can basically meet the needs of the follow-up work and built the foundation to get an optimal feasible solution in following algorithm.

Shaking: Shaking is a key process in the variable neighborhood search algorithm design. The main purpose of the shaking process is to extend the current solution search space, to reduce the possibility the algorithm falls into the local optimal solution in the follow-solving process and to get the better solution. The set of neighborhood structures used for shaking is the core of the VNS. The primary difficulty is to find a balance between effectiveness and the chance to get out of local optimal. In the shaking execution, it first selects a neighborhood structure N_k from the set of neighborhood structures of current solution x, then according to the definition of N_k, x corresponds to change and generate a new solution x*.

There are two neighborhood structures to achieve the shaking: Insert and exchange. Insert operator denotes a certain period of consecutive nodes move from the current path to another path; exchange operator refers to interchange the two-stage continuous nodes belonging to different paths. In each neighborhood the insert operator is applied with a probability p_{insert} to both routes to further increase the extent of the perturbation, then the probability of the exchange operator is 1-p_{insert}. IVNS selects randomly an exchange operator to change path for each shaking execution. The shaking process is somewhat similar to the crossover operation of the genetic algorithm. When the process is finished, the only two paths have the exchange of information; most of the features of the current solution will be preserved, to speed the convergence of the algorithm.

Local search: In a VNS algorithm, local search procedures will search the neighborhood of a new solution space obtained through shaking in order to achieve a locally optimal solution. Local search is the most time-consuming part in the entire VNS algorithm framework and decides the final solution quality so computational efficiency must be considered in the design process of local search algorithm. Two main aspects are considered in the design of local search algorithms: local search operator and the search strategy. Based on the previous studies, this study selects 2-opt and 3-opt as a local search operator in order to obtain the good quality local optimal solution in a short period, they are shown in Fig. 1. According to the probability, one of the two operators is selected in each local search process. The parameter p_2-opt represents the probability of selection for 2-opt, similarly, the probability of selection for 3-opt can be expressed as 1-$p_{2\text{-opt}}$. This mixed operator can develop optimization ability for 2-opt and 3-opt and expand the solution space of the algorithm.

There are mainly two search strategies: First-improvement and best-improvement in local search algorithm. The former refers to access the neighborhood solution of the current x solution successively in the solution process, if the current access neighborhood solution x_n is better than x, to make $x = x_n$ and update neighborhood solution. To repeat these steps until all the neighborhood solutions of x are accessed. Finally, x will

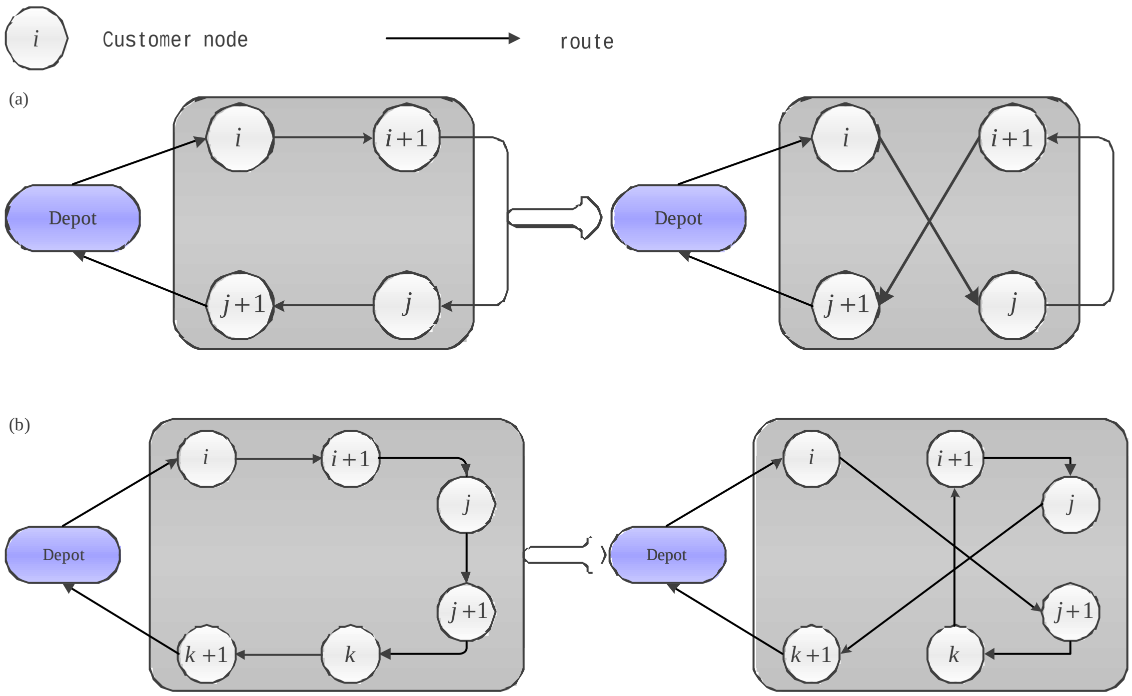

Fig. 1(a-b): 2-opt and 3-opt strategy, (a) 2-opt and (b) 3-opt

be obtained as a local optimal solution. The latter refers to traverse all of the neighborhood solution of current x solution in the solution process, to select the optimum neighborhood solution x_n as a local optimal solution. In this study, we adopt the best-improvement strategy, it enables the algorithm to achieve a better balance in the solution quality and run time.

Later optimization process: In order to accelerate the convergence speed and improve the solution quality, the later optimization process is proposed in the IVNS algorithm. After the local search is completed, if the local optimal solution xl is better than the global optimal solution xb, that is f (xl)<f (xb), the later optimization process will be continue to be implemented in order to seek a better global optimal solution. The algorithm of later optimization process which was proposed by Gendreau *et al.* (2006) is suitable for solving the traveling salesman problem and the vehicle routing problem with time windows. The algorithm processes can be simply described as follows:

Algorithm 1: Steps of Later optimization process:

Step 1: There are some assumptions. The path will be optimized is r, its length is n, its value of the evaluation function is c. The final optimized path is r* and the value of the evaluation function is c* and r* = r, c* = c and k = 1

Step 2: The Unstring and a String process (Gendreau *et al*., 2006) is respectively performed for the k-th customer in the r, the optimized path r' can be obtained and its value of the evaluation function value is c'

Step 3: If c'<c *, some processes are carried out, they are r* = r', r = r ', c* = c', c = c ' and k = 1, jump to Step 2; otherwise, k = k+1

Step 4: If k = n+1, the algorithm will be terminated; otherwise, jumps to Step 2

Acceptance decision: The last part of the heuristic concerns the acceptance criterion. Here we have to decide whether the solution produced by VNS will be accepted as a starting solution for the next iteration. To avoid that the VNS becomes too easily trapped in local optima, due to the cost function guiding towards feasible solutions and most likely complicating the escape of basins surrounded by infeasible solutions, we also allow to accept worse solutions under certain conditions. This is accomplished by utilizing a Metropolis criterion like in simulated annealing for inferior solutions x* and accepts them with a probability of Eq. 15, depending on the cost difference to the actual solution x of the VNS process and the temperature T. We update T every n_T iterations by an amount of $T_{n+1} = \delta T_n$, where q_0 a random number on the interval [0,1], Where δ is settable cooling coefficient and an initial temperature value is $T_0 = 10$:

Table 1: Comparison of solution quality from the different methods

			Gendreau *et al.* (2006)		Choi and Tcha (2007)		Lee *et al.* (2008)		Imran *et al.* (2009)		IVNS	
No	n	Best	Cost	CPU	Cost	CPU	Cost	CPU	Cost	CPU	Cost	CPU
G3	20	961.03	961.03	164.00	961.03	0.00	961.03	59.00	961.03	21.00	961.03	0.00
G4	20	6437.33	6445.10	253.00	6437.33	1.00	6437.33	79.00	6437.33	18.00	6437.33	1.00
G5	20	1007.05	1007.05	164.00	1007.05	1.00	1007.05	41.00	1007.05	13.00	1007.05	2.00
G6	20	6516.47	6516.47	309.00	6516.47	0.00	6516.47	89.00	6516.47	22.00	6516.47	2.00
G13	50	2406.36	2406.36	724.00	2406.36	10.00	2408.41	258.00	2406.36	252.00	2406.36	21.00
G14	50	9119.03	9125.65	1033.00	9119.03	51.00	9160.42	544.00	9119.03	274.00	9119.03	19.00
G15	50	2586.37	2606.72	901.00	2586.37	10.00	2586.37	908.00	2586.37	303.00	2586.72	23.00
G16	50	2720.43	2720.43	815.00	2728.14	11.00	2724.33	859.00	2741.50	253.00	2720.43	20.00
G17	75	1734.53	1734.53	1022.00	1734.53	207.00	1745.45	1488.00	1745.33	745.00	1743.76	52.00
G18	75	2369.65	2412.56	691.00	2369.65	70.00	2373.63	2058.00	2369.65	897.00	2369.65	50.00
G19	100	8659.74	8685.71	1687.00	8661.81	1179.00	8699.98	2503.00	8665.12	1613.00	8664.81	175.00
G20	100	4039.49	4166.73	1421.00	4042.59	264.00	4043.47	2261.00	4066.94	1595.00	4039.49	108.00
Average		4046.46	4065.70	765.33	4047.53	150.33	4055.33	928.92	4051.85	500.50	4047.68	39.42
BestNum*			6.00		9.00		5.00		8.00		9.00	
ARPD**			0.03		0.03		0.22		0.13		0.12	

*ARPD-average relative percentage deviation, **BestNum-the number of best solution

$$SA(x^*, x) = \begin{cases} x^* & \text{if} \quad q_0 > \exp(\frac{f(x^*) - f(x)}{T}) \\ x & \text{if} \quad q_0 \le \exp(\frac{f(x^*) - f(x)}{T}) \end{cases} \quad (15)$$

NUMERICAL EXPERIMENTS

Problem data and experimental setting: In order to assess the performance of the improved variable neighborhood search algorithm to solve MHFVRPTW, the experiments analyze and compare with other existing algorithms. IVNS algorithm is implemented by the C # language and the main configuration of the computer is Intel Core i3 1.8 GHz, 2 GB RAM and Window XP. In this experiment, the data sets from the literature (Choi and Tcha, 2007) are used. The benchmark problem is used to test the performance of the algorithm and it is 12 of 20 issues proposed by Golden *et al*. Here, the largest instance has 100 customers. For each instance, we define the Relative Percentage Deviation (RPD) and its computation equation is as follows:

$$RPD = \frac{cost_i - best_i}{best_i} \times 100 \quad (16)$$

where, cost t_i and $best_i$ denote, for the ith instance, the cost found by our heuristic and the best known solution respectively. The average deviation is then computed over all instances in the data set.

The initial values of the various parameters for IVNS algorithm are set as follows:

- The parameter settings for simulated annealing accepted criteria are initial temperature T = 10, every n_T = n/10 generation to update temperature $T_{n+1} = 0.9 \times T_n$, n_t = 1000 to end the algorithm
- The parameters value of the Shaking operation are as follows: $p_{insert} = 0.2$, $p_{cross} = 0.15$, $p_{icross} = 0.1$
- The $p_{2\text{-}opt}$ value is 0.5 in local search

Numerical results: The remaining 12 questions proposed by Golden are considered to be as the benchmark problem. The experiment results are shown in Table 1, where Best denotes best solution, row Average is the average solution of all the problems, row ARPD represents average RPD, the last row BestNum given algorithm to obtain the number of the best solution. The results in Table1 show that we proposed algorithms can get nine best solutions, ARPD is 0.12. The algorithm of Taillard and Lee obtains five best solutions, ARPD are 0.14 and 0.22, respectively and they are similar to our algorithm. Grendreau, Wassan and Osman with their algorithms have found six known best solution of the problem. The ARPD of Brandao, Wassan and Osman, respectively are 0.03 and 0.51. Choi and Tcha (2007), Brandao, Imran find the nine best solutions of the questions, ARPD are 0.05, 0.72 and 0.13 and one is slightly better than ours; two others are somewhat higher than our algorithm. However, on the issue of problem G20, the difference between the results of the three algorithms for Taillard, Imran and IVNS are small, ARPD 0.14, 0.13 and 0.12, respectively.

CONCLUSION

We presented a Variable Neighborhood Search (VNS) metaheuristic for The Multi Depot Heterogeneous Vehicle Routing Problem with Time Windows (MDHVRPTW) a generalized variant of the classical VRP. In the algorithm, a clustering algorithm is utilized to allocate customers in the initial solution construction phase, a hybrid operator of insert and exchange are used to achieve the shaking process, the Best-improvement strategy is adopted and it can make the algorithm to achieve a better balance in the solution quality and running time. The performance of our method is shown when comparing our best solution values to the benchmark instances proposed by Golden. The results indicate that the proposed algorithm is effective in solving the MDHVRPTW.

ACKNOWLEDGMENTS

Funding for this research is supported by the National Science Foundation of China (Grant No. 71301152, 71271013 and 71301011), National Social Science Foundation of China (Grant No. 11AZD096), the National Key Technology R and D Program of the Ministry of Science and Technology (Grant No. 2013BAK04B02), the Basic Scientific Research Business Projects No. 55126Z-2735. and China Postdoctoral Science Foundation (Grants No. 2013T60091, 2012M520008).

REFERENCES

Adibi, M.A., M. Zandieh and M. Amiri, 2010. Multiobjective scheduling of dynamic job shop using variable neighborhood search. Expert Syst. Appl., 37: 282-287.

Choi, E. and D.W. Tcha, 2007. A column generation approach to the heterogeneous fleet vehicle routing problem. Eur. J. Oper. Res., 34: 2080-2095.

Gendreau, M., F. Guertin, J.V. Potvin and R. Seguin, 2006. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Transport. Res. Part C, 14: 157-174.

Hansen, P., J. Brimberg, D. Urosevic and N. Mladenovic, 2007. Primal-dual variable neighborhood search for the simple plant-location problem. Informs J. Comput., 19: 552-564.

Hansen, P., N. Mladenovic and D. Perez-Britos, 2001. Variable neighborhood decomposition search. J. Heuristics, 7: 335-350.

Imran, A., S. Salhi and N.A. Wassan, 2009. Variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. Eur. J. Oper. Res., 197: 509-518.

Lee, Y.H., J.I. Kim, K.H. Kang and K.H. Kim, 2008. A heuristic for vehicle fleet mix problem using Tabu search and set partitioning. J. Oper. Res. Soc., 59: 833-841.

Salhi, S., N. Wassan and M. Hajarat, 2013. The fleet size and mix vehicle routing problem with backhauls: Formulation and set partitioning-based heuristics. Transport. Res. Part E: Logistics Transport. 56: 22-35.

Samanta, S. and M.K. Jha, 2011. Multi depot probabilistic vehicle routing problems with a time window: Theory, solution and application. Int. J. Oper. Res. Inform. Syst., 2: 17-41.

Subramanian, A., P.H. Vaz Penna, E. Uchoa and L.S. Ochi, 2012. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. Eur. J. Oper. Res., 221: 285-295.