SCATTER SEARCH ALGORITHM FOR HETEROGENEOUS FLEET VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND LOADING COST

ศิริชัย ยศวังใจ และขวัญนิธิ คำเมือง*

Sirichai Yodwangjai and Kwanniti Khammuang*

Abstract

The Vehicle Routing Problem (VRP) is one of the main problems in the fields of logistics operation management. Most VRP variants aim to minimize the total cost which includes fixed and variable costs. In some practical situations, the weight of goods is one of the constraints that impact the overall cost of transportation. This issue implies another cost relating to the weight of goods referred to as the loading cost. This paper studies a heterogeneous VRP with time window constraints and loading cost consideration. A family of scatter search (SS) algorithms with the 3-Opt/Opt* local search improvement method is proposed. Besides the well-known local search, Sequencing by Loading Cost (SLC) and a new heuristic, Insertion by Loading Cost (ILC) methods are also augmented with 3-Opt/Opt* and investigated. Computational experiments of the algorithms with benchmark problems are conducted. The results show that enhancing 3-Opt/Opt* with SLC and ILS benefits the quality of the solutions.

Keywords: Vehicle routing problem, Heterogeneous fleet, Time windows, Loading cost, Scatter search

---

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อำเภอเมือง จังหวัดพิษณุโลก 65000

Faculty of Engineering, Naresuan University, Muang District, Phitsanulok Province 65000

*corresponding author e-mail: kwannitik@nu.ac.th

Received: 17 July 2020; Revised: 24 August 2020; Accepted: 29 August 2020

Introduction

Vehicle Routing Problem (VRP), which was introduced (Dantzig & Ramser, 1959) in their study on the truck dispatching problem, is an important problem in the fields of transportation, distribution and logistics. VRP is similar to the well-known Travailing Salesman Problem (TSP), for that TSP is a special case of VRP where there is only one vehicle with no constraint on its capacity. VRP can be classified into various problem types based on the characteristics of practical situations considered.

One of the standard forms of VRP is the Capacitated Vehicle Routing Problem (CVRP), where each vehicle has its capacity limitation. CVRP attempts to find the routes for vehicles to minimize the total transportation cost while satisfying the customers' demands. Other real-life scenarios also establish some variations of the problem, for example, VRP with Time Windows (VRPTW) which the vehicles must arrive at the customers' locations within the corresponding time windows, and if the vehicle delivers to the customer later than the window, some penalty cost is usually incurred (Bräysy & Gendreau, 2002). Another significant variant of VRP considers the cases where there are multiple vehicle types called Heterogeneous VRP (HVRP) instead of homogeneous VRP where there is only one type of vehicle. In HVRP, fleet vehicles of different types have various capacities, associated costs, and the number of vehicles for each type can differ Jiang et al. (2014). In practice, the problem with the limited available number of vehicles is called Heterogeneous Fleet Vehicle Routing Problem (HFVRP). Thus, HFVRP that requires delivery within customer time windows is noted as HFVRP with Time Windows (HFVRPTW) problem.

In general, transportation cost in VRP usually consists of two parts: fixed and variable costs. Fixed cost depends on the number of vehicles used. Variable cost is normally a function of the distance traveled by the vehicles. However, only a few studies consider loading cost (LC) which is the cost associated with the weight of goods the vehicles needed to transport. Most of the aforementioned researches considered LC as constant or as a part of variable cost and therefore neglected to include LC in their objective functions. Thus, solving VRP without considering the loading cost could, in some cases, only provide an approximate and impractical solution. Most researches on VRP tend to consider only partial cost of transportation, namely fixed and variable costs. In some practical scenarios, the weight of goods is also one of the important factors that can affect the cost of delivery. As goods are being delivered to customers, the weight of goods on the vehicle is decreased. Thus, to correctly calculate the total transportation cost, loading cost which depends on the weight of goods transported needs to be considered. Based on the above practicality consideration, we studied the HFVRPTW with loading cost (HFVRPTWLC).

VRP with loading cost consideration has just been considered recently. Kuo (2010) presented a VRP intending to minimize fuel consumption which related to vehicle speed and loading weight. Tang et al. (2010) proposed a scatter search method for VRP with loading cost. Apart from investigating only single depot scenarios, multiple depot VRP with loading cost was studied by Zhang et al. (2011) applying a SS algorithm and by Kuo and Wang (2012) with variable neighborhood search and simulated annealing.

The remainder of this paper is organized as follows. A mathematical programming model of the problem is presented next. The SS algorithm is then explained. In the next two sections, two loading cost-based heuristics, SLC and ILC are proposed. Experimental results are then presented. Conclusion and future works are provided in the last section.

Materials and Methods

In this section, we present a mathematical programming model for the HVRPLC. Note that the HVRPLC is a special case of the general VRP for which a model is given in Tang et al. (2010). We assume that vehicles are heterogeneous. Each customer has a time window for receiving service. Let us define a Graph, $G = (V, A)$ as a transportation network where $V = \{0, 1, 2, ..., N\}$ is a vertex set (vertex 0 corresponds to the depot), and the set of vertexes $i = \{1, 2, ..., N\}$ corresponds to the customers. In this graph, a distance d_{ij} and travel time t_{ij} are associated with every arc $(i, j) \in A$, where A is the arcs set. Each customer i has a demand q_i, a service time s_i and a time window $[e_i, l_i]$, where e_i and l_i are the earliest time and the latest time to start the service of customer i. In the case where the vehicle to serve customer i arrives prior to e_i, it must wait at customer i location until time e_i. Note that d_0 = 0; e_0 = 0 and l_0 = T, where T is the latest time in which vehicles can return to the depot.

Sets and indices

i, j index for customers.

m index for vehicle types.

k index for vehicles.

N set of customers.

N_m set of vehicle of type m.

V_T set of vehicle types.

Parameters

t_i the service time of customer i.

t_{ij} the travel time from customer i to j.

e_i the earliest arrival time for customer i.

l_i the latest arrive time for customer i.

q_i the demand of customer i.

d_{ij} the distance from customer i to j.

p_i the penalty cost of unabling to deliver product to customer i on time per unit time later than e_i.

Cd_m the cost of traveling per unit distance of vehicle of type m.

Cl_m the cost of delivering product per unit-weight per unit distance.

Cf_m the fixed cost of assigning vehicle of type m to deliver products.

Q_m the capacity of vehicle type m.

Decision variable

$$x_{ijkm} = \begin{cases} 1, & \text{if the vehicle } k \text{ of type } m \text{ travels from customer } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$$

y_{ijkm} the load of the vehicle k of type m travels from customer i to j

LT_i the amount of time that vehicle arrives at customer i later than l_i

$$\text{MinZ} = \sum_{i \in N} \sum_{j \in N} d_{ij} \sum_{k \in N_m} \sum_{m \in V_T} x_{ijkm} (Cd_m + Cl_m y_{ijkm}) + \sum_{m \in V_T} Cf_m \sum_{j \in N} \sum_{k \in V_m} x_{0jkm} + \sum_{i \in N} p_i LT_i \tag{1}$$

Subject to

$$\sum_{j \in N, j \neq i} \sum_{k \in N_m} \sum_{m \in V_T} x_{ijkm} = 1 \qquad ; \forall i \in N, i \neq 0 \tag{2}$$

$$\sum_{i \in N, i \neq j} \sum_{k \in N_m} \sum_{m \in V_T} x_{ijkm} = 1 \qquad ; \forall j \in N, j \neq 0 \tag{3}$$

$$\sum_{i \in N} q_i \sum_{j \in N} x_{ijkm} \leq Q_m \qquad ; \forall k \in N_m, m \in V_T \tag{4}$$

$$\sum_{j \in N, j \neq 0} x_{0jkm} \leq 1 \qquad ;\forall k \in N_m, m \in V_T \tag{5}$$

$$\sum_{j \in N} x_{0jkm} - \sum_{i \in N} x_{i0km} = 0 \qquad ;\forall k \in N_m, m \in V_T \tag{6}$$

$$\sum_{j \in N} \sum_{k \in N_m} \sum_{m \in V_T} (x_{jikm} y_{jikm} - x_{ijkm} y_{ijkm}) = q_i \qquad ; \forall i \in N \tag{7}$$

$$0 \leq y_{ijkm} \leq Q_m x_{0jkm} \qquad ;\forall i, j \in N, k \in N_m, m \in V_T \tag{8}$$

$$s_i + t_i + t_{ij} - M(1 - x_{ijkm}) \leq s_j \qquad ;\forall i, j \in N \ , k \in N_m, m \in V_T \tag{9}$$

$$\sum_{j \in N} \sum_{k \in N_m} x_{ojkm} \leq |N_m| \qquad ;m \in V_T \tag{10}$$

$$s_i \geq e_i \qquad ;\forall i \in N \tag{11}$$

$$s_i - l_i \leq LT_i \qquad ;\forall i \in N \tag{12}$$

$$s_i \geq 0 \qquad ;\forall i \in N \tag{13}$$

$$LT_i \geq 0 \qquad ;\forall i \in N \tag{14}$$

$$x_{ijkm} = \{0,1\} \qquad ;\forall i, j \in N, k \in N_m \ , m \in V_T \tag{15}$$

$$y_{ijkm} \geq 0 \qquad ;\forall i, j \in N, k \in N_m \ , m \in V_T \tag{16}$$

The objective function (1) minimizes the total cost which includes variable and fixed costs, loading cost and penalty cost for late delivery. Constraint (2) and (3) guarantee that each customer is served by exactly one vehicle. Constraint (4) is the capacity constraint, which states that for each route, the accumulated demand from the customers who are served by a vehicle must not exceed the vehicle capacity. Constraint (5) and (6) guarantee that vehicles start from and return to the depot. Constraint (7) and (8) show the logical relationship between the demand of customer i and the vehicle loads on the two arc linking customer i . Constraint (9) specifies the time relationship between two customers who are served by the same vehicle. Constraint (10) limits the number of vehicles available for each vehicle type. Constraints (11)-(14) are related to time windows and defines the amount of late time. Sign restrictions on decision variables are specified in Constraints (15)-(16).

Scatter search algorithm (SS) is a population-based evolutionary algorithm which was proposed and developed as an algorithmic framework by Glover (1998). The SS framework has been widely studied because its flexibility and it can be easily applied to many diverse optimization problems. The SS framework has been used to solve many complex combinatorial optimization problems, and it has been shown to be more effective than other metaheuristics in some circumstances (Ranjbar et al., 2009; Guo & Tang, 2015)

The general framework of SS consists of five steps namely, Diversification Generation, Improvement, Reference Set Update, Subset Generation and Solution Combination. In the first phase, Diversification Generation method generates a set of diverse initial trial solutions. In step 2, Improvement method attempts to refine the trial solution to better objective values. Reference Set Updated method in the next step, generates and maintains a reference set which comprises $RefSet1$ and $RefSet2$ which are sets of high quality and high diversity solutions, respectively. In step 4, subset of $RefSet$ are generated to provide a pool of solutions for step 5 solution combination where solutions in the pool are combined to create new trial solutions from the subset.

Diversification generation method, an easy, straightforward and efficient approach to create diverse initial trial solutions is applying random insertion method. Usually the method gives reasonably good feasible solutions. To increase the diversity in the solution pool, $PopSol$, we used random insertion method to generate initial solutions which include both feasible and infeasible solutions. In case of the latter where there exists some vehicle which its load exceeding the capacity, a repair method is utilized to transform to a feasible solution. The notion behind the repair method is to remove the customers in the route that causes exceeded capacity from each route and attempt to insert them in another vehicle with sufficient capacity.

Solution Improvement solution, for each solution in the initial trial solution population and each new solution generated after the solution combination method, the selected improvement method is applied to enhance the quality of the solutions. In this research a well-known local search (LS) method for VRP, namely 3-Opt/Opt*, is used. In our study both intra-route and inter-route improvements of 3-Opt/Opt* are possible.

Reference set, reference set is a subset of trial solutions in $PopSol$. The main notion of creating a reference set is to provide both high quality and high diversity solutions. In this study, the reference set consists of two subsets, a subset of the best solutions, $RefSet1$ and a subset of diverse solutions, $RefSet2$. The initial reference set consists of b_1 best solutions, selected from $PopSol$ and b_2 diverse solutions from $PopSol \backslash RefSet1$.

In order to obtain $RefSet2$, we need to define the diversity between solutions. We applied the method of Zhang et al. (2011) to calculate the diversity of two solutions x_1 and x_2 as $d(x_1,x_2)=1-\frac{2\times e_c}{e_1+e_2}$ where e_1 and e_2 are the number of arcs in x_1 and x_2 respectively and e_c represents the number of common arcs between both solutions. Note that the diversity is in the range [0,1]. If there is no common arc then the largest diversity is obtained, that is $d(x_1,x_2)=1$. On the other hand, if all arcs are exactly the same for both solutions or there is no diversity at all, then $d(x_1,x_2)=0$, the smallest possible diversity.

After calculating diversity values for each possible pair of solutions in $RefSet1$ and $PopSol \backslash RefSet1$, the solution, x' is chosen to be a member of $RefSet2$ where $x' = (i \mid Max(\underset{j}{Min}(d(i,j))))$ such that $i \in PopSol \setminus RefSet1$ and $j \in RefSet1$, and removed from $PopSol$. The procedure is repeated until the number of members in $RefSet2$, b_2, has been reached.

Reference set update method, in order to improve the quality of the solutions, during each iteration $RefSet$ is updated in the following manner. The newly obtained solutions from the solution combination step, Sol_{new} are compared to the solutions in $RefSet$. First Sol_{new} and $RefSet$ are ascendingly and descendingly ordered, respectively denoted by Sol_{new}^{asd} and $RefSet^{desd}$. Let $ReplSize$ be the order such that the $ReplSize^{th}$ solution in the ordered Sol_{new}^{asd} is smaller than the $ReplSize^{th}$ solution from the ordered $RefSet^{desd}$ and the $(ReplSize+1)^{th}$ solution in the ordered Sol_{new}^{asd} is worse (larger) then the $ReplSize^{th}$ solution in $RefSet^{desd}$. Replacing the first $ReplSize$ solutions of $RefSet^{desd}$ in $RefSet$ with the first $ReplSize$ solutions in Sol_{new}^{asd} results in a new and updated $RefSet_{new}$. Then $RefSet^{new}$ and $RefSet$ are compared if there is a difference between the two sets, $RefSet$ is updated by removing all solutions in $RefSet2$ and create new b_2 $RefSet2$ solutions by the same method utilized when building the initial $RefSet2$.

Subset generation method, one of the important concepts in SS is developing new solutions from $RefSet$ which can be achieved by generating subsets of solutions from $RefSet$ and combining those solution to create new solutions. Subset generation method is a procedure employed to create seed subsets for constructing new trail solutions. In this study, we adopt the method of Glover et al. (2003) which three types of subset generation are applied, 2-element subsets, 3-element subsets and 4-element subsets.

Solution combination method is used to generate a new solution from the combination of two or more solutions in the subsets that were created by the subset generation method. This study applies the method of Tang et al. (2010).

The notion behind the method is to maintain arcs with higher number of traversed frequency in the solutions from the subset in the new solution. For each solution, k, in the subset, all routes were transformed into a matrix A_k, if there exists a path from i to j in the solution k then $a_k(i, j) = 1$, otherwise $a_k(i, j) = 0$. All matrices of the solutions to be combined are then composed into one matrix A_{sum}. Let n_{sol} be the number of solutions to be combined, then $a_{sum}(i, j) = \sum_{k=1}^{n_{sol}} a_k(i, j)$. For each customer row, i in A_{sum}, find the column, j with maximum entry then set $a_{sum}(i, j) = 1$ and other entries in the row j be 0. If there exist more than one column with maximum entries, then select j to be the column with highest value of q_j / d_{ij}. For each customer column, j, if there is no location set in the column, select the row i with the maximum entry and set $a_{sum}(i, j) = 1$ and other entries be 0. Tie breaks in the similar previous manner.

Sequencing by loading cost (SLC), since loading cost is one of the components in the total cost to be minimized, we propose another improvement method adapted from Tang et al. (2010) called Sequencing by Loading Cost (SLC). SLC is performed on solutions after one of the aforementioned improvements has been applied. The notion of SLC is to sequence the customers within each route based on their loading cost/distance ratio. The ratio, (q_j / d_{ij}), provides each route with loading per distance sequence priority. The SLC procedure is as follows:

We consider each route in the solution representation. Let i be the current position considered and n_r be the number of customers in the route. Customer in i^{th} position is represented by C_i. Note that $C_0 = 0$ which represents the depot. Let A and U be the set of assigned and unassigned customers respectively. Initially $A = \varnothing$ and $U = \{C_1, C_2, ..., C_n\}$. Also let $i = 0$. For each route, we perform the following

Step 1: Calculate the loading cost/distance ratio, q_j / d_{ij}, for all $j \in U$.

Step 2: Find j with the largest ratio, $j_{\max}$, and remove it from U and then add it to A.

Step 3: Put $j_{\max}$ in the $(i+1)^{th}$ position of the route.

Step 4: If $U=\varnothing$ then the complete sequence of the route is obtained. Otherwise, let $i=i+1$ and go back to Step 1.

Insertion by Loading cost (ILC), we propose another improvement method, Insertion by Loading Cost (ILC), that is similar to SLC. Instead of deterministically sequencing the customers on the route according to the loading cost/distance ratio, ILC applies probabilistic reordering assigning the chance of being selected as the next customer by the ratio. Customer with the higher ratio has more chance of being assigned as the next destination in the route. Providing randomness into the search by such selection could lead to more exploration to avoid being trapped in local optimal. The probability of customer j being assigned after customer i, p_{ij} is calculated by Equation (18) Note the customer with the lowest ratio would have no chance of being selected.

$$DF_{ij}=\frac{q_j/d_{ij}-\min(q_j/d_{ij})}{\max(q_j/d_{ij})} \quad (17)$$

$$P_{ij}=\frac{DF_{ij}}{\sum_{j\in U}DF_{ij}} \quad (18)$$

After all the routes have been assigned, the first customers in a randomly selected pair of routes are swapped since the process might lead to lower overall loading cost.

Results

In this section, we describe the experiment setting for evaluating the quality of solutions for the proposed approaches. In this study we consider SS with 3-Opt/Opt* and SLC as the search methods in the improvement method phase. Then we propose new versions of SS by applying SLC and ILC after each new solution obtained from 3-Opt/Opt* referred to as 3-Opt/*+SLC and 3-Opt/*+ILS. The four methods are tested on a set of problem instances. The set consists of seven instances from Augerat et al. (1995), two instances from Christofides & Eilon (1969), one instance from Christofides et al. (1979) and six instances from Solomon (1987). The number of customers, which also represents problem size, in these instances, ranges from 50 to 100 customers. Since in their original form, there are no time windows and heterogeneous fleet information, the instances need to be augmented with these details. The time windows are uniformly distributed and, for heterogeneous fleet information, we followed that of Jiang et al. (2014). Relevant cost information which depends on vehicle type also needs to be provided.

The setting of parameters affects the quality of the solutions of met heuristics. Here the parameters are empirically set by a set of tuning instances. For each tuning instance, the termination criterion is specified at 10000 searched solutions and the experiment is repeatedly conducted 10 times with different random number seeds. For our SS algorithms, parameters that need to be specified include the number of trial solution ($PSize$), the number of solution in $RefSet1$ and $RefSet2$ (b_1 and b_2) and the number of improvement (nLS). Based on the tuning experiments, $PSize$, b_1, b_2 and nLS are set at 100, 10, 10 and 10 respectively.

Table 1 Computational results of local search

No.	SLC		3-Opt/*		3-Opt/*+SLC		3-Opt/*+ILC	
	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.
P1	125,933	131,343	113,026	115,383	**111,544**	**115,307**	115,065	125,115
P2	186,030	195,470	165,295	173,955	**162,709**	**168,495**	175,949	183,719
P3	84,641	87,590	78,862	84,678	78,196	79,739	**73,143**	**76,662**
P4	208,083	258,095	143,397	154,852	**139,576**	**146,254**	151,890	165,216
P5	158,184	165,459	129,8480	134,785	**129,347**	**132,765**	133,761	155,368
P6	202,438	260,294	152,181	**170,188**	**151,266**	182,245	190,453	195,407
P7	257,437	277,840	194,791	198,878	192,576	197,384	**191,840**	**196,168**
P8	77,229	94,447	68,690	80,297	65,471	68,401	**63,120**	**66,512**
P9	127,307	132,037	105,581	112,568	**101,832**	109,075	102,897	**108,865**
P10	221,321	254,509	153,866	171,881	151,192	186,880	**148,315**	**170,744**
P11	26,446	27,489	26,299	27,809	26,633	26,725	**26,259**	**26,575**
P12	67,341	74,908	**62,461**	**64,649**	64,148	67,100	64,889	69,276
P13	181,033	199,002	164,744	179,872	**159,817**	**171,972**	170,923	178,737
P14	29,126	29,978	28,186	29,024	28,113	28,305	**27,510**	**28,198**
P15	250,815	272,104	179,345	187,525	**179,173**	193,987	180,886	**185,767**
P16	138,781	150,079	**115,794**	**122,166**	134,485	137,857	122,429	132,533

The four versions of SS namely 3-Opt/*, SLC, 3-Opt/*+SLC and 3-Opt/*+SLC are tested by conducting experiments on 16 problem instances. The results are shown in Table 1 with both average and minimum total costs. The best results of the total cost are presented in bold text. The results show that, in terms of average total cost, 3-Opt/+ILC outperforms other methods. 3-Opt/*+ILC provides the best solutions in 8 out of 16 instances. However, 3-Opt/* is not far behind achieving the best solutions in 5 instances where as 3-Opt/* and SLC have 3 and 0 best solutions, respectively. It is better than the local search +SLC.

Discussions

From the aspect of minimum total cost, 3-Opt*/SLC and 3-Opt*/ILC also dominates providing best solutions in 9 and 6 instances, respectively. This indicates that applying loading cost-based heuristic, such as SLC, or general VRP based heuristics (3-Opt*/SLC) to HFVRPTWLC might not generate high quality solution. Combining both type of heuristic, as in 3-Opt/*+ILC and 3-Opt/*+SLC can increase the quality of the solution up to 20.72% (in instance P8, average total cost). However, the combined heuristics, 3-Opt/*+ILC and 3-Opt/*+SLC, do not indicate significant difference in their performance. However, in some large-sized instance, the solutions obtain from 3-Opt/*+SLC is significant better that 3-Opt/*+ILC as shown in instance P4, P5 and P6.

Conclusions

This paper proposes a new method for solving heterogeneous fleet vehicle routing problem with time windows and loading costs. The problem considers the weight of goods that a set of vehicles transport as a type of cost to be included in total cost with fixed, variable and penalty cost. We applied a scatter search algorithm for solving the problem. Since local search plays an important role in the effectiveness of both the performance of SS and in solving VRP from the literature, we utilize a well-known local search method, 3-Opt/* with SS. Another search method designed specifically to tackle the loading cost aspect of the problem, SLC, is also proposed and investigated. To improve the performance of the algorithms, two new search methods, 3-Opt/*+SLC and 3-Opt/*+SLC are introduced. The four versions of SS are empirically tested in terms of their solution quality. The parameters for the SS algorithms were set by nine tuning problems. The results from the experiments show that 3-Opt/*+SLC and 3-Opt/*+SLC improve the solutions from 3-Opt/* and SLC in most cases.

In future works, other algorithms, both exact and heuristics, could be applied to tackle VRPTWLC. In the current transportation service operation, customer satisfaction is of highest priority. Customers expect shorter transportation lead time and orders could be altered anytime. Therefore, it is also interesting to investigate the dynamic version of the problem.

Acknowledgments

The authors are very grateful to the Department of Industrial Engineering, Faculty of Engineering, Naresuan University for supporting this research. Finally, the authors would like to thank the reviewers for their valuable comments and recommendations which enabled the improvement of the quality of this paper.

Reference

Augerat PJ, Belenguer E, Benavent A. et al. *Computational Results with A Branch And Cut Code For The Capacitated Vehicle Routing Problem*. Rapport de recherche- IMAG, 1995.

Bräysy O, Gendreau M. Tabu Search heuristics for the Vehicle Routing Problem with Time Windows. *Top*. 2002; 10(2): 211-237.

Christofides N, Eilon S. An algorithm for the vehicle dispatching problem. *Operational Research Quarterly*. 1969; 20: 309-318.

Christofides N, Mingozzi A, Toth P. The vehicle routing problem. *Combinatorial Optimization, Chichester: Wiley*; 1979: 315-338.

Dantzig GB, Ramser JH. The Truck Dispatching Problem. *Management Science.* 1959; 6(1): 80-91.

Kuo Y. Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering.* 2010; 59: 157-165.

Kuo Y, Wang CC. A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. *Expert Systems with Applications*. 2012; 39: 6949-6954.

Glover F. A template for scatter search and path relinking. *Artificial Evolution: Third European Conference AE '97.* Heidelberg, Springer Berlin Heidelberg; 1998. 1-51.

Glover F, Laguna M, Marti R. Scatter Search. *Advances in Evolutionary Computing: Theory and Applications*. Berlin, Heidelberg, Springer Berlin Heidelberg; 2003.

Guo Q, Tang L. An improved scatter search algorithm for the single machine total weighted tardiness scheduling problem with sequence-dependent setup times. *Applied Soft Computing.* 2015; 29: 184-195.

Jiang J, Ng KM, Poh KL. et al. Vehicle routing problem with a heterogeneous fleet and time windows. *Expert Systems with Applications.* 2014; 41(8): 3748-3760.

Ranjbar M, De Reyck B, Kianfar F. A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling. *European Journal of Operational Research.* 2009; 193(1): 35-48.

Solomon M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research.* 1987; 35(2): 254-265.

Tang J, Zhang J, Pan Z. A scatter search algorithm for solving vehicle routing problem with loading cost. *Expert Systems with Applications*, 2010; 37(6): 4073-4083.

Zhang J, Tang J, Fung RYK. A scatter search for multi-depot vehicle routing problem with weight-related cost, *Asia-Pacific Journal of Operational Research*, 2011; 28(3): 323-348.