Algorithms for the multi-objective vehicle routing problem with time windows
Peer-reviewed author version

The 5th International Congress on Logistics and SCM Systems

ICLS2009
Seoul Korea

Overcoming Economic Crisis through
Collaborative Global Logistics and Supply Chains

June 2-5, 2009
COEX Conference Center

Hosted by : KSCM, KITA

Sponsors : MKE Ministry of Knowledge Economy, IITA Institute for Information Technology Advancement
GSL Inha University Graduate School of Logistics
e-LRC Northeast Asia e-Logistics Research Center of Incheon University
Institute of Supply Chain & Logistics Management, Hanyang University
Amway Amway Korea, pantos Pantos Logistics

[Vehicle Routing and Transportation 3]

D16(icls112)

Korea's Port Development Strategy to be a World Top-class Hub Port in Global SCM

Gim Jin-Goo(LSE INSTITUTE, Korea)

D17(icls039)

Algorithm for the Multi-Objective Vehicle Routing Problem with Time Windows

Tharinee Manisri(Sripatum University, Thailand),

Anan Mungwattana(Kasetsart University, Thailand), Gerrit K. Janssens(Hasselt University, Belgium)

D18(icls102)

Key Elements of Cross-border Transportation Infrastructure: A Comprehensive Approach of Logistics Infrastructure Development

Pichet Sooksaksun, Wutthipong Moungnoi, Santi Charoenpornpattana(King Mongkut's University of Technology Thonburi, Thailand)

D19(icls108)

Container Terminal Location Model Using Set-covering Method

Suk Tae Bae, Ki Wook Lee, Chang Sung Ha,Heung Suk Hwang(TongMyong Univ., Korea)

D20(icls055)

Innovative load carrier management solution for seaport terminals
-based on positioning, identification and communication technologies-

Bernd Scholz-Reiter, Marc-André Isenberg, Anne Virnich, Mehmet-Emin Özsahin(University of Bremen, Germany)

D21(icls113)

A Study of e-RTGC Introduction Effects in the Container Terminal: Based on D Container Terminal Case Study

Hyung Rim Choi, Yong Sung Park, Moo Hong Kang, Seung Hong Lee, Hee Yoon Kim,

Ki Nam Choi(Dong-A University, Korea)

Algorithm for the Multi-Objective Vehicle Routing Problem with Time Windows

Tharinee Manisri [*A], **Anan Mungwattana**[*B], **and Gerrit K. Janssens**[*C]

[*A] **Sripatum University, Thailand, e-mail:tharinee_i@hotmail.com**
[*B] **Kasetsart University, Thailand, e-mail:fenganm@ku.ac.th**
[*C] **Hasselt University, Belgium, e-mail: gerrit.janssens@uhasselt.be**

Abstract

This paper focuses on an algorithm for the vehicle routing problem with time windows (VRPTW). It involves servicing a set of customers, with earliest and latest time deadlines, a constant service time including when the vehicle arrives to the customers. The demands are served by capacitated vehicles with limited travel times to return to the depot. The purpose of this research is to develop a hybrid algorithm that includes a heuristic, a local search and a meta-heuristic algorithm to solve optimization problems with multiple objectives. The first priority aims to find the minimum number of vehicles required and the second priority aims to search for the solution that minimizes the total travel time. The algorithm performances are measured with two criteria: quality of solution and running time.

A set of well-known benchmark data and the genetic algorithm are used to compare the quality of solution and running time of the algorithm, respectively. The algorithm is applied to solve the Solomon's 56 VRPTW benchmarking problems which have 100-customer instances. The results show that 22 solutions are better than or competitive as compared to the best solutions of the Solomon benchmark problem instances. The running time results display that the hybrid algorithm has higher performance than the genetic algorithm when the number of customers less than 25 nodes.

Keywords: Vehicle routing problem with time windows, Heuristic, Local search, Meta-heuristic

1. Introduction

The vehicle routing problem (VRP) is an operational decision problem for the delivery of goods from a depot to customers by a fleet of vehicles. The vehicle routing problem with time windows (VRPTW) is an extension of the VRP with earliest, latest, service times for customers and travel times.

The objective is to minimize the number of vehicles and the total travel time to service the customers by using an evolutionary hybrid algorithm. This paper proposes a multi-objective algorithm that incorporates a heuristic, local search and a meta-heuristic for solving the multi-objective optimization in VRPTW. The algorithm is designed by the modified push-forward insertion heuristic (MPFIH), a λ-interchange local search descent method (λ-LSD) and tabu search (TS). The route is constructed based on the MPFIH as initial solution which is improved by using the λ-LSD and TS. The constraints of the problem are to service all the customers within the earliest and latest service time of the customer without exceeding the route time of the vehicle and overloading the vehicle. The route time of the vehicle is defined as the sum of the waiting times, the service times and the travel times. A vehicle that reaches a customer before the earliest time, after the latest time and after the maximum route time incurs waiting time, tardiness time and overtime, respectively. The total of the customer demands in each route can not exceed the total capacity of the vehicle.

The rest of this paper is organized as follows. Section 2 reviews relevant VRPTW and algorithms. Section 3 presents tools and the methods to solve this problem. Section 4 presents the results and discussion. Finally, conclusions and future work are formulated in section 5.

2. Literature Review

The VRPTW arises in retail distribution, school bus routing, mail and newspaper delivery, airline and railway fleet routing and scheduling. It is well-known and complex combinatorial problem with considerable economic significance [1]. Savelsbergh [2] has shown that finding a feasible solution to the traveling salesman problem with time windows (TSPTW) is a NP-complete problem. Therefore the VRPTW is more complex as it involves servicing customers with time windows using multiple vehicles that vary with respect to the problem. By this case, almost researchers tend to heuristic and meta-heuristic methods which often produce optimal or near optimal solutions in a reasonable amount of computer time. Thus, there is still a considerable interest in the design of new heuristics for solving large-sized practical VRPTW.

Evaluation of any heuristic and meta-heuristic method is subject to the comparison of a number of

criteria that relate to various aspects of algorithm performance [3]. Examples of such criteria are running time, quality of solution, ease of implementation, robustness and flexibility [4]. Almost all algorithms for the VRPTW use route construction, route improvement or methods that integrate both route construction and route improvement. Solomon [5] designed and analyzed a number of route construction heuristics, namely: the savings, time-oriented nearest neighbor insertion and a time oriented sweep heuristic for solving the VRPTW. In his study, the time-oriented nearest neighbor insertion heuristic has shown to be very successful. Berger and Barkaoui [1] proposed a parallel version of a new hybrid genetic algorithm for the VRPTW. This approach is based upon the simultaneous evolution of two populations of solutions focusing on separate objectives subject to temporal constraint relaxation. Bräysy and Gendreau [3] presented a survey of the research on the VRPTW. Both traditional heuristic route construction methods and recent local search algorithm are examined in Part I. Meta-heuristics are general solution procedures that explore the solution space to identify good solutions and often embed some of the standard route construction and improvement heuristics [6]. Recently, several researches involve algorithms to solve the multi-objective VRPTW. The primary objective is defined as the minimization of the number of routes or vehicles. Minimization of the total travel cost is the secondary objective. Qi and Sun [7] proposed an improved algorithm based on the ant colony system (ACS), which hybridized with randomized algorithm (RACS-VRPTW). For this multi-objective problem, Ombuki *et al.*[8] presented a genetic algorithm solution using the Pareto ranking technique. An advantage of this approach is that it is unnecessary to derive weights for a weighted sum scoring formula. An evolutionary algorithm for the VRPTW was developed by incorporating various heuristics for local exploitation in the evolutionary search and the concept of Pareto's optimality [9].

All approaches in the literature are quite effective, as they provide solutions competitive with the well-known benchmark data, thus the benchmark Solomon's 56 VRPTW instances with 100 customers [10].

3. Tools and Methods

3.1 Tools

The experiments for the research are run on personal computer, Pentium 4 3.40 GHz. and using MATLAB computing software.

3.2 Notation

K : total number of vehicles, $k = 1,2,...K$

K_{LB} : lower bound of the number of vehicles,

where $K_{LB} = \dfrac{\sum\limits_{i=2}^{N} d_i}{q_k}$

N : total number of customers, including the depot

C_i : customer i , where $i = 2,3...,N$

C_1 : depot

d_i : demand of customer i

D_k : total demand for the vehicle k

q_k : capacity of vehicle k

t_{ij} : travel time between customer i to customer j

where $i, j = 1,...,N$, $i \neq j$ and $i, j = 1$ is depot

e_i : earliest arrival time at customer i

l_i : latest arrival time at customer i

A_i : arrival time to customer i

b_i : service time at customer i

w_{ij} : waiting time between customer i and j

where $w_{ij} = \max[e_j - (A_i + t_{ij}),0]$, $i, j = 2,...,N$ and $i \neq j$

M_k : maximum route time, where $k = 1,2,...K$

R_k : vehicle route k , where $k = 1,2,...K$

W_k : total waiting time for vehicle k , where $k = 1,2,...K$

B_k : total service time for vehicle k ,where $k = 1,2,...K$

O_k : total overtime for vehicle k ,where $k = 1,2,...K$

L_k : total tardiness for vehicle k ,where $k = 1,2,...K$

T_k : total travel times for vehicle k ,where $k = 1,2,...K$

Tot_k : total travel time for vehicle k ,or $Tot_k = T_k + W_k + B_k$ where $k = 1,2,...K$

α : penalty weight factor for the waiting time

γ : penalty weight factor for the tardiness time

η : penalty weight factor for the overtime

We consider a set of vehicles, K and a set of customer nodes, C_i. We identify C_1 as the depot node and $C = C_i \cup C_1$ represent the set of all nodes. Let x be the set of the decision variables, they are evaluated using the function $F(x)$ as equation (1):

$$F(x) = T_k + (\alpha \times W_k) + (\gamma \times L_k) + (\eta \times O_k) \quad (1)$$

3.3 Methods

In this paper develops the hybrid algorithm. There are two phases of this algorithm. The first phase is route construction heuristic, namely, the modified push-forward insertion heuristic (MPFIH). The MPFIH is a heuristic method for inserting a customer into a route based on push-forward insertion method of Solomon [5] and Thangiah [11][15]. It is an efficient method for computing the insertion of a new customer into the route. Let us assume a route $R_k = \{C_{ik}, ..., C_{mk}\}$ where C_{ik} is the first set of customer and C_{mk} is the last set of customer in each route k. The earliest arrival and latest arrival time are defined as e_{ik}, l_{ik} and e_{mk}, l_{mk} respectively. The number of routes k in this method is defined as the minimum of number of vehicles that satisfies the total customer demand. The feasibility of inserting a set of customers into route R_k is checked by inserting the customer between all the edges in the current route and selecting the edge that satisfies the vehicle capacity. The MPFIH algorithm is shown below.

Step1: Sort the customer nodes which have e_i and l_i by ascending and descending method, respectively

Step2: Construct the initial matrix, R_k, where $k = K_{LB}$

Step3: Construct the set of C_{lk} and C_{mk} which the first k minimum, e_i and the first k maximum, l_i, respectively

Step4: Remove the customer nodes that have been selected to matrix, R_k

Step5: Select the set of C_{ik} which the next k minimum, e_i

Step6: Check the feasible route, each row of matrix, R_k that satisfy the constraints,

$$D_k = \sum_{i=l}^{m} d_i \leq q_k, \; Tot_k \leq M_k \text{ and } L_k = 0$$

If all rows satisfy the constraints go to step7, else go to step9

Step7: Insert the set of C_{ik} between set of C_{lk} and C_{mk} then repeat step4 to step6

Step8: If all of set C_{ik} has been inserted to routes or matrix, R_k then the algorithm terminate, else go to step5

Step9: Select the remainder, C_i which the next minimum, e_i

Step10: Check the feasible route, each the remainder row of matrix, R_k that satisfy the constraints,

$$D_k = \sum_{i=l}^{m} d_i \leq q_k, \; Tot_k \leq M_k \text{ and } L_k = 0$$

If the remainder rows satisfy the constraints go to step11, else go to step14

Step11: Insert C_i in the remainder routes or rows of matrix, R_k

Step12: Remove the customer nodes that have been selected and then repeat step9 to step12

Step13: If all of C_i has been inserted to routes or matrix, R_k then the algorithm terminates, else go to step14

Step14: Construct a new route or row of matrix, R_{k+i}, where $i = 1, 2, ..., n$ and then repeat step9 to step13

The second phase is the route improvement method. This algorithm applies local search and a meta-heuristic based on the concept of iteratively improving the solution to a problem by exploring neighboring ones. To design a λ-interchange local search descent method (λ-LSD), one typically needs to specify the following choices: how an initial feasible solution is generated, what move-generation mechanism to use, the acceptance criterion and the stopping test [3]. The λ-LSD is a type of neighborhood search that the set of all neighbors generated by the LSD for a given integer λ equal to 1 and 2. The move generation mechanism creates the neighboring solutions by the move operators (0, 1), (1, 0), (1, 1), (0, 2), (2, 0), (1, 2), (2, 1) and (2, 2). Here attribute could refer, for example, The operator (0, 1) on routes (R_p, R_q) indicates a shift of one customer from route q to route p. The operator (0, 1), (1, 0), (2, 0) and (0, 2) indicates a shift of one or two customers between two routes. The operator (1, 1), (1, 2), (2, 1) and (2, 2) indicate an exchange of a customer between two routes.

It is a sequential search which selects all possible combinations of different pair of routes. The first generation mechanism was introduced by Osman and Christofides [12]. If the neighboring solution is better, it replaces the current solution and the search continues. The acceptance strategy, the first best (FB) is used to selects the first neighbor that satisfies the pre-defined acceptance criterion.
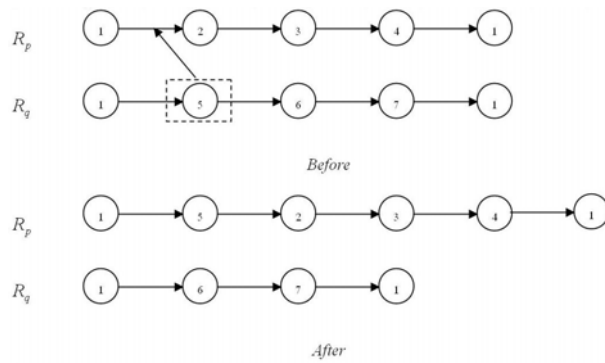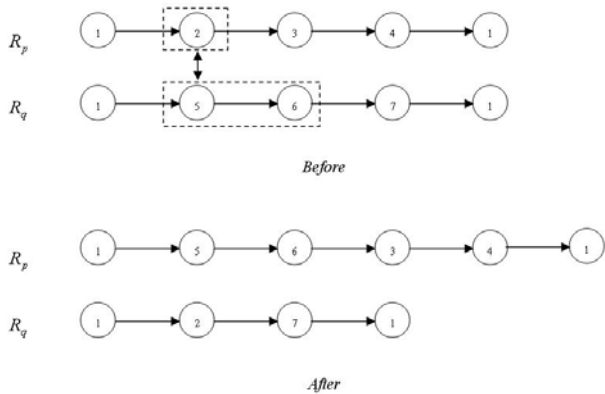
Fig. 1 The move operator (0, 1)

Fig. 2 The move operator (1, 2)

Then the TS is used as a diversification method to prevent that the algorithm falls into a local optimum. The TS is used to swap node or re-arranges a sequence of customers for each route. It is a memory-based search strategy which guides the local search descent method (LSD) to continue its search beyond local optimum [13][14]. When a local optimum is encountered, a move to the best neighbor is made to explore the solution space, even if it may cause of deterioration in the objective function value in equation (1). The TS seeks the best available move that can be determined in a reasonable amount of time. If the neighborhood is large or its elements are expensive to evaluate, candidate list strategies are used to help restrict the number of solutions examined on a given iteration. This hybrid algorithm for the VRPTW can be summarized as follows:

Step1: Construct the travel times matrix, where using Euclidean distances

Step2: Set the penalty weight factor parameters: $\alpha = 0.01$, $\gamma = 0.1$ and $\eta = 0.05$

Step3: Set the parameters for λ-LSD and TS, the number of iterations = 100 and the length of the tabu list =5

Step4: Obtain an initial MPFIH solution, x_0

Step5: Improve x_0 using the λ-LSD with the first-best selection strategy and prevent local

optima by using TS

Step6: Evaluate the fitness function

$\Delta f = F(x') - F(x_0)$, when x' is a possible solution that satisfies the constraints.

If $\Delta f > 0$ then $x = x'$ else $x = x_0$

Step7: If the stopping criterion is found then terminate the algorithm else go to step6.

The algorithms' performance is measured by two indicators. The first one refers to the quality of solution and the second one refers to the computer run time. The quality of the solution is compared with the best solution published in literature. The computer run time is hard to compare because there are many constraints must to considering. According to the type of computer, the type of computing software and the environments between runs are used. We select the best known algorithm, GA for benchmark test computer run time. GA is an efficient meta-heuristic method for a range of general applications. We design a GA, using MATLAB computing software and the same type of personal computer. We construct a simple GA involves three types of operators, thus, selection, crossover and mutation in order to solve VRPTW problems. The comparison shows CPU(s) by using the Solomon's 56 VRPTW benchmark instances with 100 customers.

4. Results and Discussion

To implement the algorithm, we created a source code using MATLAB computing software. We tested the algorithm on 6 types of Solomon's VRPTW benchmarking problems including R1, R2, C1, C2, RC1 and RC2. The experimental runs on 56 VRPTW instances. All instances have 25, 50 or 100 customer nodes and a single depot node. First, the quality of the solution is shown in Tables 1-3. The comparison results are separated to two objective functions, the minimum number of vehicles and the minimum total travel times as follows.

Table 1 The hybrid algorithm

Problems	Number of customers			
	25	50	100	All
R1	4.83	8.33	14.58	9.25
	482.13	840.82	1391.43	904.79
R2	2.44	4.33	6.82	4.69
	487.19	848.61	1321.58	915.85
C1	3.33	5.78	12.78	7.30
	289.42	637.04	1755.68	894.05
C2	2.00	3.13	6.88	4.09
	279.29	595.30	1332.43	755.51
RC1	3.75	8.25	14.75	8.92
	394.56	864.74	1584.88	948.06
RC2	2.50	5.29	7.63	5.13
	449.14	972.84	1555.16	993.23

Note. For each column two average results for Solomon's benchmarks are presented. First row in each problem is the average number of vehicles and second row is the average total travel times. Column "All" is the average results for all instances.

Table 2 The best solutions

Problems	Number of customers			
	25	50	100	All
R1	4.92	7.75	13.08	8.58
	463.37	766.13	1178.80	802.77
R2	2.89	4.11	3.09	3.34
	381.93	634.03	941.98	672.60
C1	3.00	5.00	10.00	6.00
	190.59	361.69	826.70	459.66
C2	2.00	2.75	3.00	2.61
	214.44	357.50	587.38	393.92
RC1	3.25	6.50	12.38	7.38
	350.24	730.31	1341.39	807.31
RC2	2.88	4.43	4.88	4.04
	325.53	585.24	1048.97	656.20

From Table 1 and Table 2 illustrate the result of the hybrid algorithm is effective, as it provides solutions competitive with best solutions, as well as new solutions that are not biased toward the number of vehicles. There are some new solutions that better than Solomon problem instances. They are shown in Table 3.

Table 3 New best-computed solutions for some Solomon benchmark problem instances

Problems	Best solutions		New best solutions	
	Vehicles	Travel Times	Vehicles	Travel Times
R101.25	8	617.1	7*	613.2*
R102.25	7	547.1	5*	494.7*
R110.25	4	444.1	4	433.5*
R111.25	5	428.8	4*	471.3
R102.50	11	909	9*	932.9
R103.50	9	772.9	8*	823.3
R101.100	20	1637.7	17*	1915.5
R102.100	18	1466.6	17*	1694.3
R201.25	4	463.3	3*	577.1
R203.25	3	391.4	2*	468.3
R207.25	3	316.6	2*	457
R210.25	3	404.6	2*	513.1
R203.50	5	605.3	4*	822.2
R210.50	4	645.6	3*	767.7
C205.50	3	359.8	2*	493.8
C206.50	3	359.8	2*	574.4
RC101.25	4	461.1	4	439.4*
RC203.25	3	326.9	2*	462.2
RC204.25	3	299.7	2*	406.5
RC206.25	3	324	2*	488.8
RC207.25	3	298.3	2*	403.2
RC203.50	4	555.3	3*	780.3

Note. * is the new best objective

The results from Table 3 show 22 new best solutions. There are 20 solutions in the first objective (minimum number of vehicles) and 4 solutions in the second objective better than or competitive as compared to the best solutions in Solomon's benchmark problem instances.

The computer run time comparison between the hybrid algorithm and GA is shown in Fig. 3.
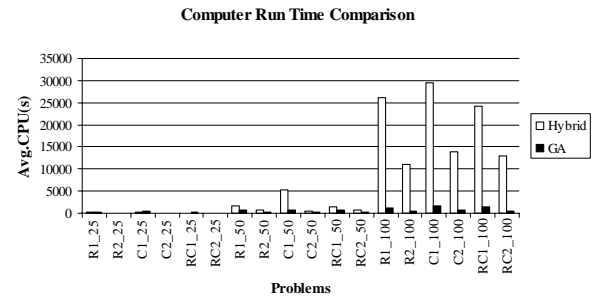

Fig. 3 Computer run time comparison

The results show a trend. The hybrid algorithm shows higher performance than the GA when the number of customers is lower than 25 nodes. The performance of the algorithm is lower than the GA when the number of customers increases over 50 nodes. The number of customers is an important factor in the performance of the hybrid algorithm but it has little effect in the GA. It is reasonable cause because of the main structure of the hybrid algorithm is local search algorithm, otherwise, GA is random search. This result demonstrates the effectiveness of the hybrid algorithm in the quality of solution more than running time. However, if the problem has the numbers of customers not exceed 25 nodes. The algorithm might be hold in this case and more effectiveness than GA.

In addition to the results, the types of problem which have a significant effect to computer run time of the algorithm, are of Type1: R1, C1 and RC1 (short scheduling horizon) and of type2: R2, C2 and RC2 (long scheduling horizon). The algorithm consumes more computer run time for Type1 than of Type2.

5. Conclusions and Future work

The modeling of VRPTW aims to optimize a multi-objective problem by using the hybrid algorithm. The results are compared according to two criteria, the quality of solution and computer run time. The quality of solution of the algorithm is effective, as it provides solutions competitive with the best solutions in the Solomon benchmark problem instances. In addition it provides the 20 new best solutions in the first priority objective that is proposed by this research.

The running time criterion, the experiments show clearly that the algorithm is higher performance than

GA when the number of customers is lower than 25 nodes. The performance of the algorithm decreases rapidly when the number of customers is over than 50 nodes. In addition to the types of benchmarking problems, there is significant effect to the computer run time.

For future work, we will improve this hybrid algorithm by using the meta-heuristic techniques, thus, simulated annealing algorithm, ant colony algorithm or GA to solve larger scale VRPTW problems, i.e. $n = 200$ to 1000 to illustrate its performance when the number of customers increases.

References

[1] J. Berger and M. Barkaoui, "A parallel hybrid genetic algorithm for the vehicle routing problem with time windows", Computers and Operations Research, Vol. 31, pp. 2037-2053, 2004

[2] M. W. P. Savelsbergh, "Local search for routing problems with time windows", Annals of Operations Research, Vol. 4 (1-4), pp.285-305, 1985

[3] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, Part I: Route construction and local search algorithms", Transportation Science, Vol. 39 (1), pp.104-118, 2005a

[4] J. F. Cordeau, M. Gendreau, G. Laporte, J. Y. Potvin and F. Semet, "A guide to vehicle routing heuristics", Journal of the Operational Research Society, Vol. 53, pp.512-522, 2002

[5] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time windows constraints", Operations Research, Vol. 35 (2), pp.254–265, 1987

[6] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, Part II: Metaheuristics", Transportation Science, Vol. 39 (1), pp.119-139, 2005*b*

[7] C. Qi and Y. Sun, "An improved ant colony algorithm for VRPTW", Computer Science and Software Engineering, Vol. 1, pp.455-458, 2008

[8] B. Ombuki, B. J. Ross and F. Hanshar, "Multi-objective genetic algorithms for vehicle routing problem with time windows", Appl. Intell., Vol 24(1), pp. 17-30, 2006.

[9] K. C. Tan, Y. H. Chew and L. H. Lee, "A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows", Computational Optimization and Applications, Vol. 34(1), pp.115-151, 2006

[10] M. M Solomon, "VRPTW benchmark problems", Available Source: http://web.cba.neu.edu/~msolomon/problems.htm , March 24, 2008.

[11] S. R. Thangiah, J. Y. Potvin and S. Tong, "Heuristic approaches to vehicle routing with backhauls and time windows", International Journal on Computers and Operations Research, Vol. 23(11), pp.1043-1057, 1996

[12] I. H. Osman and N. Christofides, "Capacitated clustering problems by hybrid simulated annealing and tabu search", International Transactions in Operational Research, Vol. 1(3), pp. 317-336, 1994

[13] F. Glover, "Tabu Search-Part I", ORSA Journal on Computing, Vol. 1, pp.190-206, 1989

[14] F. Glover, "Tabu Search-Part II", ORSA Journal on Computing, Vol. 2, pp.4-32, 1990

[15] S. R. Thangiah, "A Hybrid Genetic Algorithms, Simulated Annealing and Tabu Search Heuristic for Vehicle Routing Problems with Time Windows", Practical Handbook of Genetic Algorithms, Vol.3: Complex Structures, L. Chambers (Ed.), CRC Press, pp.347-381, 1999