Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (4): 633-638, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.633.638
© 2013 Asian Network for Scientific Information

Harmony Search Algorithm for Vehicle Routing Problem with Time Windows

[1]Esam Taha Yassen, [1]Masri Ayob, [1]Mohd Zakree Ahmad Nazri and [2]Zulkifli Ahmad
[1]Data Mining and Optimisation Research Group (DMO), Centre for Artificial Intelligent (CAIT), Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
[2]School of Linguistics and Language Studies, Faculty of Social Sciences and Humanities, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Abstract: Harmony Search Algorithm (HSA) is a meta-heuristic algorithm inspired by the musical process to solve different optimization problems. It has been successfully applied to solve many combinatorial optimization problems such as university course timetabling and standard vehicle routing problems but has not been applied to solve Vehicle Routing Problem with Time Windows (VRPTW). Therefore, this work investigates the performance of HSA in solving the vehicle VRPTW. The performance of HSA is tested using Solomon VRPTW benchmarks. The obtained results demonstrate that the HSA matching the best known result in one instance and obtained promising results in other tested instances. This proves that HSA is a promising method for solving VRPTW.

Key words: Harmony search algorithm, population-based approach, nature inspired algorithm

INTRODUCTION

The Vehicle Routing Problem (VRP) is a significant class of NP-hard combinatorial optimization issue which was proposed by Dantzig and Ramser (1959). VRP is identified as the matter of serving a number of customers by a number of vehicles, with the aim of reducing the cost of allocating the goods (Fig. 1).

There are many variations of VRP, such as: the Capacitated Vehicle Routing Problem (CVRP), the Vehicle Routing Problem with Pickup and Delivery (VRPPD), the Stochastic Vehicle Routing Problem (SVRP), the Site-Dependent Vehicle Routing Problem (SDVRP) and the Vehicle Routing Problem with Time Windows (VRPTW), (Fig. 2).

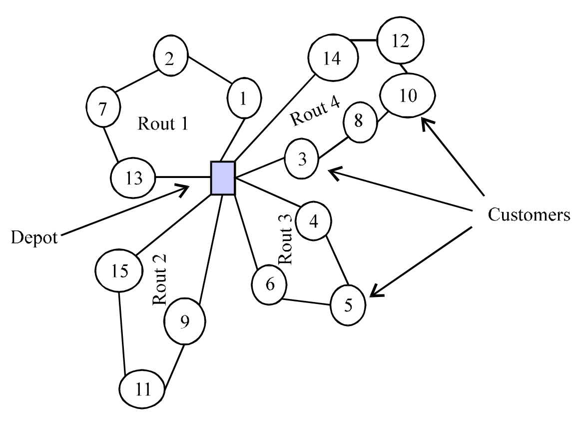

Fig. 1: A solution for an instance of a VRP with 15 a customers and 4 vehicles

The VRPTW is a significant and widespread extended form of VRP (Braysy and Gendreau, 2005). The time windows restriction is the most importance aspect in the VRPTW because it replicates the real life. The aim of VRPTW is to produce a feasible solution to serve all customers at the least cost while respecting the enforced restrictions. These restrictions are:

- Each customer should be visited exactly once during its time window

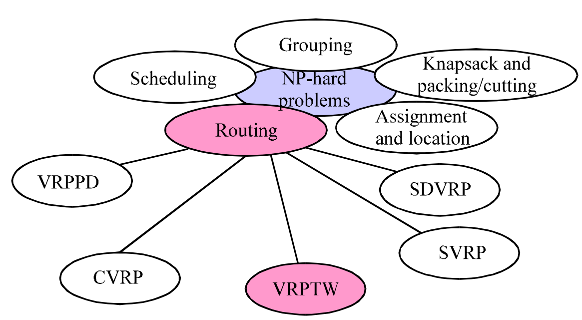

Fig. 2: VRP problem variants

Corresponding Author: Masri Ayob, Data Mining and Optimization Research Group (DMO), Center of Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia
Tel: +603 8921 6741 Fax: +603 8921 6184

- Each vehicle should begin and end at the depot
- The total demands of all customers in each route must not be exceed the capacity of vehicle

Over the years, VRPTW gets the attention of many researchers and a lot of solving approaches have been suggested for the VRPTW. These approaches can be classified into two main types: exact methods and heuristics methods. Exact methods have the ability to get the best solutions and guarantee their optimality (Talbi, 2009). Many exact methods have been proposed to handle VRPTW (Kolen *et al*., 1987; Desrosiers *et al*., 1984). Yet, because of the growth in the problem size and the VRPTW is NP-hard, exact methods are not useful options as they are only suitable to deal with a problem with a small size (Talbi, 2009).

Hence, many researchers adopted the heuristics and meta-heuristic approaches in solving the VRPTW. Heuristics/meta-heuristic approaches can obtain a good quality solution in a reasonable time but without guarantee the optimality of these solutions (Talbi, 2009). Meta-heuristic algorithms are classified into local search (i.e., single based) or population based. Examples of the local searches for VRPTW are: the greedy randomized adaptive search procedure (Kontoravdis and Bard, 1995), the simulated annealing (Czech and Czarnas, 2002) and the tabu search (Garcia *et al*., 1994). Many researchers adopt population heuristic approaches to solve VRPTW, for example: genetic algorithm (Braysy and Gendreau, 2001), scatter search (Russell and Chiang, 2006) and particle swarm optimization (Gong *et al*., 2012). To promote the search ability of meta-heuristics, many hybrid meta-heuristics were presented for the VRPTW, for examples, Bent and van Hentenryck (2004), Homberger and Gehring (2005), Zhang *et al*. (2008) and Yu *et al*. (2011).

A recent nature inspired algorithm is Harmony Search Algorithm (HSA). HSA is a population based algorithm that has a local search-based components (Lee and Geem, 2004). HSA is different from other meta-heuristics in the following aspects:

- HSA generate a new solution by joining various parts of the solution candidates from different solution regions. This makes it very good in exploration than other meta-heuristics
- HSA can freely deal with each component variable as it produces a new vector. However, genetic algorithms fail to do so as they have to preserve the gene structure
- HSA mix various features of current meta-heuristic algorithms (Lee and Geem, 2004) by means of using the harmony memory. As in tabu search algorithm, HSA has the ability of maintaining the history of former vectors. Furthermore, it is analogous to genetic algorithm in its ability of handling numerous solution vectors at the same time. Similar to simulated annealing algorithm, HSA changes its parameters value dynamically

According to the previous studies, HSA has been successful applied in tackling many challenging optimization problems such as, university course timetabling (Al-Betar *et al*., 2012), vehicle routing (Geem *et al*., 2005), Sudoku Puzzle (Geem, 2007) and other optimization problems. This motivates us to investigate the performance of HSA for the VRPTW.

PROBLEM DESCRIPTION

VRPTW is defined as follows: assume that G(V, E) is an undirected graph, where V represents a set of nodes V = {0, 1, 2,..., n}, node 0 represents the depot and nodes 1, 2,..., n represents customers. E is a set of edges, $E = \{(i,j): i \neq j \text{ and } i, j \in V\}$ and each edge is associated with the travel distance cost $c_{ij} = c_{ji}$ and $c_{ji}>0$. Each customer, except the depot, has a specific demand, service time and time windows which must be known in advance. In VRPTW, there are a fixed number of vehicles (v). Each vehicle has a capacity and this capacity is known in advance. If the vehicle arrives at the customer c_i before the beginning of its start time window e_i the vehicle must wait until its time window begins. On the other hand, the vehicle is unable to serve customer c_i if it arrived after the end of the time window l_i of that customer. Suppose the following variables:

v = Number of serving vehicles
n = Number of customer nodes (excluding the depot node)
q_i = Demand of costumer c_i
Q_k = Capacity of the kth vehicle
s_i = Service duration of costumer c_i
t_i = The arrive time at customer c_i
t_{ij} = Travel time from customer c_i to customer c_j
W_i = Waiting time at customer c_i
e_i = Starting time window for customer c_i
l_i = End time window for customer c_i

The VRPTW can be described mathematically as follows (Gong *et al*., 2012):

$$x_{ij}^k = \begin{cases} 1 & \text{if vehicle k travelled directly from customer } c_i \text{ to } c_j \\ 0 & \text{Otherwise} \end{cases}$$

$$y_i^k = \begin{cases} 1 & \text{if customer } c_i \text{ is served by vehicle } k \\ 0 & \text{Otherwise} \end{cases}$$

The quality of the solution S is measured as:

$$f(S) = \min \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{v} t_{ij} x_{ij}^k \quad (1)$$

s.t.:

$$\sum_{i=0}^{n} x_{ij}^k = y_j^k \quad \forall k = 1,\ldots,v\ ,\ \forall j = 1,\ldots,n \quad (2)$$

$$\sum_{j=0}^{n} x_{ij}^k = y_i^k \quad \forall k = 1,\ldots,v\ ,\ \forall i = 1,\ldots,n \quad (3)$$

$$\sum_{i=0}^{n} y_i^k \times q_i \le Q_k \quad \forall k = 1,\ldots,v \quad (4)$$

$$\sum_{k=1}^{v} y_i^k = 1 \quad \forall i = 1,\ldots,n \quad (5)$$

$$\sum_{k=1}^{v} y_0^k = v \quad (6)$$

$$t_i + W_i + s_i + t_{ij} = t_j \ \forall i, j = 0, 1, 2,\ldots, n,\ i \neq j \quad (7)$$

$$e_i \le t_i \le l_i \ \forall i, = 0, 1, 2,\ldots, n \quad (8)$$

$$W_i = \max \{e_i - t_i, 0\} \ \forall i = 0, 1, 2,\ldots, n \quad (9)$$

Constraints 2-3 guarantee that each vehicle enters and leaves from any customer provided that it serves that particular customer. Constraint 4 confirms that the vehicle capacity is not exceeded. Constraint 5 confirms that each customer is served once only. Constraint 6 proposes that each vehicle must start from the depot. Constraints 7-9 signify the time constraints; to make sure that time window is not violated.

HARMONY SEARCH ALGORITHM FOR VRPTW

Harmony Search Algorithm (HSA) is a population-based meta-heuristic method inspired by the musical improvisation practice in solving optimization problems (Geem *et al.*, 2001). In the musical improvisation process, a group of musicians play their instruments, step by step while learning from the preceding knowledge, to produce a good harmony. Improvisation process is similar to the process of creating a new solution in the optimization process and this solution is estimated using the objective function (Das *et al.*, 2011). The pitch of each musical instrument in the musician term denotes a decision variable in the optimization problems. The improvisation process task is to identify the group of the pitch that the musicians will play. These pitch are joined together to build a harmony. In optimization, each variable during the optimization procedure is allocated with a value at a time where combining these variable will generate a new solution *S* for the problem. Just like the improvisation process, where the harmony is repeatedly be adjusted to develop its quality; solutions are also repeatedly be improved in the optimization process to improve their quality, f(S) (Alia and Mandava, 2011). Figure 3 illustrates the flowchart of the basic HSA which has five components:

Step 1: Initialization of the HSA parameters: In this step, the HSA parameters are specified. These parameters are: the Harmony Memory Size (HMS), the Harmony Memory Consideration Rate (HMCR), the Pitch Adjustment Rate (PAR) and the Number of Improvisations (NI)

Step 2: Initialization of harmony memory (HM): In this step, the Harmony Memory (HM) is filled with a set of solutions which are generated randomly. The number of generated solutions is identical to HMS. After that, these solutions are arranged depending on the values of objective function, f(S)

Step 3: Improvisation process: A new harmony (solution) is generated in this step as follows (with regard to VRPTW): first create an empty solution S. Next, generate a random number r between zero and one. If r is less than the HMCR, choose one route from HM and add it to S. If not, generate an empty route and a group of customers are randomly allocated to this route as long as the assignment does not disrupt any

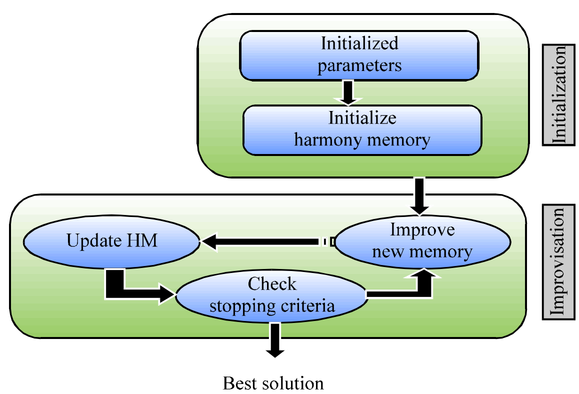

Fig. 3: Basic HSA flowchart

constraint. The route is then added to S. Next, any route that has been chosen from the HM is changed due to the PAR as follows: if the generated random number is less than PAR, randomly choose two customers from the current route and interchange their locations (if it does not disrupt VRPTW constrains). The improvisation process will be ended if the size of the new solution S (in terms of the number of routes) is equal to the largest one in HM. If S is unfeasible in the sense that there is some missed or repeated customers, repeated customers are deleted, whilst missed customers are either allocated to any exist route that can accommodate them or generate a new route for them

Step 4: **HM updating:** In this step, the new solution S which is generated from improvisation process, is added into HM if it is better than the worst solution in HM in the term of the quality, f(S)

Step 5: **Termination process:** This step tests the termination criterion of HSA that is characterized by the Number of the Improvisations (NI). The HSA will terminate if the maximum number of NI is attained. Otherwise, the improvisation and updating steps are repeated. HSA will return the best quality solution in HM when the stopping condition is reached

EXPERIMENTAL SETUP AND RESULTS

This section, discussed the properties of the selected Solomon's VRPTW benchmark instance that are utilized to assess the performance of HSA. In addition, the parameters setting of HSA will also be discussed. HSA is implemented in Java and executed on a PC running Mac OS with 2.8 GHz Quad-Core Intel Xeon processor and 6 GB RAM.

Solomon's VRPTW benchmark: Solomon's VRPTW benchmark dataset (Solomon, 1987) is adopted to evaluate the performance of HSA. Twelve instances called R101, R102, R201, R202, C101, C109, C201, C206, RC101, RC102, RC201 and RC202 are selected. Table 1 shows the characteristics of these instances:

Parameters setting: This section present the parameter settings for HSA. There are four parameters in HSA: HMS, HMCR, PAR and NI.

The parameter PAR is computed by the Eq. 10:

$$PAR(gn) = PAR_{max} - \frac{(PAR_{max} - PAR_{min})}{NI} \times gn \qquad (10)$$

where, gn and NI represent the current iteration and the maximum number of iteration, respectively. During the improvisation process, the value of PAR will be changed dynamically in descending manner (Taherinejad, 2009). From other works, PAR_{max} and PAR_{min} are set to 0.9 and 0.3, respectively (Das *et al*., 2011). The suitable values of other parameters (HMS, HMCR and NI) are fixed according to the preliminary test. Table 2 shows the best results (quality of solution, f(S)) over 10 runs with different parameter values for the HMS, HMCR and NI. The best values which are adopted for these parameters are: HMS = 20, HMCR = 0.7 and NI = 1000.

Table 1: Characteristics of selected Solomon's VRPTW instances

Dataset	No. of customers	No. of vehicles	Capacity of vehicle	Distribution of customers	Width of time window
R1-01	100	25	200	Random	Small Time Windows
R1-02	100	25	200	Random	Small Time Windows
R2-01	100	25	1000	Random	Large Time Windows
R2-02	100	25	1000	Random	Large Time Windows
C1-01	100	25	200	Cluster	Small Time Windows
C1-09	100	25	200	Cluster	Small Time Windows
C2-01	100	25	700	Cluster	Large Time Windows
C2-06	100	25	700	Cluster	Large Time Windows
RC1-01	100	25	200	Random /Cluster	Small Time Windows
RC1-02	100	25	200	Random /Cluster	Small Time Windows
RC2-01	100	25	1000	Random /Cluster	Large Time Windows
RC2-02	100	25	1000	Random /Cluster	Large Time Windows

Table 2: Results of different HSA parameter values

	HMS				HMCR				NI		
Instances	10	20	30	Instances	0.6	0.7	0.8	Instances	500	1000	2000
R101	2103.70	2074.47	2088.89	R101	2095.49	2074.47	2056.56	R101	2087.99	2074.47	2086.81
R201	1548.25	1496.51	1544.35	R201	1542.11	1496.51	1512.29	R201	1524.8	1496.51	1528.99
C101	1720.52	1661.05	1703.99	C101	1727.96	1661.05	1706.84	C101	1698.68	1661.05	1726.21
C201	683.23	591.56	824.89	C201	742.89	591.56	666.38	C201	809.85	591.56	603.88
RC101	2287.16	2292.89	2244.20	RC101	2241.98	2292.89	2267.29	RC101	2201.64	2292.89	2254.90
RC201	1777.79	1744.75	1765.92	RC201	1846.68	1744.75	1792.41	RC201	1823.57	1744.75	1807.54
Average	1686.78	1643.54	1695.37	Average	1699.52	1643.54	1666.96	Average	1691.09	1643.54	1668.06

Table 3: Results of HSA

		HSA			
Instances	Best-known	Best	AV	Std.	PD
R1-01	1645.79	2061.79	2137.62	33.85	25.28
R1-02	1486.12	1803.25	1962.83	42.43	21.34
R2-01	1252.37	1532.24	1584.98	29.61	22.35
R2-02	1191.70	1329.84	1460.78	34.30	11.59
C1-01	828.94	1640.07	1761.54	62.33	97.85
C1-09	828.94	1406.84	1586.32	55.53	69.72
C2-01	591.56	591.56	780.19	149.47	0.00
C2-06	588.49	800.13	894.26	43.35	35.96
RC1-01	1696.94	2154.70	2303.16	52.36	26.98
RC1-02	1554.75	2033.93	2136.89	40.67	30.82
RC2-01	1406.91	1749.43	1885.26	52.82	24.35
RC2-02	1365.65	1564.13	1734.14	49.66	14.53

The values at column best-known, Best and AV are the quality of solution, f(S)

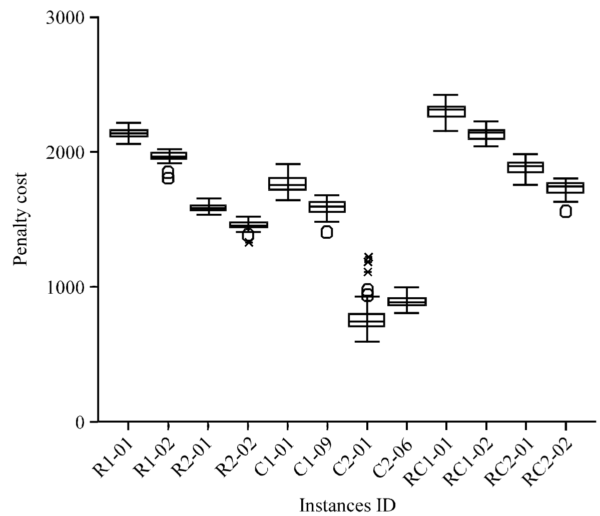

Fig. 4: Box plot of solution distribution for HSA

Computational result: Table 3 illustrates the results obtained by HSA for VRPTW over 51 independent runs. The presented results are the best solution (Best), average solution (Average), standard deviation (Std) and the Percentage Deviation (PD) from the best known results (Gong *et al*., 2012).

These results show that HSA matches the best known result in C2-01 instance as soon as, for other tested instances, HSA obtain promising results. The box plots of solutions distribution over 51 runs for HSA are presented in Fig. 4.

CONCLUSION

There are many characteristics in harmony search algorithm make it very effective method for solving NP-hard optimization problems. This work applied harmony search algorithm for solving VRPTW. The performance of harmony search algorithm has been tested on twelve Solomon benchmark instances. Results showed that the HSA matching the best known result in C2-01 instance and obtained promising results in other tested instances. To support this, we plot the box-whiskers plot of solutions distribution.

ACKNOWLEDGMENT

The authors wish to thank Ministry of Higher Education for supporting this work under the FRGS Research Grant Scheme (FRGS/1/2012/SG05/UKM/02/11).

REFERENCES

Al-Betar, M.A., A.T. Khader and M. Zaman, 2012. University course timetabling using a hybrid harmony search metaheuristic algorithm. IEEE Trans. Syst. Man Cybernet. Part C: Appli. Rev., 42: 664-681.

Alia, O.M. and R. Mandava, 2011. The variants of the harmony search algorithm: An overview. Artif. Intell. Rev., 36: 49-68.

Bent, R. and P. van Hentenryck, 2004. A two-stage hybrid local search for the vehicle routing problem with time windows. Transport. Sci., 38: 515-530.

Braysy, O. and M. Gendreau, 2001. Genetic algorithms for the vehicle routing problem with time windows. Arpakannus, 1: 33-38.

Braysy, O. and M. Gendreau, 2005. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. Transport. Sci., 39: 104-118.

Czech, Z.J. and P. Czarnas, 2002. Parallel simulated annealing for the vehicle routing problem with time windows. Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-Based Processing, January 9-11, 2002, Canary Islands, Spain, pp: 376-383.

Dantzig, G.B. and J.H. Ramser, 1959. The truck dispatching problem. Manage. Sci., 6: 80-91.

Das, S., A. Mukhopadhyay, A. Roy, A. Abraham and B.K. Panigrahi, 2011. Exploratory power of the harmony search algorithm: Analysis and improvements for global numerical optimization. IEEE Trans. Syst. Man Cybern. B: Cybern., 41: 89-106.

Desrosiers, J., F. Soumis and M. Desrochers, 1984. Routing with time windows by column generation. Networks, 14: 545-565.

Garcia, B.L., J.Y. Potvin and J.M. Rousseau, 1994. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. Comps Opns. Res., 21: 1025-1033.

Geem, Z.W., 2007. Harmony search algorithm for solving sudoku. Proceedings of the 11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems and 17th Italian Workshop on Neural Network, September 12-14, 2007, Vietri sul Mare, Italy, pp: 371-378.

Geem, Z.W., C. Tseng and Y. Park, 2005. Harmony search for generalized orienteering problem: Best touring in china. Adv. Nat. Comput., 3612: 741-750.

Geem, Z.W., J.H. Kim and G.V. Loganathan, 2001. A new heuristic optimization algorithm: Harmony search. Simulation, 76: 60-68.

Gong, Y.J., J. Zhang, O. Liu, R.Z. Huang, H.S.H. Chung and Y.H. Shi, 2012. Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev., 42: 254-267.

Homberger, J. and H. Gehring, 2005. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. Eur. J. Oper. Res., 162: 220-238.

Kolen, A.W., A.R. Kan and H.W.J.M. Trienekens, 1987. Vehicle routing with time windows. Oper. Res., 35: 266-273.

Kontoravdis, G. and J.F. Bard, 1995. A GRASP for the vehicle routing problem with time windows. ORSA J. Comp., 7: 10-23.

Lee, K.S. and Z.W. Geem, 2004. A new structural optimization method based on the harmony search algorithm. Comput. Struct., 82: 781-798.

Russell, R.A. and W.C. Chiang, 2006. Scatter search for the vehicle routing problem with time windows. Europ. J. Operat. Res., 169: 606-622.

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operat. Res., 35: 254-265.

Taherinejad, N., 2009. Highly reliable harmony search algorithm. Proceedings of the European Conference on Circuit Theory and Design, August 23-27, 2009, Antalya, pp: 818-822.

Talbi, E.G., 2009. Metaheuristics: From Design to Implementation. John Wiley and Sons, Hoboken, New Jersey.

Yu, B., Z.Z. Yang and B.Z. Yao, 2011. A hybrid algorithm for vehicle routing problem with time windows. Expert Syst. Appli., 38: 435-441.

Zhang, Q., T. Zhen, Y. Zhu, W. Zhang and Z. Ma, 2008. A hybrid intelligent algorithm for the vehicle routing with time windows. Proceedings 4th International Conference on Intelligent Computing Advanced Intelligent Computing Theories and Applications: With Aspects of Theoretical and Methodological Issues, September 15-18, 2008, Shanghai, China, pp: 47-54.