Research Journal of Recent Sciences ____ ISSN 2277-2502
Vol. **4(2),** 30-35, February **(2015)** *Res.J.Recent Sci.*

Using an Ant Colony approach for Solving capacitated Vehicle Routing Problem with time Windows

Maryam Razavi,* and Abbas Toloie Eshlaghy
Department of industrial management, Science and Research Branch, Islamic Azad University, Tehran, IRAN

Available online at: www.isca.in, www.isca.me
Received 3rd December 2013, revised 21st February 2014, accepted 25th April 2014

Abstract

In this paper, a capacitated vehicle routing problem with time windows (CVRPTW) is presented. In this work, a new idea for calculating the heuristic value to improve the performance of ant colony algorithms when solving capacitated vehicle routing problems with hard time windows is proposed. The performance of the model and the heuristic approach are evaluated by Solomon's VRPTW benchmark problems. The results show that in 14 problem instances, our solutions are better than the best solutions reported for the VRPTW by other researchers in both total traveling distance and number of used vehicles. Our solutions superiority over the best solutions published in the literature are in instances R1, R2 and Particularly RC2 such a way that the average number of used vehicles are considerably less.

Keywords: Capacitated vehicle routing problem with time windows (CVRPTW), ant colony optimization (ACO), combinatorial optimization problems, metaheuristics.

Introduction

Transportation of goods is an important task in the society of today. Large amounts of money are spent daily on logistics. So, Attempting to reduce the amount of money spent on transportation as even small improvements can lead to huge improvements in absolute terms.

Vehicle routing problems (VRPs) are central to logistics management which consist of a fleet of vehicles and a set of customers to be visited. The cost of traveling between each pair of customers and between the depot and each customer is given. Our task is to find a route for each vehicle, starting and ending at the depot, such that all customers are served by exactly one vehicle, and such that the overall cost of the routes are minimized, subject to side constraints. The most common operational constraints impose that the total demand carried by a vehicle at any time does not exceed a given capacity, service time windows set by customers are not violated and the total duration of any route is not greater than a prescribed time window of depots. Taking into consideration the time windows of the customers and the depot(s) and bounded capacity of vehicles extends the problem into the capacitated vehicle routing problem with time windows (CVRPTW).

In transportation management, the VRP have a significant economic importance because of many practical applications so, many researchers have focused to develop solution approaches for these problems. Single Depot VRP, SDVRP, was first proposed in 1959 by Dantzig and Ramser[1]. In 1987 Solomon added time-window constraints to the classical VRP and introduced a set of well known benchmark problems now known as ''Solomon Instances''[2]. Laporte described both exact and approximate algorithms for VRP problem[3]. Larsen used exact approach using Dantzig–Wolfe decomposition for the VRPTW[4]. Branch-and-cut algorithms for the CVRP was developed by Lysgaard et al.[5]. The vehicle routing problem has been and is still an enriched research topic for researchers. The VRP is a hard combinatorial optimization problem and because of their difficulty they are called NP-hard problems. So, a large number of heuristic algorithms have been proposed in the last decade. Good metaheuristics for the Capacitated VRP, CVRP, were developed such as parallel tabu search by Taillard[6]. Various heuristic methods also have been used for VRPTW. For example, Chiang and Russell solved VRPTW with simulated annealing[7], Ombuki et al., Berger et al. used genetic algorithm for VRPTW[8,9]. Cordeau et al. introduced exact and heuristic methods on VRPTW[10]. Bräysy and Gendreau (2005a,b) focuses on local search and Metaheuristics for VRP with TW[11,12]. An inspiring family of heuristics for CVRP is represented by the adaptive large neighborhood search (ALNLS) of Pisinger and Ropke which used ruin and recreate moves[13]. Chand et al. presented a genetic algorithm based approach is designed to resolve a bi-criteria CVRP in which the number of vehicles and the total distance are minimized[14]. Wang and Li designed a hybrid genetic algorithm for a multi-objective VRPTW in order to minimize the total distance and maximize client satisfaction by fulfilling time-window requirements[15]. Tavakkoli-Moghaddam et al. have proposed a VRP with hard time windows using a simulated annealing (SA) approach and the considered criteria are fleet cost, routes cost, and violation of hard time windows penalty to minimize[16]. Tan et al., Ombuki et al. and Ghoseiri and Ghannadpour, have considered the VRPTW as a biobjective optimization problem, minimizing the number of vehicles and the total travel distance by using of genetic algorithm for solving the standard Solomon's benchmark to evaluate the quality of the proposed algorithm[17,18].

Let G =(N,A) be a graph where $N=\{0,1,...,n\}$ is the set of nodes and $A=\{(i,j)\mid i,j\in N, i\neq j\}$ is the set of arcs. The depot is represented by node 0 and the other nodes in N corresponds to customers. Each customer has a known demand ($q_i \geq 0$) and a hard time window $[e_i, l_i], e_i \geq 0, l_i \geq 0$. The time window at the depot $[e_0, l_0]$, corresponds to the scheduling horizon. We assume that the service at the customers cannot start before or after the time windows. If a vehicle arrives early at a customer, the vehicle should wait until the customer time window open and a customer can not be served outside its time window. Each arc $(i,j)\in A$ has a parameter d_{ij} which is the distance of that arc. Each customer must be assigned to exactly one of the k vehicles and the total size of each route assigned to each vehicle must not exceed the vehicle capacity Q_k .The aim is to construct a set of vehicle routes in order to Minimize the distance traveled by the vehicles and total number of vehicles used to serve the customers. Therefore a solution requiring fewer routes is always considered better than a solution with more routes by fulfilling the following requirements:Vehicle capacity constraints are observed. Time window constraints are considered. Each customer is met by each vehicle exactly once.Each vehicle starts its journey from depot and ends at the depot

This paper proposes a new idea to improve the performance of ant colony algorithms when solving capacitated vehicle routing problems with time windows. The paper is further organized as follows: This part offers a brief review of the Literature of the vehicle routing problems and formally describes the Capacitated Vehicle Routing Problem with Time Windows (VRPTW). Next section presents the ant colony meta-heuristic based ant colony system, introduce the new idea for calculation of its parameters. Then the performance of the model and the heuristic approach are evaluated using solomon's instances and at last conclusions are stated.

Material and Methods

Two classes of algorithms are available for the solution of combinatorial optimization problems: exact and approximate algorithms. Exact algorithms are guaranteed to find the optimal solution but in the case of NP-hard problems, exact algorithms need, in the worst case, exponential time to find the optimum and in this case the performance of exact algorithms is not satisfactory. Approximate algorithms, often also loosely called heuristic methods, seek to obtain good, that is, near-optimal solutions at relatively low computational cost without being able to guarantee the optimality of solutions. A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. The use of metaheuristics has significantly increased the ability of finding very high-quality solutions to combinatorial optimization problems in a reasonable time. Ant colony optimization is a metaheuristic in which a colony of artificial ants cooperates in finding good solutions to difficult discrete optimization problems. In this section, we first present the ant colony optimization and the general algorithm. Then, we detail the elements of the ant colony algorithm adapted to the CVRPTW.

Ant colony optimization: The principle of ACO algorithms (Dorigo and Gambardella, 1997) is based on the way ants search for food[19]. Each ant consider pheromone trails left by all other ant colony members which preceded its route, the pheromone trail being a signal, a smell left by every ant on its way. This pheromone evaporates with time, and therefore the probabilistic value of each route for each ant changes with time. When ants construct their routes, the path to the food will be characterized by higher pheromone traces and thus all ants will follow the same path. So, ACO can be used to find a solution to the shortest path problem. In VRP, a solution is described in terms of paths through depots to customers in accordance with the problems' constraints.

In this algorithm, imitating the ants' feeding behavior, a number of artificial ants with the described characteristics search for good quality solutions of the optimization problem. Artificial ants are the main part of ACO which concurrently build a tour. At each construction step, ant k applies a probabilistic action choice rule, called random proportional rule, to decide which node to visit next. For ant *k*, the probabilistic transition rule is indicated by p_{ij}^k which represents the probability of choosing to move from node *i* to node *j* (which is not met yet: N_i^k), and is given by:

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}.\eta_{ij}^{\beta}}{\sum_{l\in N_i^k}\tau_{il}^{\alpha}.\eta_{il}^{\beta}} & if\ j\in N_i^k) \\ 0 & otherwise \end{cases} \qquad (1)$$

where β and α are respectively parameters controlling the importance of the trail τ_{ij} and the actual attractiveness η_{ij} (set it to $1/d_{ij}$ for classic VRP) for of the arc (*i,j*). In this article for the problem with hard time window the new formula for calculating the heuristic value η_{ij} is proposed as follows:

$$\eta_{ij} = \{[(\max\{t_{ij}(t_i), e_j - t_i\})^a * (l_j - t_i)^b]^{1/(a+b)}\}^{-1}$$

Where: $t_{ij}(t_i)$ is the arrival time to *j* at time t_i

t_i is the departure time from node *I*, $[e_j, l_j]$ is the time window of customer *j*. e_j and l_j are the earliest and the latest time of servicing customer *j* respectively.

The heuristic value η_{ij} for CVRPTW is composed of two parts: The first part is consist of $t_{ij}(t_i)$ (the distance to the customer) and $e_j - t_i$ (urgency of servicing). If the vehicle meets customer *j* before starting of its time window, $e_j - t_i$ will choose. This means that it should wait until customer *j* becomes ready for servicing and the second part is $l_j - t_i$ which is the time distance between departure time from customer *i* and the latest time of servicing customer *j*. This formula is acquired from the idea that the closer clients, which their time windows are met, are more attractive to choose based on equation 1 and the positive exponents a and b are parameters determining the relative importance of first part versus second part.

Due to ant colony system algorithm, ant k selects customer *j* according to the so called pseudorandom proportional rule,

Given by $j = \begin{cases} \arg\max_{l \in N_i^k} \{\tau_{il}\eta_{il}^{\beta}\} & if \; q \le q_0 \\ J & otherwise \end{cases}$

(exploitation) where q is a random variable uniformly distributed in $[0\ 1]$ and $q_0, 0 \le q_0 \le 1$ is a parameter and *J* is a random variable selected according to the probability distribution given by equation(1) (exploration). Tuning the parameter q_0 determines the degree of importance of exploration versus exploitation and the choice of whether to concentrate the search of the system around the best-so-far solution or to explore other tours.

When ant *k* construct its route, a local pheromone updating is performed on the pheromone matrix, according to the following rule:

$$\tau_{ij} = (1-\varphi).\tau_{ij} + \varphi.\tau_0 \quad (2)$$

Where the value of τ_0 is set to be the same as the initial value for the pheromone trails and $\varphi \ (0 \le \varphi \le 1)$ is a parameter regulating pheromone evaporation. The effect of the local updating rule is that each time an ant moves from node *i* to *j* its pheromone trail τ_{ij} is reduced, so that the arc becomes less desirable for the following ants and makes an increase in the exploration of arcs that have not been visited yet.

Once the m ants of the colony have completed their computation, the best known solution (best-so-far or best-iteration) is used to globally modify the pheromone trail by applying evaporation:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\,\Delta\tau_{ij}^{best} \quad (3)$$

where $\Delta\tau_{ij}^{best} = 1/L_{best}$ and $\rho \ (0 \le \rho \le 1)$. The ant which is allowed to add pheromone may be either the best-so-far, in which case $\Delta\tau_{ij}^{best} = 1/L_{bs}$ where L_{bs} is the length of the best so far tour, or the iteration-best, in which case $\Delta\tau_{ij}^{best} = 1/L_{ib}$, where L_{ib} is the length of the iteration-best tour.

ACO procedure: We propose the following ant colony optimization algorithm based on ant colony system: Set parameters, initialize pheromone trails
Best objective function = ∞
For each arc (i, j)
$\tau_{ij} = \tau_0$
End For
While (termination condition not met) do
For k= 1 to m
While (Ant k has not completed its solution)
Select the next customer j;
Employ local pheromone update
End While
Objective function= length of the current solution;
If (Objective function < Best Objective function)
Best Objective function = Objective function
Best Solution = current solution;
End If
End For
For each move (i, j) in solution Best Solution
Employ global pheromone update
End For
End While

Results and Discussion

This section is described the experimental evaluation of the proposed algorithm on solomon benchmarks. The algorithms have been coded in Matlab 2010a, and all the tests have been carried out on a 2.2GHz Intel Celeron CPU. Tests were performed on Solomon's instances for the VRPTW. These instances are euclidean and the travel time between two customers is identical to the corresponding euclidean distance. These instances are categorized in three classes, the customers are randomly generated (R), clustered (C) or both clustered and randomly generated (RC). Problem sets R1, C1 and RC1 have a short scheduling horizon and cannot service many customers at one time. In contrast, the sets R2, C2 and RC2 have a long scheduling horizon permitting many customers to be serviced by the same vehicle.

Solutions for each problem data are gathered then averaged for each problem type and the result is reported in the table-1.

Table-1
Solutions for each 56 benchmark problems (Solomon, 1987)

	R 1			**C1**			**RC1**	
	Distance	NV		Distance	NV		Distance	NV
R 101	1469	15	C101	829	10	RC101	1613	13
R 102	1394	13	C102	828	10	RC102	1477	12
R 103	1220	11	C103	828	10	RC103	1324	11
R 104	1164	10	C104	827	10	RC104	1284	11
R 105	1378	13	C105	829	10	RC105	1564	12
R 106	1289	12	C106	828	10	RC106	1379	11
R 107	1146	10	C107	829	10	RC107	1334	11
R 108	1113	10	C108	828	10	RC108	1240	11
R 109	1210	11	C109	829	10	**avg**	**1401.88**	**11.50**
R 110	1139	10	**avg**	**828.33**	**10.00**			
R 111	1151	11						
R 112	1102	10						
avg	**1231.25**	**11.33**						
	R 2		**C2**		**RC2**			
	Distance	NV		Distance	NV		distance	NV
R 201	1153	3	C201	590	3	RC201	1337	3
R 202	1047	3	C202	591	3	RC202	1132	3
R 203	932	2	C203	600	3	RC203	1028	3
R 204	846	2	C204	590	3	RC204	860	2
R 205	1058	3	C205	589	3	RC205	1231	3
R 206	960	2	C206	588	3	RC206	1205	3
R 207	897	2	C207	588	3	RC207	1063	3
R 208	809	2	C208	588	3	RC208	990	3
R 209	979	2	**avg**	**590.5**	**3**	**avg**	**1105.75**	**2.88**
R 210	971	2						
R 211	892	2						
avg	**958.55**	**2.27**						

According to the quality of the solution generated through this proposed algorithm, the best solution known of a number of problem instances have been improved. For example, the best known solution for problem R101 is reported as 1608 and 18 which are minimum distance and minimum number of vehicles, respectively. Whereas the solution with proposed algorithm is 1483 and 14. The new best solution value by our proposed algorithm for this problem is the following: (0 shows depot number and the other numbers 1 to 100 indicate demands number.)

Route 1: 0 59 92 95 98 61 16 85 99 96 94 6 89 0, **Route 2:** 0 14 42 15 2 57 87 97 37 91 100 0, **Route 3:** 0 72 39 23 75 22 41 73 21 74 56 0, **Route 4:** 0 5 83 45 82 7 88 31 69 1 50 76 0, **Route 5:** 0 27 52 30 51 81 33 79 3 77 68 0, **Route 6:** 0 63 62 11 19 47 36 49 64 0, **Route 7**: 0 65 71 9 78 34 35 66 0, **Route 8**: 0 28 12 40 53 26 54 55 25 0, **Route 9:** 0 44 38 86 84 17 60 18 0, **Route 10:** 0 67 43 13 93 0, **Route 11:** 0 90 10 32 20 70 0, **Route 12:** 0 8 46 48 0, **Route 13:** 0 29 24 80 0, **Route 14:** 0 4 0, **Route 15:** 0 58 0

Table-2 shows the comparison between the best solutions Found in the literature for problems whose solutions are considerably improved and our solutions. The solutions obtained by the proposed algorithm are better in both total traveling distance and number of used vehicles than those in the literature.

H - J. Homberger, RT - Y. Rochat and E.D. Taillard, LLH - H. Li, A. Lim, and J. Huang, HG - J. Homberger and H. Gehring, RGP - L.M. Rousseau, M. Gendreau and G. Pesant, WL - Woch M., Lebkowski P, TBGGP - E. Taillard, P. Badeau, M. Gendreau, F. Geurtin, and J.Y. Potvin, BBB - J. Berger, M. Barkaoui and O. Bräysy, MBD - D. Mester, O. Bräysy and W. Dullaert, GCC - Agnieszka Debudaj-Grabysz, Zbigniew J.Czech and Piotr Czarnas, CC - Z. J. Czech and P. Czarnas.

Table-2
Comparison between the best solutions Found in the literature and the solutions obtained by the proposed algorithm

	The best solution found in the literature			**Proposed algorithm**	
Problem	Distance	NV	Authors	Distance	NV
R 101	1650.80	19.00	H	1469	15
R 102	1486.86	17.00	RT	1394	13
R 103	1292.67	13.00	LLH	1220	11
R 201	1252.37	4.00	HG	1153	3
R 202	1191.70	3.00	RGP	1047	3
R 203	939.50	3.00	WL	932	2
RC101	1696.95	14	TBGGP	1613	13
RC102	1554.75	12	TBGGP	1477	12
RC105	1629.44	13	BBB	1564	12
RC106	1424.73	11	BBB	1379	11
RC201	1406.94	4	MBD	1337	3
RC202	1365.64	3	GCC	1132	3
RC203	1049.62	3	CC	1028	3
RC205	1297.65	4	MBD	1231	3

Table 3 compares some of the route construction algorithms by different authors regarding to the average number of vehicles and average total distance[22-28]. The results from the comparison can be summarized as follows:

Conclusion

This paper presented ant colony algorithm approach to the capacitated vehicle routing problem with time windows. The proposed algorithm has been performed on the benchmark Solomon's 56 VRPTW 100-customer instances, which obtained 14 routing solutions better than the best solutions reported for the VRPTW by other researchers in both total traveling distance and number of used vehicles as shown in table 2 and the rest are competitive as compared to the best solutions published in literature. The different results have been more in instances R1, R2 and Particularly RC2 such a way that the average amount of total traveling distance and number of used vehicles as illustrated in table 3 have represented the superiority of the solution of proposed algorithm with the best solutions published specially in the average number of used vehicles.

Moreover, the most significant contribution of this paper is the new formula for calculating the heuristic value η_{ij} (an ant colony parameter) in case of hard time windows which is acquired from the idea that the closer clients, which their time windows are met, are more attractive to choose. Future work will be directed towards expanding the model with the addition of other constraints, for example the impact of future and real-time information on the solution quality.

Table-3
Comparison of average number of vehicles and average total distance between of the route construction algorithms by different authors and the proposed algorithm

	R 1	**R 2**	**C1**	**C2**	**RC1**	**RC2**
Thompson et al. (1993)	13.00	3.18	10.00	3.00	13.00	3.71
	1356.92	1276.00	916.67	644.63	1514.29	1634.43
Potvin et al. (1995)	13.33	3.27	10.00	3.13	13.25	3.88
	1381.90	1293.40	902.90	653.20	1545.30	1595.10
Bräysy (2001a)	12.17	2.82	10.00	3.00	11.88	3.25
	1253.24	1039.56	832.88	593.49	1408.44	1244.96
Homberger & Gehring, (2005)	12.08	2.82	10.00	3.00	11.50	3.25
	1211.67	950.72	828.45	589.96	1395.93	1135.09
Pisinger & Röpke, (2005)	11.92	2.73	10.00	3.00	11.50	3.25
	1212.39	957.72	828.38	589.12	1387.12	1123.49
Mester et al., (2007)	12.00	2.73	10.00	3.00	11.50	3.25
	1208.18	954.09	828.38	589.12	1387.12	1119.70
Pisinger & Ropke, (2007)	12.03	2.75	10.00	3.00	11.60	3.25
	1215.16	965.94	828.38	589.86	1385.86	1135.46
Nagata, Braysy & Dullaert, (2010)	11.92	2.73	10.00	3.00	11.50	3.25
	1210.34	951.71	828.38	589.86	1384.30	1119.43
Cordeau & Maischberger, (2012)	12.00	2.73	10.00	3.00	11.50	3.25
	1209.19	951.17	828.38	589.86	1385.90	1120.53
proposed algorithm	**11.33**	**2.27**	**10.00**	**3.00**	**11.50**	**2.88**
	1231.25	**958.55**	**828.52**	**590.50**	**1401.88**	**1105.75**

References

1. Dantzig G., Ramser B. and Hubert J., The Truck Dispatching Problem, *Management Science*, **6(1)**, 80–91 **(1959)**
2. Solomon M.M., Algorithms for the vehicle routing and scheduling problems with time window constraints, Oper Res, **35(2)**, 254–265 **(1987)**
3. Laporte G., The vehicle routing problem : An overview of exact and approximate algorithms, Europe Journal of Operational Research, **59(3)**, 345–358 **(1992)**
4. Larsen J., Parallelization of the vehicle routing problem with time windows. Ph.D. thesis, Department of Mathematical Modelling, Technical University of Denmark, Lynghy, Denmark **(1999)**
5. Lysgaard J., Letchford A.N. and Eglese, R.W., A new branch-and-cut algorithm for the capacitated vehicle routing problem, *Math Program*, **100**, 423–445 **(2004)**
6. Taillard R.E., Parallel iterative search methods for vehicle routing problems, *Networks*, **23**, 661–73 **(1993)**
7. Chiang W.C. and Russel R.A., Simulated annealing metaheuristic for the vehicle routing problem with time windows, *Annals of Operations Research*, **63**, 3–27 **(1996)**
8. Ombuki B., Ross B.J., Hanshar F., Multi-objective genetic algorithms for vehicle routing problem with time windows, *App Intell*, **24(1)**,17–30 **(2006)**
9. Berger J., Barkaoui M., A parallel hybrid genetic algorithm for the vehicle routing problem with time windows, *Computers and Operations Research,* **31**, 2037–2053 **(2004)**
10. Cordeau J.F., Gendreau M., Laporte G., Potvin J.Y. and Semet, F., A guide to vehicle routing heuristics, *Journal of the Operational Research Society*, vol. **53(5)**, 512–522 **(2002)**
11. Bräysy O., Gendreau M., Vehicle routing problem with time windows, part I: route construction and local search algorithms, *Transp Sci,* **39**,104–118 **(2005a)**
12. Bräysy O. and Gendreau M., Vehicle routing problem with time windows, part II: metaheuristics, *Transp Sci*, **39**,119–139 **(2005b)**
13. Pisinger D. and Ropke S., A general heuristic for vehicle routing problems, *Comput Oper Res*, **34**,2403– 2435 **(2007)**
14. Chand P., Mishra B.S.P. and Dehuri S., A multi objective genetic algorithm for solving vehicle routing problem, *Int J Info Tech Knowl Mgmt*, **2**,503–506 **(2010)**
15. Wang C.H. and Li C.H., Optimization of an established multi-objective delivering problem by an improved hybrid algorithm, *Expert Syst Appl*, **38(4)**,4361–4367 **(2011)**
16. Tavakkoli-Moghaddam R., Safaei N. and Shariat M.A., A multi-criteria vehicle routing problem with soft time windows by simulated annealing, *J Indus Eng*, **1(1)**,28–36 **(2005)**
17. Tan K.C., Chew Y.H. and Lee L.H., A hybrid multi-objective evolutionary algorithm for solving vehicle routing problem with time windows, *Comput Opt Appl*, **34(1)**,115–15 **(2006)**
18. Ghoseiri K. and Ghannadpour S.F., Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm, *Appl Soft Comput*, **10**,1096–1107 **(2010)**
19. Dorigo M. and Gambardella L.M., Ant Colony System: A cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation*, **1**, 53–66 **(1997)**
20. Solomon Benchmark Problems http://web.cba.neu.edu/msolomon/problems.html
21. Best Known Solution http://www.sintef.no/TOP/Problems/VRPTW/Solomon-benchmark/100- customers/
22. Thompson P.M. and Psaraftis H.N., Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems, *Operations Research*, **41**, 935–946 **(1993)**
23. Potvin J.Y. and Rousseau J.M., An Exchange Heuristic for Routing Problems with Time Windows, *Journal of the Operational Research Society,* **46**, 1433–1446 **(1995)**
24. Homberger J. and H. Gehring, A two-phase hybrid metaheuristic for the vehicle routing problem with time windows, *European Journal of Operational Research*, **162 (1)**, 220-238 **(2005)**
25. Mester D., Bräysy O. and Dullaert W., A Multi-parametric Evolution Strategies Algorithm for Vehicle Routing Problems, *Expert Systems with Applications*, **32(2)**, 508-517 **(2007)**
26. Pisinger D, Ropke S, A General Heuristic for Vehicle Routing Problem, Department of Computer Science, University of Copenhagen **(2005)**
27. Nagata Y., Bräysy O. and Dullaert W., A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows, *Comput Oper Res*, **37**,724–737 **(2010)**
28. Cordeau J.F. and Maischberger M., A parallel iterated tabu search heuristic for vehicle routing problems, *Computers and Operations Research*, **39(9)**, 2033–2205 **(2012)**