Research Journal of Recent Sciences ISSN 2277-2502
Vol. **4(3),** 34-40, March **(2015)** *Res.J.Recent Sci.*

A two Phase Approach for solving Dynamic 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 1st January 2014, revised 8th May 2014, accepted 15th October 2014

Abstract

In this paper, a two phase algorithm for solving the dynamic capacitated vehicle routing problems with soft time windows is proposed. In phase one an ant colony optimization algorithm is used to find solution for static data of problem. In second phase, an improved heuristic algorithm based on insertion heuristic is used to solve the problem in presence of dynamic arrivals of new customer orders. The proposed algorithm has been performed on the Solomon R and RC problems. The results are evaluated through a measure denoted as the value of information. Evaluating the solution by two factors (objective function and no. of vehivles) indicate that the results of the proposed algorithm are approximately equal to the solution of static problem for degree of dynamism of 10% in problem R1,R2,RC1 and appropriate for the other situation.

Keywords: Dynamic, capacitated, vehicle, routing, windows (DCVRPTW), ant colony optimization (ACO), insertion heuristic, metaheuristics, degree of dynamism.

Introduction

The traditional vehicle routing problem (VRP) consists of constructing routes for the vehicles with minimum travel time in a way that each customer is visited exactly once. In this type of problem, all information relevant to the planning of the routes is assumed to be known before the routing process begins and does not change after the routes have been constructed. In real world, this assumption is not true in all cases. For example, some customer requests are revealed over the planning horizon. So, the dynamic vehicle routing problem (DVRP) can be described as a routing problem in which information about the problem can change during the optimization process[1,2]. The dynamic vehicle routing problem considering dynamic requests or time dependent travel times has been solved by tabu search in Ichoua et al., Attanasio et al., Ichoua et al., Kergosien et al. and Nguyen et al.[3-7]. Haghani and Jung, Barkaoui and Gendreau have proposed genetic algorithm for DVRP[8,9]. Malandraki and Daskin and Fleischmann presented a heuristic algorithm for solving time-varing travel times in vehicle routing problem[10,11], Chen et al. and Donati et al. presented a multi ant colony algorithm based approach for solving time dependent vehicle routing problem[12,13]. Yang et al. and Bent and Van Hentenryck have focused to develop solution approaches for dynamic vehicle routing problem with dynamic requests[14,15].

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 correspond to customers. Each customer has a known demand ($q_i \geq 0$) and a soft 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 can start before or after the time windows. If a vehicle arrives earlier or later than customer time window, a customer can be served outside its time window, but appropriate penalties should be considered which reflect the time windows violations and customer dissatisfaction. 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. Dynamic data which are the arrivals of new customer orders reveal throughout the planning horizon. When a new customer order arrives, the system should attempt to update the route plan immediately to determine where the order should be inserted and then reoptimize the route plan. 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 and the penalties of time windows violations. 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 an approach to solve the capacitated vehicle routing problems with time windows and dynamic customer requests. The main contribution of this paper is using a new approach to solve the DVRP. This approach is implemented in two phase. In phase one an ant colony optimization algorithm is used to find solution for static data of problem. In second phase, when new customers are arriving, an improved heuristic algorithm based on insertion heuristic is used to solve the dynamic capacitated vehicle routing problem with soft time windows (DCVRPSTW) in presence of real-time data.

Methodology

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 and the elements of the ant colony algorithm adapted to the static CVRPTW. Then, we propose an improved heuristic algorithm based on insertion heuristic to solve the dynamic CVRPTW.

Ant colony optimization: The principle of ACO algorithms is based on the way ants search for food[12]. 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:

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

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 soft 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 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 means that appropriate penalties should be considered which reflect the time windows violations and customer dissatisfaction. 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 constructs 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 and the aim is to construct a set of vehicle routes for static data in order to minimize the total time traveled by minimum number of vehicles and the penalties of time windows violations. The objective function is shown below:

$$F = \sum_{k\in K}(\lambda_1 \sum_{p=1}^{m_k} t_{i_{p-1}^k i_p^k} + \lambda_2 \sum_{p=1}^{m_k-1}(t_{i_p^k} - l_{i_p^k})^+ + \lambda_3 \sum_{p=1}^{m_k-1}(e_{i_p^k} - t_{i_p^k})^+$$

Where,

$$\sum \lambda_1 + \lambda_2 + \lambda_3 = 1 \ , \lambda_1, \lambda_2, \lambda_3 \ge 0 \quad and \quad (x-y)^+ = \max(0, x-y)$$

The first part in the objective function is total traveling time, the second and third part are the penalties of time window violation which are considered when the vehicle arrives to each customers sooner or later respectively.

Set parameters, initialize pheromone trails

```
Best objective function = ∞
For each arc (i, j)
τij = τ0
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
```

Insertion Heuristic: One of the most successful sequential insertion heuristics is called I1which is proposed by Solomon[13]. A route is first constructed with a “seed” customer and the remaining unrouted customers are added into this route until it is full with respect to the scheduling horizon and/or capacity constraint.

The seed customers are chosen to be the farthest one from the depot or the unrouted customer with the lowest starting time for service or it can be selected randomly. After initializing the current route with a seed customer, the method defines two criteria $c_1(i,u,j)$ and $c_2(i,u,j)$ to select customer u for insertion between customers i and j in the currently constructed route[14].

Let $(i_0, i_1, i_2, ..., i_m)$ be the current route and i_0, i_m are depot. For each unserved customer u, the minimum insertion cost is computed as $c_1(i(u),u,j(u)) = \min[c_1(i_{p-1},u,i_p)\ p = 1,2,...,m]$, where the insertion of u between i_{p-1}, i_p is feasible.

$$c_1(i,u,j) = \alpha_1 c_{11}(i,u,j) + \alpha_2 c_{12}(i,u,j) \quad \alpha_1 + \alpha_2 = 1\ ,\ \alpha_1, \alpha_2 \ge 0$$

$$c_{11}(i,u,j) = d_{iu} + d_{uj} - \mu d_{ij} \ , \mu \ge 0$$

$$c_{12}(i,u,j) = t_{ju} - t_j$$

d_{iu}, d_{uj}, d_{ij} : distances between customers i and u, u and j and i and j respectively. μ : Parameter savings in distance t_{ju} : the new time for service to begin at customer j if u is inserted on the route. t_j : The beginning of service before insertion.

Then, the best customer u* is the one that $c_2(i(u^*), u^*, j(u^*)) = \max[\ c_2(i(u), u, j(u) \mid u$ is unrouted and route is feasible].

When no more customers with feasible insertions can be found, the method starts a new route, unless it has already routed all customers.

$$c_2(i,u,j) = \lambda d_{ou} - c_1(i,u,j), \quad \lambda \ge 0$$

d_{ou} : Distance from the depot to the unrouted customer u. λ determines the relative importance of d_{ou} and $c_1(i,u,j)$ according to the best insertion place for an unrouted customer.

Improved Insertion Heuristic: In phase one for static data which are revealed before starting route construction, an ant colony optimization algorithm is used to find solution for problem. In second phase, an improved insertion heuristic is used to insert newly arrived customer to the existing route but for this insertion we take into account the three constraints created by time window. The insertion cost included three parts: the cost of increasing travel time because of inserting new customer in route $c_1(i,u,j)$, the cost of arriving earlier or later than starting and ending time window respectively $c_2(i,u,j)$ and the similar time cost of subsequent point $c_3(i,u,j)$.

According to the insertion heuristic I1 explained above, $c_1(i,u,j) = \alpha_1(t_{iu} + t_{uj} - t_{ij})$ is the cost of increasing travel time.

The cost of violating time window of the unrouted customer is $c_2(i,u,j) = \alpha_2(e_u - t_u)^+ + \alpha_3(t_u - l_u)^+$.

Where: $(x-y)^+ = \max(0, x-y)$, $t_u = t_i + s_i + t_{iu}$, t_u: the arrival time to u, s_i: service time of u (e_u, l_u): time window of newly arrived customer. The arrival of the vehicle to node j and the subsequent points would cause delay (D) after Inserting u between i and j. $D = t_{iu} + s_u + t_{uj} - t_{ij}$

$$c_3(i,u,j) = \sum_{r \geq j} \cos t(r)$$

$$\cos t(r) = \begin{cases} -\alpha_2 D & t_r \leq e_r - D \\ -\alpha_2(e_r - t_r) & e_r - D < t_r \leq e_r \ \& \ t_r \leq l_r - D \\ 0 & e_r < t_r \leq l_r \ \& \ t_r > l_r - D \\ \alpha_3(t_r + D - l_r) & e_r < t_r \leq l_r \ \& \ t_r > l_r - D \\ \alpha_3(D) & t_r > l_r \\ -\alpha_2(e_r - t_r) + \alpha_3(t_r + D - l_r) & t_r \leq e_r \ \& t_r > l_r - D \end{cases}$$

So, the insertion cost will be: $c(i,u,j) = c_1(i,u,j) + c_2(i,u,j) + c_3(i,u,j)$ $\alpha_1 + \alpha_2 + \alpha_3 = 1$, $\alpha_1, \alpha_2, \alpha_3 \geq 0$

Improved Insertion Heuristic algorithm: Step1: Consider the K routes generated from ant colony algorithm, Step2: Each time a new customer (u) arrived, check the feasibility of inserting u in each routes in terms of capacity of route. Step3: Best insertion cost function $c(i,u,j) = \infty$. Step4: In each feasible route insert u between every two nodes of it and compute the Best insertion cost function $c(i,u,j)$. Step5: Find the best insertion point with minimum best insertion cost function $c(i,u,j)$ and place the customer between node i, j Step 6: If all of the customer arrived or vehicles were finished or depot time window was ended, go to the Step7; else go to Step 2, Step7: End

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. In order to adjust the benchmark instances with our problem, we consider three degree of dynamism: low, medium and high. In low dynamic condition (D.D=0.1), 10% of customers are randomly selected and then arrived into the problem in a time related to the earliest time of servicing of that customer and the others are entered at time t=0. For example for customer i which is selected to enter, the arrival time of customer i is calculated by the following formula:

$arrival\, time\, of\ customer\, i = e_i * r$ r is a random number with uniform distribution.

In medium and high degree of dynamism, the process is the same but 30% and 50% of customers are selected to be entered after time t=0.

In order to evaluate the performance of solution approaches on dynamic routing problems Mitrovic-Minic et al. propose a measure denoted as the value of information[19]. The value of information of solution approach on instance I is defined as:

$$V(I) = \frac{Z(I) - Z(I^o)}{Z(I)}$$

Where $Z(I)$ is the objective function of proposed approach with dynamic requests and $Z(I^o)$ is the objective function when all the data are arrived at time t=0 (static).

Solutions for each problem data for three dynamic condition are gathered then averaged for each problem type and the results are reported in the table 1 to 4. Here we apply the proposed algorithm for the problem type R1, R2, RC1 and RC2.

According to the solution generated through this proposed algorithm, we compare the average solution of each problem type in the case of dynamic arrival of request for three degree of dynamism with the solution of static problem. If all the information known in advance, $Z(I) = Z(I^o) \Rightarrow V(I) = 0$. So, the larger values of $V(I)$ indicate the lower performance of the proposed algorithm for dynamic problem and the smaller values of $V(I)$ indicate the higher performance of the proposed algorithm for dynamic problem. In the tables above, the $V(I)$ index of objective function, in the worst case is 0.29 for RC1 and D.D=0.5. It means that when 50% of customer request revealed after route construction, the proposed algorithm tried to find the best place for dynamic requests. After finding the best routes the percentage of closeness of its objective function with the objective function of the problem with static data, shows the quality of the proposed algorithm. For example, in problem R1,R2 and RC1 the solutions are approximately the same for D.D=0.1 and D.D=0 and when the degree of dynamism of problem increases, the differences of solutions are increased but these augmentation are appropriate.

Conclusion

This paper presented a two phase algorithm to solve the capacitated vehicle routing problems with time windows and dynamic customer requests. In phase one an ant colony optimization algorithm is used to find solution for static data of problem. In second phase, when new customers are arriving, an improved heuristic algorithm based on insertion heuristic is used to solve the problem in presence of real-time data.

Table-1
Solutions for R1 problem in three degree of dynamism

	D.D=0		D.D =0.1		V(I)		D.D =0.3		V(I)		D.D =0.5		V(I)	
	F	NV	F	NV	F	NV	F	NV	F	NV	F	NV	F	NV
R 101	1041	12	1201	12	0.13	0.00	1251	15	0.17	0.20	1380	14	0.25	0.14
R 102	852	11	931	13	0.08	0.15	1100	14	0.23	0.21	1113	15	0.23	0.27
R 103	715	11	750	11	0.05	0.00	739	12	0.03	0.08	838	12	0.15	0.08
R 104	580	10	595	11	0.03	0.09	594	11	0.02	0.09	673	11	0.14	0.09
R 105	719	11	711	12	-0.01	0.08	801	13	0.10	0.15	851	13	0.16	0.15
R 106	762	11	767	12	0.01	0.08	813	12	0.06	0.08	925	14	0.18	0.21
R 107	605	10	617	11	0.02	0.09	714	12	0.15	0.17	714	12	0.15	0.17
R 108	476	10	521	10	0.09	0.00	534	10	0.11	0.00	606	11	0.21	0.09
R 109	685	11	723	11	0.05	0.00	753	13	0.09	0.15	920	13	0.26	0.15
R 110	589	11	613	11	0.04	0.00	674	12	0.13	0.08	745	12	0.21	0.08
R 111	567	11	598	11	0.05	0.00	697	13	0.19	0.15	723	14	0.22	0.21
R 112	490	10	506	10	0.03	0.00	611	11	0.20	0.09	647	11	0.24	0.09
avg	673.42	10.75	711.08	11.25	0.05	0.04	773.42	12.33	0.13	0.13	844.58	12.67	0.20	0.15

Table-2
Solutions for RC1 problem in three degree of dynamism

	D.D =0		D.D =0.1		V(I)		D.D =0.3		V(I)		D.D =0.5		V(I)	
	F	NV	F	NV	F	NV	F	NV	F	NV	F	NV	F	NV
RC101	869	12	1022	12	0.15	0.00	1072	13	0.19	0.08	1301	14	0.33	0.14
RC102	739	12	788	12	0.06	0.00	940	13	0.21	0.08	982	13	0.25	0.08
RC103	612	11	656	11	0.07	0.00	734	11	0.17	0.00	810	12	0.24	0.08
RC104	585	11	621	11	0.06	0.00	643	11	0.09	0.00	678	11	0.14	0.00
RC105	766	11	853	12	0.10	0.08	1012	13	0.24	0.15	1155	15	0.34	0.27
RC106	706	11	732	12	0.04	0.08	863	13	0.18	0.15	993	14	0.29	0.21
RC107	584	11	633	11	0.08	0.00	827	13	0.29	0.15	861	13	0.32	0.15
RC108	529	11	567	11	0.07	0.00	713	12	0.26	0.08	788	13	0.33	0.15
avg	673.75	11.25	734.00	11.50	0.08	0.02	850.50	12.38	0.21	0.09	946.00	13.13	0.29	0.14

Table-3
Solutions for R2 problem in three degree of dynamism

	D.D =0		D.D =0.1		V(I)		D.D =0.3		V(I)		D.D =0.5		V(I)	
	F	NV	F	NV	F	NV	F	NV	F	NV	F	NV	F	NV
R 201	1862	3	2094	3	0.11	0.00	2030	3	0.08	0.00	2217	3	0.16	0.00
R 202	1434	3	1534	3	0.07	0.00	1371	3	-0.05	0.00	1420	4	-0.01	0.25
R 203	889	3	1048	4	0.15	0.25	1121	3	0.21	0.00	1243	5	0.28	0.40
R 204	622	2	758	2	0.18	0.00	801	2	0.22	0.00	896	2	0.31	0.00
R 205	1048	3	1242	3	0.16	0.00	1310	3	0.20	0.00	1494	3	0.30	0.00
R 206	962	3	990	3	0.03	0.00	1093	4	0.12	0.25	1110	3	0.13	0.00
R 207	722	2	723	2	0.00	0.00	738	3	0.02	0.33	807	3	0.11	0.33
R 208	420	2	439	2	0.04	0.00	443	2	0.05	0.00	474	3	0.11	0.33
R 209	986	3	1125	3	0.12	0.00	1235	3	0.20	0.00	1145	3	0.14	0.00
R 210	926	3	975	4	0.05	0.25	980	3	0.06	0.00	980	4	0.06	0.25
R 211	750	2	765	2	0.02	0.00	740	3	-0.01	0.33	691	3	-0.09	0.33
avg	965.55	2.64	1063.00	2.82	0.09	0.06	1078.36	2.91	0.10	0.09	1134.27	3.27	0.15	0.19

Table-4
Solutions for RC2 problem in three degree of dynamism

	D.D =0		D.D =0.1		V(I)		D.D =0.3		V(I)		D.D =0.5		V(I)	
	F	NV	F	NV	F	NV	F	NV	F	NV	F	NV	F	NV
RC201	1780	3	2173	3	0.18	0.00	2166	3	0.18	0.00	2206	3	0.19	0.00
RC202	1371	3	1861	3	0.26	0.00	1918	4	0.29	0.25	2013	4	0.32	0.25
RC203	990	3	1648	3	0.40	0.00	1707	3	0.42	0.00	1821	3	0.46	0.00
RC204	957	2	869	3	-0.10	0.33	878	4	-0.09	0.50	885	4	-0.08	0.50
RC205	1457	3	2128	3	0.32	0.00	2062	3	0.29	0.00	2285	4	0.36	0.25
RC206	1244	3	1485	3	0.16	0.00	1471	3	0.15	0.00	1417	3	0.12	0.00
RC207	1309	3	1287	3	-0.02	0.00	1392	3	0.06	0.00	1421	3	0.08	0.00
RC208	673	3	713	3	0.06	0.00	649	3	-0.04	0.00	877	3	0.23	0.00
avg	1222.63	2.88	1520.50	3.00	0.20	0.04	1530.38	3.25	0.20	0.12	1615.63	3.38	0.24	0.15

The proposed algorithm has been performed on the adjusted Solomon R and RC problems. The results are evaluated through a measure denoted as the *value of information*. Evaluating the solution by two factors (objective function and no. of vehivles) indicate that the results of the proposed algorithm are appropriate and in some case are great. For example the results of the proposed algorithm are approximately equal to the solution of static problem for degree of dynamism of 10% in problem R1,R2,RC1 and appropriate for the other situation.

References

1. Psaraftis H., Dynamic vehicle routing: status and prospects, *Annals of Opertations Reasearch*, **61**, 143–164 **(1995)**

2. Ghiani G., Guerriero F., Laporte G. and Musmanno R., Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies, *European Journal of Operational Research*, **151 (1)**, 1–11 **(2003)**

3. Ichoua, S., Gendreau, M., Potvin J.Y., Diversion issues in real-time vehicle dispatching, *Transportation Science,* **34 (4)**, 426–438 **(2000).**

4. Attanasio A., Cordeau J.F., Ghiani G. and Laporte G., Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem, *Parallel Computing*, **30(3)**, 377–387, **(2004).**

5. Ichoua S., Gendreau M. and Potvin J.Y., Exploiting knowledge about future demands for real-time vehicle dispatching, *Transportation Science*, **40(2)**, 211–225 **(2006)**

6. Kergosien Y., Lent Ch., Piton D. and Billaut J.C., A tabu search heuristic for the dynamic transportation of patients between care units, *European Journal of Operational Research,* **214**, 442–452 **(2011)**

7. Nguyen P.K, Crainic T.G. and Toulouse M., Atabu search for Time-dependent Multi-zone Multi-trip Vehicle Routing Problem with Time Windows, *European Journal of Operational Research,* **(2013)**

8. Haghani A. and Jung S., A dynamic vehicle routing problem with time-dependent travel times, *Computers and Operations Research*, **32(11)**, 2959 – 2986 **(2005)**

9. Barkaoui M., Gendreau M., An adaptive evolutionary approach for real-time vehicle routing and dispatching, *Computers and Operations Research*, **40,** 1766–1776 **(2013)**

10. Malandraki C. and Daskin M., Time dependent vehicle routing problems: formulations, properties and heuristic algorithm, *Transportation Science*, **26(3**), 185–200 **(1992)**

11. Fleischmann B., Gietz M. and Gnutzmann S., Time-varying travel times in vehicle routing, *Transportation Science*, **38(2)**, 160–73 **(2004)**

12. Chen B., Song S. and Chen X., Multi-Ant Colony System for Vehicle Routing Problem with Time-Dependent Travel Times, *Proceedings of the IEEE International Conference on Automation and Logistics,* 18–21 **(2007)**

13. Donati A.V., Montemanni R., Casagrande N., Rizzoli A.E. and Gambardella L. M., Time dependent vehicle routing problem with a multi ant colony system, *European Journal of Operational Research,* **185**, 1174–1191 **(2008)**

14. Yang J., Jaillet P. and Mahmassani H., Realtime multi vehicle truckload pickup and delivery problems, *Transportation Science*, **38(2)**, 135–148 **(2004)**

15. Bent R. and Van Hentenryck P., Scenariobased planning for partially dynamic vehicle routing with stochastic customers, *Operations Research*, **52(6)** 977–987 **(2004b)**

16. 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)**

17. Solomon M.M., Algorithms for the vehicle routing and scheduling problems with time window constraints,

Oper. Res, **35**, 254–265 **(1987)**

18. Solomon Benchmark Problems http://web.cba.neu.edu/_msolomon/problems.html, **(2014)**

19. Mitrovic-Minic S. and Laporte G., Waiting strategies for the dynamic pickup and delivery problem with time windows, *Transportation Research Part B: Methodological*, **38(7)**, 635–655 **(2004)**