http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (20): 5701-5706, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.5701.5706

A Hybrid Ant Colony Algorithm for Vehicle Routing Problem with Time Windows

Wenxue Ran, Li Liu and Guomin Yang
Department of Logistics, Yunnan University of Finance and Economics,
Kunming, China

Abstract: Vehicle routing problem is proved as a NP-Hard problem. VRPTW involves the routing of a set of vehicles with limited capacity from a central depot to a set of geographically dispersed customers with known demands and pre-defined time windows. In this study, a new hybrid algorithm based on Ant Colony Optimization (ACO) and Genetic Algorithm (GA) is provided to solve the vehicle routing problem with time windows. The proposed algorithm combines the advantages of ACO and GA. Then a mathematical programming formulation for the problem is given. The performance of the model and the heuristic approaches are evaluated using instances generated from a set of classic examples. Results of numerical experiments show that the proposed algorithm is an effective tool in solving VRPTW problem.

Key words: Vehicle Routing Problem (VRP), time window, hybrid ant colony algorithm

INTRODUCTION

Vehicle Routing Problem, which is firstly put forward by Dantzig and Ramser (1959), is one of the hot-topics and difficult problems in logistics distribution. The VRP can be simply described as the goal of finding out the most optimal distribution route under certain constraints (such as, the number of vehicles, the maximum load, the quantity demand, time windows) to meet customer demand, to realize the goal of the shortest route, the lowest cost, the least time-consuming and the least number of vehicles and the like. Essentially speaking, VRP is an optimization problem, as well as a decision problem. One of the most common extensions of VRP is Vehicle Routing Problem with Time Windows (VRPTW) in practice. In VRPTW, vehicles service all the customers during given time interval without violating capacity constraints (Lang and Hu, 2010). The objective of the problem is to find a set of minimum cost routes for vehicles from a central supply depot. In real life many problems can be concluded as VRPTW, such as postal deliveries, running of train and bus, goods delivery from e-business and the like.

Rochat and Taillard (1995) proposed the adaptive memory, which turned out to be very effective for Tabu Search applications in the VRP. Tas *et al*. (2013) studies a vehicle routing problem with soft time windows and stochastic travel times. They developed a model that considers both transportation costs and service costs and used a Tabu search method to solve this model. Vidal *et al*. (2013) studied a large class of time-constrained vehicle routing problems, accounting for penalized in feasible solutions with respect to time-window and duration constraints.

As VRPTW belongs to NP-Hard problem, currently almost all the literatures focus on constructing heuristic algorithm, such as Heuristic and meta-heuristic approaches (Braysy and Gendreau, 2005; Gendreau and Tarantilis, 2010; Gehring and Homberger, 1999) relative to their computational efficiently and the quality of the solutions obtained. Nguyen *et al*. (2011) develop a hybrid genetic approach.

Baldacci *et al*. (2011) reported the optimal/best-known solutions provided for the well-known VRPTW instances with deterministic parameters.

This study presents a new hybrid ant colony algorithm, that is, combines with genetic algorithm and ant colony algorithm. First, genetic algorithm derived initial solution, then ant colony algorithm to calculate its exact solution, which combines the advantages of each algorithm and uses it to solve VRPTW and gets the prominent efficiency in the verify cases.

DESCRIPTION OF VRPTW ANDMATHEMATICAL MODEL

Based on VRP, the constraint of time windows is introduced into VRPTW, that is to say, the goods should be delivered within the time windows, otherwise the customers will reject the goods or ask the company for

Corresponding Author: Wenxue Ran, Department of Logistics, Yunnan University of Finance and Economics, Kunming, China

early or late penalty. Therefore, hard time windows and soft time windows for VRPTW are proposed. The hard time windows VRPTW means each distribution service must be completed within the required time, the one which doesn't satisfy the time windows will be the infeasible solution. The soft time windows s VRPTW refers to a task which can't be completed within the required time and thus will receive a certain penalty.

The VRPTW of the distribution center can be generally described as follows: Supposing that in a distribution center there are m vehicles, which provide distribution service for n customers and each customer's requirement is g_I (i = 1, 2, Y, N), max $g_i \leq \max q_k$, the load capacity for per car is ω_I (k = 1, 2, .., M). The time spent on completing the distribution service of customer i (loading or unloading) is t_i and distribution service i must be completed during the time windows [t_{ei}, t_{li}], t_{ei} is the earliest starting time which allowed by task i, t_{li} is the latest finishing time which allowed by task i. If the vehicle arrives i before t_{ei}, the vehicle must wait, which increases the time cost. If the vehicle arrives i after t_{li}, the delivery is delayed and company must pay a penalty. Supposing the amount of distribution center is W, t_{ij} represents the time from customer i to j, which including the waiting time. C_{ij} denotes the transportation cost from customer i to j. C_0 is the opportunity cost caused by waiting. C_p is the penalty cost which caused by delaying. Therefore the mathematical model of VRPTW can be described as follows:

$$\min Z = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{M} C_{ij}x_{ijk} + C_0\sum_{i=1}^{n}\max(t_{e_i} - t_i, 0) + C_p\sum_{i=1}^{n}\max(t_i - t_{l_i}, 0) \quad (1)$$

s.t.

$$\sum_{i=1}^{n}\omega_i y_{ik} \leq W \qquad k \in [1,m] \quad (2)$$

$$\sum_{k=1}^{n} y_{ik} = 1 \qquad i \in [1,n] \quad (3)$$

$$\sum_{i=1}^{n} x_{ijk} = y_{ik} \qquad j \in [1,\cdots,n], \forall k \quad (4)$$

$$\sum_{j=1}^{n} x_{ijk} = y_{ik} \qquad i \in [1,\cdots,n], \forall k \quad (5)$$

$$t_{ei} \leq t_i \leq t_{li} \quad (6)$$

And

$$x_{ijk} = \begin{cases} 1 & \text{vehicle k from customer i to customer j} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$y_{ik} = \begin{cases} 1\text{vehicle k from customer i to customer j} \\ \text{otherwise} \\ 0 \end{cases} \quad (8)$$

$$t_i = \sum_{i=0}^{n}\sum_{k=0}^{m} x_{ijk}(t_i + t_{ij} + t_j) \qquad j = 1, \cdots, n \quad (9)$$

where, Eq. 1 is the objective function. Equation 2-9 are the specific constraints in which Eq. 2 is the constraint of the vehicle's rated load. Equation 3 represents that the delivery service is specified by clients. Equation 4 and 5 limit that the same vehicle repeatedly pass by the same node, are the constraints of the same vehicle repeatedly pass through the same client, on that condition the route selected must be a loop. Equation 6 is the time windows constraint. Equation 7 and 8 represent whether the transportation service will be implemented; formula-9 represents the time when the goods are delivered to customer i.

HACO FOR VRPTW

This study constructs a new type of HACO, which is a combination of improved ant colony algorithm and improved genetic algorithm. From the following benchmark instances we can see that the HACO takes the full advantage of ACO and GA, finally obtains the best solution.

Improved GA: The solution obtained by GA is the main source of initial pheromone of ant colony algorithm. The efficiency of ACO depends partly on the accuracy of solution obtained by GA. We can improve the accuracy of solution by the following steps:

- **Genetic algorithm encoding:** The client node is numbered sequentially from1 to N. The number of distribution center is denoted by 0 and there are N client nodes, 7 Number 0 is used as a separator of each route and inserted into the chromosome which satisfies the criteria. Given that there are five client nodes, that is G = 1 2 3 4 5, when these five customers can be visited by only one vehicle which costs the least, then G = 0 1 2 3 0 4 5 0. The distribution route can be shown as follows:
  - **Route 1:** DC→client 1→client 2→client 3→DC
  - **Route 2:** DC→client 4→client 5→DC

- **Population initialization:** Randomly generate N nodes which denote the customer number. According to the constraints of vehicle capacity and demand, the number 0 is inserted into the chromosome. Perform the judgment when searching until all of the nodes are visited. Thus a new chromosome is constructed. Repeat the above steps until the populations are satisfied
- **Determine the fitness function:** When the population size is determined, a proper fitness function must be selected to ensure the GA algorithm evaluates in a better direction. The greater the fitness function value of an individual is, the better the performance is. The objective function value of each chromosome, which is denoted as X_g, can be obtained by executing Eq. 4. If the solution is infeasible, the objective function value can be set to a very large number, which is abandoned in the process of constructing solution. Thus the feasible solution will be find out quickly. We can get the fitness value of each chromosome by implementing the following equation:

$$f_g = \frac{1}{X_g} \tag{10}$$

The above equation shows that the smaller the X_g is, the larger the f_g is, thus the better the performance of chromosome is and the corresponding solution is much closer to optimum.

- **Selection operation of GA:** The selection operation adopted in this study is steady replication, which replaces the worst parent solution with offspring solution in each iteration, leaving the optimal solution in the next generation
- **Crossover and recombination of chromosome:** The crossover operation in GA algorithm can not only maintain the excellent performance of the parent generation, prevent stagnation and prematurity, but also increase the diversity of population and searching space. The crossover method which operates in sequence is adopted in this study. The final solution is compared with the parent solution, leaving the optimal solution. Specific steps are as follows:

- Assuming that there are two sets of genetic codes, T_1 and T_2. The initial position and crossover length of the two sets are randomly generated during the operation of algorithm:

$$T_1{:}|R_1|R_2|R_3 \tag{11}$$

$$T_2{:}\ S_1|S_2|S_3 \tag{12}$$

- Find out the genes which need to crossover in T_1 and T_2 and operate R_2 and S_2. Insert S_2, which is in front of R_2, intoT_1. Insert R_2, which is in front of S_2, into T_2. Thus, two sets of genetic code T_3 and T_4 are generated. That is:

$$T_3{:}|R_1|S_2|R_2|R_3 \tag{13}$$

$$T_4{:}\ S_1|R_2|S_2|S_3 \tag{14}$$

- Delete the other genes of the same except for 0 in R_1, R_2, R_3 and S_2 (the genes are good ones at the either end. When operating recombination, these genes cannot be destroyed). Delete the other genes of the same except for 0 in S_1, S_2, S_3 and R_2. Thus T_5 and T_6 are generated
- Compare the results of T_1, T_2, T_5, T_6 and the best two sets of code are selected to keep in the chromosome. That is:

T_1: 1 0|2 3|4 0 5 6 7
T_2: 7 0|6 5|0 4 3 2 1
T_3: 6 5|1 0 2 3 4 0 5 6 7
T_4: 2 3|7 0 1 6 5 4 3 2 1
T_5: 6 5 1 0 2 3 4 0 7
T_6: 2 3 7 0 1 6 5 4 1

- **Mutation of chromosome:** In GA algorithm, mutation, which is randomly generated, is an important means of enhancing chromosome diversity. The genes of non-protective are randomly selected to perform exchange operation. Results are compared before and after of chromosomes swapping. Thus the excellent chromosomes are selected into the next generation

Improved ACO: On the condition that the initial pheromone is asymmetrical, in order to avoid getting into local optimum, Max-Min Ant algorithm is adopted to improve the accuracy and efficiency of the method:

- **Determination of the objective function:** The objective function for VRPTW is also the ACO's. As shown in type-1
- **Transition rules:** The running of ACO is a process of continuously "exploring", but it is because of such mechanism that limits the convergence speed of the algorithm. To solve this problem, q_0 0 [0, 1) is introduced in this study. Before selecting route, q is

randomly generated, then ant m select the next route according to the following equation:

$$j=\begin{cases}(16) & q\ge q_0\\ \arg\max_{j\in m_{allowed}}\left\{\left[\tau_{ij}(t)\right]^{\alpha}\times\left[\delta_{ij}(t)\right]^{\beta}\right\} & q<q_0\end{cases}\tag{15}$$

When $q\ge q_0$, exploratory search of basic ACO is the transition rule of ant m. When $q<q_0$, ant m selects the route of higher probability as its route. Then deterministic search continues on this route.

Selection mode of q_0:

- Firstly, the value of q_0 is large from the beginning, that is, the probability of a deterministic search is big. Thus searching for local optimal solutions speed up
- Decrease the value of q_0 at run time until it becomes a very small number. The probability of exploratory searching of ant m increases. Thus the population space of searching for optimal solution expands
- While the algorithm runs, q_0 returns to pre-set value, which accelerates the convergence speed:

$$P_{ij}^{m}(t)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\times\left[\delta_{ij}(t)\right]^{\beta}}{\sum\left[\tau_{ij}(t)\right]^{\alpha}\times\left[\delta_{is}(t)\right]^{\beta}}\text{if } j\in m_{allowed}\\ 0\end{cases}\tag{16}$$

where, $m_{allowed}$ is a collection of next distribution client allowed to select in the optimization process of ant m. α is the information heuristic parameter which indicates the importance for remaining pheromone. β is the expected inspired parameter which denotes the importance of pheromone in selecting the route. The bigger the value of β is, the closer the state transition probability of ant m will get to greedy rules. $\boldsymbol{\delta}_{ij}$ (t) is heuristic function, which shows the degree of expectation from client i to client j of ant m. For , d_{ij} is the distance between two adjacent clients. For ant m, the smaller the d_{ij} is, the bigger the $\boldsymbol{\delta}_{ij}$ (t) is, therefore the bigger the $P_{ij}^{m}(t)$ is

- **Selection of ρ:** ρ is a constant in ACO and the size of ñ has a direct effect on searching for global solution and convergence rate. Thus parameter ñ will be adjusted as follows in HACO:

$$p(t+1)=\begin{cases}\max\left[\mu\times p(t),p_{min}\right], r=r_{max}\\ p(t),\text{otherwise}\end{cases}\tag{17}$$

where, $\mu\in(0, 1)$ is pheromone polatility which is a constant. In order to avoid the phenomenon of reducing the speed of convergence because the value of ρ is small, ρ_{min} has been preset, which ensure the obtained solution is a better one. r means the times that evolutionary cycle does not continuously happen. r_{max} is a constant. When r reaches pre-set value r_{max}, value of ñ decreases. Then r recounts, repeat this operation until ρ reached the minimum value ρ_{min}

- **Settings of τ_0:** During the process of search ,the range of possible pheromone trails on each route component is limited to a maximum minimum interval $[\tau_{min},\tau_{max}]$.If beyond this range, the pheromones in the route will be forcefully set as τ_{min} or τ_{max}. Assuming that $\tau_0\in[\tau_{min}, \tau_{max}]$, this study benchmarks elitist strategy to pre-set 'Elitist ants'. The number of 'Elitist ants' is used to dynamically adjust τ_{min} and τ_{max}

Given that L_s denotes the path length of global optimal solution, so the increased maximum information amount on the route is $1/L_s$ after each iteration. Values of τ_{min} and τ_{max} must be updated when updating the optimal solution. It is because that τ_{max} has a negative relationship with parameters ñ and L_s and a negative relationship with the number of elitist ants τ. The way to determine τ_{min} and τ_{max} is as follows.:

- After running Genetic Algorithm (GA) and Ant Colony Algorithm (ACO) does not began to run, the value of τ_{min} and τ_{max} are determined by the following equation:

$$\tau_{max}=\frac{1}{2(1-\rho)}\times\frac{1}{L_s}\tag{18}$$

$$\tau_{min}=\frac{\tau_{max}}{20}\tag{19}$$

- After updating the pheromone, the value of τ_{max} is determined by the following equation:

$$\tau_{max}=\frac{1}{2(1-\rho)}\times\frac{1}{L_s}+\frac{\sigma}{L_s}\tag{20}$$

Calculation of τ_{min} is the same as above

- **Pheromone updating:** The update mode in Max-Min Ant algorithm is global update, besides, the pheromone of best ant can only be updated. The update rules can be described as follows:

$$\tau_0^{new}=(1-\rho)\tau_0^{old}+\sum_{\mu=1}^{\sigma-1}\Delta\sigma_0^{\mu}+\sigma\Delta\tau_0^{*}\tag{21}$$

$$\tau_0^{new} = \tau_0^{new} + \tau_0^{new} \times \frac{L_{new} - L_{old}}{L_{old}} \quad (22)$$

When route (v_i, v_j) is ranked as the best choice of ant μ, the increment of pheromone is $\Delta\tau_0^i$, that is $\Delta\tau_0^i = (\tau\text{-}\mu/L_i)$, where L_i is the path length of ant μ. All of the pheromone concentration on the current route has been increased by $\tau\Delta\tau_0^*$, that is $\tau\Delta\tau_0^* = 1/L^*$,where L^* denotes the current best path length

Elitist ants updates pheromone by using Eq. 19. In the process of iteration, the offspring solution which is better than parent solution will be motivated, while the worse solution in offspring will be punished.

Operation of HACO:

- When running ACO algorithm, the initial pheromone should be set based on current optimal solution which is searched by GA algorithm. The GA algorithm will firstly calculate out a current optimal solution and the corresponding path length, denoted as L^*. Then L^* is used to replace the maximum and minimum values obtained by L_S using Eq. 18 and 19:

$$\tau_0 = \tau_{min}+c \quad (23)$$

  c is a constant, which is set as 30% of τ_{min}
- The initial population of ACO algorithm will select some optimal solutions from GA algorithm according to special proportion. The pheromone concentration generated by the selected solutions can be set as τ_{max}

SIMULATION OF HACO

Given that the distribution center will assign 5 vehicles to visit 20 customers and each vehicle would have a maximum load average of 8 tons and maximum mileage of 50 kilometers. The vehicle starts from distribution center at 8:00 AM and there are 10 minutes to unload. The time windows of the client is [9:00-12:00], the coordinate of the distribution center is (14.5, 513.0 km). This example is depicted in Table 1.

In GA algorithm, L (0) is the initial population, Gen is the maximum iterations, P_c is the crossover probability, P_m is the mutation probability. Here, L (0) = 50, Gen = 100, P_c = 0.8 and P_m = 0.006. In addition, the steady reproduction ratio is 5% and there are two variants.

In ACO algorithm, Q is the capacity of a vehicle, N_{cmax} is the maximum number of iterations, m is the number of ants, σ is the number of elite ants, M is the number of vehicles and N is the number of customers. Here, Q =, N_{cmax} = 200, m = 12, σ = 4, M = 5, N = 20, α = 1.0, β = 2.0, ρ_{min} = 0.2.

We use Matlab 11.0 as the programming language. All the meta-heuristic approaches are tested on a PC with Dual-Core 3.2G CPU and RAM 4 GB under Microsoft Windows XP.

Table 2 and 3 respectively shows the simulation results of the HACO and literature [9], respectively.

From Table 2 and 3 we can see that the shortest distance, obtained by HACO, is 107.8km,the shortest path length is 110.3 km, the average number of vehicles is 3.6 and the average evolutional generation for the first time to search the final solution is 42,while in literature[9],the corresponding value, obtained by improved GA, is 108.6, 112.5, 3.7, 71 km, respectively.

Table 1: Coordinates and demand of 20 clients

ID	1	2	3	4	5	6	7	8	9	10
X axis (km)	12.8	18.4	15.4	18.9	15.5	3.9	10.6	8.6	12.5	13.8
Y axis (km)	8.5	3.4	16.6	15.2	11.6	10.6	7.6	8.4	2.1	5.2
Demand (t)	0.1	0.4	1.2	1.5	0.8	0.8	1.7	0.6	1.2	0.4
Earliest service time	9:00	9:00	9:00	9:00	9:00	9:00	9:00	9:00	9:00	9:00
Latest service time	12:00	12:00	12:00	12:00	12:00	12:00	12:00	12:00	12:00	12:00
ID	11	12	13	14	15	16	17	18	19	20
X axis (km)	6.7	14.8	1.8	17.1	7.4	0.2	11.9	13.2	6.4	9.6
X axis (km)	16.9	2.6	8.7	11	1.0	2.8	19.8	15.1	5.6	14.8
Demand(t)	0.9	1.3	1.3	1.9	1.7	1.1	1.5	1.6	1.7	1.5
Earliest service time	9:00	9:00	9:00	9:00	9:00	9:00	9:00	9:00	9:00	9:00
Latest service time	12:00	12:00	12:00	12:00	12:00	12:00	12:00	12:00	12:00	12:00

Table 2: Computational results of HACO (10 iterations)

Computation sequence	1	2	3	4	5	6	7	8	9	10	Average value
Minimum distance	113.0	109.6	110.2	111.7	110.4	111.2	109.1	109.6	107.8	10.4	110.3
Number of vehicleapp:	4	3	3	4	3	4	4	3	4	4	3.6
Addword: No. of vehicle Iterations	76	45	15	64	51	64	27	23	46	4	42

Table 3: Computational results of literature [7] (10 iterations)

Computation sequence	1	2	3	4	5	6	7	8	9	10	Average value
Minimum distance	114.7	114.4	112.7	113.0	112.2	115.4	109.6	114.4	108.6	109.6	112.5
No. of vehicleapp: Addword:	4	4	3	3	3	5	4	4	4	3	3.7
No. of vehicle Iterations	51	48	79	87	106	93	36	65	113	33	71

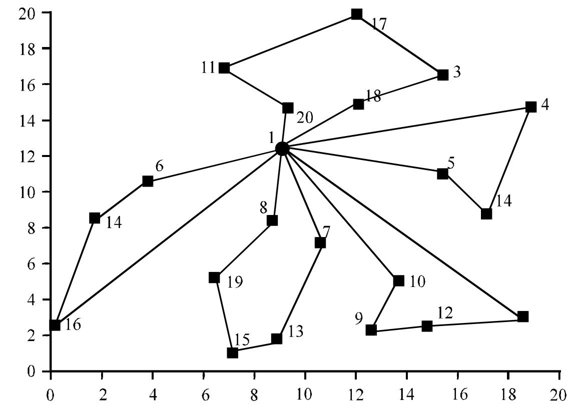

Fig. 1: Results of HACO

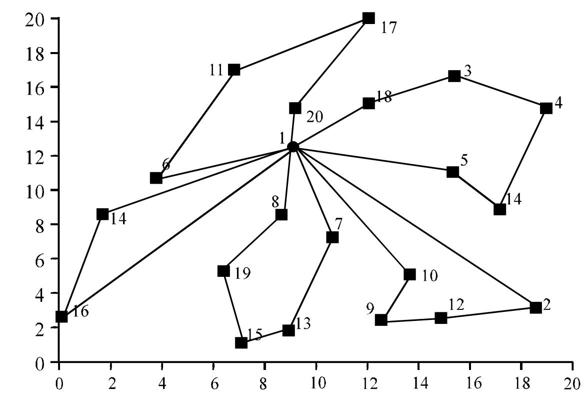

Fig. 2: Results of improved GA

From the above results we can see that, compared with improved GA published in literature[9], the HACO algorithm proposed by this study is more effective.

The following two graphs are the route choices of the HACO algorithm and improved GA algorithm. Figure 1 shows the optimal solution of HACO algorithm. The route selection results of improved GA algorithm is depicted in Fig. 2.

CONCLUSION

In this study, we have proposed a new hybrid approximate solution algorithm for VRPTW by combining ant colony algorithm and genetic algorithm. The numerical experiments for the instances clearly verify that the proposed method is an effective tool to solve the vehicle routing problem with time windows.

REFERENCES

Baldacci, R., A. Mingozzi and R. Roberti, 2011. New route relaxation and pricing strategies for the vehicle routing problem. Oper. Res., 59: 1269-1283.

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.

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

Gehring, H. and J. Homberger, 1999. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. Proceedings of EUROGEN99-Short Course on Evolutionary Algorithms in Engineering and Computer Science, Reports of the Department of Mathematical Information Technology, Series A. Collections, No. A 2/1999, (EUROGEN'99), University of Jyvaskyla, Finland, pp: 57-64.

Gendreau, M. and C.D. Tarantilis, 2010. Solving large-scale vehicle routing problems with time windows: The state-of-the-art. Technical Report CIRRELT-2010-04, February 2007, CIRRELT, Montreal, Canada.

Lang, M.L. and S.J. Hu, 2010. Study on the optimization of physical distribution routing problem by using hybrid genetic algorithm. Chin. J. Manage. Sci., 10: 51-56.

Nguyen, P.K., T.G. Crainic and M. Toulouse, 2011. A hybrid genetic algorithm for the periodic vehicle routing problem with time windows. Technical Report CIRRELT-2011-25, April 2011, CIRRELT, Montreal, Canada.

Rochat, Y. and E.D. Taillard, 1995. Probabilistic diversification and intensification in local search for vehicle routing. J. Heurist., 1: 147-167.

Tas, D., N. Dellaert, T. van Woensel and T. de Kok, 2013. Vehicle routing problem with stochastic travel times including soft time windows and service costs. Comput. Oper. Res., 40: 214-224.

Vidal, T., T.G. Crainic, M. Gendreau and C. Prins, 2013. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Comput. Oper. Res., 40: 475-489.