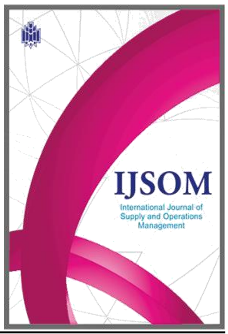
International Journal of Supply and Operations Management

IJSOM

May 2018, Volume 5, Issue 2, pp. 134-151
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com



Solving a Dynamic Vehicle Routing Problem with Soft Time Windows Based on Static Problem Resolution by a Hybrid Approach

Bouchra Bouziyane [a,*], Btissam Dkhissi [a] and Mohammad Cherkaoui [a]

[a] *National school of applied sciences, Abdelmalek Essaadi University, Tetuan, Morocco*

Abstract
More and more companies in routing industry are interested in dynamic transportation problems that can be found in several real-life scenarios. In this paper, we addressed a dynamic vehicle routing problem with soft time windows (D-VRPSTW) in which new requests appear at any point during the vehicle's route. We presented a mathematical formulation of the problem as well as a genetic algorithm hybridized with a variable neighborhood search (VNS) metaheuristic designed for the considered problem. Then, using the time discretization in intervals with new features, we focused on the proposed solution method to solve each partial static problem. We extended the dynamic vehicle routing problem (D-VRPSTW) by considering several objective functions, i.e. minimizing the transportation time by producing better planning, improving the quality of service by minimizing the delay time for each customer, and minimizing time loss by increasing the stopping time for each vehicle. The solution quality of this method has been compared against the existing results on benchmark problems.

Keywords: Optimization; Dynamic Vehicle Routing Problem (DVRP); Hybridization; Genetic Algorithm; Variable Neighborhood Search (VNS).

1. Introduction

Vehicle routing optimization has beneficial effects on the companies operating on different sectors of activity, such as the distribution of the highest quality products. Tirkolaee et al. (2017) studied a special case of perishable products. They formulated a robust multi-trip vehicle routing problem with intermediate depots and time windows, and presented a mixed-integer linear programming model. Another important application of vehicle routing other than the distribution of perishable products is the urban waste collection, which belongs to the set of arc routing applications. Tirkolaee et al. (2018a) have recently developed a mixed-integer linear programming model and presented a hybrid algorithm to solve the multi-trip capacitated arc routing problem. The same subject is addressed by extending the planning period to many days and considering drivers and crew's working time (Tirkolaee et al., 2018b).

The current literature available on vehicle routing problems is mainly concerned with static problems, where all data are known in advance. The increasing awareness of just-in-time supply systems with apparition of new advances in communication and information technologies have recently led the researchers to focus on dynamic vehicle routing problem. In DVRP (also referred to as online or real-time VRP), the inputs are revealed or updated continuously (such as new customer requests arriving at any point during the vehicle's route), and are received as time progresses and must be dynamically incorporated into an evolving schedule. Kilby et al. (1998) who believed that their work can be seen as the starting point in the field of dynamic problems analyzed many aspects related to dynamic vehicle routing problems, like business model choices and measures of dynamicity of a problem. They considered a DVRP, where dynamism arises from newly arriving requests. As pointed recently by Pillac et al. (2013) in their comprehensive review of the DVRP,

Corresponding author email address: bouziyaneensa@gmail.com

the technological development has encouraged the interest in dynamic vehicle routing problems, but the critical aspect of dynamic vehicle routing is due to incomplete and unknown information during the planning horizon.

By referring to practical logistics problems, uncertainty in demand is most prevalent in the DVRPTW and DVRP (e.g. Moretti et al.,2009 ; Novoa et al., 2009) ; Wen et al., 2010 ; Cao et al., 2010 ; Lei et al., 2011; Moghaddam et al., 2012; Hong, 2012; Hu et al., 2013). Uncertainty in travel times is also present. For instance, Taniguchi and Shimamoto (2004) have proposed a genetic algorithm (GA) to solve a DVRPTW with dynamic travel times. The problem is solved as a static VRPTW at each time slot, and its resolution is based on a traffic simulator that provides updated travel time information.

Developing an algorithm that can deal with dynamicity and time windows is very useful for delivery companies involved in the procedure of daily routing. Several works have addressed this practical problem when new demands appear during operation time and each customer must be served within a predefined time window. For example, Yang et al. (2017) proposed a multiple ant colony algorithm combined with powerful local search procedures, and Chen et al. (2017) proposed a hybrid heuristic algorithm that combines the harmony search (HS) algorithm and the variable neighborhood descent (VND) algorithm to solve the dynamic vehicle routing problem with time windows (DVRPTW).

The subject of VRPTW is very interesting because it describes several other real-world problems, such as school bus routing, postal deliveries, bank deliveries, security patrol service, and industrial refuse collection. So, the modelisation and the resolution of these problems have improved the supply chain for several institutions in terms of delays, money expenditures, and staff satisfaction.

Most researchers dealt with problems with hard time windows. In practice, time windows tend to be soft, so these time windows are often relaxed to allow for early or late arrivals at customer locations. Qureshi et al. (2012) presented a microsimulation-based evaluation of an exact solution approach to deal with the problem of the dynamic vehicle routing problem with soft time windows, such as the exact solution approach based on column generation, and travel time data obtained using a microsimulation software, VISSIM. The routes are re-optimized whenever a dynamic traffic event occurs.

Other authors have chosen to treat VRP with respect to the environmental aspect. The green VRP was introduced in Mirmohammadi et al.'s (2017) study, where some specifications like the periodic VRP, the time-dependent urban traffic, and time windows were highlighted. The objective function is to minimize the total amount of carbon dioxide emissions, earliness and lateness penalties costs, and costs of used vehicles. Cimen et al. (2017) proposed an approximate dynamic programming-based heuristic to address a time- dependent capacitated vehicle routing problem with stochastic vehicle speeds and environmental concerns. Later, a new multi-objective vehicle routing problem under a stochastic uncertainty was presented by Rabbani et al. (2018) who solved by a meta-heuristic algorithm, namely simulated annealing to minimize the total transportation cost, traffic pollution, and customer dissatisfaction, and maximize the reliability of vehicles.

Qureshi (2014) pointed out that metaheuristics, such as tabu search (TS) and the genetic algorithms (GAs) have been successfully used for the static VRPTW. On the other hand, adapting metaheuristics for the dynamic version of this problem poses some problems because of high compilation time. But in many cases, such as in the case of a parallel implementation of an adaptive tabu search proposed by Gendreau et al. (1999), the implementation of metaheuristics gives good results and yields a much better optimization compared to local optimization-based heuristics. They treated dynamic vehicle routing problem with soft time windows whose resolution is based on tabu search heuristic, designed for the static version of the problem, then adapted to the dynamic case to be solved as a series of static problems and finally implemented on a parallel platform to increase computational effort.

Hybridization is a trend in many studies carried out in metaheuristics over the past decade. It leverages the cumulative benefits of different metaheuristics. Subsequently, we addressed several studies in detail, where the resolution is based on hybridization of metaheuristics and a local search method. Khouadjia et al. (2012) examined dynamic vehicle routing problem DVRP with dynamic requests proposed two approaches based on particle swarm optimization (PSO) and variable neighborhood search (VNS), and introduced a new set of benchmarks for evaluating the proposed methods as well as existing benchmarks in the literature. Their resolution consists of partitioning the working day into time slices, then applying the proposed methods. The research object of Qiuyun et al. (2013) is the dynamic vehicle routing problem with time windows for distribution goods, which take into consideration a random demand and dynamic network. The problem has many objectives, i.e. maximizing the number of customers, and minimizing customer waiting time and total vehicle driving distance which is treated as a multiobjective optimization problem. The resolution is based on dynamic hill-climbing local search operator and genetic hybrid algorithm, while a standard test data from Solomon are used for simulation experiment. Barkaoui et al. (2013) presented a new hybrid genetic approach (HGA-LCS) to address the DVRPSTW. The basic scheme consists of the two concurrently evolving populations of solutions to minimize customer service denial, lateness or temporal constraint violation, and total distance travelled. Combining variations of key concepts inspired by routing techniques and search strategies to define new hybrid genetic operators, the proposed approach also exploits the least commitment routing policy to serve the scheduled customers to potentially improve solution quality. Euchi et al. (2015) also proposed a hybrid metaheuristic approach for the dynamic vehicle routing problem based on 2-opt local search, called artificial ant colony hybridized with a 2-opt local search. To solve the dynamic delivery and pickup problem, they applied the artificial ant colony for each static problem, obtained by the division of the customers into multiple equal time slices. Another hybrid method for solving VRPTW was presented by Yassen et al. (2017) in which the harmony search algorithm is hybridized with five local search algorithms, namely hill climbing, simulated annealing, record to record, reactive tabu search, and great deluge. So, the most difficult task for them is how to design an adaptive selection mechanism to adaptively select a suitable local search algorithm.

Genetic algorithms represent a powerful tool for different optimization problems. Using the genetic algorithm and simulated annealing algorithm, Rabbani et al. (2015) treated a multi-objective problem to find an efficient compromise solution, which consists of delivering perishable products to customers with minimum cost by considering different soft time windows for each customer. Mirabi et al. (2016) highlighted the vehicle routing problem with soft time window by considering several depots and flexible end depot in each route. They aimed to minimize travelled distance as well as the latest and earliest arrival time penalties. Concerning the method of resolution, the customers are divided into groups by the clustering method, and then a genetic algorithm is applied to find the best route. Genetic algorithms are applied for the resolution of problems other than vehicle routing problems, such as an efficient genetic algorithm to solve the intermodal terminal location problem proposed by Oudani et al. (2014). In other studies, the genetic algorithm was hybridized with local search methods. Larioui et al. (2015) used a memetic algorithm to solve the vehicle routing problem by minimizing the total travelled distance with respect of time window constraints of suppliers and customers. The problem is to transport products from the suppliers to customers via a cross-dock. Mańdziuk et al. (2016) presented a memetic approach for solving vehicle routing problem with dynamic requests. The process consists of receiving new orders, dividing the working day into equal-length time slices, and solving a static version in each time slice by the implementation of genetic algorithm enhanced with the local memetic optimization. The problems of dynamic vehicle routing are solved by other methods, such as article swarm optimization used in the work of Okulewicz and Mańdziuk (2017), and ant colony system (Kuo et al., 2016) considered the uncertain service time.

The effectiveness of local search method has been proven, so some authors are based solely on these methods to solve the dynamic vehicle routing problem. The problem studied in the work of Armas and Melián-Batista (2015) included several real constraints, such as heterogeneous fleet of vehicles, multiple and soft time windows, and customers priorities. This leads to the definition of a dynamic rich vehicle routing problem with time windows. Therefore, they proposed a metaheuristic procedure based on variable neighborhood search to solve this particular problem. Another example of this type of method is a metaheuristic procedure based on an adaptive large neighborhood search proposed by Chen et al. (2018) to solve the dynamic vehicle routing problem (DVRP) with limited vehicles and hard-time windows.

In this paper, we looked at a practically important variant of vehicle routing problems (dynamic vehicle routing problem with soft time windows (D-VRPSTW)), and we aimed to build an approach that can at the same time take into consideration the dynamicity of the problem and the time windows of customers.

The VRP and its variants are classified as NP-hard problems. Hence, the use of exact optimization methods does not appear experimentally efficient to solve large instances of the problems. So, we will use heuristics and meta-heuristics for finding good solutions, also, we will combine some of these methods, which could be termed hybrid methods.

To solve this problem, we proposed the discretization method based on the hybrid algorithm, which combines a genetic algorithm with the variable neighborhood search (VNS), proposed to solve the static part of the problem. We aimed to minimize the tradeoff between the sum of the delays, the stopping time, and the total transportation cost. This technique can potentially yield near-optimal solutions to many difficult optimization problems.

This is a multi-criteria vehicle routing problem, which is modeled as a single criterion problem by using weights that reflect the importance of each criterion. All problem parameters, such as demand locations and time windows are assumed to be known with certainty.

This paper first aimed to present a hybrid optimization method that minimizes the expressed objectives, then second to focus on the dynamic part of this problem. The paper is then structured as follows. Section 2 presents a mathematical model for the dynamic vehicle routing problem with soft time windows. Steps to resolve the dynamic problem by the hybrid method are presented in section 3, such as requests collected during each time slice and served at the end of this time slot. In the second part of this section, the heuristic proposed to serve dynamic customers at each time t after we receive their requests is presented. The experimental results are reported in Section 4. Lastly, conclusion and future work are presented in Section 5.

2. Problem statement and formulation

In this section, we present a mathematical formulation for the D-VRPSTW. The objective is to minimize transportation time as well as the latest and the earliest arrival time. We use scalarization for solving this multiobjective problem, and express the various objectives in a single unit, then the problem will be solved as a single objective problem using weighted sum objective function.

We must respond to requests from *n* customers *(i,j*=1,2,..*n)*, some of whom are available at the beginning, and others appeared during the planning horizon using *k (k*=1,2,..*K)* vehicles starting from a single depot and returning to it at the end (0 denotes the depot).

Since the considered problem is dynamic, the complete scheduling horizon is divided into various time slots and the mathematical model describes a static partial problem corresponding to each time slice.

Parameters and sets used in modeling this problem are listed below:

Parameters:

i,j	Index of customers
k	Index of vehicles
Q	Vehicle capacity
q_i	Demand of i^{th} customer
s_i	Service time
$[e_i, l_i]$	Service time window of customer i
d_{ij}	Distance between customers *i* and *j*
t_{ij}	Travel time between customers *i* and *j*
α_i	Weighting coefficients
T	Length of working day
A_i	Arrival time at customer *i*
D_i	Departure time from customer *i*
y_{ik}	Goods quantity in the vehicle k visiting customer i
0_{lk}	Last customer served by the vehicles *k* at time slice *l*
Ts	Beginning of the planning horizon

Sets:

N	Set of network nodes
K	Set of vehicles
$T = \{T_0, T_1, T_2, \ldots, T_{n_{ts}}\}$	Division of working day into n_{ts_l} time slices with different length.
N_{T_l}	Set of customers whose orders are received at the previous time slices

Assumptions:

- Travel time between customers is their euclidean distance $t_{ij} = d_{ij}$.
- It is not necessary to serve all static customers in the first time slice, and then switch to serve dynamic customers. This is illustrated in figure 1:

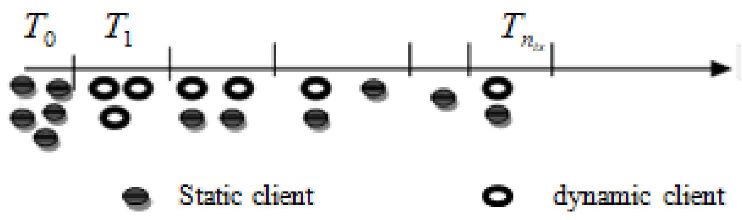


Figure 1. Time discretization in intervals

T_0 Ts may be different from the value 0. So, we start serving customers (static and dynamic) when the beginning of their windows time have been achieved before.

- We assumed that the vehicles leave the depot fully loaded. The vehicles are not all available at the beginning of the horizon, but in a progressive way during the planning horizon, as shown in figure 2:

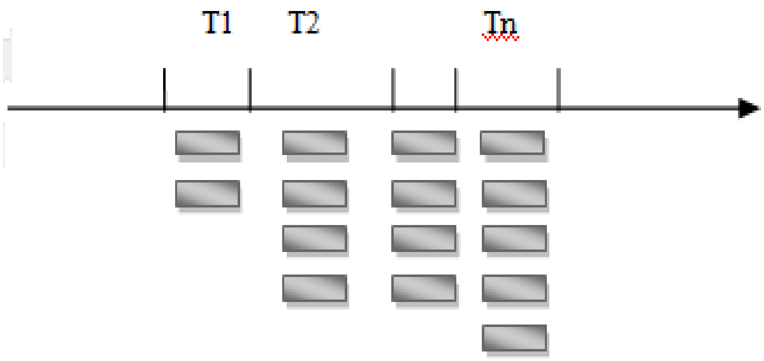


Figure 2. Use of vehicles resources during the planning horizon

- If a vehicle has currently no further requests assigned, the vehicle waits at its last serviced location for further requests assigned to its tour. Once the order is given to vehicle, it leaves this serviced location.
- If the end of the previous interval is larger than the end of service time, the service beginning time at the new customer is the end of this interval plus the travel time between this customer and the new customer. If not, the service beginning time at the new customer is the end of service at the last customer plus the travel time between this customer and the new customer.
- We assume that objectives are equally important (all the weights are equal). We present a three-objective model with the same dimension (time).

Decision variables

x_{ijk} 1, if there is travel from *i* to *j* by the vehicle *k* and 0 otherwise.

z_{ik} 1,if vehicle *k* visit customer *i* and 0 otherwise.

$x_{0_{lk}ik}$ 1,if there is travel from 0_{lk} (last customer served by *k* in time slice *l*) and customer *i* by *k*.

Mathematical model:

Now, the problem can be mathematically formulated as follows:

$$\text{Minimize} \quad f = f1 + f2 + f3 \tag{1}$$

$$f1 = \sum_{k \in K} \sum_{i,j \in V_l} t_{ij} x_{ijk} \qquad f2 = \sum_{k \in K} \sum_{i,j \in V_l} \max(0, e_j - A_j) x_{ijk} \qquad f3 = \sum_{k \in K} \sum_{i,j \in V_l} \max(0, A_j - l_j) x_{ijk}$$

Subject to

$$\sum_{k\in K}\sum_{i\in N_{T_l}} x_{ijk} = 1 \qquad \forall j \in N_{T_l} \tag{2}$$

$$\sum_{j\in N_{T_l}} x_{jik} = \sum_{j\in N_{T_l}} x_{ijk} \qquad \forall i \in N_{T_l} \forall k \in K \tag{3}$$

$$\sum_{i\in N_{T_l}} x_{0_{lk}ik} \leq 1 \qquad \forall k \in K \tag{4}$$

$$\sum_{i\in N_{T_l}} x_{0_{lk}ik} = \sum_{i\in N_{T_l}} x_{i0k} \qquad \forall k \in K \tag{5}$$

$$\sum_{j\in N_{T_l}} q_j z_{jk} \leq Q \qquad \forall k \in K \tag{6}$$

$$(Q - y_{0_{lk}k}) + q_{0_{lk}} + \sum_{i\in N_{T_l}} q_i z_{ik} \leq Q \qquad \forall k \in K \tag{7}$$

$$Ts + \sum_{i,j\in N_{T_l}\cup\{0,0_{lk}\}} x_{ijk} t_{ij} + \sum_{i\in N_{T_l}} z_{ik} s_i \leq T \qquad \forall k \in K \tag{8}$$

$$y_{jk} = (y_{ik} - q_i)x_{ijk} \qquad \forall i, j \in N, \forall k \in K \tag{9}$$

$$D_i = [\max(A_i, e_i)] + s_i \qquad \forall i \in N_{T_l} \tag{10}$$

$$D_i + t_{ij} + s_j \leq D_j \qquad \forall i, j \in N, \forall k \in K \tag{11}$$

$$x_{ijk}, z_{ik}, x_{0_{lk}ik} \in \{0,1\} \qquad \forall i, j \in N, \forall k \in K \tag{12}$$

The objective function (1) in this model is to minimize the transportation time. The first part attempts to minimize the total travel time between customers, the second part minimizes the total sum of stopping time while the third part includes minimization of total sum of delay time. Constraint (2) ensures that each customer appeared in the time slice T_l is visited exactly once. Constraints (3) denote the flow balance for each vehicle, that is, it controls input to output from each intermediate node constituting two arcs. Constraints (4) and (5) state that each vehicle starts as the last customer has been visited at the previous time slice and returns to the depot at the end of his route. The quantity requested by a set of customers served in a time slice by a vehicle must not exceed the capacity of the vehicle. This is ensured by the constraint (6). Constraint (7) indicates the capacity constraint of the k^{th} vehicle, and constraint (8) checks the maximum time limitation for each vehicle. Constraint (9) updates the vehicle load. The temporal coherence of tours is respected by constraints (10) and (11). Finally, decision variables are defined as binary in constraint (12).

3. Dynamic vehicle routing problem with soft time windows

Our algorithm is designed to generate high quality solutions by applying some improvements that focus on two points which are mentioned as follows:

- The strategy of discretization we have been following is different from that proposed in literature, because we used different lengths of time slices. Also, we can delay the start of the planning horizon of static requests until we have other dynamic requests, and we use gradually the available vehicles as explained before in Figure 2.

- The hybridization of the genetic algorithm which allows a strong diversification, and the VNS which is based on the local search, and would correspond to a marked intensification.

The hybrid algorithm adapted in this work was organized on two stages. The first stage aims to serve customers who are known in advance at the beginning of the planning horizon (but not necessarily all). In the second stage, we serve, at the end of each time slot, the requests revealed dynamically over time, in addition to some static requests not yet serviced.

In this paper, we consider the vehicle routing problem with soft time window constraints, which is closer to the reality, and we consider that the dynamic event is the appearance of dynamic customers. New request must be inserted into a single tour without modification of order of the visited customers and with a minimum of delay. The demand of the new customer is unknown, and the time of her appearance is also unknown. This problem belongs to the category of dynamic and deterministic vehicle routing problems, where the part or all of the input is unknown and revealed over time, meaning that the complete instance is only known at the end of the planning horizon. For these problems, vehicle routes are redefined in an ongoing fashion, requiring technological support for real-time communication between the vehicles and the decision maker (e.g., mobile phones and global positioning systems).

Customers have soft time windows constraints, so we can collect the new requests that arrived during a given interval of time, and serve them at the end of this interval. The idea of discretization in intervals was first proposed by Kilby et al. (1998), but in our work, we use different length of time slices, depending on the number of customers appeared and the time window widths.

The problem is solved as a static problem at the end of each time slice by a hybrid genetic algorithm that will be detailed in this section. The only difference is that for the partial static problem, we use vehicles having different starting locations.

The choice of customers to be served in a given time interval is constrained. A constraint of time window (start time) is introduced to decrease the number of customers served prior to start time window. In order to improve the results, we added another constraint depending on the end of the window time. So, customers whose temporal windows are wide are served at the end, and give way to customers whose temporal windows are narrow.

3.1. The hybrid method proposed to solve the partial static problem

The dynamic problem resolution already presented is based on a hybrid algorithm. This solution method combines the advantages of an evolutionary method which is excellent for detecting good regions in the search space with the advantages of a local search method which explore efficiently promising areas. The hybrid algorithms allowed obtaining the better results, in our case the genetic algorithm method and the variable neighborhood search method (VNS) were hybridized in order to improve the solutions obtained.

A genetic algorithm works by building a population of chromosomes which is a set of possible solutions to the optimization problem. Each solution in the initial population is the permutation of n positive integers, such that each integer is corresponding to a customer. We use a single line to represent each solution. It is a representation of several tours served by a set of vehicles. Figure 3 illustrates how we can obtain vehicles tours from an element of the population:

Customer i	1	3	2	7	5	6	4
q_i	10	7	2	12	6	5	11

The solution can be presented as follows figure:

Vehicle 1	0_{l1}	1	3	2
Vehicle 2	0_{l2}	7	5	
Vehicle 3	0_{l3}	6	4	

Figujre 3. An example of encoding of a solution

According to the example, we must serve seven customers in a given time interval, each with a request q_i . Knowing that the capacity of vehicle is 20, the chromosome may be broken into three part. Customers are listed in their order of visitation, vehicles depart from 0_{lk} (the last customer served by vehicle *k* at the previous time slice), and return to the depot if the vehicle cannot serve the next customer (luck load)or if there are no other customers waiting.

Constructive methods produce acceptable solutions starting from an empty initial solution and inserting, at each stage, a component in the current partial solution. Constructive methods are distinguished by their speed and simplicity. We obtain very quickly a feasible solution to a given problem without using highly sophisticated techniques. So, we use a greedy constructive heuristic to generate a 50% of the initial population, the greedy method starts with one customer and moves systematically to the nearest customer that has not yet been visited. The rest of the population is generated randomly.

The performance of genetic algorithms is affected by genetic operators. We present thereafter the crossover and mutation operators applied to the initial population in each time slot.

There are several different types of crossover, i.e. crossover point, linear order crossover (LOX), order crossover (OX). Our approach to resolve this problem uses order crossover which can be explained as follow: a swath of consecutive alleles from parent *P1* drops down, and the remaining values are placed in the child *S1* in the order which appear in parent *P2*. If you desire a second child *S2* from the two parents, flip Parent *P1* and Parent *P2*.

Once the two solutions (chromosomes) *S1* and *S2* were obtained, and considering the customers' requests (shown between parentheses), and vehicle capacity (20), we can build tours with customers to serve in that time slot. We obtain the vehicles tours as explained in figure.4.

We know that the mutation is a genetic operator used to maintain genetic diversity from one generation of a population to the next. For the permutation encoding already used, we apply order changing which select two numbers (customer 5 and customer 1) and exchange them. Figure 4 illustrates the passage from solution *S1* in the form of tours to solution encoded by chromosome, which is then mutated:

V 1	0_{l1}	2	5	7
V 2	0_{l2}	6	1	
V 3	0_{l3}	4	3	

↓ ↓

S1	2 (2)	5 (6)	7 (12)	6 (5)	1 (10)	4 (11)	3 (7)

S1	2 (2)	1 (10)	7 (12)	6 (5)	5 (6)	4 (11)	3 (7)

V 1	0_{l1}	2	1
V 2	0_{l2}	7	6
V 3	0_{l3}	5	4
V4	0_{l4}	3	

Figure 4. Mutation operator applied to S1

According to this example, we find that the solution obtained requires the availability of a fourth vehicle to serve all customers waiting to be served in that time slot.

After we apply the crossing and mutation, we must determine individuals who will be present in the following population. For this reason, the child who improves the objective function will replace poor quality solution in the population at each iteration.

The proposed approach is also based on a metaheuristic called variable neighborhood search, where the principle is the change of neighborhoods during the search. This approach is called hybrid genetic algorithms, or local search hybrid, or memetic algorithms because it is the hybridization of the genetic algorithm and the neighborhood search algorithm. The procedure VNS applied to our problem is detailed in the following paragraphs. The description consists of the building of an initial solution, the shaking phase, the local search method, and the acceptance decision.

The building of an initial solution: The initial solution is the best obtained by the genetic algorithm.

The shaking phase: We use three neighborhood structures, the first one uses the insertion method to generate neighbors, by inserting a randomly chosen customer in a new position also chosen randomly.

In the other neighborhood structure, two randomly selected customers are simply swapped. The third neighborhood structures (Fisher-Yates) is used to randomly permute the chromosome.

Local search method: This paper selects 2-opt as a local search operator in order to obtain the good quality local optimal solution in a short period.

The acceptance decision: If the local optimum is better than the incumbent, the latter is updated, and we continue the search with the same neighborhood structure. Otherwise, we move to another neighborhood structure.

The description of the steps of VNS, obtained by taking two neighborhoods, is shown in figure 5:

procedure VNS(initial solution s)
Best←s
Three neighborhood structures are defined N_1 (by insert), N_2 (by exchange), N_3 (Fisher-Yates).
Repeat
k←1
repeat
Shaking: choose s' ∈ $N_k(s)$
s" ←2-opt on s'
if f(s") <f(s) **then** s←s"
else k ←k+1 **end if**
Update Best
I←I+1
until $k = k_{\max}$
until ($I = I_{\max}$)
end

Figure 5. Procedure VNS

The VNS process already presented is introduced in the figure 6 with a concrete example to illustrate the mechanisms of each phase of the algorithm.

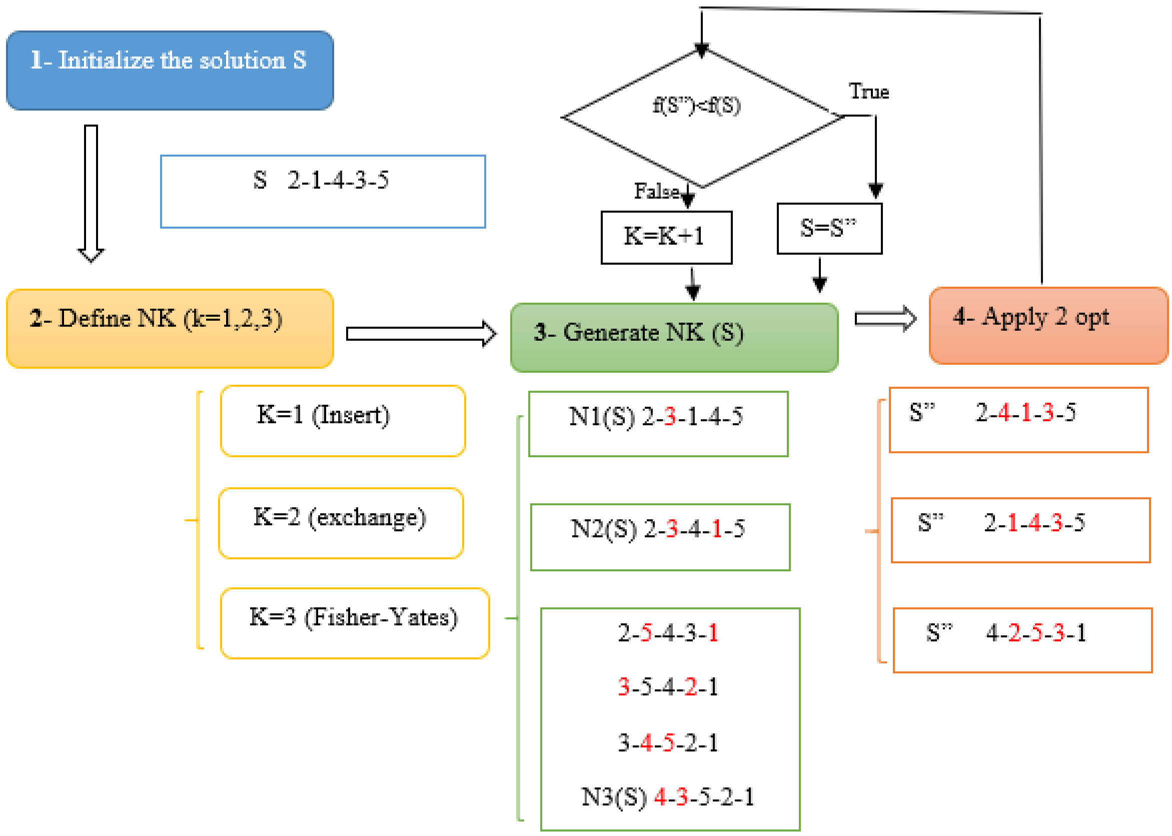


Figure 6 Example detailing the VNS process

In this figure that illustrates the process of VNS, we have detailed the shaking phase which consists of proposing 3 neighborhood structures Nk (k=1,2,3), then generate neighbors, by applying these structures on the initial solution S. We took as an example the case of 5 customers (1,2,3,4,5). For insertion (NK1), customer 3 is inserted in a new position after customer 2, while the exchange (NK2) consists of swapping two customers 1 and 3. NK3 (Fisher-Yates) makes successive permutations from the last element. The local search phase applies 2-opt, by inverting the path between customers 2 and 5.

The steps of the proposed hybrid method with time discretization in intervals are illustrated in figure.7:

l:=0;
The starting time of each vehicle is set at the depot
N_{T_l} ; static demands(orders known at the beginning of the planning horizon).
StaticProblem:=problem with some static demands
Run the Hybrid method on staticProblem;
While($N_{T_l} \neq \emptyset$)
l:=l+1;
N_{T_l} :=Orders received at the previous time slices, and some of the static demands.
PartialProblem:=problem with orders in N_{T_l}
Run the hybrid method on PartialProblem;
Update starting positions, load of vehicles;
EndWhile
Commit the depot to all the vehicles;
End

Figure 7. Hybrid method with time discretization in intervals

3.2. The proposed heuristic to serve dynamic customers at each time t

The proposed approach in the previous section is much convenient and effective in case we receive many requests in each time slot, but if we receive few requests, a heuristic is proposed to serve each customer at the time t (time of receipt of the request). The responsibility for the planning of tours must differentiate between the two cases, and apply the appropriate approach.

The steps of the insertion method (shown in figure 8) are:

- The planning of routes to serve a set of customers by applying the hybrid method (Previously presented).
- Determination of vehicle's position at the time of an appearance of dynamic customer.
- Introduce the possibilities of insertion of dynamic demand, but customers in the previous appearance of this customer should be frozen.
- Update the routing of vehicles when we apply the insertion minimizing our objective function.

Example of insertion of dynamic customers: The heuristic already presented in the previous paragraph will be applied to a simple example. The problem is to ensure delivery service to static customers {1,2,3,4,5,6,7,8} and dynamic customer {10}. The route planning served by 3 vehicles {1, 2, 3} is based on hybrid method that we proposed. For each tour, we determine the position of the vehicle, and the possibilities of insertion of customer 10 (see figure 8):

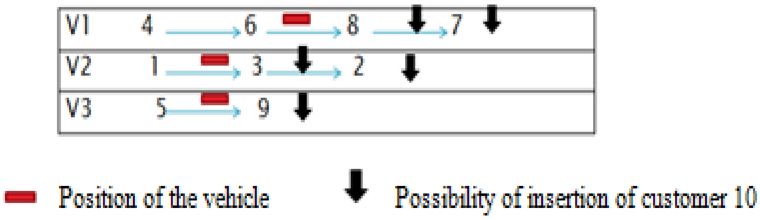


Figure 8. Example of vehicle route planning

Now, we must calculate the variation of the objective function of every possibility. If the customer 10 is inserted between customer x and customer y, the goal is to minimize the variation: $v = (V_1, V_2, V_3)$

$$\begin{cases} V_1 = d_{x,10} + d_{10,y} - d_{x,y} \\ V_2 = \sum_{y^+ \in Y} \max(0, T + A_{y^+} - l_{y^+}) \\ V_3 = \sum_{y^+ \in Y} \max(0, e_{y^+} - A_{y+} - T) \end{cases}$$

Y: set of customers to be served after the customer y, and it is included there.

In the following section, we represent briefly all steps of the adapted approach to solve a DVRPSTW with a set of static customers (static part), and with the appearance of some dynamic customers:

4. Computational results

The proposed method was coded in C++ and executed in an Intel (R) Core(TM) i5 Processor 2.67GHz with 4 Go of RAM .The 56 Solomon benchmark problems for the VRPHTW are based on six groups of problem instances with 100 customers. This section compares the results of the hybrid genetic algorithm against the results of the well-known Solomon benchmark problems for the VRPHTW for the problem set R1.

The stop duration of a vehicle at each customer allows avoiding early deliveries, but it can increase the delay time for same customers or increase the number of customers with late deliveries. We will present results for both cases, the case where the vehicle has to wait for the customer's window to start, and another where it is possible to serve customers before the start of the delivery window, which does not imply a vehicle stop.

In order to compare our results obtained on the instances of Solomon with the best known results in the literature when tested on solomon's VRPTW 100 customer instances, we assume that the system is fully static, i.e. all requests are received at time 0.

1: Initialization: generate the initial population
2: Repeat
3: Select two solution x and x '
4: Crossover operator applied on two parents x and x ' → children y1 and y2
5: for each child y faire
6: Mutation is applied to each child → y'
7: Replace one solution from the population by y' if there is improvement of the objective function
8: End For
9: Until (Stopping criterion)
10: Procedure VNS is applied on s → s' (s is the best solution obtained by the adapted Genetic Algorithm)
11: Repeat
12: Determination of vehicle's position at the time t
13: Study possibilities of insertion of dynamic customer
14: The insertion which minimizes our objective function has been applied
15: Update the routing of the vehicles
16: Update the time t and the solution s'
17: Until (the end of the planning horizon)

Figure 9 .Steps of the insertion method

The genetic algorithm parameters were set to the following suitable values determined by:

popSize = 5000 (the number of individuals in the population)
γ = 1.0 (probability of recombination)
μ = 0.05 (probability of mutation)
numGen = 500 (number of generations)
Tsize = 2 (tournament size)
Each version of the algorithm was run 30 times.

The results that illustrate the number of routes (*NV*), the distance, the average of delay time en min (*D*) and the average of stopping time en min(*S*) are shown in the following table (Table 1):

Table 1. Results of simulation

Problem	**Best Known Solutions Identified by Heuristics**		**Results obtained by the proposed approach**			
	NV	***Distance***	***NV***	***Distance***	***D***	***S***
R101	20	1637.7	17	1593.48	6.53	8.73
R102	17	1486.12	15	1363.36	4.82	7.18
R103	13	1292.68	10	1142.58	2.86	2.93
R104	9	1007.24	8	998.52	2.26	1.00
R105	14	1377.11	9	1374.24	2.82	4.30
R106	12	1251.98	10	1198.67	3.22	2.35
R107	10	1104.66	9	1095.55	3.46	2.42
R108	9	960.88	8	954.93	0.11	0.95

Problem	Best Known Solutions Identified by Heuristics		Results obtained by the proposed approach			
	NV	***Distance***	***NV***	***Distance***	***D***	***S***
R109	11	1194.73	10	1144.18	5.17	2.38
R110	10	1118.59	8	1121.89	3.27	1.14
R111	10	1096.72	9	1074.54	3.20	1.40
R112	9	982.14	8	982.96	2.07	0.73

We can observe that our results decrease the number of routes and the distance of the routes with:

- Respect of the earliest arrival time for customer;
- Delay time mostly not exceeding the average of 3.5min at each customer;
- Stop duration equal to 19 min on average for each vehicle.

If we can make an early delivery, the vehicles will not stop for waiting the start of the window, as a result, the delay time will decrease, but some customers will be served ahead of time. The total distance of the tour, the average of delay time (*D*), and the difference in time between the beginning of the time window and the delivery time (*S*) are shown in the following table (Table 2):

Table 2. Results of simulation with permission of early delivery

Problem	Best Known Solutions Identified by Heuristics		Results obtained by the proposed approach			
	NV	***Distance***	***NV***	***Distance***	***D***	***S***
*R*101	20	1637.7	10	1557.84	1.43	6.68
*R*102	17	1486.12	10	1358.40	1.98	5.80
*R*103	13	1292.68	10	1192.14	1.10	6.14
*R*104	9	1007.24	8	954.53	1.39	3.28
*R*105	14	1377.11	10	1240.36	1.71	6.61
*R*106	12	1251.98	10	1130.14	1.68	4.26
*R*107	10	1104.66	8	1015.14	1.55	3.96
*R*108	9	960.88	8	906.46	2.43	2.60
*R*109	11	1194.73	10	1092.92	1.17	5.09
*R*110	10	1118.59	8	1038.02	2.97	3.30
*R*111	10	1096.72	8	1052.22	1.90	3.61
*R*112	9	982.14	8	979.81	1.65	2.18

As shown in the above table, the average of delay time is 1.74 min, whereas this average was equal to 3.31 in the previous case, which means that the delay decreased by 47%.

On the other hand, the average of time for customers served ahead of time is 4.45. In the real world, this time is negligible and does not entail major problems for customers.

Now, we will compare the results of the hybrid genetic algorithm against other solution methods that report solution quality and computation time on benchmark problems for the VRPSTW. These benchmark problems are variations of the well-known Solomon benchmark problems for the VRPHTW.

There are some references with time and cost results for VRPTW with soft time windows. The solution results presented by Balakrishnan (1993) are denoted BAL. The solution results presented by Chiang and Russell (2004) which use two solution methods, tabu search and advance recovery are denoted by the initials TB and AR respectively. The solution results presented by Fu, Eglese, and Li (2008) using the unified tabu search method are denoted (UTS). The solution results of the new iterative route construction and improvement (IRCI) presented by Figliozzi (2010) are denoted IRCIs (short computational times) and IRCIe (extended computational times).

In the VRP with soft time windows, deliveries are still possible outside the time windows with some penalty costs. Therefore, in this type of problem, there is an allowable violation of time windows denoted $P_{\max} \geq 0$. It means each customer can be served in the interval $[e_i - P_{\max}, l_i + P_{\max}]$ with an early penalty if service time starts early or a late penalty if service starts late. $P_{\max}$ can be either 10%, 5%, or 0% of the total route duration $[l_0 - e_0]$. To facilitate comparisons, a new constraint that limits the maximum waiting time (the amount of time that a vehicle can wait at a customer location before starting service) is added: *Wmax*=10%.

The primary objective function for the VRPSTW is the minimization of the number of routes (*NV*). A secondary objective is the minimization of the number of time window violations *(%HTW*). A third objective is the minimization of total time or distance plus penalties for early or late deliveries. This is illustrated in the following table (Table 5):

Table 3. VRPSTW results for R1 problems, W=10%, P=10%

	Methods						
Problem	***BAL***	***TS***	***AR***	***UTS***	***IRCIs***	***IRCIe***	***Our approach***
*R*101							
NV	15	14	12	12	13	12	10
%HTW	62	49	8	31	43	25	19
Distance	1832	1388	1212	1376	1493	1314	1387
*R*103							
Veh.	13	11	10	10	11	10	10
%HTW	83	65	58	76	76	66	80
Distance	1657	1063	1013	1185	1274	1138	1561
*R*109							
Veh.	13	12	11	11	11	11	10
%HTW	95	84	60	75	85	93	72
Distance	1445	1154	1084	1168	1336	1393	1449
Aver.							
Veh.	13.6	12.3	11	11	11.6	11	10
%HTW	80	66	42	60	68	61.3	57
Distance	1644	1201	1103	1243	1367	1281	1465

Table 3 shows that our proposed method uses the minimum of vehicles compared to others approache

Now, we present some results for DVRPSTW. In this problem, we must integrate dynamic customers after the end of each time slice and update routes. The proposed method has been tested on the problem set C1. In table 4, we present an example of these tests:

Table 4. Results for DVRPSTW for C103

	ACSLNS		AG hybrid			Our approach		
dod	***D***	***NCS***	***D***	***Delay***	***PCS***	***D***	***Delay***	***PCS***
10	1386	99	1506	1.02	93	1506	1.02	93
30	1769	100	1630	1.16	91	1701	0.90	95
50	1786	100	1668	1.72	89	1640	1.30	92
70	1810	99	1646	1.82	94	1650	1.22	95
90	1732	100	1666	1.34	91	1666	1.34	91

dod (degree of dynamism): The ratio between the number of dynamic requests and the total number of requests.

D: The total distance traveled by all the vehicles used.

NCS: The number of customers served (The total number of customers is 100).

PCS: The percentage of customers served within their time window.

ACSLNS is a proposed approach to solve the VRPHTW (Messaoud et al., 2013), it combines two methods: ant colony algorithm (ACS) for minimizing the number of vehicles (ACSNV) and large neighborhood search (LNS) for minimizing the total distance.

AG hybrid: is our proposed approach (Bouziyane et al., 2016), which combines two methods: the genetic algorithm and the VNS method.

Our tests were carried out with 10 vehicles by treating 5 degrees of dynamics (dod), which are 10%, 30%, 50%, 70% and 90%.

According to table 4, the number of customers served outside the time window is negligible compared to the number of customers served in their time windows (PCS).

Also, compared to the distance in the case of application of the ACSLNS –DVRPHTW, our approach was able to reduce this distance, with an average of 1.4 min of delay.

The proposed heuristic is very effective in the case where we receive a few dynamic customers distributed over the planning horizon. Dynamic problems characterized by short reaction times are the most concerned by this heuristic. Table 5 presents some results generated by the proposed heuristic, compared to results of our hybrid method.

Table 5. Results of the proposed heuristic for DVRPSTW

	Our approach (Hybrid)			Our approach(heuristic)		
dod	***D***	***Delay***	***PCS***	***D***	***Delay***	***PCS***
10	1506	1.02	93	1406	0.2	95
30	1701	0.90	95	1600	1.5	95
50	1640	1.30	92	1687	1.2	91
70	1650	1.22	95	1820	0.6	97
90	1666	1.34	91	1703	0.8	95

There is an improvement in results in the case, where just 10% or 30% of customers are dynamics. The purpose of this heuristic is to minimize time between the reception and the start of service of each request. So, firefighters, ambulances, and the police, for example, are the most concerned by this proposed heuristic.

5. Conclusion

In this paper, we were able to deal with a very interesting and practical problem especially with the technological advances, which allows companies to put in place all the equipment necessary for the resolution of their dynamic problems.

The hybrid approach proposed to solve the DVRPSTW is based on the genetic algorithm and the VNS method. In this problem, we integrated new requests during the execution of planned routes. The human factor is present in selecting from the two proposed methods, one that postpones the delivery of requests received until the end of one time interval, and another that served each customer directly after receiving his request. We solved the dynamic problem based on the static resolution, the objective is to minimize the weighted sum of the delays, stops, and the total cost of transportation. The decision maker is well aware of the importance of each objective, so it is up to him to assign weighting coefficients to each part of the objective function.

To validate our approaches, a dynamic benchmark is available in the literature based on the static Solomon's benchmark for VRPTW, by revealing some of the orders only during operation time to the algorithm. The obtained results are encouraging and comparable to those found in the literature. In future studies, we will aim to use a multi-objective method to treat the VRPSTW with several antagonist objectives to optimize simultaneously. The proposed objective is to provide a set of Pareto optimal solutions by proposing an improved multi-objective local search method (MOLS) based on a hybrid approach. We extend metaheuristics through simulation to deal with real-life uncertainty that is what we call simheuristics.

References

Armas J. and Melián-Batista B. (2015). Variable Neighborhood Search for a Dynamic RichVehicle Routing Problem with time windows. *Computers & Industrial Engineering*, Vol. 85, pp. 120-131.

Badeau P., Gendreau M., Guertin F., Potvin J.Y. and Taillard E. (1995) . A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transportation Science*, Vol. 31(2), pp. 170 – 186 .

Balakrishnan N. (1993). Simple heuristics for the vehicle routing problem with soft time windows. *The Journal of the Operational Research Society* , Vol. 44, pp. 279–287.

Barkaoui M., Berger J. and Boukhtouta A. (2013). A Hybrid Genetic Approach for the Dynamic Vehicle Routing Problem with Time Windows. *American Journal of Mathematical and Management Sciences*, Vol. 28, pp. 131-154.

Bouziyane B., Dkhissi B. and Cherkaoui M. (2016). Hybrid genetic algorithm for the static and dynamic Vehicle Routing Problem with Soft Time Windows. *The 3rd IEEE International Conference on logistics Operations Management.*

Cao E. and Lai M. (2010).The open vehicle routing problem with fuzzy demands. *Expert Systems with Applications*, Vol. 37(3), pp. 2405–2411.

Chen S., Chen R.,Wang G., Gao J. and Sangaiah A.K. (2018). An adaptive large neighborhood search heuristic for dynamic vehicle routing problems.*Computers & Electrical Engineering*,in Press.

Chen S., Chen R. and Gao J. (2017). A Modified Harmony Search Algorithm for Solving the Dynamic Vehicle Routing Problem with Time Windows. *Scientific Programming*, Vol. 2017, pp. 1-13.

Chiang W.C. and Russell R.A. (2004). A metaheuristic for the vehicle-routing problem with soft time windows. *Journal of the Operational Research Society* ,Vol. 55, pp.1298–1310.

Çimen M. and Soysal M. (2017). Time-dependent green vehicle routing problem with stochastic vehicle speeds: An approximate dynamic programming algorithm. *Transportation Research Part D: Transport and Environment*, Vol. 54, pp. 82-98.

Euchi J.,Yassine A. and Chabchoub H. (2015). The dynamic vehicle routing problem:solution with hybrid metaheuristic approach.*Swarm and Evolutionary Computation*, Vol. 21, pp. 41-53

Ferrucci F. (2013). *Pro-active Dynamic Vehicle Routing*, Contributions to Management Science series. Berlin Heidelberg: Physica-Verlag Heidelberg.

Fu Z., Eglese R., LI L.Y. (2008).A unified tabu search algorithm for vehicle routing problems with soft time window.*Journal of the Operational Research Society* , Vol. 59, pp. 663-673.

Figliozzi A. (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transportation Research Part C: Emerging Technologies,* Vol. 18(5), pp. 668-679.

Gendreau M., Guertin F. , Potvin J.Y. and Taillard E. (1999). Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*, Vol. 33(4), pp. 381-390.

Hong L. (2012). An improved LNS algorithm for real-time vehicle routing problem with time windows. *Computers and Operations Research*, Vol. 39(2), pp. 151–163.

Hu X., Sun L. and Liu L. (2013). A PAM approach to handling disruptions in real-time vehicle routing problems. *Decision Support Systems*, Vol. 54(3), pp. 1380–1393.

Khouadija M.R., Sarasola B., Alba E., Jourdan L. and Talbi E.G. (2012). A comparative study between dynamic adapted PSO and VNS for the Vehicle Routing Problem with dynamic request. *Applied Soft Computing*, Vol. 12(4), pp. 1426-1439.

Kilby P., Prosser P. and Shaw P. (1998). Dynamic VRPs : a study of scenarios. CSIRO Mathematical and Information Sciences, University of Strathclyde, UK.

Kuo R.J., Wibowo B.S. and Zulvia F.E. (2016). Application of a fuzzy ant colony system to solve the dynamic vehicle routing problem with uncertain service time.*Applied Mathematical Modelling,* Vol. 40 (23-24), pp. 9990-10001.

Larioui S., Reghioui M., Elfallahi A. and Elkadiri K.E. (2015). A memetic algorithm for the vehicle routing problema with cross docking. *International Journal of Supply and Operations Management*, Vol. 2(3), pp. 833-855.

Lei H., Laporte G. and Guo B. (2011). The capacitated vehicle routing problem with stochastic demands and time windows. *Computers & Operations Research*, Vol. 38(12), pp. 1775–1783.

Mańdziuk J. and Żychowski A.(2016). A memetic approach to vehicle routing problem with dynamic requests.*Applied soft computing* , Vol. 48, pp. 522-534.

Messaoud E., El hilali Alaoui A. and Boukachour J. (2013). Hybridation de l'algorithme de colonie de Fourmis avec l'algorithme de recherche à grand Voisinage pour la résolution du VRPTW statique et dynamique. *Information Systems and Operational Research*, Vol. 51, pp. 40-50.

Mirabi M., Shokri N. and Sadeghieh A. (2016). Modeling and Solving the Multi-depot Vehicle Routing Problem with Time Window by Considering the Flexible End Depot in Each Route. *International Journal of Supply and Operations Management* , Vol. 3(3) , pp. 1373-1390.

Mirmohammadi, S. H., Babaee Tirkolaee, E., Goli, A. and Dehnavi-Arani, S. (2017). The periodic green vehicle routing problem with considering of the time-dependent urban traffic and time windows. *Iran University of Science & Technology*, Vol. 7(1), pp. 143-156.

Moghaddam B.F., Ruiz R. and Sadjadi S.J. (2012). Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Computers & Industrial Engineering*, Vol. 62(1), pp. 306–317.

Moretti Branchini R., Amaral Armentano V. and Løkketangen A. (2009). Adaptive granular local search heuristic for a dynamic vehicle routing problem. *Computers & Operations Research*, Vol. 36(11), pp. 2955–2968.

Novoa C. and Storer R. (2009). An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research,* Vol. 196(2), pp. 509–515.

Okulewicz M. and Mańdziuk J. (2017) .The impact of particular components of the PSO-based algorithm solving the Dynamic Vehicle Routing Problem. *Applied soft computing*, Vol. 58, pp. 586-604.

Oudani M., El Hilali Alaoui A. and Boukachour J. (2014). An efficient genetic algorithm to solve the intermodal terminal location problem. *International Journal of Supply and Operations Management,* Vol. 1(3), pp. 279-296**.**

Pillac V.,Gendreau M., Guére C. and Medaglia A.L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, Vol. 225(1), pp. 1–11.

Qureshi A.G., Taniguchi E. and Yamada T. (2012). A Microsimulation Based Analysis of Exact Solution of Dynamic Vehicle Routing with Soft Time Windows. *Procedia-Social and Behavioral Sciences*, Vol. 39,pp. 205-216.

Russell G., Fwa T.F. and Eiichi T. (2014). *Urban Transportation and Logistics*, Vehicle Routing and Scheduling with Uncertainty. CRC Press.

Qiuyun W., Wenbao J. and Gang Z. (2013). A Novel Model and Algorithm for Solving Dynamic Vehicle Routing Problem on Goods Distribution. *Journal of Applied Sciences,*Vol. 13, pp. 5410-5415.

Rabbani M., Ramezankhani M.J., Farrokhi-Asl H. and Farshbaf-Geranmayeh A. (2015). Vehicle routing with time windows and customer selection for perishable goods. *International Journal of Supply and Operations Management* , Vol. 2 (2), pp. 700-719.

Rabbani, M., Bosjin, S., Yazdanparast, R. and Saravi, N. (2018). A stochastic time-dependent green capacitated vehicle routing and scheduling problem with time window, resiliency and reliability: a case study. *Decision Science Letters*, Vol. 7(4), pp. 381-394.

Taniguchi E. and Shimamoto H. (2004) .Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel time. *Transportation Research,* Vol.12 (Part C), pp 235-250.

Tirkolaee E. B., Goli A., Bakhsi M. and Mahdavi I. (2017). A robust multi-trip vehicle routing problem of perishable products with intermediate depots and time windows. *Numerical Algebra, Control & Optimization*, Vol. 7(4), pp. 417-433.

Tirkolaee E. B., Alinaghian A., Hosseinabadi A. A. R., Sasi M. B. and Sangaiah A. K. (2018a). An improved ant colony optimization for the multi-trip Capacitated Arc Routing Problem, *Computers & Electrical Engineering*, in press.

Tirkolaee, E. B., Mahdavi, I., and Esfahani, M. M. S. (2018b). A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew’s working time. *Waste Management*, in press.

Wen M., Cordeau J.F., Laporte G. and Larsen J. (2010).The dynamic multi-period vehicle routing problem. *Computers & Operations Research*, Vol. 37(9), pp. 1615–1623.

Yang Z., Van Osta J.P., Van Veen B., Van Krevelen R., Van Klaveren R., Stam A., Kok J., Bäck T. and Emmerich M., (2017). Dynamic vehicle routing with time windows in theory and practice. *Natural computing*, Vol. 16(1), pp. 119-134.

Yassen E.T., Masri A., Nazri M.Z. and Sabar N.S. (2017). An adaptive hybrid algorithm for vehicle routing problems with time windows.Computers & Industrial Engineering, Vol. 113, pp. 382-391.