Ant Colony System for Dynamic Vehicle Routing Problem with Overtime

Khaoula OUADDI, Youssef BENADADA, Fatima-Zahra MHADA
Smart Systems Laboratory, Rabat IT Center
ENSIAS, Mohammed V University in Rabat, Madinat Al Irfane, BP 713, Agdal Rabat, MOROCCO

***Abstract*—Traditionally, in a VRP the vehicles return to depot before the end of the working time. However, in reality several constraints can occur and prevent the vehicles from being at the depot on time. In the dynamic case, we are supposed to answer the requests the same day of their arrival. Nevertheless, it is not always easy to find a solution, which ensures the service while respecting the normal working time. Therefore, allowing the vehicle to use additional time to complete their service may be very useful especially if we have a large demand with a limited number of vehicles. In this context, this article proposes a mathematical modeling with an Ant Colony System (ACS) based approach to solve the dynamic vehicle routing problem (DVRP) multi-tours with overtime. To test the algorithm, we propose new data sets inspired from literature benchmarks. The competitiveness of the algorithm is proved on the classical DVRP.**

Keywords—Dynamic vehicle routing problem (DVRP); multi-tours; mathematical modeling; hybrid; Ant Colony System (ACS); overtime

I. Introduction

Being defined more than 50 years ago, Vehicle Routing Problem (VRP) is one of the most classical combinatorial optimization problems. The main objective is to find the optimal path that can visit all nodes in question. For example, in a capacity VRP, these nodes are a set of customers that need to be served from a single depot with limited load capacity vehicles. Another example is the VRP with pickup and delivery, in which, the nodes can be a set of customers who will be served and from which the goods can be collected and a set of depots where vehicles get off. For each of these two examples we can define several sub-variants by adding some specifications to the problem. For example, we can find a problem with the constraint of time windows during which customers must be visited. Another example is the stochastic case where the requested quantity is unknown exactly before visiting requesting customer. Accordingly, the VRP become a very large class of problems that several studies have devoted themselves to review and classify [1], [2].

Among VRP variants, the dynamic VRP (DVRP) is relatively recent. In this variant, the information available at the beginning is incomplete and is subject to random variations over time. In other words, the starting solution is adjustable according to new data. In Fig. 1, we have a simple example of a DVRP where a vehicle has to serve a set of customers. Dynamic customers (E and F) are inserted in the new planned routes taking into account customers not yet visited (in dotted line).

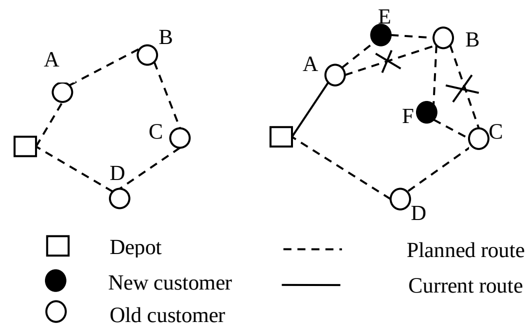

Fig. 1. Example of DVRP.

This concept, relatively new, has brought several advantages that are potentially beneficial for transportation companies; it can be helpful for companies to increase their competitiveness. As long as it allows companies to serve their customers on the same day of their requests and ensure, there is a better customer satisfaction. Besides, due to its flexibility and adaptability, DVRP is very useful in the emergency context where we need an immediate response to requests. Furthermore, the DVRP can handle dynamic travel time. This case is more present in urban areas where it is more difficult to predict network travel time because of the great congestion especially during peak hours.

All these advantages cannot be implemented without technological tools allowing real-time communication between the dispatcher and the driver. Fortunately, new communication and geographical location tools have allowed dispatchers and drivers to have a real time idea on the state of the network, and provide a real-time response to customers' requests.

That is why, Intelligent Transportation Systems [3] area set of platforms, each one is dedicated to a particular process. Among these platforms the Advanced Fleet Management Systems (AFMS) which are very useful in the DVRP case. They are specifically designed for dynamic or static business management of fleet while considering possible variations in the travel time on networks links.

As a result, since its first introduction by Psarftis [4] many variants of DVRP have been introduced and studied and literature studies have reviewed and classified them [5]-[8].

Despite all this number of variants treated, the concept of overtime still very little studied in the DVRP literature. This concept, which is widely used by transportation companies in general, is more needed in dynamic cases. Dynamically re-optimizing the current planning to insert new queries usually provides a solution that consumes more time by comparing it with a solution that considers all customers from the beginning. To verify this hypothesis we can just compare the results of the total traveled distance of the dynamic case in Kilby et al. [9] instances and the static case in those of Taillard et al. [10]. Even so, in this model, we must answer the maximum of requests the day of their arrival. If we have a limited number of vehicles, we will need more overtime especially in case of high demand. In this article, we introduce a new variant of DVRP with the concept of overtime. To this end, we will consider the case of a transportation company that has a homogeneous fleet of trucks and which responds dynamically to customers' request. Each truck can perform several tours during the day. A limited number of trucks are available to satisfy all customers 'orders. Consequently, it is not always easy to find a solution that ensures the service while respecting the normal working time. If necessary, the trucks are allowed to use overtime on the condition that it shouldn't exceed the maximum legal overtime. So, we have two objectives in this problem; Minimize the maximum overtime performed by trucks and minimize the total traveled distance. Thus, we propose a multi-objective mathematical model. To solve this problem we propose a hybrid Ant Colony System (ACS) algorithm

The rest of this paper is organized as follows: the second section presents a brief DVRP literature review. The third section presents the mathematical model. The methodology of resolution will be presented in the fourth section. Before concluding, we will present the numerical results in the fifth section.

II. DVRP: Literature Review

Going around DVRP literature we find all types of optimization methods known up to now, from the exact methods up to the metaheuristics. Otherwise, the literature addresses the DVRP according to four main perspectives: deterministic, stochastic, continuous and periodic. In the deterministic case we consider only known requests and deterministically respond to the dynamic ones. In other words, the current plan, takes into account the data actually known by the dispatcher. While in the stochastic case, we can consider stochastic data, such demand forecasts while elaborating a preliminary routing plan. In continuous optimization, the re-optimization starts at the arrival of each new request to get a new plan. While in the periodic one, the planning period is decomposed into time intervals. In this way, during each time interval, incoming requests are collected and inserted all at the end of the current interval or at the beginning of the next interval.

In order to have a global view of this research field, we will list some research works for the four main perspectives, and then we will exam those are related to the DVRP with overtime.

Let's start by works that have adopted deterministic continuous optimization. It started with Psaraftis [4], who is the first to introduce the dynamic DVRP. In this work, he used dynamic programming to deal with the dynamic dial-a-ride problem. The aim is to find the best route at each new demand. In the same perspective, Gendreau et al. [11] applied the Tabu search (TS) to solve the DVRP. As soon as a new request arrives, the algorithm saves its former results in the adaptive memory to insert the locations of the new requests. The TS was also applied, in a deterministic continuous way, by Chang et al. [12] on the DVRP with pickup, delivery and time windows.

As for the work done for the DVRP in a deterministic periodic manner, the first one we mention is the work of Chen and Xu [13] who have proposed an approach based on linear programming and dynamic column generation for the DVRP with time windows and infinite fleet. On their part, Hanshar et al. [14] decompose the working day into time intervals. Then they launch the optimization program at the beginning of the time interval. They propose a solution based on Genetic Algorithm (GA) for capacitated DVRP. The main contribution in their work was the way they represent a chromosome in dynamic optimizations. In the same perspective and using the Neighborhood Search Algorithm (NSA), Hong [15] solves the DVRP with hard time windows. It applies the withdrawal reinsertion mechanism of this algorithm to insert new queries in the already planned routes.

Another adaptation of the NSA to DVRP was proposed by Khouadjia et al. [16]. To conclude with the deterministic periodic manner, we will quote a very interesting work with ACS. It is the first application of the ACS on DVRP by Montemanni et al. [17]. In this masterpiece, the authors decomposed the planning period into time intervals. At the beginning of each time interval, the optimization program is launched to insert the incoming requests during the previous period in the planning of the rest of the day. The static problem is solved using an ordinary version of ACS. As for the other time intervals, the article applies the same algorithm with a modification of the initial rate of pheromone on the arcs of the network. As each sub-problem is potentially similar to its successor, a pheromone conservation mechanism is applied to put the weight on the arcs belonging to the previous solutions and thus reduces the execution time.

In a stochastic periodic manner, Hvattum et al. [18] have developed a heuristic approach whose principle is to divide the planning horizon into time intervals and to assign a set of promising queries to the vehicles at the beginning of each interval depending on their frequencies of occurrence in the possible stochastic scenarios. The algorithm then uses the Branch-and-Regret method to merge them in order to have a single optimal solution. Another example of the stochastic perspective, but this time in a continuous way, is that of Hemert and Poutré [19] who have used GA to solve a problem of collecting charges from customers and delivering them to a single central depot. The authors have introduced the notion of fruitful regions, where there are more probable potential customers. In the same perspectives, a more recent study of Schyns [20], aimed at optimizing the routing of refueling trucks in an airport, proposed an adaptation of the ACS to take into account the lack of visibility of the planning period and the hard time windows.

Our article will deal with the case of a deterministic periodic DVRP with overtime. After checking articles published in this area, we have found only one article that is the article of Gendreau et al. [21] which deals with a DVRP with pickup, delivery and time window by allowing the use of the overtime and without considering the capacity constraint. To solve this multi-objectives problem, the authors proposed NSA algorithm while adopting a deterministic continuous perspective.

III. Mathematical Model

The objective of this article is to solve a capacitated DVRP multi-tours that tolerate overtime. We have a homogeneous fleet of vehicles with limited capacity and a single depot. The vehicles leave the depot at the beginning of the day with a full capacity and return there to restock and start a new tour or to close their working day. In our case, we adopt the Montemmani et al. [17] approach, which divides the planning period into time intervals. Thereby, we collect all the incoming requests during a time interval, to insert them in the planning of the following period. In this way, the re-optimization is launched at the beginning of each interval and the found solution covers all the rest of the planning period.

We set T_f time limit to accept customers' requests. After this time, the incoming requests will be reported to be recorded in the next planning period. Thus, at the beginning of the planning period, we have a CVRP with a homogeneous fleet and a single depot. However, at the beginning of other time intervals, a vehicle that has already served one or more customers will have a less load as it will have a starting point other than the central depot. We will call this starting point fictitious depot. Therefore, we have a CVRP with heterogeneous fleet and several fictitious depots. In both cases, trucks are allowed to perform several tours. At the end of a time interval, if a truck is serving a customer, this later is considered as fictitious depots in the problem of the next time interval. Else, it will be on the road to a destination customer. In this case, this later is considered as fictitious depots in the problem of the next time interval. We present the mathematical model of one time interval. Thus, we put:

F: Set of depots

I: Set of Customers

F^*: Set of depots without central depot $(F\setminus\{0\})$

I^*: Set of Customers and central depot $(I\cup\{0\})$

K: Set of trucks

f: Index of depot (including fictitious depots and the central one)

i: Index of customers

k: Index of trucks

n: Maximum number of tours for a truck

0: Index of the central depot, $0 \in F$

d_{ij}: Cost (distance) of movement between i and j

t_{ij}: Travel time between i and j

Q_k: Remaining capacity of the truck k

Q: The initial capacity of trucks

T_l: Normal driving time remaining for the period l

T: Length of working period

αT: Maximal legal overtime

q_i: Quantity requested by the customer i

$$f_k=\begin{cases}1, \text{ if the truck k is initially stationned in fictitious depot } f\in F^* \\ 0 \quad \text{else}\end{cases}$$

We have a single decision variable:

$$x_{ij}^r=\begin{cases}1, \text{ if the customer j was visited after the customer i during the tour r} \\ 0 \quad \text{else}\end{cases}$$

To simplify the problem formulation, we assume that each vehicle can carry a maximum of n tours during the remainder of the day. Considering that we have K trucks in service, the maximum number of tours that can be achieved is nK. Excluding already served customers, we consider that a tour starts from the fictitious depot where the customer exists at the beginning of the current time interval.

Therefore, for every truck k, we choose to reserve n indices of tours that will be eventually assigned to it k,k+K,..,k+(n-1)K. Tours r with indices $\sum_{i=0}^{N}\sum_{j=0}^{N}x_{ij}^r=0$, refers to tours that will not actually be performed by the trucks.

So we define S_k the set of n possible tours served by the truck k.

$$S_k=\{k+qK,\ q=0..n-1\}$$

$$S_T=\bigcup_{k\in K} S_k$$

If a truck k actually makes n successive rounds, these rounds will have respectively the index k, k + K..., k +(n-1)K. The tour with the index k is the first to be performed, then k+K and finally the owner of the index k + (n-1)K. In this way, if the truck k is initially parked in a fictitious depot different from the central depot, the tour of index r = k must imperatively start from this fictitious depot. Any other tour starts from the central depot. So, if r∈{1,..,K} then, the tour r=k, starts from depot f>0. Else, r starts from the central depot. Now, we define the available capacity for the tour r as:

$$Q_r=\begin{cases}Q_k & \text{if } r\in\{1,..,K\} \\ Q & \text{else}\end{cases}$$

The travel time of a trunk k is the necessary time to serve all its tours. However the overtime for each vehicle is the additional time over the planning horizon whose vehicle needs to ensure its service. If the vehicle respects the constraint of time, the overtime is zero. It can be calculated for a trunk k as follow:

$$OT_k = \max\left(0, \sum_{r\in S_k}\sum_{i\in I\cup J}\sum_{\substack{j\in J\\ j\neq i}} t_{ij}x^r_{ij} - T_l\right)$$

The solution overtime is the maximum overtime of all vehicles. It's related to the last truck that returns definitively to the depot:

$$OT_l = \max_{0\leq k\leq K} OT_k$$

During each period of re-optimization, the mathematical model consists of minimizing the maximum overtime and the routing cost. If the problem has a solution that respects the constraints without needing overtime, then the objective in this case is to minimize the total traveled distance. Otherwise the objective will be to minimize the total overtime of the solution. The objectives of the problem can be formulated as follows:

$$\min \sum_{r\in S_T}\sum_{i\in I\cup J}\sum_{j\in J} d_{ij}\, x^r_{ij}$$

$$\min(OT_l)$$

Under the following constraints:

$$\sum_{i\in I\cup F}\sum_{\substack{j\in I^*\\ j\neq i}} x^r_{ij} \leq Q_r, \forall r\in S_T \tag{1}$$

$$\sum_{r\in S_k}\sum_{i\in I\cup F}\sum_{\substack{j\in I^*\\ j\neq i}} t_{ij}x^r_{ij} - T_l \leq \alpha T\ ,\ k\in K \tag{2}$$

$$\sum_{r\in S_T}\sum_{\substack{j\in I^*\\ j\neq i}} x^r_{ij} = 1, \forall i\in I\cup F^* \tag{3}$$

$$\sum_{\substack{i\in I\cup F\\ i\neq j}} x^r_{ij} = \sum_{\substack{i\in I^*\\ i\neq j}} x^r_{ji}\ , \forall\ j\in I^*, \forall\ r\in S_T \tag{4}$$

$$\sum_{r\in S_k}\sum_{i\in I\cup F}\sum_{\substack{j\in I^*\\ i\neq j}} x^r_{ij} \geq f_k, \forall k\in K \tag{5}$$

$$\sum_{r\in S_T}\sum_{\substack{i\in I\cup F\\ j\neq i}} x^r_{ij} = 0 \tag{6}$$

$$\sum_{i\in S}\sum_{\substack{j\in S\\ j\neq i}} x^r_{ij} \leq |S|-1, \forall r\in S_T, S\subset I, 2\leq|S|\leq |I|-1 \tag{7}$$

$$x^r_{ij}\in\{0,1\}, (i,j)\in I\cup F, r\in S_T \tag{8}$$

The first constraint (1) is made to comply with the remaining capacity of trucks in each tour. The second equation (2) restricts the overtime to a permitted maximum value. The third constraint (3) ensures that each customer is visited once and only once. The flow conservation constraint at the customer and the central depot level is carried out by (4); each truck that visits a customer must leave him after his delivery request and every truck that leaves the central depot must come back at the end of the working period. While the constraint (5) is made to ensure that trucks initially parked in a fictitious depot other than the central one must perform at least one tour. In this way we will be sure that these trucks return to the central depot at the end of the day. The sixth constraint (6) expresses that a fictitious depot can't be a destination. The constraint (7) prohibits the creation of sub-tours. Finally, the integrity constraints associated with decision variables are included in (8).

A study of Hassein and Rubinstein [22] has previously proved that an ordinary VRP is considered NP-hard when the set of customers contains more than 3 customers, so impossible to solve by an exact method. A meta-heuristic can therefore be used to solve this problem in order to find a good solution in a reduced time. The mathematical model just described is done to better describe the specificities of the problem and the objective functions.

IV. Resolution by the Hybrid ACS

A. Static Problem

To solve the problem that has just been described, we propose an approach based on the ACS. Firstly, we solve the problem of the beginning of the day. This problem, characterized by a single central depot and a homogeneous fleet, contains yesterday's requests arriving after the time T_f. The steps of our algorithm are described in Fig. 2.

First, we try to find a realizable solution that respect the normal time by minimizing only the first objective of the problem. If such solution isn't found we try to minimize the second objective without exceeding the maximal permitted overtime. Thus, the first step is affecting a positive value of τ_0 to each arc. Then, we place an ant on each customer. This allows for more diversification in the solutions found. The ants build, then, their tours. To move from one customer to another, the ant must select the clients that can visit without violating the constraints of the problem (remaining truck capacity and maximum return time). If no client responds to these constraints, the ant returns to the depot to begin a new tour. Otherwise, the ant chose the next customer to visit according to the following rule:

$$j = \begin{cases} \mathrm{argmax}_{u\in N_i^k}\left[\tau_{iu}(t).\eta_{iu}^{\beta}\right] & \text{if } q\leq q_0 \\ J & \text{if } q>q_0 \end{cases}$$

Where

q: A random variable uniformly distributed on [0,1]

q_0: A parameter in the interval [0,1]. It defines the balance diversification / intensification;

$\eta_{iu} = \frac{1}{d_{iu}}$: The visibility of the arc (i,u). It corresponds to the inverse of the distance d_{iu} between i and u.

$\tau_{iu}(t)$: The pheromone rate on arc (i,u) at the instant t.

β: The relative influence of the visibility.

N_i^k: The set of nodes that can be visited just after the position i by the ant.

$J\in N_i^k$: A randomly selected customer with probability:

$$p_{ij}^{k}(t)=\frac{\tau_{ij}(t).\eta_{ij}^{\beta}}{\sum_{u\in N_i^k(t)}\tau_{iu}(t).\eta_{iu}^{\beta}}$$

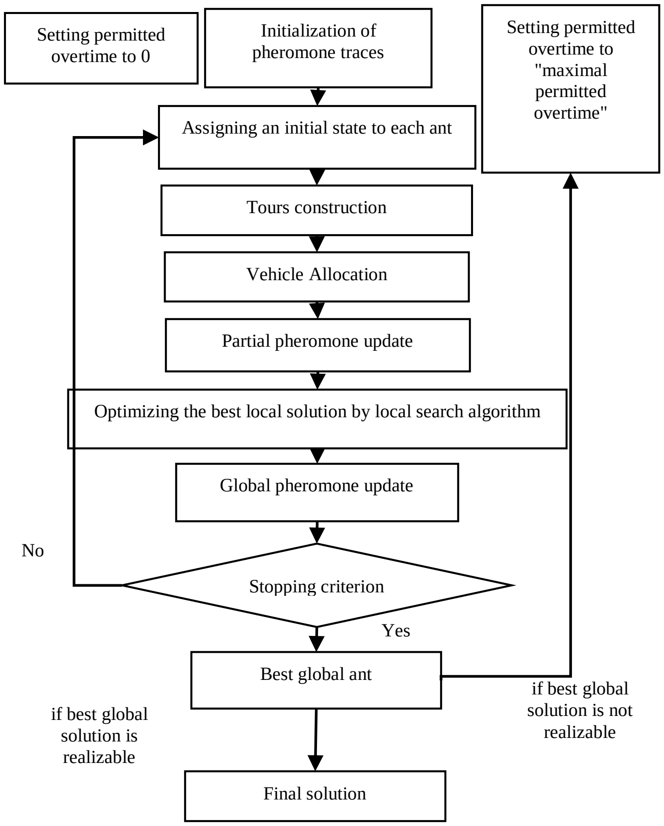

Fig. 2. Steps of the algorithm.

After building a solution by an ant k, we allocate its tours to the vehicles in order to calculate the total overtime. The procedure of vehicle allocation is described in Fig. 4. Then, a local update of pheromone is done on arcs it visited according to the following formula:

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\frac{\rho.\tau_0}{1+OV_k}$$

Where

ρ: The evaporation factor set to a positive value less than 1. This factor is intended to avoid the unlimited accumulation of the pheromone traces on the edges of the graph.

OV_k: The overtime performed by the ant k.

Before applying the global update, the best solutions of the iteration undergoes a phase of optimization by a local search intra and inter tour. The global update is then carried out as follow:

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho.\Delta\tau_{ij}(t)$$

where, the arcs (i,j) belong to the best solution S of the iteration. $\Delta\tau_{ij}=\frac{1}{L_s}$ and L_s: is the total time of S.

At the end of each iteration, the solution of the best ant undergoes an optimization phase by local search algorithm that combines an intra and inter tour optimization (Fig. 3). This procedure is the same used by Ayadi and Benadada [23]. In this phase we look to optimize, only, the total traveled distance.

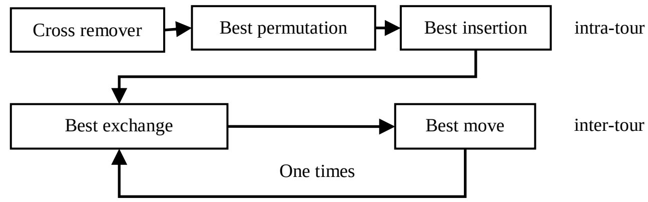

Fig. 3. Local search algorithm.

If (*ToursNumber=<TrukNumber*)
Assign a truck to each tour
End if;
Else
Rank the tours from the longest to the shortest;
Assign a truck to the *TruckNumber* tour;
While (there is "not assigned tours")
Rank the trucks according to the total length of the assigned tours in an ascending order;
Assign the first free tour to the first truck
End while
End else

Fig. 4. Vehicle allocation procedure (Static problem).

In a first step, the crossovers are eliminated and the possible permutations, that decreases traveled distance within each tour, are sought. Then, the optimal permutations carried out. Afterward, a reinsertion operator comes to see if there is any better location for a customer in its tour. The optimal reinsertion is carried out. These three operations (elimination of crossovers, permutation and reinsertion of customers) form the intra tour local search. Subsequently, we look for the possibility of exchanging two customers of two different tours. Indeed, the optimal exchange is carried. Next, we try to find the possibility of moving a customer from one tour to another without violating problem constraints. The optimal displacement is chosen to be performed. These last two operations form the local search inter tour. Finally, the intra tour search is performed once, while the inter tour search is performed twice.

B. Dynamic Problem

In the dynamic case, we have the same steps described in the static case. However, the details are different since heterogeneous fleets and many fictitious depots, beside the central one, characterize the dynamic problem. The initialization of pheromone rate is done according to the pheromone rate retention mechanism proposed by Montemanni et al. [21]:

$$\tau_{ij}=(1-\gamma_r)\tau_{ij}^{old}+\gamma_r\tau_0$$

where,

τ_{ij}^{old}: The final value of the pheromone level on the arc (i,j) of the previous problem.

γ_r: A positive factor less than 1, which regulates the conservation of pheromone level.

τ_0: The initial constant of pheromone. It initializes arcs corresponding to the new clients.

Once pheromone traces are initialized, we place an ant on each fictitious depot. Each ant creates its tours by visiting customers one after one until the capacity or the remaining time don't allow to insert a new customer, in this case it returns to the central depot.

The ant starts the new tour from the nearest unvisited fictitious depot or from the central depot if all fictitious depots are visited and there are still unvisited customers. The move, from the central depot to the fictitious depot is done without deposing pheromone.

The transition rule depends on a parameter$0\leq q_0\leq 1$, which defines the balance diversification / intensification. In order to choose the next customer j, the ant k located in the customer i use the following rule:

$$j=\begin{cases}\text{argmax}_{u\in N_i^k}\left[\tau_{iu}(t).\eta_{iu}^{\beta}\right] & \text{if } q\leq q_0\\ J & \text{if } q>q_0\end{cases}$$

With

q_0 : A parameter in [0,1] that determine the balance diversification/ intensification

q: A random variable uniformly distributed on $[0,1]$

$\eta_{iu}=\frac{1}{d_{iu}}$: The visibility of the arc (i,u). It corresponds to the inverse of the distance d_{iu} between i and u.

$\tau_{iu}(t)$: The pheromone rate on the arc (i,u) at the instant t.

β: The relative influence of the visibility.

N_i^k: The set of nodes that can be visited just after the position i by the ant k.

$j\in N_i^k$: A randomly selected customer with probability:

$$p_{ij}^k(t)=\frac{\tau_{ij}(t).\eta_{ij}^{\beta}}{\sum_{u\in N_i^k(t)}\tau_{iu}(t).\eta_{iu}^{\beta}}$$

The update of pheromone is divided into two levels: a local update and a global one. The first is done after building a solution by an ant k. We modify the pheromone of arcs visited by this ant according to the following formula:

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\frac{\rho.\tau_0}{1+OV_k}$$

OV_k: The overtime performed by the ant k after affecting vehicles to its tours. The procedure of vehicle affectation is detailed in Fig. 5.

At the end of each iteration and before applying the global update, the best solution found in this iteration undergoes a phase of optimization by a local search intra and inter tour. To this end, we use the local search algorithm of the static problem.

Assign the tours starting from fictitious depot to corresponding truck

Rank the remaining tours from the longest to the shortest;

While (there is "not assigned tour")

Rank the trucks according to the total length of the assigned tours in an ascending order;

Assign the *Firsttour* to the *Firsttruck*

Delete *Firsttour* from "not assigned tours"

end

Fig. 5. Vehicle allocation procedure (dynamic problem).

Finally, we must note that; if in a time interval of the problem we cannot find a solution without overtime, we will have one goal (minimize overtime) in the following time interval. In other words, we always try to minimize the total distance but once the algorithm ceases to find feasible solutions, the objective becomes the minimization of the overtime for all the rest of the planning period.

V. Computational Results

We have parameterized our algorithm according to the results found by Gambardella et al. [10] on the classical VRP. Thus, we set:q_0=0.9, β=1, ρ=0.1 Gambardella et al. [24] gave τ_0 the value of $\frac{1}{n*Cost(PI)}$ where Cost(PI) is the cost of a solution found by a greedy heuristic; In our case we fixed it in the static case to τ_0=0,1. This value was chosen by comparing its results with those of other values. In the dynamic case, we set $\tau_0=\frac{1}{n_d*Cost(PI)_p}$ where n_d is the number of clients of the current dynamic problem and $Cost(PI)_p$ is the cost of the previous problem; In this way, we reduce the execution time consumed by the greedy heuristic, especially since the current problem is similar to the previous one.

For the number of time intervals and the T_{co} we adopt the same parameter of Montemmanni et al. [17] (T_{co}=0.5*T and 25 time intervals). The algorithm stops after 200 iterations for all instances. It is coded in Java and executed on a machine with Intel Core i5 processor, 2.6 GHz, 8GB of RAM and Windows 8 as the operating system.

A. Static Case

The results of the static algorithm are already detailed in [25]. In this paragraph we give a reminder of these results.

We have tested our static algorithm on extracts data sets from Taillard et al. [10]. The problems used in these tests are: CMT1, CMT2, CMT3, CMT5, F134.

To avoid a zero in the denominator of η_{ij}, we have grouped customers who have the same coordinates in a single customer with a demand equal to the sum of the requests. Therefore, in F134, we have 132 customers and in CMT5, we have 199 customers. Table I presented our results referred to as OBM16, compared with those of Ayadi and Benadada [23] referred to, as AB13. While Table II presented the average of execution time of compared works, knowing that the AB13 time is normalized to our processor according to Geekbench benchmarks.

To qualify the solutions obtained, we use the Longest Trip Rate (LTR). It compares the time of the longest trip, which corresponds to the time of the truck that return the last to the depot, on the normal time horizon T:

$$LTR = \max_{k \in \{1..K\}} t(k) / T$$

With t(k) the time taken by the vehicle to make his tours.

TABLE I. NUMERICAL RESULTS OF THE STATIC CASE

Problem	T	M	AB13		OBM16	
			Distance	*LTR*	*Distance*	*LTR*
CMT1 z*=524,6	T_1	1	524,92	0,953	514.981	0.936
		2	536,56	0,998	523.768	0.991
		3	561,01	1,023	550.538	1.032
		4	547,1	1,027	536.329	1.050
	T_2	1	524,94	0,91	521.038	0.903
		2	530,79	1	512.242	0.941
		3	556,61	0,993	535.160	0.992
		4	546,43	0,985	538.937	1.003
CMT2 z*=835,2	T_1	1	835,77	0,953	842.325	0.960
		2	839,58	0,956	846.619	0.969
		3	836,18	0,964	872.351	1.043
		4	835,77	0,974	886.157	1.038
		5	839,71	0,996	845.569	1
		6	861,88	0,999	857.787	1.065
		7	885,57	1,034	869.800	1.119
	T_2	1	835,26	0,909	832.521	0.905
		2	835,26	0,912	856.500	0.934
		3	835,26	0,916	852.197	1.006
		4	835,77	0,959	875.851	0.963
		5	835,77	0,971	867.698	0.974
		6	842,28	0,997	870.636	1.003
		7	870,19	0,998	909.785	1.069
CMT3 z*=826,14	T_1	1	830	0,957	876.870	1.011
		2	828,74	0,97	865.569	1.005
		3	829,91	0,97	906.932	1.087
		4	828,74	0,982	870.609	1.027
		5	833,98	0,996	911.458	1.121
		6	867,72	1,008	986.338	1.198
	T_2	1	828,26	0,911	896.513	0.986
		2	828,26	0,913	891.445	0.991
		3	829,51	0,931	895.420	0.995
		4	829,54	0,969	871.22	0.999
		5	834,42	0,959	924.410	1.053
		6	836,56	0,994	921.07	1.082
F134 z*=1162,96	T_1	1	1166,96	0,956	1162.919	0.952
		2	1163,53	0,954	1142.409	0.945
		3	1162,97	0,964	1117.997	0.984
	T_2	1	1166,71	0,912	1137.630	0.889
		2	1163,53	0,911	1156.821	0.912
		3	1165,98	0,962	1143.229	0.950
CMT5 z*=1291,44	T_1	1	1313,22	0,968	1417.287	1.045
		2	1312,58	0,969	1479.293	1.092
		3	1313,21	0,969	1465.134	1.129
	T_2	1	1316,48	0,926	1469.505	1.034
		2	1380,38	0,972	1493.719	1.054
		3	1378,68	0,97	1467.871	1.091

TABLE II. AVERAGE OF EXECUTION TIMEIN SECONDS

Problem	Time AB13	Time OBM16
CMT1	39	11
CMT2	334	17
CMT3	3435	69
CMT5	5321	257
F134	6447	1246

B. Dynamic Casewithout Overtime

These results are already detailed in [26]. In this paragraph we give a reminder of these results.

In order to compare our algorithm with other works from the literature, we test it, first, on the benchmark of Kilby et al. [9]. To this end, we fix a single objective for the problem that is minimizing the total traveled distance. We consider five runs of the algorithm for each instance. Table III presents our results and those of Montemanni et al. [17]

TABLE III. NUMERICAL RESULTS OBTAINED BY OUR HYBRID ACS COMPARED TO THE ACS OF MONTEMANNI 2005

	Hybrid ACS			ACS Montemanni 2005		
	Best	*Averg*	*Time*	*Best*	*Averg*	*Time*
c50	685,81	708,64	0,20	631,3	681,86	4,10
c75	1077,39	1133,46	0,31	1009,36	1042,39	4,10
c100	1052,47	1078,13	1,93	973,26	1066,16	4,10
c100b	948,74	980,12	1,068	944,23	1023,6	4,10
c120	1279,32	1361,18	6,82	1416,45	1525,15	4,10
c150	1459,61	1535,27	4,564	1345,73	1455,5	4,10
c199	1827,81	1888,88	5,642	1771,04	1844,82	4,10
f71	316,6	321,05	4,336	311,18	358,69	4,10
f134	15675,0	16228,3	4,186	15135,5	16083,5	4,10
tai75a	1821,53	1916,60	0,398	1843,08	1945,2	4,10
tai75b	1555,78	1641,10	0,258	1535,43	1704,06	4,10
tai75c	1556,16	1632,73	0,662	1574,98	1653,58	4,10
tai75d	1514,16	1553,36	0,734	1472,35	1529	4,10
tai100	2225,45	2392,01	0,962	2375,92	2428,38	4,10
tai100	2384,79	2446,73	0,73	2283,97	2347,9	4,10
tai100	1662,35	1720,50	1,184	1562,3	1655,91	4,10
tai100	2008,47	2094,65	2,966	2008,13	2060,72	4,10
tai150	3368,62	3465,25	4,636	3644,78	3840,18	4,10
tai150	3082,21	3254,41	5,658	3166,88	3327,47	4,10
tai150	2842,62	2968,38	5,7	2811,48	3016,14	4,10
tai150	3253,3	3321,26	3,004	3058,87	3203,75	4,10
Total	51598,2	53642,2	55,95	50876,2	53794,0	86,24

The hybrid ACS outperforms the ACS Montemmani 2005 on 16 instances. The execution time is also lower than that of Montemanni 2005 with a percentage of 35% knowing that the Montemmani time presented in Table III is normalized to our processor according to Geekbench benchmarks.

C. Dynamic Case with Overtime

To test our algorithm, we have made some modifications on Kliby et al. [9] benchmark data set. Since, they are characterized by a big number of available trucks that is 50 for each instance, we can't have overtime with these instances.

We propose a data set for the DVRP with overtime using the twenty one basic Kilby problems; seven problems of Christofides et al. [27] (C), two of Fisher (F) [28] and twelve of Rochat and Taillard [29] (Thai). We use the same demands and truck capacities of the basic problems. Instances are generated by proposing several values of m (the number of available vehicle) and a restricted values of time horizon $T=[1,1*z^*/m]$, with z^* is the value of the best solution found by Rochat and Thaillard [29] for the VRP problem. The arrival time of customer requests is proportional to the arrival time set by Kilby et al. [9]. We set the maximum allowed overtime for each instance to one quarter of the normal time horizon. Time and distance are considered equivalent. The algorithm stops after 200 iterations for all instances. Table IV presents the value of T for m=1 and the size of each instance.

TABLE IV. DVRP INSTANCES

Problem	T	Size
CMT1	577	50
CMT2	919	75
CMT3	909	100
CMT4	1131	150
CMT5	1421	199
CMT11	114	120
CMT12	902	100
F71	266	71
F134	12979	134
Thai75a	1780	75
Thai75b	1479	75
Thai75c	1420	75
Thai75d	1502	75
Thai100a	2245	100
Thai100b	2134	100
Thai100c	1547	100
Thai100d	1739	100
Thai150a	3361	150
Thai150b	3000	150
Thai150c	2595	150
Thai150d	2910	150

For each instance, we tested m between 1 and 5. In Table V we write, just, the instances to which we found at least one feasible solution (which respects the maximum time plus the maximum permitted overtime). For each instance three runs of hybrid ACS are considered.

T is the normal maximum time while m represents the number of used vehicles. We consider the average of the total distance traveled and the overtime as well as the minimum of these two values obtained during the three executions. Time represent the average of the execution time in (min, second). A feasible solution is any solution that does not exceed the normal time plus the maximum overtime allowed. Instances denoted by * are instances in which only one feasible solution has been found, while those denoted by ** are instances in which two feasible solutions have been found. For both CMT4 and CMT5, we don't find any feasible solution. Therefore, they are not noted on the table. For the other instance, we found three feasible solutions.

TABLE V. NUMERICAL RESULTS OF DYNAMIC CASE WITH OVERTIME

Problem	T	m	Average		Best		Time
			Distance	*Over*	*Distance*	*Over*	
CMT1	577	1	629,95	52,95	625,38	48,38	0,66
	289	2	673,85	53,87	658,15	40,79	0,42
	192	3	672,24	36,28	652,52	26,37	0,37
	144	4*	674,09	36	674,09	36	0,3
CMT2	919	1	1082,73	163,73	992,85	73,85	0,44
	459	2	1079,1	81,5	1064,08	73,69	0,43
	306	3	1092,22	64,306	1069,12	58,87	0,38
	230	4	1060,74	48,81	1046,83	45,17	0,44
CMT3	909	1	1015,32	106,32	988,62	79,62	3,97
	454	2	1018,86	66,10	983,59	46,13	4,71
	303	3	1064,43	59,99	1015,42	40,79	3,17
	227	4	1047,00	44,77	1021,17	39,7	2,81
	182	5*	1114,19	44,77	1114,19	44,77	2,36
CMT11	1146	1**	1429,11	286,07	1426,16	280,16	20,37
CMT12	902	1	1015,01	113,01	1007,87	105,87	3,45
	451	2	1054,03	83,32	1049,76	75,67	4,25
	301	3	1038,55	53,44	982,75	29,51	2,45
	225	4	1012,49	49,17	998,63	45,81	2,56
F71	266	1**	312,63	46,63	304,84	38,84	4,55
	133	2	322,26	28,61	314,07	24,16	2,16
F134	12979	1	15474,94	2682,94	15085,66	2293,66	188,51
	640	2*	15007,73	1428.80	15007,73	1428,8	15,46
Thai75a	1780	1	1982,87	202,87	1876,06	96,06	0,49
	890	2	2035,67	134,89	1990,81	108	0,36
	593	3**	2097,32	119,66	2086,23	108	0,4
Thai75b	1479	1	1540,21	61,45	1478,29	0	0,92
	740	2	1579,59	52,25	1540,43	32,02	0,46
	493	3	1659,39	66,96	1613,21	53,84	1,015
	370	4**	1679,67	76,07	1658,09	64,73	0,46
Thai75c	1420	1	1568,35	148,35	1473,16	53,16	0,40
	710	2	1686,46	141,26	1651,26	118,7	0,32
	473	3*	1617,17	70,33	1617,17	70,33	0,44
	355	4*	1712,65	80,5	1712,65	80,5	0,53
Thai75d	1502	1	1761,29	259,29	1619,28	117,28	1,14
	751	2	1610,88	65,91	1572,01	35,31	1,19
	501	3	1738,87	90,37	1665,06	64,2	1,20
	375	4**	1806,92	93,37	1761,58	93,03	1,34
Thai100a	2245	1	2513,64	268,64	2487,71	242,71	1,48
	1123	2	2683,37	221,12	2621,6	192,8	1,32
	748	3	2531,46	106,9	2377,98	50,53	2
	561	4	2557,75	106,87	2462,64	75,99	1,85
	449	5**	2653,21	99,13	2626,6	92,09	2,13
Thai100b	2134	1	2522,88	388,88	2443,42	309,42	1,85
	1067	2	2633,28	253,25	2584,29	229,49	1,39
	711	3	2462,98	118,73	2388,63	95,14	1,46
	533	4**	2528,62	106,62	2497,6	96,23	2,24
Thai100c	1547	1	1633,16	86,16	1567,05	20,05	2,656
	773	2	1692,32	80,75	1505,88	0	3,95
	516	3	1820,90	95,44	1717,18	60,27	2,45
Thai100d	1739	1	2102,63	363,63	2017,65	278,65	3,766
	869	2*	2156,86	216,7	2156,86	216,7	3,05
Thai150a	3361	1	3881,88	520,88	3772,71	411,71	5,71
	1680	2	4065,08	360,43	3887,27	274,96	4,343
	1120	3	3904,39	206,4	3819,79	164,81	5,92
	840	4	3956,27	173,6	3934,36	172,36	7,64
Thai150b	3000	1**	3436,23	436,23	3426,57	426,57	13,34
	1500	2	3405,98	212,1	3281,88	155,48	8,92
	1000	3*	3612,9	230,52	3612,9	230,52	8,4
Thai150c	2595	1	3086,49	491,49	3006,83	411,83	12,11
	1297	2*	3190,45	312,39	3190,45	312,39	9,29
Thai150d	2910	1	3416,54	506,54	3323,09	413,09	11,17
	1455	2	3459,54	289,57	3431,4	271,49	5,70

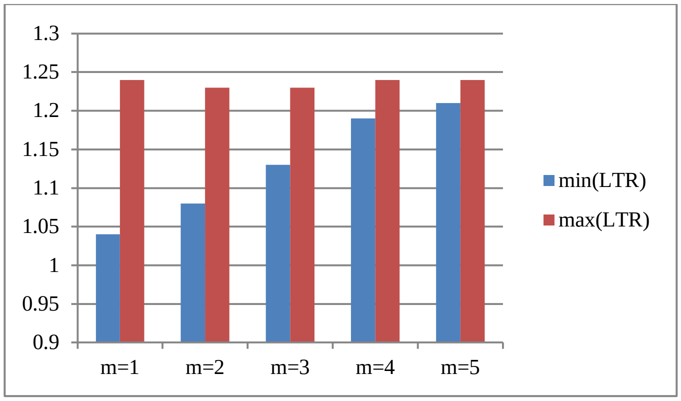

Fig. 6. Max and min of the LTR value depending on m.

The graph of Fig. 6 shows the minimum and maximum value of the LTR according to m.

A note to make is that the more the m increases, the more the minimal LTR increases. However, the maximum value is not significant compared to m. This can be justified by the fact that the maximum value of the LTR for m = 1 was obtained for the CMT11 instance which has no feasible solution for the other value of m. If we exclude this instance, the maximum value becomes 1.20 (Thai100d). Adopting this remark we can conclude that the LTR increases with m. This is also justifiable by the fact that we could not have a feasible solution for the big values of m.

Highlighted results are results where the minimum value of the total distance traveled and the minimum value of the overtime do not match the same solution. We have five instances where there is no such correspondence on 62instances. This gives a percentage of 3,1%. Thus in 96,9% a solution that minimizes the overtime minimizes also the distance. This being stated, we can conclude that the two objectives of our problem are proportional on 96,9%

D. Practice Use

For an industrialist what counts from all what is said is having an optimal or near optimal solution in a practical time. Our algorithm was able to give a near optimal solution for the DVRP with overtime in an execution time that does not exceed 6.50 (min, second) on average. To take advantage of these results, we are working on a software project that runs the same algorithm but with adaptable and comfortable interfaces for managers and industrialists while allowing to have results for a static problem in case of need.

VI. Conclusion

To conclude, this article introduces a new variant of the DVRP that is the multi-tours DVRP with overtime. The article gives a mathematical model of the problem with a hybrid ACS resolution. The results of the static and dynamic algorithm are competitive, comparatively to other works from literature. To test our algorithm on the dynamic case with overtime, we have proposed new benchmarks inspired from the very famous ones. Results have shown that the two objectives of the problem are proportional on 96,9%.

Several avenues exist for future works; a future goal is to work with another metaheuristics on the same problem to compare its results with those given by the hybrid ACS. Another research direction includes introducing new constraints such customer time windows, as well as considering stochastic data while providing the first planning of the day.

References

[1] Braekers K., Ramaekers K. & Van Nieuwenhuyse I., "The vehicle routing problem: State of the art classification and reveiw".Computers and Industrial Engineering, 99, 300-313, (2016)

[2] Eksioglu B., Volkan Vural A., & Reisman A., "The vehicle routing problem: A taxonomic review", Computers and Industrial Engineering, 57(4), 1472-1483, (2009)

[3] Crainic T.G., Gendreau M., & Potvin J.-Y., " Intelligent freight-transportation systems : assessment and the contribution of operations research", Transportation Research Part C: Emerging Technologies, 17 (6), 541–557, (2009)

[4] Psaraftis H., "A dynamic-programming solution to the single vehicle many-tomany immediate request dial-a-ride problem", Transportation Science, 14 (2), 130–154, (1980)

[5] Pillac V., Gendreau M., Guéret C., & Medaglia A.,"A review of dynamic vehicle routing problems", European Journal of Operational Research, 225(1), 1-11, (2013)

[6] Ferrucci F., Bock S., &Gendreau M. "A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods ". European Journal of Operational Research, 225(1), 130-141, (2013)

[7] Hu, T. "Evaluation Framework for Dynamic Vehicle Routing Strategies Under Real-Time Information". Transportation Research Record, 1774(1), 115-122, (2001)

[8] Tirado G, Hvattum L, Fagerholt K, Cordeau J. " Heuristics for dynamic and stochastic routing in industrial shipping ". Computers & Operations Research, 40(1), 253-263, (2013)

[9] Kilby P., Prosser P.and Shaw P. "Dynamic VRPs:A study of scenarios". APES-06-1998, University of Strathclyde, U.K, (1998)

[10] Taillard, E. D., Laporte, G.and Gendreau, M."Vehicle routing with multiple use of vehicles", Journal of Operationeal Research Society, 47(8), 1065-1070, (1996)

[11] Gendreau M., Guertin F., Potvin J.-Y., Taillard E., "Parallel tabu search for realtime vehicle routing and dispatching ", Transportation Science, 33 (4), 381–390, (1999)

[12] Chang M.S., Chen S., Hsueh C., "Real-time vehicle routing problem with time windows and simultaneous delivery/pickup demands", Journal of the Eastern Asia Society for Transportation Studies, 5, 2273–2286, (2003)

[13] Chen Z., Xu H., "Dynamic column generation for dynamic vehicle routing with time windows", Transportation Science, 40 (1), 74–88, (2006)

[14] Hanshar F.,& Beatrice M., "Dynamic vehicle routing using genetic algorithms", Applied intelligence, 27(1),89-99, (2007)

[15] Hong, L."An improved LNS algorithm for real-time vehicle routing problem with time windows", Computers & Operations Research, 39(2), 151-163, (2012)

[16] Khouadjia M., Sarasola B., Alba E., Jourdan L., and Talbi E."A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests", Applied Soft Computing, 12(4), 1426-1439, (2012)

[17] Montemanni R., Gambardella L.M., Rizzoli A.E., Donati A.V., "Ant colony system for a dynamic vehicle routing problem", Journal of Combinatorial Optimization,10 (4), 327–343, (2005)

[18] Hvattum L.M.,Lokketangen A.,Laporte G., "Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic ", Transportation Science, 40 (4), 421–438, (2006)

[19] Hemert J. and Poutré J. "Dynamic routing problems with fruitful regions: models and evolutionary computation", in Xin Y. Edmund K.

José A. Jim S. Juan J. John A. Jonathan E. Peter Tiňo P. Ata Kabán A. and Hans-Paul S (eds), Parallel Problem Solving from Nature - PPSN VIII, Berlin, Springer, 692-701, (2004)

[20] Schyns M., "An ant colony system for responsive dynamic vehicle routing", European Journal of Operational Research, 245(3),704-718, (2015)

[21] Gendreau M., Guertin F., Potvin J., & Séguin R., "Neighborhood search heuristic for dynamic vehicle dispatching problem with pick-ups and delivries", Transportation Research Part C (14), 157-174, (2006)

[22] Hassin R., Rubinstein S., "On the complexity of the k-customer vehicle routing problem", Operations Research Letters (33), 71-76, (2005)

[23] Ayadi R., Benadada Y.,"Memetic algorithm for a Multi-Objective vehicle routing problem with multiple trips", International Journal of Computer Science and Applications, (IJCSA),(10), 72– 91, (2013)

[24] Gambardella,L.M., Taillard, E. and Agazzi, G."MACS-VRPTW: a multipleant colony system for vehicle routing problems with time windows", editors New Ideas in Optimization, McGraw-Hill, 63-76, (1999)

[25] Ouaddi K., Benadada Y., Mhada F.Z., "Multi period dynamic vehicles routing problem: Literature review, modelization and resolution", The 3rd IEEE International Conference on Logistics Operations Management (GOL'16), DOI: 10.1109/GOL.2016. 7731706, (2016)

[26] Ouaddi K.., Benadada Y., Mhada F.Z., "Ant Colony system approach for dynamic vehicles routing problem multi tours", Proceedings of the 2nd international Conference on Big Data, Cloud and Applications, 109, (2017)

[27] Christofides, N., Mingozzi, A. and Toth, P.,"The vehicle routing problem", Combinatorial Optimization. Wiley, Chichester, 315-338, (1979)

[28] Fisher, M,"Optimal Solution of Vehicle Routing Problems Using Mimimum k-trees". Operations Research, 42(4), 626-646, (1994)

[29] Rochat, Y. and Taillard, E.,"Probabilistic Diversification and Intensification in Local Search for Vehicle Routing". Publication CRT-95–13, (1995)