

Memetic algorithm for multi-tours dynamic vehicle routing problem with overtime (MDVRPOT)**Khaoula Ouaddi^{a*}, Fatima-Zahra Mhada^a and Youssef Benadada^a**^a*Mohammed V University in Rabat ENSIAS, Morocco***CHRONICLE***Article history:*

Received February 5 2020
 Received in Revised Format
 February 28 2020
 Accepted March 5 2020
 Available online
 March 5 2020

Keywords:

DVRP
Memetic algorithm
Multi-tours
Overtime

ABSTRACT

After three decades of its introduction, the dynamic vehicle routing problem (DVRP) remains a fertile field for new studies. The technological evolution, which continues to progress day by day, has allowed better communication between different actors of this model and a more encouraging execution time. This encouraged researchers to investigate new variants of the DVRP and use more complicated algorithms for the resolution. Among these variants is the multi-tour DVRP (MTDVRP) with overtime (MTDVRPOT), which is the subject of this article. This paper proposes an approach with a memetic algorithm (MA). The results obtained in this study are better than those of the ant colony system (ACS) applied to the same problem and published in an earlier paper.

© 2020 by the authors; licensee Growing Science, Canada

1. Introduction

Dantzig and Ramser (1959) introduced a wide class of problems, the objective of which is to determine the best roads linking two or more points under certain constraints. Since then, many researchers have studied different possible variants of this problem. These variants are defined depending on the nature, constraints, and objectives of the problem. In the classical version, the objective is to determine the optimal set of routes that should deliver to a set of customers from a single depot with a limited fleet of trucks with limited capacities. These customers have varying demands for the same products. Golden et al. (1977) introduced the term ‘vehicle routing’ and called this variant ‘the generic vehicle routing problem’ (VRP). By imposing that the delivery to each customer is accomplished in a limited time window, a new VRP variant namely, VRP with time window was introduced by Solomon (1983). If there are several depots at which the trucks begin their tours, the multi-depot VRP is used, which was introduced by Kulkarni and Bhave (1985). If the fleet of trucks is inhomogeneous, the heterogenous VRP (HVRP) is employed, which was introduced by Golden et al. (1984). Thus, researchers have defined and treated a large number of VRPs according to the demands of the transport market and ability of the technological context (Eksioglu et al. (2009)).

* Corresponding author
 E-mail: khaoula.ouaddi@gmail.com (K. Ouaddi)

The evolution of communication technologies and geolocation systems simplify the real-time management of a fleet of vehicles. These tools make it possible to locate the current position of the vehicles, communicate with drivers in real time, receive new requests more easily, and even estimate the routing time according to the current state of the network (Crainic et al. (2009)). In this context, a new VRP variant has emerged and received growing interest: the dynamic VRP (DVRP). In this model, unlike in the static one (Psaraftis et al., 2016)), the routes planned at the beginning of the day are not definitive; thus, they can be changed if necessary. Several scenarios can impose this change; for instance, a customer wants to modify or cancel his or her order, the roads between two customers are blocked or very congested, or an urgent request arrived. Nowadays, DVRPs have become a class of different problems that are very important in modern transport models. However, DVRPs contain variants that have been less studied despite their importance in real life: for example, the multi-tour DVRP (MTDVRP). Despite the many researchers studying this variant in the static VRP version, research studies of the multi-trip DVRP are rare. In the multi-tour VRP (VRPM) version, the fleet is generally small. In this case, it is not always possible to serve all customers during the legal working time. As a result, many studies dealing with VRPMs in the static version allow vehicles to use overtime. The dynamic case is more complicated and requires more time to complete the service. Hence, overtime must also be tolerated in this case.

The topic of this study is the MTDVRP with overtime (MTDVRPOT), which is a multi-objective problem. The first objective is to minimize the total travelled distance and the second is to minimize the maximal performed overtime. Therefore, an approach based on the memetic algorithm (MA) is proposed. First, the competitiveness of this algorithm is demonstrated by testing it on known benchmarks from literature. Subsequently, the same algorithm is applied to the benchmark that was already proposed in a previous study with an ant colony system (ACS) (Ouaddi et al. (2018)) to compare the results.

The remainder of this paper is organized as follows: Section 2 presents a literature review, and Section 3 describes the investigated problem. The solution method is explained in the fourth section. Before the conclusion, the numerical results are presented in Section 5.

2. Literature Review

Each DVRP can be modelled in series of static VRPs. However, the characteristics of the problem change over time. In this article, the case of a capacitated MTDVRPOT is discussed. This problem can be decomposed into a series of static VRPs such that the first one is a capacitated VRPM with a homogeneous fleet and overtime, whereas the other problems comprise capacitated VRPMs with heterogeneous fleets, multi-depots, and overtime. Before talking about DVRP works, we present a brief review of this static VRP variants.

2.1 Static VRP

It should be noted that the capacity constraint is pervasive in most VRP studies. Nonetheless, Letchford and Eglese (1998) and Jaillet and Wagner (2008) presented counter-examples. In its simplest form, the static capacitated VRP (CVRP) contains a single depot with a homogeneous fleet. Several researchers have studied this problem variant (Taillard, 1993; Eydi & Javazi, 2012; Stewart & Golden, 1984; Gillett & Miller, 1974; Gendreau et al., 2008). The VRPM is another VRP with capacity; however, in this variant, vehicles are permitted to make several tours during the same day instead of only one. This variant was introduced for the first time by Fleischmann (1990). Many researchers have used the concept of overtime for the VRPM. However, it has been treated differently from one study to another. Taillard et al. (1996) proposed a method based on the Taboo search (TS) algorithm to solve the VRPM by allowing the use of overtime with a penalty payment. The objective is to minimize the total travelled distance and overtime penalty. To test their method, they proposed benchmarks with a restricted fleet and limited time. Several researchers have studied this problem in the form defined by Taillard et al. (1996) and used its benchmarks to validate their approaches. In addition to the total travelled distance, researchers who have based their tests on these benchmarks have used another comparison rate to compare infeasible solutions, i.e. the longest trip (LTR). The LTR reflects the degree of use of the maximal overtime, i.e. the longer

the LTR, the greater is the maximal overtime and vice versa. Brandão and Mercer (1998) were among the first to use Taillard benchmarks for their tests. They based their approach on the same metaheuristic TS and treated the same problem. By using the same benchmarks, Petch and Salhi (2003) tested a multi-phase heuristic problem based on a generalized gain algorithm (Yellow, 1970) and local search. The main difference between this problem and the previous ones is the objective function, which includes the minimization of the maximal performed overtime instead of the minimized total overtime. Furthermore, Olivera and Viera (2007) presented a formulation based on a path, which is a single feasible tour. For the resolution, they proposed an approach based on the TS and adaptive memory. Another mathematical formulation was proposed by Mingozzi et al. (2013). This time, the authors considered a list of possible schedules. A schedule reflects all the tours made by a single vehicle. The authors used an exact method based on the Lagrangian relaxation. They tested it on 52 instances from the Taillard et al. (1996) benchmarks with a maximum of 120 clients and solved 42 instances to optimality. Moreover, Ayadi & Benadada (2013) developed an MA for a VRPM that tolerates overtime. Therefore, they proposed two mathematical formulations (by arc and by path) and adopted the multi-line chromosome representation. The tests were also made based on the benchmarks of Taillard et al. (1996). Cheikh et al. (2015) used an approach based on the variable neighborhood search (VNS) algorithm to solve the VRPM. The algorithm could find a minimal number of non-feasible solutions as compared with those used in other studies in literature.

Since its introduction by Golden et al. (1984), the VRP with heterogeneous fleet (HVRP) has established a fertile research field. The vehicles do not have the same characteristics or are of different types. In general, its purpose is to determine the composition of the fleet and vehicle tours. There are two major HVRP types: the Fleet Size and Mix Vehicle Routing Problem (FSM) introduced by Golden et al. (1984) and characterized by an unlimited heterogenous fleet and the Heterogeneous Fixed Fleet Vehicle Routing Problem (HF) introduced by Taillard (1999) with a limited heterogenous fleet. Koç et al. (2016) provided a more detailed classification (with five essential variants) of this problem:

- 1) FSM with fixed or variable vehicle costs. This problem was introduced by Ferland & Michelon (1988) and noted FSM (F, V).
- 2) FSM with fixed vehicle costs, introduced by Golden et al. (1984) and noted FSM (F)
- 3) FSM with variable vehicle costs, introduced by Taillard (1999) and noted FSM (V)
- 4) HF with fixed or variable vehicle costs, introduced by Li et al. (2007) and noted HF (F, V)
- 5) HF with variable vehicle cost introduced by Taillard (1999) and noted HF (V)

To the best of the authors' knowledge, only three studies of the VRPM with heterogeneous fleet (HVRPM) have been conducted: Brandão & Mercer (1997), Prins (2002), and Seixas & Mendes (2013). The first researchers investigated a real case with a heterogeneous fleet. The distribution was done with the time windows constraint, and the possibility to reuse vehicles and overtime was tolerated. The multi-objective problem was solved via a three-phase TS algorithm. In addition, Prins (2002) proposed adaptations of some classical heuristics to deal with the HVRPM. The objective was to minimize the total travel time and number of used vehicles. Seixas & Mendes (2013) proposed a column generation algorithm, a constructive heuristic, and a TS-based approach to solve the HVRPM with time windows.

The multi-depot VRP (MDVRP) is another VRP variant. The objective of this problem is determining the routes of several vehicles from several depots to serve a set of customers at minimal transportation costs. This review focuses on the multi-tour MDVRP with a heterogeneous fleet. Several researchers have studied the MDVRP with a heterogeneous fleet; however, few have investigated the multi-tour MDVRP. Salhi & Sari (1997) studied the first case. Their problem is characterized by a fixed number of depots that have unlimited capacity and a limited number of vehicle types; however, each type has an unlimited number of vehicles. The vehicles have a known capacity and fixed or variable costs (FSM (F, V)). The objective is determining the composition of the fleet and vehicle tours at minimal cost. Salhi et al. (2014) investigated the same problem with a VNS algorithm. In addition, Bolaños et al. (2018) used a modified genetic algorithm, and Xu et al. (2012) studied the MDVRP with a heterogeneous fleet and

time window. Dondo & Cerdá (2007) and Irnich (2000) investigated MDVRPs with heterogeneous fleets, time windows, and pickup and delivery.

However, only Levy et al. (2014) studied the multi-tour MDVRP (heterogeneous fleet). The objective is determining the tours that satisfy the fuel delivery constraints while allowing vehicles to be replenished from any depot when required. The authors used two methods: the variable neighborhood descent (VND), which provides better results than the second method, the VNS algorithm.

2. Dynamic VRP

According to Pillac et al. (2013), researchers have treated the DVRP based on four broad perspectives.

	Periodic	Continuous	Stochastic	Deterministic	Problem type	Resolution method
Psaraftis (1980)	√			√	DR	O
Benyahia & Potvin (1998)		√		√	IC	GA
Gendreau et al. (1999)		√		√	IC-TW-PD	TS
Ichoua et al. (2000)		√		√	IC-TW-PD	TS
(Ichoua et al., 2003)		√		√	IC-TW-TM-PD	TS
Haghani & Jung (2005)		√		√	C-HF-PD-TW-TM	GA
Montemanni et al. (2005)	√			√	C	AS
Chen & Xu (2006)	√			√	C-TW	O
Hvattum et al. (2006)	√		√		C-PD	O
Gendreau et al. (2006)		√		√	IC-O-TW-PD	NS
Hanshar & Ombuki-Berman (2007)	√			√	C	GA, TS
Cheung et al. (2008)		√		√	C	GA
Beaudry et al. (2010)		√		√	DR	TS-O
Hemert & Poutré, (2010)		√	√		C-PD	GA
Hong (2012)	√			√	C-TW	NS
Azi et al. (2012)	√		√		C-TW-MT	NS
Khouadjia et al. (2012)	√			√	C	PS
Albareda-Sambola et al. (2014)	√		√		C	NS
Lin et al. (2014)		√		√	C-TW-PD	NS
Barkaoui et al. (2015)	√			√	C-TM-TW-PD	GA
Schyns (2015)		√	√		C-HF-TW	ACS
Mańdziuk & Żychowski (2016)	√			√	C	MA
Ouaddi et al. (2018)	√			√	C-MT-O	ACS

The first (the deterministic perspective) considers only known requests and responds reactively to dynamic ones (e.g. Montemanni et al., 2005; Ouaddi et al., 2018; Khouadjia et al., 2012). The second takes advantage of the forecasts to provide a prior plan (e.g. Schyns, 2015; Albareda-Sambola et al., 2014), which is a stochastic approach. Other approaches perform the optimization throughout the day and maintain data on best solutions in an adaptive memory (Taillard et al., 2001). Whenever new request appears, this data is used to update the routing plan. They call this method ‘continuous optimization’ plan (e.g. Gendreau et al., 1999; Ichoua et al., 2000; Ichoua et al., 2003). In addition, other approaches decompose the planning period into time intervals. Thus, re-optimization is periodically made at the beginning or end of each interval (e.g. Khouadjia et al., 2012; Mańdziuk & Żychowski, 2016). This DVRP model was first introduced by Kilby et al. (1998) while presenting a test benchmark for the capacitated DVRP. Furthermore, the DVRP can handle variations in the travel time. This case occurs in urban areas where the network travel time is difficult to predict because of the congestion, in particular during peak hours (Lin et al. (2014)). Table 1 presents a summary of the various DVRP studies. We adopt these notations to describe problem types: Capacitated (C), Incapacitated (IC), Time Windows (TW), Pick-up and Delivery (PD), Dial-a-ride (DR), Multi Tour (MT), Travel time (TM), Overtime toleration (O) and Heterogeneous Fleet (HF). To describe resolution method, we use: Taboo Search (TS), Genetic Algorithm (GA), Memetic algorithm (MA), Particle Swarm (PS), Ant Colony System (ACS), Neighborhood search (NS), Other (O). There are only two scientific articles (Azi et al., 2012; Ouaddi et al., 2018) on the MTDVRP. The idea of the first is to fix the routes of vehicles at the beginning. For dynamic queries, the decision to consider only their acceptance or rejection depends on their proximity to the already planned routes. To solve this problem, the authors developed an adaptation of the large neighborhood search algorithm, which exploits the forecasts of requests to create a population of likely future scenarios based on which the routes will be created. The second paper presents a previous study

of the same problem with an ACS algorithm (Ouaddi et al., 2018). The concept of overtime is barely present in the DVRP literature. To the best of the authors' knowledge, apart from their previous study (Ouaddi et al., 2018), there is only one study that tolerated the use of overtime. In this study, a DVRP with pickups, delivery, and time window without capacity constraint is investigated. Thus, new studies that cover this DVRP variant with other resolution methods are required to determine the best solutions. The contribution of this article is an approach that is based on an MA for the MTDVRPOT.

3. Problem Description

In this study, the case of a transportation company that delivers to a set of customers from a single depot is investigated. Customers can make their requests during the day, and the company must answer the maximal number of requests the same day. After the time limit T_f , new requests will be postponed to the next day. The workday is divided into several equal time periods. Thus, incoming requests during a period are gathered to insert them together at the end of this period. At the beginning of the day, the trucks start at the central depot with a routing plan containing the undelivered requests of the previous day. At the end of the first period, the current positions of the trucks are marked. If the truck has delivered to the customer or is in route for delivery, this customer will be considered a fictitious depot in the following period. An optimization is conducted to determine the new routes that will contain the new and remaining customers from the previous period. Therefore, a multi-tour model is proposed, which tolerates the use of overtime. Thus, the trucks can make multiple tours during the same working day and use extra time (overtime) when necessary. The objective is minimizing the total travelled distance and maximal performed overtime. This problem is multi-objective. Therefore, the total travelled distance with zero overtime is minimized first. If no feasible solution can be found, the overtime is minimized. The mathematical model of each sub-problem (problem of a single period) was already presented in detail in a previous paper (Ouaddi et al., 2018). This paper provides a summary.

We use the following notations:

O: Index of central depot

F: Set of depots

I: Set of customers to be served (fictitious depots are not included)

K: Number of trucks

n: Maximum number of tours for a truck

d_{ij} : Distance between customer i and customer j

t_{ij} : Travel time between customer i and customer j

Q_k : Remaining capacity of the truck k

Q: The initial capacity of trucks

T_p : Normal travel time remaining for the period p

T: Length of the working day

αT : Maximal legal overtime

q_i : Quantity requested by the customer i.

$$f_k = \begin{cases} 1, & \text{if the truck k is initially stationed in a fictitious depot} \\ 0, & \text{else} \end{cases} \quad (1)$$

$$x_{ij}^r = \begin{cases} 1, & \text{if the customer j was visited after the customer i during the tour r} \\ 0, & \text{else} \end{cases} \quad (2)$$

To distinguish the tours of each truck, the set of n possible tours served by the truck k is defined as S_k ;

$$S_k = \{k + qK, q = 0..n-1\} \quad (3)$$

And

$$S_T = \bigcup_{k \in K} S_k \quad (4)$$

Q^r is the available quantity for the tour r . If this tour starts at the central depot, Q^r is equal to the initial capacity of the vehicle (Q). If the tour starts at a fictitious depot, Q^r is equal to the remaining capacity of the vehicle at the fictitious depot at the beginning of the current period. In this case, r is the first tour of this vehicle ($r \in \{1, \dots, K\}$). Thus,

$$Q^r = \begin{cases} Q_k & \text{if } r \in \{1, \dots, K\} \\ Q & \text{else} \end{cases} \quad (5)$$

The objective function is:

$$\min \sum_{r \in S_T} \sum_{i \in IU} \sum_{j \in J} d_{ij} x_{ij}^r \quad (6)$$

$$\min(OT^P) \quad (7)$$

where

$$OT^P = \max_{1 \leq k \leq K} (OT_k) \quad (8)$$

and

$$OT_k = \max \left(0, \sum_{r \in S_k} \sum_{i \in IU} \sum_{\substack{j \in J \\ j \neq i}} t_{ij} x_{ij}^r - T_p \right) \quad (9)$$

OT^P and OT_k refer to the maximal overtime that should be minimized and the overtime of the truck indexed by k . The constraints of this problem are modelled as follows:

$$\sum_{i \in IU \cup F} \sum_{\substack{j \in (IU \cup \{0\}) \\ j \neq i}} q_i x_{ij}^r \leq Q^r, \quad \forall r \in S_T \quad (10)$$

$$\sum_{r \in S_k} \sum_{i \in IU \cup F} \sum_{\substack{j \in (IU \cup \{0\}) \\ j \neq i}} t_{ij} x_{ij}^r - T_p \leq \alpha T, \quad \forall k \in \{1, \dots, K\} \quad (11)$$

$$\sum_{r \in S_T} \sum_{\substack{j \in (IU \cup \{0\}) \\ j \neq i}} x_{ij}^r = 1, \quad \forall i \in IU(F \setminus \{0\}) \quad (12)$$

$$\sum_{\substack{i \in IU \cup F \\ i \neq j}} x_{ij}^r = \sum_{\substack{i \in (IU \cup \{0\}) \\ i \neq j}} x_{ji}^r, \quad \forall j \in (IU \cup \{0\}), \quad \forall r \in S_T \quad (13)$$

$$\sum_{r \in S_k} \sum_{i \in IU \cup F} \sum_{\substack{j \in (IU \cup \{0\}) \\ i \neq j}} x_{ij}^r \geq f_k, \quad \forall k \in \{1, \dots, K\} \quad (14)$$

$$\sum_{r \in S_T} \sum_{\substack{i \in I \cup F \\ j \neq i}} x_{ij}^r = 0 \quad \forall j \in (F \setminus \{O\}) \quad (15)$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} x_{ij}^r \leq |S| - 1, \quad \forall r \in S_T, S \subset I, 2 \leq |S| \leq |I| - 1 \quad (16)$$

$$x_{ij}^r \in \{0, 1\}, (i, j) \in I \cup F, \quad \forall r \in S_T \quad (17)$$

Constraint (10) is made to respect the remaining capacity of the trucks in each sub-problem. Eq. (11) restricts the overtime to a permitted maximum. Each customer is visited once, which is ensured by constraint (12). Eq. (13) represents the flow conservation constraint at the customer and central depot; each truck that visits a customer must leave him or her after his delivery request, and every truck that leaves the central depot must come back at the end of the working period. Constraint (14) ensures that the trucks initially parked in a fictitious depot other than the central one performs at least one tour. Consequently, these trucks return to the central depot at the end of the day. Constraint (15) states that a fictitious depot cannot be a destination, and constraint (16) prohibits the creation of sub-tours. Finally, the integrity constraints associated with decision variables are included in (17).

4. Memetic Algorithm

The genetic algorithm is a well-known metaheuristic and widely used to solve VRPs. It can be combined with other heuristics to obtain better results within a reasonable time. For instance, MAs use the same solving process of the genetic algorithm in combination with local search operators. In this study, an MA is used, the steps of which are explained in detail in the remainder of this section.

First, the initial population is created based on the insertion heuristic. If the number of solutions generated by this heuristic is below the number required to complete the initial population, the random insertion heuristic is used to generate new individuals. The generated tours are allocated to the available vehicles depending on the capacity and remaining time period of the work day of each vehicle. The individuals of the initial population are treated with a local search heuristic to improve their quality. Subsequently, two individuals are selected from the initial population. Depending on the probability of crossover (P_c), it is decided whether these individuals undergo the crossover step or go directly to the mutation stage. Depending on the number of available vehicles, two crossover types are employed. If this number is over one, Several Vehicle Crossover is applied, or else, One Vehicle Crossover is used. If the crossover of one vehicle is terminated, and the resulting individuals no longer respect the constraints, a correction procedure and the local search heuristic are applied before the mutation. The mutation is applied with mutation probability. The resulting individuals are corrected with the correction procedure, and the tours are reassigned to the vehicles and improved again by the local search heuristic. For the static problem of the first time slot, the algorithm of Ayadi and Benadada (2013) is applied. The same algorithm is used for the crossover, mutation, and correction procedure of the dynamic algorithm. In addition, the methods used by Ouaddi et al. (2018) are adopted for local research and vehicle affectation.

4.1 Chromosome conception and population initialization

Owing to the MTDVRP, chromosomes of the population should effectively reflect the essential information of a multi-tour and multi-depot solution, i.e. the order of client visits in each tour, the fictitious depot at which the tour starts, and the assignment of the tours to the trucks. For this reason, multi-line coding is adopted. Each line represents the complete journey of a single truck for the remaining period of the current day. A trip is the set of tours made by the same vehicle. The first box in the line represents the depot at which the truck begins its journey. If the truck starts at the central depot, this box exhibits a 0 . Otherwise, this box is filled by the index f of the fictitious depot. In addition, a delimiter (marked 0) denotes the central depot to separate the tours. Fig. 1 shows a coding example of the solution for $K = 2$.

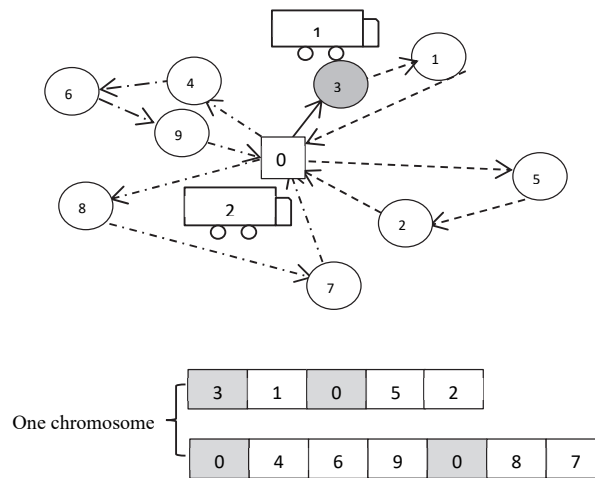


Fig. 1. Chromosome representation

4.1.1 Insertion Heuristic

The principle of this heuristic is to determine all feasible insertions of new customers in the previous period remaining solution by inserting a customer each time. Thus, all feasible insertions of the first dynamic customer are determined, and in each solution based on these insertions, the feasible insertions of the second and following dynamic customers are determined. By applying the same procedure iteratively to the rest of the dynamic customers, a set of feasible solutions is constructed. This can be used as the initial population for the memetic algorithm. If the number of solutions exceeds the number required to complete the initial population, the individuals of the initial population are randomly selected from the determined solutions (Algorithm 1).

Algorithm 1

$C = \{c_1, c_2, \dots, c_n\}$ list of new customers

$S = \{t_1, t_2, \dots, t_m\}$ Tours of the remaining solution from previous period

Find all possible insertions of c_1 in S . S_1 is the union of resulting solutions

For each i from 2 to n

Find all possible insertions of c_i in each solution of S_{i-1}

End for

Return S_n

4.1.2 Random Insertion Heuristic

This heuristic is only used if the number of solutions generated by the insertion heuristic is below the number required to complete the initial population. In this method, a tour starting at each fictitious depot is created. Then, customers are randomly selected. For each selected customer, the first feasible insertion in one of the previously created tours is determined. If no insertion is possible, a new tour, starting at the central depot and visiting this customer, is created. The algorithm stops when all customers have been inserted (Algorithm 2).

Algorithm 2

 $C = \{c_1, c_2, \dots, c_n\}$ set of all not served customers $D = \{d_1, d_2, \dots, d_m\}$ set of fictitious depotsFrom each fictitious depot of D we create an empty tour returning to central depot $T = \{t_1, t_2, \dots, t_m\}$ set of created toursFor each c_i de C ,If we found t_j from T where we can insert c_i $t'_j = t_j \cup c_i$ $T = (T|t_j) \cup t'_j$ $C = C|c_i$

Else

Create tour t_{m+1} starting from the central depot, visiting c_i and going back to central depot $T = T \cup t_{m+1}$ $m=m+1$

End if

End for

Return T

4.2 Crossover

The Crossover is a crucial operation in a genetic algorithm. This operator combines two chromosomes called 'parents' to produce children. In this study, two crossover methods are used, which depend on the number of available vehicles. The first is applied for a single vehicle (Cross over One Vehicle), and the second is used for more vehicles (Cross over Several Vehicles).

4.2.1 Cross Over Several Vehicles

For several vehicles, one of the parents is considered the receiver and the second the donor. The crossover is applied to generate the first child. Subsequently, the roles are reversed to generate the second child. The tours of the vehicle that consumes the most time are removed from the receiving parent, while the first box that represents the depot is retained if it is fictitious. Hence, the first part of the child is obtained. At the donor parent level, only customers that have been deleted from the receiver are retained. Consequently, the boxes indicating fictitious depots are marked by 0, and the second part of the child is obtained. The two parts are combined to form the new child, which will be treated with the local search heuristic. The example in Fig. 2 illustrates this crossover.

4.2.2 Cross Over One Vehicle

In this case, each chromosome is represented by a single line. The first case presents a fictitious depot in which the vehicle is initially stationed. Other cases exhibit the visited customers. The tours are separated by case 0, and two random tours are chosen. The first tour belongs to the first parent and the second to the second parent. Customers of the first tour replace those of the second in the second parent and customers of the second tour replace those of the first in the first parent while keeping the same fictitious depot of the parent. Each of the generated children receives the correction procedure described in Section 4.4. The example of Fig. 3 illustrates this crossover.

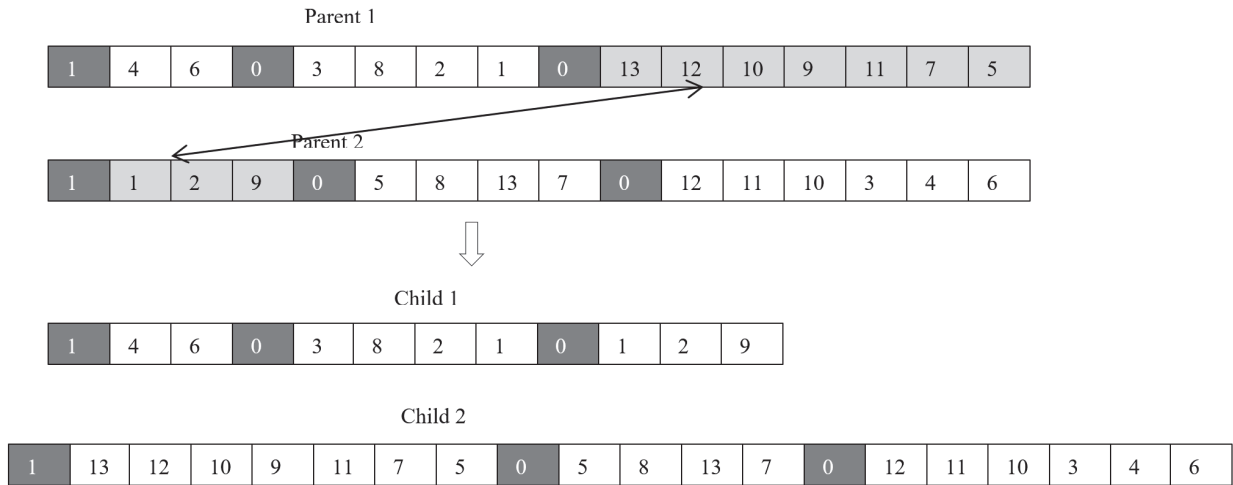


Fig. 2. One Vehicle Cross over

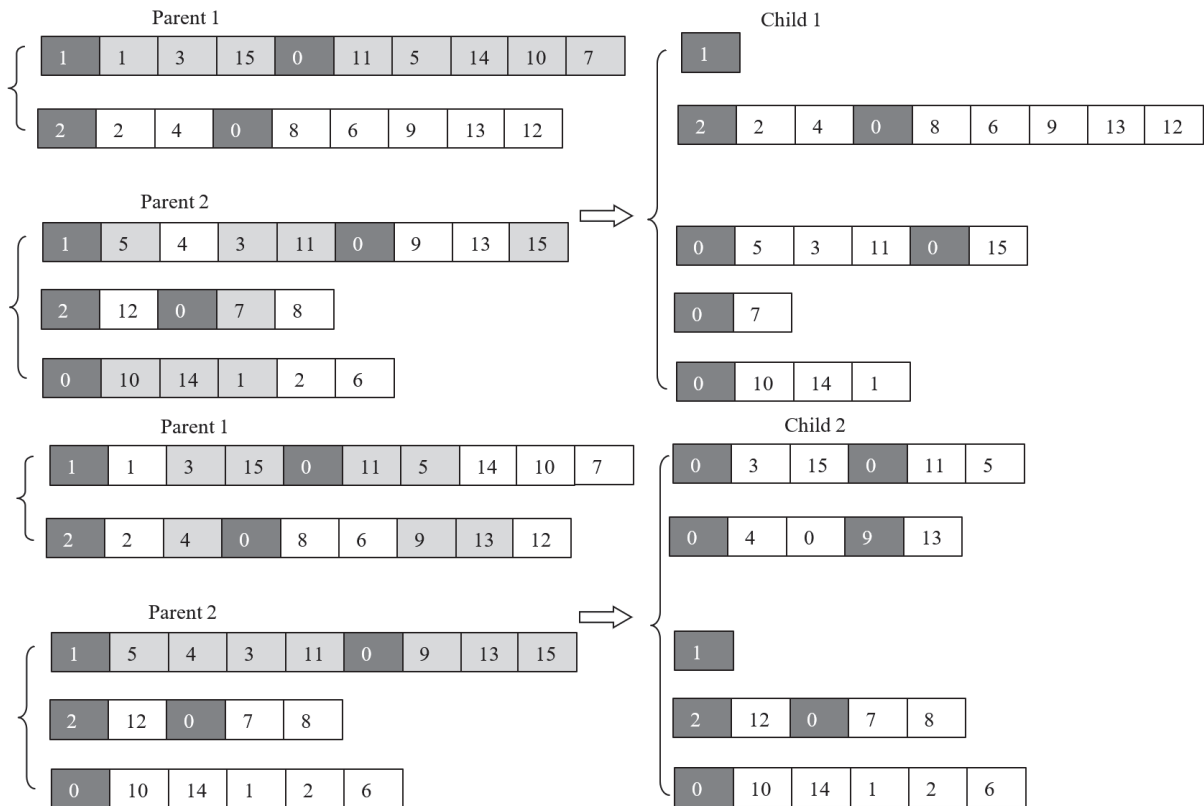


Fig. 3. Several Vehicles Cross over

4.3 Mutation

Two random operations are used in the mutation phase to obtain a population with new characteristics and thus extend the search area:

Random exchange: this operator chooses two customers randomly from two different tours and exchanges their positions.

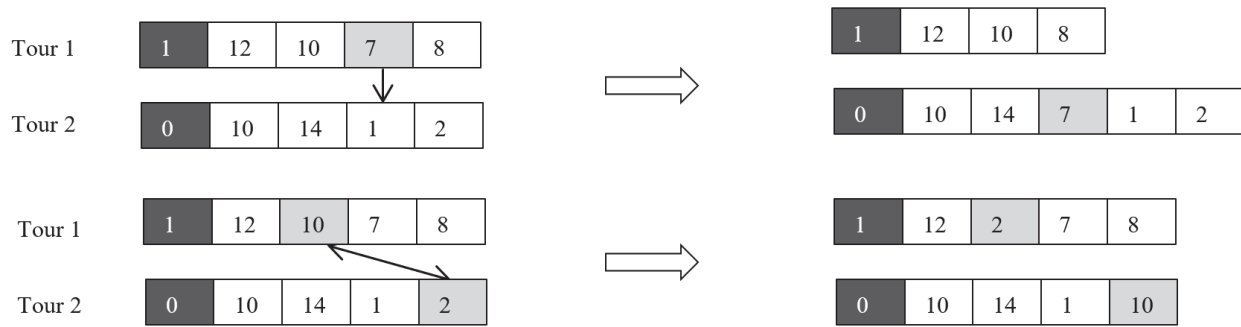


Fig. 4. Mutation operator

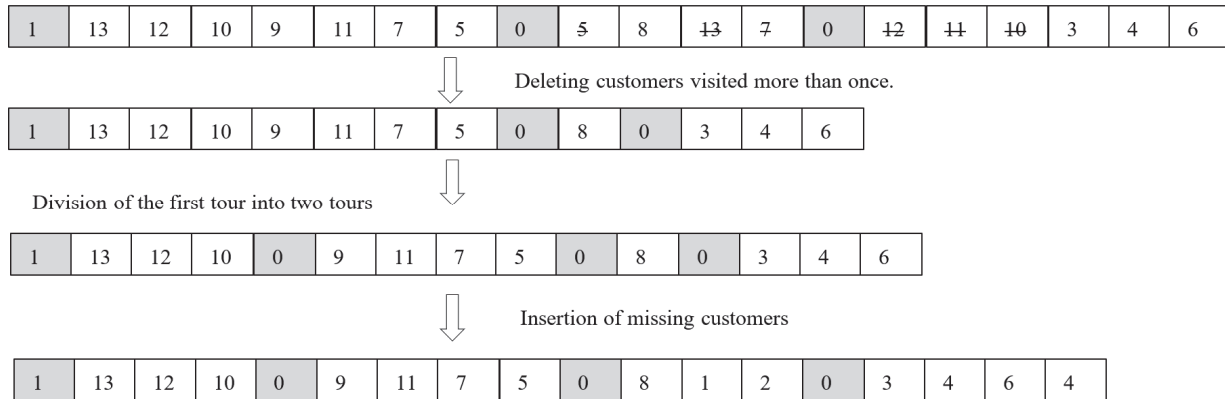


Fig. 5. Correction procedure

Random insertion: this operator chooses two tours randomly. A customer is selected from the first tour and inserted into the second at a randomly chosen position. The example in Figure 4 represents the mutation operator.

4.4 Correction Procedure

This procedure is applied to the individuals resulting from the one-vehicle crossover or mutation phase. First, the redundant customers are removed, and the order of the visits to the customers in each tour is improved to minimize the routing time. If the routing time or quantity of tours exceeds the time limit or capacity of the vehicle, it will be divided into two or more tours. If some customers are not included in any tour, they are inserted into one of the existing tours. If no insertion is feasible for a customer, a new dedicated tour starting at the central depot is created. Finally, the Clarke and Wright algorithm is used to combine short tours. Fig. 5 shows the correction procedure applied to the children of the one-vehicle crossover in Fig. 3.

4.5 Evaluation

To compare the solutions, their objective function is determined. Therefore, the total travelled distance and maximal overtime performed by each individual are determined. The individuals are compared as follows: If one or both individuals exceeds/exceed the time horizon, the one that achieves minimal overtime is the best. If both individuals respect the time constraint, the best individual is that with minimal travel costs.

5. Computational Results

First, the results obtained by applying the MA to the instances presented by Kilby et al. (1998) are presented to assess its performance based on the classical DVRP. Subsequently, the results obtained by applying the MA to the instances of Ouaddi et al. (2018) and those obtained by the ACS of the same study (Ouaddi et al., 2018) are compared.

5.1 Setting of Memetic Algorithm

The size of the population is set to 100 for the static part of the problem (first period of the day) and to 20 for the dynamic part of the problems (for the rest of the day). For the static part, the population size (100) allows a better exploration of the search area because the execution time of this phase is not strictly limited. For the remaining dynamic problems, generating feasible and different individuals for the initial population takes more time. If the size of the initial population is considerably large, the speed of the algorithm may be decreased. Preliminary tests have shown that the number 20 allows a balance between the execution time and exploration of the search area. In addition, preliminary tests with several values for the mutation and crossover probabilities are performed. Results of Table 1 (for the C50 instance) show that $P_m=0.4$ for mutation probability and $P_c=0.9$ for cross over probability are the most appropriate.

Table 1
Preliminary tests on the probability of cross over and mutations

	P_m	P_c	P_m	P_c	P_m	P_c	P_m	P_c
	0.1	0.9	0.1	0.7	0.4	0.9	0.4	0.7
1	648,8		627,64		563,46		618,48	
2	641,89		619,12		616,59		616,22	
3	636,05		609,23		617,27		630,4	
4	616,39		620,23		593,76		624,53	
5	652,57		621,11		624,08		611,67	
Min	616,39		609,23		563,46		611,67	
Average	639,14		619,466		603,032		620,26	

The algorithm stops after 200 iterations or if it realizes 100 iterations without any improvement in the objective function. The algorithm is coded in Java and executed on a machine with Intel Core i7 for the classical DVRP tests. In addition, the same machine used by Ouaddi et al. (2018) (Core i5 processor) is used to perform the MTDVRPOT tests.

5.2 Results of Classical DVRP

As shown in Table 2, the MA provides better results than Hanshar's genetic algorithm for 13 instances (62%) and Montemanni's algorithm for 19 instances (90%). In addition, the algorithm does not exceed 8.5 min.

Table 2
Results of MA on classical DVRP compared to Hanshar 2007 and Montemmani 2005

	MA	Hanshar 2007	Montemmani 2005		MA	Hanshar 2007	Montemmani 2005
c50	563,46	570,89	631,3	tai75c	1473,97	1440,54	1574,98
c75	977,73	981,57	1009,36	tai75d	1391,19	1399,83	1472,35
c100	967,02	961,1	973,26	tai100a	2208,31	2232,71	2375,92
c100b	889,88	881,92	944,23	tai100b	2211,63	2147,7	2283,97
c120	1258,19	1303,59	1416,45	tai100c	1486,75	1541,28	1562,3
c150	1282,55	1348,88	1345,73	tai100d	1767,57	1834,6	2008,13
c199	1623,09	1654,51	1771,04	tai150a	3327,58	3328,85	3644,78
f71	316,13	301,79	311,18	tai150b	3057,77	2933,4	3166,88
fl34	1381,27	1552,88	1513,55	tai150c	2668,34	2612,68	2811,48
tai75a	1776,56	1782,91	1843,08	tai150d	3061,13	2950,61	3058,87
tai75b	1434,1	1464,56	1535,43				

The GAP between a solution s and solution s' allows to measure the relative difference by comparing their objective functions.

$$GAP(s/s') = 100 * \left(\frac{f(s)}{f(s')} - 1 \right)$$

In this case, the objective is minimizing the total travelled distance. Thus, the higher the GAP, the better is the quality of solution s' compared to that of s . The lower the GAP, the better is the solution s compared to that of s' . Later, $f(s)$ will be replaced by the value of the solution determined by the MA, while $f(s')$ will be replaced by the value of the solution determined by the second algorithm. In this study part, the GAP is used to compare the solutions found by OBM with those found by the methods of Hanshar 2007 and Montemanni 2005. Table 3 presents the maximal and minimal GAP related to these two algorithms. The maximal GAP does not exceed 5% in the best case, while the minimal GAP reaches up to approximately -12%. Figure 7 in Appendix B describes the tours of the solution that exhibits a lower GAP compared to that of Hanshar 2007. It is a solution of the f134 instance.

Table 3

GAP of memetic algorithm related to Hanshar 2007 and Montemanni 2005

GAP	Hanshar 2007	Montemanni 2005
Max	4,75%	1,59%
Min	-11,05%	-11,97%

5.3 Results of MTDVRPOT

In this study part, the data set of Ouaddi et al. (2018) is used for the MTDVRPOT. The data set is inspired by the 21 problem of Kilby et al. (1998). A similar approach to that of Taillard et al. (1996) is adopted to generate the instances, and the same demands and truck capacities of the basic problems are used. In addition, the instances are generated by proposing several values of m (number of available vehicles) and restricted values of the time horizon $T=[1,1*z^*/m]$, where z^* is the value of the best solution determined by Rochat & Taillard (1995) for the classical static VRP. The arrival time of the customer requests is proportional to the arrival time proposed by Kilby et al. (1998). Moreover, the maximal allowed overtime for each instance is set to one quarter of the normal time horizon, and the time and distance are considered equivalent. For each instance, m between 1 and 5 is applied. To distinguish the problems of this data set from the original problems (Kilby et al., 1998), the first letter of each problem is capitalized. For example, instance C100-3 corresponds to the instance with $m = 3$ of the problem C100. A solution is feasible if the realized overtime does not exceed the maximal allowed overtime. In this section, the results obtained by the MA are compared with those obtained by the ACS (Ouaddi et al., 2018). Three runs of both algorithms are considered for each instance. Table 4 provides the number of feasible solutions found by each algorithm, and table 7 in Appendix A provides the detailed computational results.

Table 4

Number of feasible solutions found by MA and ACS algorithm

	MA	ACS		MA	ACS
C50	4	4	Tai75c	3	4
C75	5	4	Tai75d	5	4
C100	5	5	Tai100a	5	5
C100b	5	4	Tai100b	5	4
C120	2	1	Tai100c	5	3
C150	5	0	Tai100d	5	2
C199	5	0	Tai150a	5	4
F71	3	2	Tai150b	5	3
F134	4	2	Tai150c	5	2
Tai75a	5	3	Tai150d	5	2
Tai75b	5	4			

The MA obtains feasible solutions for 96 of 105 instances, and the ACS obtains only 62 feasible solutions. Amongst all the instances, there is only one for which the ACS algorithm can determine a solution whereas the MA cannot (Thai75c-4). Furthermore, the MA provides better results than the ACS algorithm for both objectives of the problem with 94 instances, and the ACS provides better results than the MA for only two instances (C100 -2, Thai75c -1). For C100-2, the solution provided by the ACS is better in terms of distance; however, the overtime achieved by the MA is better. In addition, the ACS provides a better solution for the two objectives in the Thai75c-1 instance. Table 5 presents the maximal and minimal GAPs related to the ACS for the two objectives of the problem.

Table 5

GAP related to the ACS algorithm

GAP	Distance	Overtime
Min	-92,11%	-100%
Max	2,5%	69,49%

As already stated in Section 4.5, the best solution should exhibit the shortest overtime if it exceeds 0. Otherwise, the best solution should exhibit the minimal total travelled distance. Thus, a solution may be better if it exhibits a minimal overtime despite its non-minimal distance. Regarding coherence, two objectives are coherent for a given instance if the best solution (among the three executions done by each algorithm) for this instance is a solution that achieves the minimal distance and overtime. Otherwise, the objectives are not coherent for this instance. This case is only true if the best solution does not exhibit the minimal total travelled distance. Fig. 6 shows the number of instances in which the objectives are coherent in both algorithms. The two objectives are coherent for 88 from 96 instances solved by the MA and for 54 from 62 instances solved by the ACS.

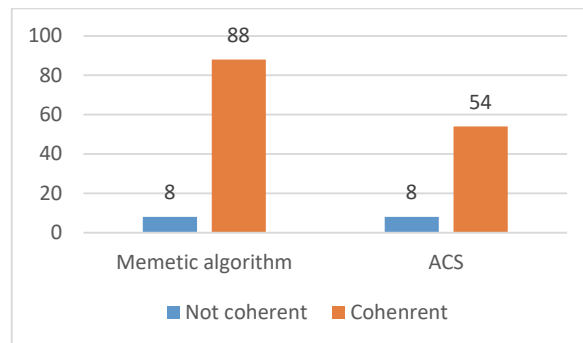


Fig. 6. Coherence of problem objectives

Because both algorithms have been executed on the same machine, their execution times can be compared. Table 6 shows the average execution time of the instances of each problem and both algorithms. Neither algorithm dominates in terms of execution time. For several problems, the ACS performs much faster than the MA. This is the case for the problems: C75, Tai75b, Tai75c, Tai75d, Tai100a, Tai100b, Tai100d, Tai150a, Tai150b and Tai150c, in which the GAP exceeds -50% and achieves up to 208%. These problems constitute 47% of the total number of problems. In addition, there are problems for which the MA performs much faster (C150, F134). In these instances, the GAP is below -50%. However, their number (9% of the total number of problems) remains minimal compared to the number of instances in which the ACS obtains results faster than the MA. Furthermore, the two algorithms consumed comparable execution time ($-50% < \text{GAP} < 50%$) for the rest of the problems. It should be noted that two problems (C100b, C120) are not included in this comparison because the ACS could not find any solution for any of their instances.

Table 6

Average of execution time in second

	MA	ACS	GAP		MA	ACS	GAP
C50	42,4	40,7	4,1%	Tai100a	302,7	112,7	168,5%
C75	78,2	42,5	84%	Tai100b	288,9	117,3	146,2%
C100	199	211	-5,6%	Tai100c	252,8	202,5	24,8%
C150	607,9	-	-	Tai100d	389,7	206	89,1%
C199	1253,7	-	-	Tai150a	584,6	328,5	77,9%
C120	513,3	1237	-58,5%	Tai150b	979,1	614,4	59,3%
C100b	196,3	193	1,7%	Tai150c	1038,6	647	60,5%
F71	145,4	205,9	-29,3%	Tai150d	528,4	517,1	2,1%
F134	1266,3	6138,8	-79,3%				
Tai75a	62,4	42	48,5%				
Tai75b	156,6	55	184,7%				
Tai75c	125,4	40,6	208,8%				
Tai75d	201,3	81,4	147,2%				

6. Conclusion

In this article, an MA for the MTDVRPOT is proposed, which provides very competitive results for the capacitated DVRP and MTDVRPOT. For the first problem, tests on the benchmarks presented by Kilby et al. (1998) were conducted, and the results were compared with the results of Hanshar & Ombuki-Berman (2007) and Montemanni et al. (2005); 62% of the results of this study are better. Regarding the second problem, tests on datasets that were proposed in a previous study were performed. According to the results, the MA is quantitatively and qualitatively better than the ACS. In the future, other metaheuristics will be tested on the same problem to realize a comparative study of different approaches. Adding more constraints to the problem (e.g. soft time windows) could be a subject for future studies. In that case, more objectives (e.g. maximizing customer satisfaction) could be added to the objective function, and the problem could be treated based on a multi-objective aspect to provide decision-makers with a set of good solutions for each objective.

References

- Albareda-Sambola, M., Fernández, E., & Laporte, G. (2014). The dynamic multiperiod vehicle routing problem with probabilistic information. *Computers and Operations Research*, 48, 31–39.
- Ayadi, R., & Benadada, Y. (2013). Memetic algorithm for a multi-objective vehicle routing problem with multiple trips. *International Journal of Computer Science and Applications*, 10(2), 72–91.
- Azi, N., Gendreau, M., & Potvin, J. Y. (2012). A dynamic vehicle routing problem with multiple delivery routes. *Annals of Operations Research*, 199(1), 103–112.
- Barkaoui, M., Berger, J., & Boukhtouta, A. (2015). Customer satisfaction in dynamic vehicle routing problem with time windows. *Applied Soft Computing Journal*, 35, 423–432.
- Beaudry, A., Laporte, G., Melo, T., & Nickel, S. (2010). Dynamic transportation of patients in hospitals. *OR Spectrum*, 32(1), 77–107.
- Benyahia, I., & Potvin, J. Y. (1998). Decision support for vehicle dispatching using genetic programming. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 28(3), 306–314.
- Bolaños, R. I., Escobar, J. W., & Echeverri, M. G. (2018). A metaheuristic algorithm for the multi-depot vehicle routing problem with heterogeneous fleet. *International Journal of Industrial Engineering Computations*, 9(4), 461–478.
- Brandão, J., & Mercer, A. (1997). A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research*, 100(97), 180–191.
- Brandão, J., & Mercer, A. (1998). The Multi-Trip Vehicle Routing Problem. *Journal of the Operational Research Society*, 49, 799–805.
- Cheikh, M., Ratli, M., Mkaouar, O., & Jarboui, B. (2015). A variable neighborhood search algorithm for the vehicle routing problem with multiple trips. *Electronic Notes in Discrete Mathematics*, 47, 277–284.

- Chen, Z. L., & Xu, H. (2006). Dynamic column generation for dynamic vehicle routing with time windows. *Transportation Science*, 40(1), 74–88.
- Cheung, B. K. S., Choy, K. L., Li, C. L., Shi, W., & Tang, J. (2008). Dynamic routing model and solution methods for fleet management with mobile technologies. *International Journal of Production Economics*, 113(2), 694–705.
- Crainic, T. G., Gendreau, M., & Potvin, J. Y. (2009). Intelligent freight-transportation systems: Assessment and the contribution of operations research. *Transportation Research Part C: Emerging Technologies*, 17(6), 541–557.
- Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, 6(1), 80–91.
- Dondo, R., & Cerdá, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, 176(3), 1478–1507.
- Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers and Industrial Engineering*, 57(4), 1472–1483.
- Eydi, A., & Javazi, L. (2012). A novel heuristic method to solve the capacitated arc routing problem. *International Journal of Industrial Engineering Computations*, 3(5), 767–776.
- Ferland, J. A., & Michelon, P. (1988). The vehicle scheduling problem with Multiple Vehicle Types. *Journal of Operational Research Society*, 39(6), 577–583.
- Fleischmann, B. (1990). *The vehicle routing problem with multiple use of the vehicles*. Fachbereich Wirtschaftswissenschaften, Universität Hamburg, Germany.
- Gendreau, M., Guertin, F., Potvin, J.-Y., & Taillard, É. (1999). Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*, 33(4), 381–390.
- Gendreau, M., Guertin, F., Potvin, J. Y., & Séguin, R. (2006). Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Transportation Research Part C: Emerging Technologies*, 14(3), 157–174.
- Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2008). A Tabu Search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51(1), 4–18.
- Gillett, B. E., & Miller, L. R. (1974). A Heuristic Algorithm for the Vehicle-Dispatch Problem. *Operations Research*, 22(2), 340–349.
- Golden, B., Assad, A., Levy, L., & Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers and Operations Research*, 11(1), 49–66.
- Golden, B. L., Magnanti, T. L., & Nguyen, H. Q. (1977). Implementing vehicle routing algorithms. *Networks*, 7(2), 113–148.
- Haghani, A., & Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers and Operations Research*, 32(11), 2959–2986.
- Hanshar, F. T., & Ombuki-Berman, B. M. (2007). Dynamic vehicle routing using genetic algorithms. *Applied Intelligence*, 27(1), 89–99.
- Hemert, J. I. va., & Poutre, J. A. La. (2010). *Dynamic Routing Problems with Fruitful Regions: Models and Evolutionary Computation*. 692–701.
- Hong, L. (2012). An improved LNS algorithm for real-time vehicle routing problem with time windows. *Computers and Operations Research*, 39(2), 151–163.
- Hvattum, L. M., Løkketangen, A., & Laporte, G. (2006). Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science*, 40(4), 421–438.
- Ichoua, S., Gendreau, M., & Potvin, J.-Y. (2000). Diversion Issues in Real-Time Vehicle Dispatching. *Transportation Science*, 34(4), 426–438.
- Ichoua, S., Gendreau, M., & Potvin, J. Y. (2003). Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144(2), 379–396.
- Irnich, S. (2000). Multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles. *European Journal of Operational Research*, 122(2), 310–328.
- Jaillet, P., & Wagner, M. R. (2008). Generalized online routing: New competitive ratios, resource

- augmentation, and asymptotic analyses. *Operations Research*, 56(3), 745–757.
- Khouadjia, M. R., Sarasola, B., Alba, E., Jourdan, L., & Talbi, E. G. (2012). A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests. *Applied Soft Computing Journal*, 12(4), 1426–1439.
- Kilby, P., Prosser, P., & Shaw, P. (1998). Dynamic VRPs: A Study of Scenarios. *Report APES-06-1998*, (April), 1–11.
- Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, 249(33), 1–14.
- Kulkarni, R. V., & Bhave, P. R. (1985). Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, 20(1), 58–67.
- Letchford, A. N., & Eglese, R. W. (1998). The rural postman problem with deadline classes. *European Journal of Operational Research*, 105(3), 390–400.
- Levy, D., Sundar, K., & Rathinam, S. (2014). Heuristics for routing heterogeneous unmanned vehicles with fuel constraints. *Mathematical Problems in Engineering*.
- Li, F., Golden, B., & Wasil, E. (2007). A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers and Operations Research*, 34(9), 2734–2742.
- Lin, C., Choy, K. L., Ho, G. T. S., Lam, H. Y., Pang, G. K. H., & Chin, K. S. (2014). A decision support system for optimizing dynamic courier routing operations. *Expert Systems with Applications*, 41(15), 6917–6933.
- Mañdziuk, J., & Żychowski, A. (2016). A memetic approach to vehicle routing problem with dynamic requests. *Applied Soft Computing Journal*, 48, 522–534.
- Mingozzi, A., Roberti, R., & Toth, P. (2013). An exact algorithm for the multitrip vehicle routing problem. *INFORMS Journal on Computing*, 25(2), 193–207.
- Montemanni, R., Gambardella, L. M., Rizzoli, A. E., & Donati, A. V. (2005). A new algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System. *Second International Workshop on Freight Transportation and Logistics*, 1(1), 1–4.
- Olivera, A., & Viera, O. (2007). Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers and Operations Research*, 34(1), 28–47.
- Ouaddi, K., Benadada, Y., & Mhada, F.-Z. (2018). Ant Colony System for Dynamic Vehicle Routing Problem with Overtime. *International Journal of Advanced Computer Science and Applications*, 9(6), 306–315.
- Petch, R. J., & Salhi, S. (2003). A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics*, 133(1–3), 69–92.
- Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1), 1–11.
- Prins, C. (2002). Efficient Heuristics for the Heterogeneous Fleet Multitrip VRP with Application to a Large-Scale Real Case. *Journal of Mathematical Modelling and Algorithms*, 1(2), 135–150.
- Psaraftis, H. N. (1980). A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem. *Transportation Science*, 14(2), 130–154.
- Psaraftis, H. N., Wen, M., & Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1), 3–31.
- Rochat, Y., & Taillard, É. D. (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1), 147–167.
- Salhi, S., & Sari, M. (1997). A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research*, 103(1), 95–112.
- Salhi, Said, Imran, A., & Wassan, N. A. (2014). The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers and Operations Research*, 52, 315–325.
- Schyns, M. (2015). An ant colony system for responsive dynamic vehicle routing. *European Journal of Operational Research*, 245(3), 704–718.
- Seixas, M. P., & Mendes, A. B. (2013). Column generation for a multitrip vehicle routing problem with time windows, driver work hours, and heterogeneous fleet. *Mathematical Problems in Engineering*,

2013.

Solomon. (1983). Vehicle Routing and Scheduling with Time Window Constraints: Models and Algorithms. *Ph.D. Dissertation, Dept. of Decision Sciences, University of Pennsylvania.*, 130(2), 556.

Stewart, W. R., & Golden, B. L. (1984). A Lagrangean relaxation heuristic for vehicle routing. *European Journal of Operational Research*, 15(1), 84–88.

Taillard, D., Gambardella, L. M., & Gendreau, M. (2001). *Adaptive memory programming: A unified view of metaheuristics*. 135.

Taillard, É. (1993). Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8), 661–673.

Taillard, É. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO (Recherche Opérationnelle/ Operations Research)*, 33, 1–14.

Taillard, É., Laporte, G., & Gendreau, M. (1996). Vehicle routing with multiple use of vehicles. *Journal of the Operational Research Society*, 47(8), 1065–1070.

Xu, Y., Wang, L., & Yang, Y. (2012). A New Variable Neighborhood Search Algorithm for the Multi Depot Heterogeneous Vehicle Routing Problem with Time Windows. *Electronic Notes in Discrete Mathematics*, 39, 289–296.

Yellow, P. (1970). A Computational modification to the savings method of vehicle scheduling. *Operational Research Quarterly*, 12, 281–283.

Appendix A

Table 7 provides detailed computational results provided by MA compared to those of the ACS. T is the normal working time while m represents the number of used vehicles. A feasible solution is a solution whose overtime does not exceed the maximal allowed overtime. Instances denoted by * are instances in which only one feasible solution was found, while those denoted by ** are instances in which two feasible solutions were found. We leave the cells empty for the instances to which we could not find any feasible solution. For the other instances, three feasible solutions were found. We consider three executions for each instance. If the objectives are coherent, the minimum distance and overtime are the values of the best-found solution. Otherwise, the two objectives are not coherent (the best solution does not match to the solution with minimal traveled distance). These instances are highlighted into dark shaded cells. We use Table 8 to represent their best-found solutions.

Table 8

Results of the memetic algorithm on multi-tour DVRP with overtime compared to those obtained by the ACS

	m	T	Memetic algorithm				Ant colony system algorithm			
			Best		Average		Best		Average	
			Dist	Over	Dist	Over	Dist	Over	Dist	Over
C50	1	577	565,4	0	579,3	8,8	625,3	48,3	629,9	52,9
	2	289	600	11,2	620,5	22,1	658,1	40,7	673,8	53,8
	3	192	613,1	15,2	644,2	21,9	652,5	26,3	672,2	36,2
	4	144	620,5	21,6	642,8	27,3	674*	36*	674*	36*
	5	115								
C75	1	919	911,4	0	920,3	5,4	992,8	73,8	1082,7	163,7
	2	459	931,5	7,3	962,1	22,5	1064	73,6	1079,1	81,5
	3	306	930,8	8,8	948,9	15,2	1069,1	58,8	1092,2	64,3
	4	230	952,1	10,8	983,2	18,3	1046,8	45,1	1060,7	48,8
	5	184	964,4	9,7	985,4	14,9				
C100	1	909	977,4	68,4	986,3	77,3	988,6	79,6	1015,3	106,3
	2	454	991	44	1023,2	58,4	983,5	46,1	1018,8	66,1
	3	303	1008,3	34,3	1057,8	50,5	1015,4	40,7	1064,4	59,9
	4	227	1005,3	26,5	1026,4	40,3	1021,1	39,7	1047	44,7
	5	182	1052	35,5	1067,8	41,5	1114,1*	44,7*	1114,1*	44,7*
C150	1	1131	1304,4	173,4	1313,7	182,7				
	2	565	1249,4	59	1272,1	70,7				
	3	377	1271,9	50,6	1279	51,2				
	4	282	1257,8	32,1	1290,5	41,4				
	5	226	1364,4	52,9	1377	53				

C199	1	1421	1549,3	128,3	1565	144				
	2	710	1589,8	85,1	1613,3	97,2				
	3	473	1598,1	59,1	1609,2	63,1				
	4	355	1633,1	53,7	1641	56,4				
	5	284	1621,3	40,5	1642,4	45,1				
C120	1	1146	1133,1	0	1163,9	22,2	1426,16**	280,16**	1429,115**	283,115**
	2	573	1363*	108,6*	1363*	108,6*				
	3	382								
	4	287								
	5	229								
C100b	1	902	865,9	0	870,1	0	1007,8	105,8	1015	113
	2	451	917,5	7,8	925,1	13	1049,7	75,6	1054	83,3
	3	301	867,6	0	871,3	0,7	982,7	29,5	1038,5	53,4
	4	225	873,3	0	882,7	0	998,6	45,8	1012,4	49,1
	5	180	987,8	27,2	1002,4	29,5				
F71	1	266	277,4	11,4	297,9	31,9	304,8**	38,8**	312,6**	46,6**
	2	133	294,6	15	305,6	20,1	314	24,1	322,2	28,6
	3	89	295	15,9	309	18				
	4	66								
	5	53								
F134	1	1279	1189,5	0	1255,3	12,1	1508,5	229,3	1547,4	268,2
	2	639	1214,2	0	1252,9	2	1500,7*	142,8*	1500,7*	142,8*
	3	426	1267,1	0	1323,3	16,9				
	4	319	1348,8	23,3	1382,7	32,1				
	5	255								
Thai75a	1	1780	1735,1	0	1758	0	1876	96	1982,8	202,8
	2	890	1740,1	0	1774,9	5,1	1990,8	108	2035,6	134,8
	3	593	1704,2	0	1780,2	13,1	2086,2**	108**	2097,3**	119,6**
	4	445	1819	12,1	1850	21,8				
	5	356	1860,2	26,5	1958,2	49,5				
Thai75b	1	1479	1356,7	0	1414	9,6	1478,2	0	1540,2	61,4
	2	740	1358,6	0	1366,6	0	1540,4	32	1579,5	52,2
	3	493	1431,1	0	1433,1	0	1613,2	53,8	1659,3	66,9
	4	370	1581,2	40,4	1622,9	44,7	1658**	64,7**	1679,6**	76**
	5	296	1611,8	71,5	1666,7	82,4				
Thai75c	1	1420	1510	90	1566,1	146,1	1473,1	53,1	1568,3	148,3
	2	710	1524,9	52,6	1531,9	56,1	1651,2	118,7	1686,4	141,2
	3	473	1524,1	56,5	1603,3	74,2	1617,1*	70,3*	1617,1*	70,3*
	4	355					1712,6*	80,5*	1712,6*	80,5*
	5	284								
Thai75d	1	1502	1449,6	0	1502,6	18	1619,2	117,2	1761,2	259,2
	2	751	1453,5	0	1475,4	4,6	1572	35,3	1610,8	65,9
	3	501	1386	0	1416,8	0	1665	64,2	1738,8	90,3
	4	375	1421,3	0	1442	1,7	1761,5**	93**	1806,9**	93,2**
	5	300	1496	13,8	1544,9	32,8				
Thai100a	1	2245	2100	0	2179,9	9,3	2487,7	242,7	2513,6	268,6
	2	1123	2106,5	0	2145,2	0	2621,6	192,8	2683,3	221,1
	3	748	2130,6	0	2247,8	14	2377,9	50,5	2531,4	106,9
	4	561	2169,2	0	2265,9	18,1	2462,6	75,9	2557,7	106,8
	5	449	2322,5	29,3	2349,3	36,3	2626,6**	92**	2653,2**	99,1**
Thai100b	1	2134	2087,9	0	2106	0	2443,4	309,4	2522,8	388,8
	2	1067	2079,9	0	2131,9	8	2584,2	229,4	2633,2	253,2
	3	711	2146,1	7,4	2195,3	22,6	2388,6	95,1	2462,9	118,7
	4	533	2130,6	1,1	2160,4	12,3	2497,6**	96,2**	2528,6**	106,6**
	5	427	2189,1	30,9	2217,9	43,5				
Thai100c	1	1547	1444,7	0	1465,5	0	1567	20	1633,1	86,1
	2	773	1440,8	0	1466	0	1505,8	0	1692,3	80,7
	3	516	1470,8	0	1515,8	2,4	1717,1	60,2	1820,9	95,4
	4	387	1652,7	27,7	1660,2	29,6				
	5	309	1558,8	13,1	1672,4	35,7				
Thai100d	1	1739	1767	28	1783,3	44,3	2017,6	278,6	2102,6	363,6
	2	869	1750,8	6,5	1800,3	31,8	2156,8*	216,7*	2156,8*	216,7*
	3	580	1755,5	5,7	1818,8	26,6				
	4	435	1776,1	14,2	1867,1	35,2				
	5	348	1820	27,9	1882,4	38,4				
Thai150a	1	3361	3464,5	103,5	3534,3	173,3	3772,7**	411,7**	3881,8**	520,8**
	2	1680	3437,3	39,1	3470,2	57,1	3887,2	274,9	4065	360,4
	3	1120	3200,3	0	3402,2	32,7	3819,7*	164,8*	3904,3*	206,4*
	4	840	3411,5	17,9	3460,5	33,4	3934,3	172,3	3956,2	173,6
	5	672	3296,8	0	3357,9	7,9				
Thai150b	1	3000	2943,4	0	3013,1	32	3426,5	426,5	3436,2	436,2
	2	1500	3079,8	40	3135,3	67,8	3281,8	155,4	3405,9	212,1

	3	1000	2973,6	0	3014,1	11,6	3612,9	230,5	3612,9	230,5
	4	750	2947,5	0	3029,8	17,8				
	5	600	3173,8	41,1	3186	47,9				
Thai150c	1	2595	2498,2	0	2599,2	50,5	3006,8	411,8	3086,4	491,4
	2	1297	2481,2	0	2584,2	15,3	3190,4*	312,3*	3190,4*	312,3*
	3	865	2681,5	29,6	2720,6	42,4				
	4	649	2557,1	0	2613,3	10,4				
	5	519	2561,1	0	2747,6	36,3				
Thai150d	1	2910	2875,7	0	2894,4	0	3323	413	3416,5	506,5
	2	1455	2927,3	8,9	2943	16,6	3431,4	271,4	3459,5	289,5
	3	970	2903,7	1,1	2920,8	8,9				
	4	727	2913,7	3,5	2930,5	7,5				
	5	582	2967,2	11,8	3003,4	21,5				

Table 8

Best solution of highlighted instances

	m	T	Memetic		Hybrid Ant colony	
			Dist	Over	Dist	Over
C75	3	306	942,9	8,8		
C75	4	230			1069,3	45,1
C100	2	454			989,7	46,1
C150	3	377	1279,4	50,6		
C150	5	226	1389,6	52,9		
C199	4	355	1633,8	53,7		
C100b	2	451			1052,9	75,6
C100b	4	225			1010,6	45,8
C100b	5	180	1022,3	27,2		
F134	1	1279			1549,2	270
Thai75b	4	370	1627	40,4		
Thai75d	4	375			1852,2	93
Thai100a	5	449	2330,4	29,3		
Thai150a	4	840			3946,6	172,3
Thai150b	5	600	3197,4	41,1		
Thai150d	2	1455			3435,1	271,4

Appendix B

Fig. 7 presents the tours of the solution that exhibits a lower GAP compared to that of Hanshar 2007 (section 5.2).

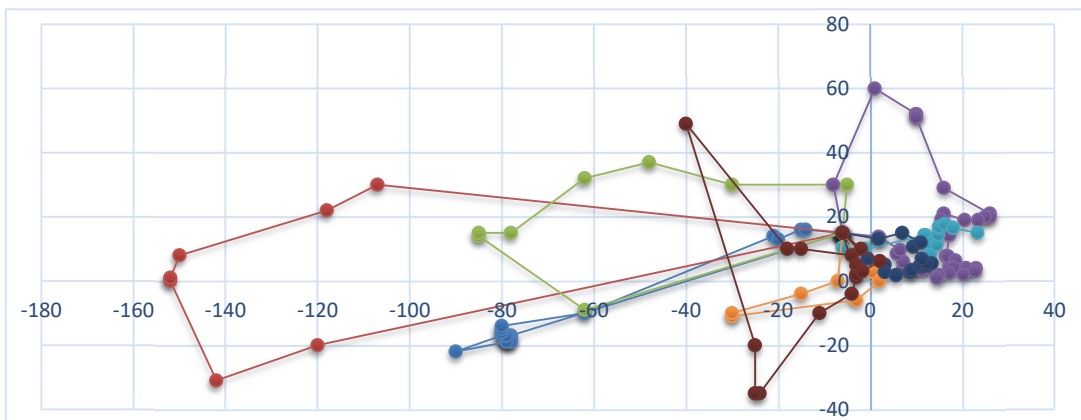


Fig. 7. Solution that exhibits a lower GAP compared to that of Hanshar 2007 (F134)



© 2020 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).