

Memetic algorithm for the dynamic vehicle routing problem with simultaneous delivery and pickup

Amina Berahhou^{a*}, Youssef Benadada^a and Khaoula Bouanane^a

^aSmart Systems Laboratory, ENSIAS, Mohammed V University, Rabat, Morocco

CHRONICLE

Article history:

Received January 24 2022
Received in Revised Format
March 21 2022
Accepted June 3 2022
Available online
June, 3 2022

Keywords:

DVRP
DVRPSDP
Local search
Memetic algorithm
Reverse logistics

ABSTRACT

In recent years, the Vehicle Routing Problem (VRP) has become an important issue for distribution companies. Also, the rapid development of communication means and the appearance of reverse logistics have given rise to new variants of the VRP. This article deals with an important variant of the VRP which is Dynamic Vehicle Routing Problem with Simultaneous Delivery and Pickup (DVRPSDP), in which new customers appear during the working day and each customer requires simultaneous delivery and pickup. A Memetic Algorithm (MA) that combines Genetic Algorithm (GA) and local search procedure have been proposed to solve the problem. The performance of the algorithm is evaluated with the tests carried out on a set of benchmarks found in the literature. The proposed memetic algorithm is very efficient and gives many good solutions.

© 2022 by the authors; licensee Growing Science, Canada

1. Introduction

In recent years, the world has experienced an economic, technological and environmental evolution. So, transportation companies, large and small, are seeking to design an efficient distribution system that increases their competitiveness and satisfies their customers. The first challenge encountered by these companies is the protection of the environment which has become a main objective of most countries, so companies should adopt environmentally friendly approaches and integrate environmental preservation into all stages of supply chain management. Among the measures to be taken we have the collection of reusable or recyclable packaging, empty packaging and end-of-life products, and deposit them in the recycling depots. The second challenge is the emergencies, it consists in the appearance of urgent customers when the working day has already started and the vehicles are in circulation, so the new customers must be inserted in the distribution plan as soon as possible. Combining the two challenges, i.e. the environmental protection and the emergencies issue, will give rise to a dynamic and inverse distribution logistics that aims to optimize the delivery of products to customers and the simultaneous pickup of reusable or empty packaging in order to recycle them, in a context where new customers who have a demand for delivery and collection appear when the working day has already started, which requires their integration into planned tours or creating new tours. In this context, a very important variant of the VRP is created and studied: The Dynamic Vehicle Routing Problem (Sáez et al., 2008; Berbeglia et al., 2010; Zachariadis et al., 2010; Mitrović-Minić & Laporte, 2004; Pura & Laporte, 2008; Ropke & Pisinger, 2006; Sheridan et al., 2013; Srour et al., 2018) with Simultaneous Delivery and Pickup (DVRPSDP). Fig. 1 shows an example of DVRPSDP.

The rest of this paper is organized as follows: Section 2 presents a detailed literature review of DVRPSDP. In section 3 the description of the problem is introduced and a mathematical model is proposed. Section 4 is reserved to the description of the proposed algorithm. Section 5 presents the parameter of the algorithm, the benchmark used for the problem and the numerical results. The conclusion is presented in section 6.

* Corresponding author

E-mail: amina_berahhou@um5.ac.ma (A. Berahhou)

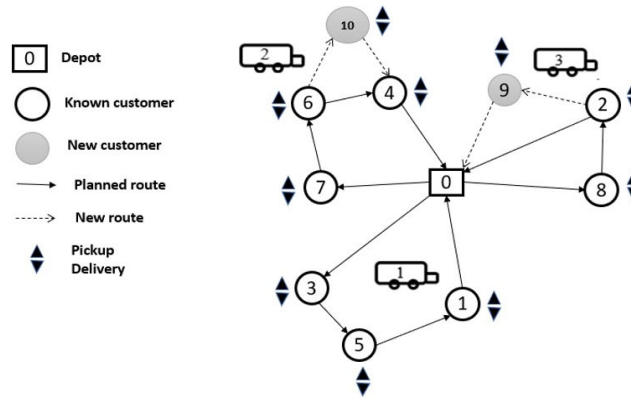


Fig. 1. Illustration of DVRPSDP

2. Literature Review

The VRP (Barkaoui et al., 2015) is a classical problem of combinatorial optimization. It consists of determining the routes for a set of vehicles available at the depot to serve a set of customers dispersed geographically. Each vehicle begins its tour from the depot and returns to the same depot. VRP has several variants including DVRP, VRPSDP and DVRPSDP. The main objective of our study deals with the arrival of new customers with delivery and collection demands (DVRPSDP) which is an important problem encountered by the distribution systems of large companies but unfortunately it is not very studied by researchers. To our knowledge, there are no studies in the literature that address the dynamic case of VRPSDP in the periodic and deterministic case. In this section, we will therefore be content to review the VRPSDP and DVRP literature.

2.1 VRPSDP

The VRPSDP is an extension of VRP in which the customer visited has a quantity to deliver and a quantity to be collected. When the vehicle finishes its tour, it returns to the depot to deposit what it has collected. The VRPSDP was introduced by Min (1989), he proposed a mathematical model for this variant, and developed a resolution approach based on the construction of Clusters of customers, then Assigning trucks/drivers to clusters and finally Creating the route structure. And then the VRPSDP became an objective of several research due to its importance in the protection of the environment. Several methods and techniques have been proposed in the literature to solve VRPSDP. We will mention the most relevant ones. For example, Dethloff (2001) described the important role played by the VRPSDP in reverse logistics. He suggested 2 an insertion-based heuristic which is based on three insertion criteria: Travel Distance (TD), Residual Capacity (RC), and Radial overload, to solve the problem. Angelelli and Mansini (2002) were the first to develop an exact approach based on the Branch and Price methods to solve the VRPSDP with time windows (VRPSDPTW). The same technique is used by Dell'Amico et al. (2006) to solve the VRPSDP. Chen and Wu (2006) have solved the VRPSDP using the insertion-based procedure to create a good initial solution, then a hybrid heuristic based on the record-to-record travel, tabu lists, and route improvement procedures are suggested to improve the initial solutions. Subramanian et al. (2007) suggested a greedy randomized adaptive search procedure that uses a variable neighborhood descent method to perform the local search for solving the VRPSDP. Gajpal and Abad (2009) proposed an approach to solve the VRPSDP, this approach is based on an Ant Colony System (ACS) algorithm that uses 3 local search schemes: 2-opt scheme, the customer insertion/interchange multi-route scheme, and the sub-path exchange multi-route scheme. Ai and Kachitvichyanukul (2009) developed a Particle Swarm Optimization (PSO) algorithm (Goksal et al., 2013; Lan & Xia, 2015) to solve the VRPSDP. Subramanian et al. (2010) solved the VRPSDP by using a multi start heuristic which consists in a variable neighborhood descent procedure, a random neighborhood ordering procedure and an Iterated Local Search (ILS) framework. A heuristic algorithm is proposed by Liu et al. (2013) to solve VRPSDPTW. The first heuristic is based on a GA, and the second one is a Tabu Search (TS) that is based on route assignment, an augmented cost function, route re-optimization, and attribute-based aspiration levels. Aguiar et al. (2014) solved capacitated VRPSDP with heterogeneous fleet by meta-heuristic PSO Discrete Adapted (PSODA). Hu et al. (2015) addressed the VRPSDP, in a context where the quantity to be delivered to customers is known at the beginning but the quantity to be collected is uncertain and known when the vehicle delivers to customer, and also the goods to be delivered and collected are incompatible. They proposed the VNS to solve the VRPSDP, as well as other methods to deal with uncertainty pickup. Kalayci and Kaya (2016) proposed an ACS empowered variable neighborhood search algorithm to solve VRPSDP. Zhao et al. (2018) has studied the disruption caused by the arrival of new pickup customers (Catay, 2010) on the original plan of VRPSDP. They established a model for the problem, and proposed a new dispatching method (Gendreau et al., 2006) using tabu search algorithm to solve the problem. Bouanane et al. (2020) carried out a taxonomic survey of the VRPSDP, conducting a comprehensive taxonomic classification of the VRPSDP literature published since 1989. Tao et al. (2020) studied the disruption of VRPSDP. They proposed a mathematical model to identify the disruption caused by the change of the original distribution and they developed a hybrid quantum bacterial foraging scheduling algorithm to solve the

problem. Christopher et al. (2021) compares the Variable Neighborhood Descent algorithm (VND) and TS algorithm in solving VRPSDP.

2.2 DVRP

The Dynamic Vehicle Routing Problem (DVRP), is the problem where new customers appear when the working day has already started and vehicle tours are planned, so the new customers must be included in the working day. The DVRP with capacity and time duration constraints was introduced by Kilby et al. (1998). Later, the DVRP became the object of several researches and several extensions, among these researches we have the work of Montemanni et al. (2005), they gave a description of the DVRP, and proposed the approach of dividing the DVRP into a sequence of static VRPs. Each one is solved with ACS. The same algorithm is used by Schyns (2015) to solve the capacitated DVRP with TW (DVRPTW) with split delivery and heterogeneous fleets. Necula et al. (2017) also proposed the resolution of the DVRPTW using an ant colony optimization algorithm. Chang et al. (2003) proposed Tabu Search algorithm (TS) to solve Dynamic Pickup (Mes et al., 2010) and delivery problems with TW(DPDPTW), in a deterministic and continuous perspective, i.e., the optimization is applied after the arrival of each new request. Hanshar et al. (2007) suggested a GA to solve the DVRP using the approach adopted by Montemanni et al. (2005), the proposed algorithm has given good results in terms of minimizing travel costs. The GA is also used by Elhassania et al. (2014) to solve DVRP, they combine the GA with some heuristics for the construction of the initial population and the crossover. Luo et al. (2011) treated the DVRP with stochastic (Swihart & Papastavrou, 1999) requests for emergencies. They developed a hybrid algorithm based on Clarke and Wright saving algorithms with the vehicle capacity and TW to build the initial solution, and the TS to optimize the solution. Khouadja et al. (2012) developed a PSO and a Variable Neighborhood Search (VNS) to solve the DVRP. Hong (2012) proposed an NS-based approach to solve the DVRP with hard time windows. He used the remove-reinsert mechanism to insert new customers in the planned tours. Euchi et al. (2015) proposed the resolution of the DVRP with a hybrid meta-heuristic approach that uses an Artificial Ant Colony based on 2-opt local search. Mandziuk and Zychowski (2016) developed a Memetic Algorithm (MA) to solve the DVRP, the proposed algorithm was able to give 14 new best results for the 21 instances tested. Ouaddi et al. (2018) treats the DVRP multi tours with overtime, they proposed a mathematical model for the problem and developed an ACS algorithm to solve it. Recently, Ouaddi et al. (2020) proposed a MA to solve multi-tours DVRP with overtime. Rios et al. (2021) presents a detailed review of the DVRP literature of the last seven years (2015–2021), they are interested in applications and solution methods.

3. Problem Formulation

The problem studied in this paper consists in satisfying the customers who appear when the planning of the day is already realized, and the vehicles have started the distribution. Each new customer has two kinds of demand: distribution demand and collecting demand. To serve the new customers there are two ways: either use the vehicles available at the depot, or insert the new customers into the existing vehicle tours. We adopted the approach proposed by Montemanni et al. (2005) which consists of dividing the working day in time slices (periods). During the first period, vehicles start their tours from the depot, and serve the customers received in the previous day after T_f (Time limit for receipt customers). During the following period, each vehicle begins its tour from the last customer it visited in the previous period. The capacity of the vehicle is the remaining capacity in the vehicle after serving the last customer of the previous period. The start time begins after serving the last customer of the previous period. The mathematical model of each period is already described in Berahhou and Benadada (2020).

Notation

T : is the length of the working day.

C_{T_p} : Set of new customer locations.

V_d : Set of vehicles available at the depot.

V_c : Set of vehicles in circulation (vehicles that have already visited customers).

$V = V_d \cup V_c$: Set of all vehicles.

p : Index of the period.

0 : Index of the depot.

O_{pk} : is the last customer visited by the vehicle k at the period T_{p-1} .

D_{pk} : is the total quantity delivered to customers already committed to vehicle k at the end of the period T_{p-1} .

P_{pk} : is the total quantity collected from customers already committed to vehicle k at the end of the period T_{p-1} .

T_{pk} : is the end of the serving time for the last customer committed to vehicle k at the period T_{p-1} .

L_{0k} : Load of vehicle in circulation $k \in V_c$ when leaving the depot.

D_j : Delivery quantity of customer $j \in C_{T_p}$.

P_j : Pick-up quantity of customer $j \in C_{T_p}$.

s_j : Service time of customer j .

C_{ij} : Travel costs between customer i and customer j .

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

Q: The maximum capacity of a vehicle.

M: Large number.

$O_p = \{O_{pk}, \forall k \in V_c\}$ is the set of the positions for the vehicles k at the beginning of the period T_p .

$N_p = C_{T_p} \cup \{0, O_p\}$ is the set of all nodes of the period T_p , and $c_p = \text{card}(N_p)$.

$$x_{ij}^k = \begin{cases} 1 & \text{if vehicle } k \text{ travels from customer } i \text{ to customer } j \\ 0 & \text{Otherwise} \end{cases}$$

$$y_i^k = \begin{cases} 1 & \text{if vehicle } k \text{ visits customer } i \\ 0 & \text{otherwise} \end{cases}$$

L_{pk} : Load of vehicle $k \in V$ at the beginning of the period T_p .

L_j : Load of vehicle after having serviced customer $j \in C_{T_p}$.

V_j : Variable used to prohibit sub tours; can be interpreted as position of node $j \in C_{T_p}$ in the route.

The objective function is

$$\text{Minimize } z = \sum_{k \in V} \sum_{i, j \in N_p} C_{ij} x_{ij}^k \quad (1)$$

Under the following constraints:

$$\sum_{j \in C_{T_p}} x_{O_{pk}j}^k \leq 1, \quad \forall k \in V \quad (2)$$

$$\sum_{i \in C_{T_p}} x_{O_{pk}i}^k = \sum_{i \in C_{T_p}} x_{i0}^k, \quad \forall k \in V \quad (3)$$

$$\sum_{i \in C_{T_p}} y_i^k \leq M * \sum_{j \in C_{T_p}} x_{j0}^k, \quad \forall k \in V \quad (4)$$

$$\sum_{k \in V} \sum_{i \in N_p} x_{ij}^k = 1, \quad \forall j \in C_{T_p} \quad (5)$$

$$\sum_{i \in N_p} x_{is}^k = \sum_{j \in N_p} x_{sj}^k, \quad \forall k \in V, \forall s \in C_{T_p} \quad (6)$$

$$L_{pk} = L_{0k} - D_{pk} + P_{pk}, \quad \forall k \in V_c \quad (7)$$

$$L_{pk} = \sum_{i \in N_p} \sum_{j \in C_{T_p}} D_j x_{ij}^k, \quad \forall k \in V_d \quad (8)$$

$$L_j \geq L_{pk} - D_j + P_j - M(1 - x_{O_{pk}j}^k) \quad \forall j \in C_{T_p}, \forall k \in V \quad (9)$$

$$L_j \geq L_i - D_j + P_j - M \left(1 - \sum_{k \in V} x_{ij}^k \right) \quad \forall i, j \in C_{T_p} \quad (10)$$

$$L_{pk} \leq Q, \quad \forall k \in V \quad (11)$$

$$L_j \leq Q, \quad \forall j \in C_{T_p} \quad (12)$$

$$D_{pk} + \sum_{j \in C_{T_p}} D_j x_{O_{pk}j}^k + \sum_{i \in C_{T_p}} D_i \sum_{j \in N_p} x_{ij}^k \leq L_{0k}, \quad \forall k \in V_c \quad (13)$$

$$T_{pk} + \sum_{i, j \in N_p} t_{ij} x_{ij}^k + \sum_{i \in C_{T_p}} s_i y_i^k \leq T, \quad \forall k \in V \quad (14)$$

$$\sum_{k \in V} \sum_{j \in C_{T_p}} x_{0j}^k \leq \text{Card}(V_d) \quad (15)$$

$$V_j \geq V_i + 1 - c_p * \left(1 - \sum_{k \in V} x_{ij}^k \right), \quad \forall i, j \in C_{T_p}, i \neq j \quad (16)$$

$$x_{ij}^k, y_i^k \in \{0, 1\} \quad \forall i, j \in C_{T_p} \cup \{0, O_{pk}\}, k \in V \quad (17)$$

$$V_j \geq 0, \quad \forall j \in C_{T_p} \quad (18)$$

The mathematical model obtained is a mixed linear program. The objective function (1) minimizes the total cost of the tours. Constraints (2), (3) and (4) impose that all vehicles start their tours from the last customer that have been committed to it and return to the depot. Constraints (5) require that each customer is visited once and only once by one vehicle. The flow conservation constraints are carried out by the constraints (6). Constraints (7) represent the load of vehicles in circulation at

the beginning of the period, and the constraints (8) represent the load of vehicles coming from the depot. The Vehicle load after visiting the first customer of the period is limited by constraints (9). Constraints (10) ensure the vehicle load on route. Constraints (11) and (12) express that the quantity loaded in a vehicle does not exceed capacity limits. Constraints (13) ensure the availability of deliveries. Constraints (14) assure that the total duration of each route does not exceed a predefined limit. Constraints (15) impose that the number of vehicles added must not exceed the number of vehicles available in the depot. Finally, constraints (16) are Subtours breaking constraints.

4. The Proposed Memetic Algorithm

The DVRPSDP is a problem related to the VRP which is NP-Complex, the Cplex solver is used to validate the proposed mathematical model and to solve small instances, but it requires a long execution time. So, in order to find a good solution for large problem instances in a reasonable time, we will propose a meta-heuristic solution. The memetic algorithm (MA) nominated by Moscato (1989), is part of the family of evolutionary algorithms and is based on the mechanisms of natural evolution; it is inspired by Darwin's theory of species evolution. It consists in combining the GA, first introduced by Holland (1975), with local search heuristics to improve the solution. It is widely used to solve several optimization problems, because it gives good solutions when the exact methods cannot give a solution within a reasonable time, among these problems we find the VRP for which the MA has given good results. Our algorithm was inspired from the MA used by Ayadi and Benadada (2013) to solve the multi-objective VRP with multiple trips. Fig. 2 represents the life cycle of MA.

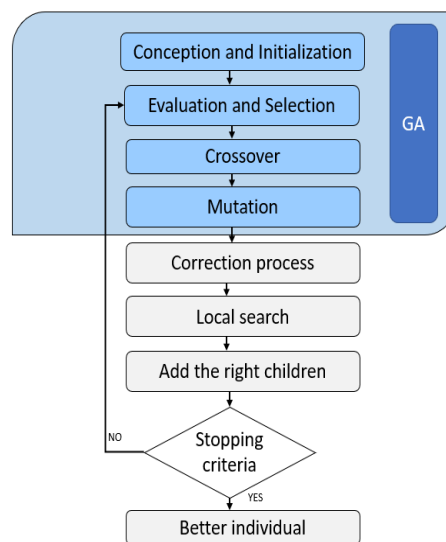


Fig. 2. Life cycle of the memetic algorithm.

4.1 Coding the solution

For each period T_p , we represent a chromosome by an array. Each chromosome represents the customers with the order in which they are visited by a vehicle. The first box contains the point of departure of the vehicle. If it starts from the central depot the box contains 0, otherwise it contains the last customer visited in the precedent period. Fig. 3 shows the coding of a solution for 6 customers and 2 vehicles.

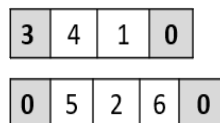


Fig. 3. Chromosome representation

4.2 Initialization of the population

The first step of the GA consists in building a population of individuals that will be the basis of future generations. To generate the initial population, we use the nearest neighbor method first, if the number of solutions obtained is less than the number necessary to build the initial population, we use a random insertion method to generate new solutions.

4.2.1 Nearest neighbor method or k nearest neighbors

This method is widely used to solve the VRP, because it is fast in execution. First, we look for the nearest customer to the depot, and add him to the tour. Then, the closest customer of the last customer visited is searched and inserted at the end of

the tour, and so on until the maximum capacity or time is reached. Then, a new tour is started and the addition of customers is done in the same way until all customers are inserted.

4.2.2 Random insertion method

This method starts with the creation of a tour, then customers are randomly selected and added to the tour one by one. If adding a customer violates the constraints of the problem a new tour is created. The algorithm stops when all customers are added.

4.3 Selection and Evaluation

After the construction of the initial population, it is necessary to select the individuals that will participate in the reproduction. We will use the selection by tournament which is widely used by evolutionary algorithms. It consists in choosing randomly a subset of individuals, then selecting the best individual according to its fitness function. The process is repeated until the desired number of individuals is obtained (Miller & Goldberg, 1995). The fitness function chosen for our problem is the total distance traveled to visit all customers.

4.4 Crossover

The crossover is a genetic operator that combines two chromosomes called parents to produce a new one called child. The importance of the crossover is that the chromosome created can be better if it takes the best features of each parent and is applied with a crossover probability p_c . Due to the nature of DVRP, traditional crossover methods such as Partially-Mapped-Crossover (PMX) are not adaptable for its resolution, so a new version of crossover is represented in Figure 4.

The Best Cost Route Crossover with priority to Capacity Constraint (BCRCPCC) method is a combination of Best Cost Route Crossover (BCRC) and capacity priority. First it checks the remaining capacity in the vehicle, making sure if it can meet the quantity requested by this customer. If no vehicle can meet this need, we create a new tour using an empty vehicle that is available at the depot. If a vehicle is chosen the customer will be inserted into the location that minimizes the total cost of each parent with the respect of the constraint. This requires calculating the insertion cost of each customer in an area of the tour.

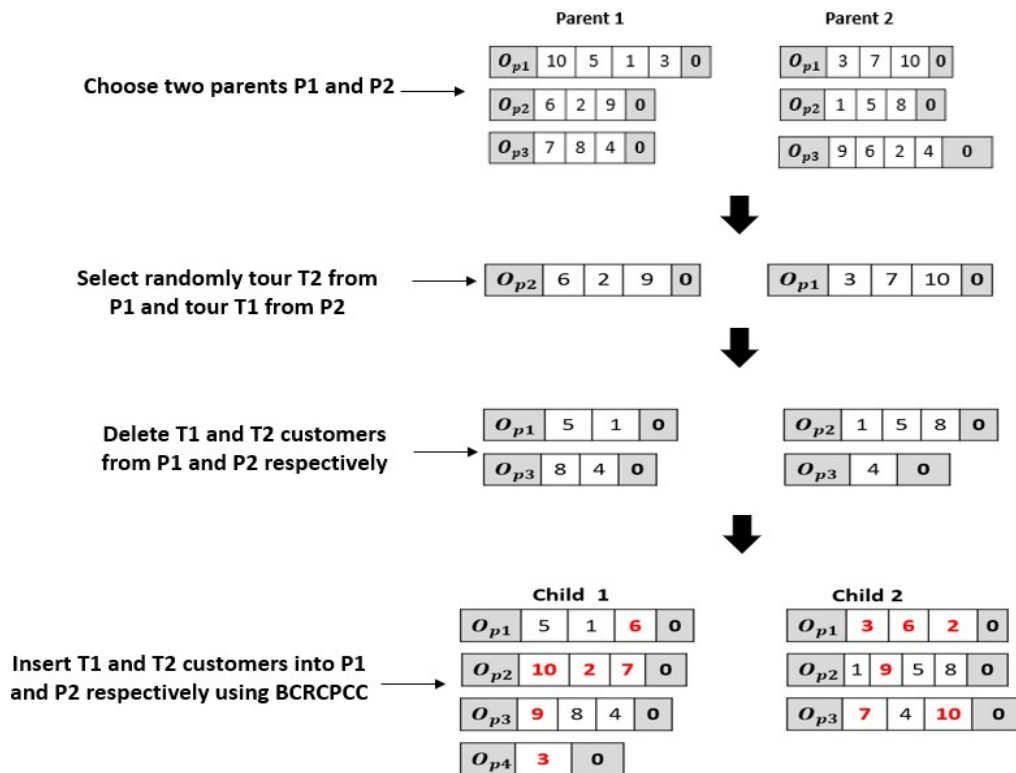


Fig. 4. The steps of the new crossover

4.5 Mutation

The mutation is a genetic operator that allows exploration of the research space, and avoids that the algorithm converges too quickly towards a local optimum. It is applied, with a probability of mutation, to the chromosomes generated by the crossing step, or directly to the selected parents if the crossing is not performed. Our algorithm will use two mutation methods, it

depends on the number of vehicles used, the first one is used in the case of a single vehicle and the second one is used in the case of several vehicles.

4.5.1 One vehicle mutation

In the one vehicle mutation there is only one line so the mutation consists in choosing randomly two cut points, and then reverse the order of visit of the customers who are between the 2 chosen points. Fig. 5 shows an example of one vehicle mutation.

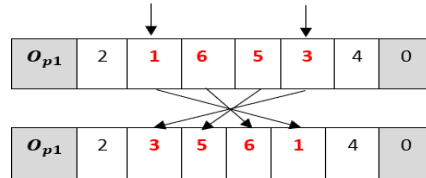


Fig. 5. One vehicle mutation

4.5.2 Several vehicles mutation

In several vehicles' mutation, two customers are randomly selected from two different tours and they are exchanged. Fig. 6 shows an example of several vehicles' mutation.



Fig. 6. Several vehicles mutation

4.6 Correction process

After applying the crossover and mutation operations on the parents, the constraints are violated in most cases. Therefore, it is necessary to apply a correction procedure to check the feasibility of the generated children. This procedure launches a local search to improve the order in which customers are visiting. Then the tours that exceed the capacity of the vehicles or the maximum service time are partitioned according to the position that guarantees the best total cost of the two new tours. Finally, the short turns are combined using the Clark and Wright method (Clarke & Wright, 1964). Fig. 7 shows the steps of the correction process.

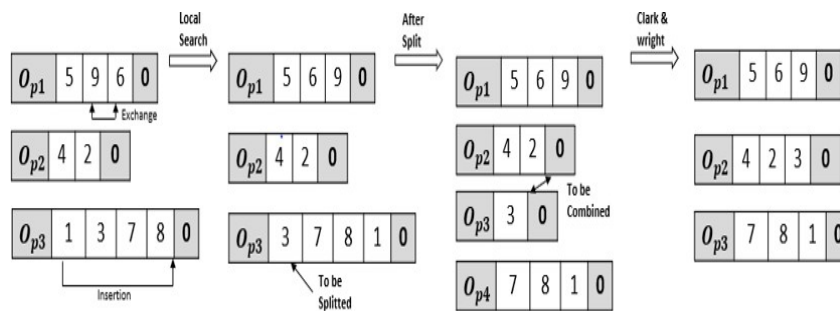


Fig. 7. The steps of correction process

4.7 Local Search

Local search is applied to each solution of the initial population and to each new child obtained through crossing and mutation operations. The proposed local improvement procedure uses two basic movements, highly responsive in solving the vehicles routing problems; reintegration and exchange. In order to accelerate research, the strategy of first improvement is adopted

rather than the best improvement. The procedure begins with the improvement of each individual's tour to be processed separately from the other tours. Then, local research is applied between the individual's tours two by two.

4.7.1 Local Search Intra-tour

Local search intra-tour consists in applying the reintegration and exchange operators on each tour separately from the other solution tours. The algorithm of local search intra-tour is detailed below

Algorithm 1

```

1: For each tour  $R_i$  of  $\{R_0, R_1, R_2, \dots, R_p\}$ 
2:   For each pair of customers  $c_i$  of  $R_i$  and  $c_j$  of  $R_{i+1}$ 
3:     If the exchange between  $c_i$  and  $c_j$  reduces the cost and all constraints are verified
4:       Exchange  $c_i$  and  $c_j$ 
5:       If the new solution is better than the original solution
6:         Save the exchange and go to 1
7:   For each customer  $c_i$  of  $R_i$ ,
8:     Insert  $c_i$  in the best position of  $R_i$ .
9:     If the new solution is better than the original solution
10:      Save the movement and go to 1
11: End

```

4.7.2 Local Search Inter-tour

In the local search inter-tour, the exchange and reintegration operators are applied between two different tours of the solution. First of all, the solution tours are sorted from the longest to the shortest in terms of distance. Indeed, the longest tours are the first to improve, considering that they are the first to be responsible for the high cost of the solution. The algorithm of local search inter-tour is detailed below:

Algorithm 2

```

1: While the maximum number of movements is not reached
2:   Sort tours from the longest to the shortest
3:   For each tour  $R_i$  of  $\{R_0, R_1, R_2, \dots, R_p\}$ 
4:     For each tour  $R_j$  of  $\{R_{i+1}, R_{i+2}, \dots, R_p\}$ 
5:       For each pair of customers  $c_i$  of  $R_i$  and  $c_j$  of  $R_j$ 
6:         If the exchange between  $c_i$  and  $c_j$  reduces the cost and all constraints are verified.
7:           Exchange  $c_i$  and  $c_j$ 
8:           If the new solution is better than the original solution
9:             Save the exchange and go to 1
10:      For each customer  $c_i$  of  $R_i$ ,
11:        If the total quantity of  $c_i$  and  $R_j$  does not exceed the capacity of the vehicle
12:          Insert  $c_i$  in the best position of  $R_j$ .
13:          If the new solution is better than the original solution
14:            Save the movement and go to 1
15: End

```

5. Computational Results

5.1 Parameter tuning with Taguchi method

The parameters of the MA have a great impact on the results obtained. The quality of the solution changes with the variation of parameter values. In order to choose the best parameters for our algorithm we adopted the Taguchi method. The Taguchi experimental design is widely used for optimization problems; it helps to identify the best parameters for the algorithms. Taguchi method is based on two tools: Orthogonal Array (OA) and signal to noise (ratio S/N). The Taguchi method based on OA allows to reduce the number of experiments; the experimental schemes are constructed from the different levels of the parameters. In the ratio S/N, S represents the variable response signal and N is the standard deviation noise. So, the S/N ratio represents the strength of change in the response variable. The objective of Taguchi method is to maximize the S/N ratio (Azadeh et al. (2017); Sarbijan et al. (2022)).

Parameters of the proposed MA considered for tuning are: population size, crossover probability and mutation probability. Different levels of these parameters for the tuning process are shown in Table 1.

Table 1
Different levels of MA parameters used for tuning

MA parameters	Levels			
	1	2	3	4
Population Size	50	100	150	200
Crossover probability	0.5	0.6	0.7	0.8
Mutation probability	0.01	0.04	0.07	0.1

By studying the different designs of the Taguchi method, we choose the L16 design using Minitab 21 software. More details about Taguchi method and the different designs can be found in the book written by Roy (2010). For each scheme, the instance of 50 customers was tested, and 10 runs of the MA is performed for each design. The algorithm stops after 200 iterations without any improvement in the objective function. The proposed algorithm is coded in Java and tested on a machine with Intel Core I7 processor, 2Ghz, 32GB of RAM and the operating system windows 10. The results obtained are summarized in Table 2.

Table 2
Results obtained from different designs of Taguchi method

Design	MA Parameters			Distance
	Population Size	Crossover probability	Mutation probability	
1	50	0.5	0.01	576.16
2	50	0.6	0.04	575.34
3	50	0.7	0.07	578.94
4	50	0.8	0.10	573.73
5	100	0.5	0.04	575.25
6	100	0.6	0.01	577.15
7	100	0.7	0.10	580.08
8	100	0.8	0.07	581.15
9	150	0.5	0.07	576.22
10	150	0.6	0.10	573.87
11	150	0.7	0.01	570.39
12	150	0.8	0.04	569.32
13	200	0.5	0.10	571.84
14	200	0.6	0.07	571.63
15	200	0.7	0.04	572.12
16	200	0.8	0.01	568.37

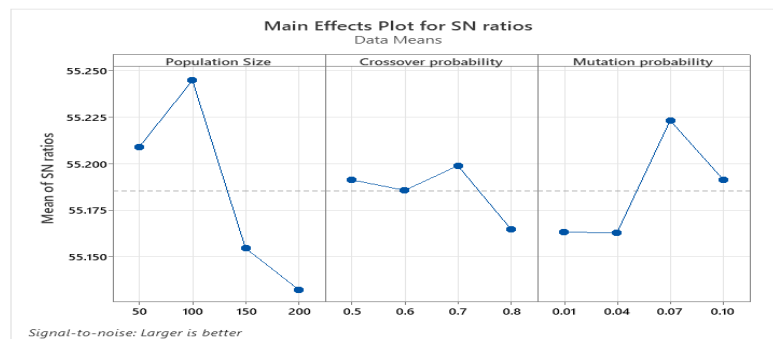


Fig. 8. S/N ratio plot obtained from Taguchi method

According to the S/N ratios diagram of the Taguchi design represented in Fig. 8, the optimal values of the parameters of the MA are the maximum values of the diagram. Table 3 represents the optimal value of each parameter.

Table 3
Optimal Values for MA parameters

Parameters	Optimal values
Population Size	100
Crossover probability	0.7
Mutation probability	0.07

5.2 Benchmarks

The DVRP and DVRPSDP was tested on the instances of Christofides and Beasley (1984) and Christofides (1979) (7 instances), Taillard (1993) (12 instances), and Fisher et al. (1981) (2 instances) proposed by Kilby et al. (1998) and extended by Montemanni et al. (2005). The arrival time of each customer is proportional to the arrival time proposed by Kilby et al. (1998). Concerning the number of periods and T_f , we used the same parameter of Montemanni et al. (2005) (25 periods and $T_f = \frac{T}{2}$). The number of vehicles is set to 50 for each problem. To generate the delivery and pickup demands of each client from their original demand, we adopted the method used by Salhi and Nagy (1999). For each client i of X_i and Y_i coordinates, and original demand d_i , we calculate the ratio r_i , and then the delivery demand D_i and pickup demand P_i are calculated as follows:

$$r_i = \min\left(\frac{X_i}{Y_i}, \frac{Y_i}{X_i}\right), \quad D_i = r_i * d_i \quad \text{and} \quad P_i = (1 - r_i) * d_i$$

Each instance is divided as sets X and Y. A set X of instances is created, and another set Y of instances is generated by exchanging the delivery and pickup demand for each customer.

5.3 Results

To our knowledge, there are no studies in the literature that address the dynamic case of VRPSDP with periodic re-optimization, so our results have not been compared with other works. Thus, before giving the numerical results obtained by the application of the MA to large instances of the DVRPSDP, we started by solving small instances of the VRPSDP with Cplex and then with the MA. This allowed us, on the one hand, to validate the mathematical model, and on the other hand, to test the performance of the MA. Our test is performed on the data set A of Augerat et al. (1995) by considering only the first 10, 15, 20, 25 and 30 customers. According to the results presented in Table 4, we can see that for the small instances of 10, 15, 20, 25 customers the Cplex solver was able to give the optimal result but this requires a long time that goes up to 7h for 25 customers, however the MA was able to give results very close to the optimal value in a time that does not exceed 1 minute. For the problem of 30 customers, Cplex could not give a solution because of the running out of memory. We conclude that the MA has shown its efficiency by giving solutions very close to the optimal value and in less than one minute of CPU time.

Table 4
Comparison of Cplex and Memetic algorithm results for Static VRPSDP

Nb Customers	Cplex Solver		Memetic Algorithm	
	Distance	CPU(S)	Distance	CPU(S)
10	428	963	429.80	2
15	441	1505	442.10	6
20	450	2726	451.89	8
25	464	23100	466.23	8
30	-	-	496.63	10

5.3.1 Results of Dynamic VRPSDP

Table 5 presents the results obtained by applying the MA on Montemanni et al. (2005) data set.

Table 5
Memetic Algorithm results for VRPSDP and DVRPSDP

Instance	VRPSDP- T_f		DVRPSDP-T		VRPSDP- T_f		DVRPSDP-T		
	Distance	CPU	Distance	CPU	Distance	CPU	Distance	CPU	
c50X	410.87	0.2	557.23	1.6	c50Y	418.32	0.4	571.42	1.9
c75X	819.25	0.4	954.08	2.0	c75Y	809.52	0.4	936.94	2.3
c100X	861.46	0.7	1008.16	3.3	c100Y	885.74	0.9	979.09	3.7
c100bX	576.31	1.2	869.42	4.5	c100bY	586.31	1.4	876.84	5.3
c120X	859.24	1.7	1221.68	6.4	c120Y	912.64	1.5	1285.22	6.3
c150X	917.382	1.9	1376.47	7.4	c150Y	804.85	1.5	1261.97	7.0
c199X	1008.98	4.5	1464.25	8.4	c199Y	1018.30	4.3	1475.18	8.2
f71X	209.14	1.3	301.67	3.3	f71Y	213.29	1.4	304.09	3.7
f134X	928.07	2.5	1304.88	8.2	f134Y	924.38	2.6	1304.27	7.4
tai75aX	1254.52	0.9	1618.21	2.4	tai75aY	1243.51	1.4	1607.74	3.5
tai75bX	979.06	1.6	1366.29	3.7	tai75bY	894.60	1.4	1280.25	2.8
tai75cX	983.64	1.9	1376.46	2.9	tai75cY	1019.88	2.5	1409.62	3.7
tai75dX	836.72	2.4	1223.50	4.7	tai75dY	917.46	2.6	1301.84	5.1
tai100aX	1509.64	3.3	2151.28	5.2	tai100aY	1580.44	3.6	2219.54	5.7
tai100bX	1634.38	3.2	2139.51	5.1	tai100bY	1715.13	3.8	2218.10	5.7
tai100cX	921.76	2.6	1376.19	4.4	tai100cY	964.35	2.7	1416.82	5.4
tai100dX	1263.08	3.6	1675.45	6.7	tai100dY	1348.63	4.3	1765.22	7.6
tai150aX	2667.72	6.2	3384.83	8.3	tai150aY	2587.40	5.8	3302.08	8.6
tai150bX	2142.90	6.0	2713.61	8.9	tai150bY	2273.98	6.4	2854.17	8.9
tai150cX	1905.21	5.2	2438.16	9.1	tai150cY	1994.45	5.9	2527.90	9.7
tai150dX	2335.56	6.9	2932.77	8.8	tai150dY	2371.80	6.9	2962.67	9.2

The table shows the distance for the first period T1(static case) and the total distance of all periods T (dynamic case). We notice that for all the instances tested the MA was able to provide a feasible solution in a reasonable time that does not exceed 10 min.

5.3.2 Results of classical DVRP

As we are the first to study the DVRPSDP with periodic reoptimization, to the best of our knowledge, we could not compare our results with other works. Thus, to better evaluate the performance of the proposed algorithm, we will apply the MA to the classical case of DVRP. Next, we compare our MA with the Ant Colony System (DVRP-ACS) developed by Montemanni et al. (2005), and with GA (DVRP-GA) proposed by Hanshar and Beatrice (2007). The results of table 6 indicate that the MA was able to find a new best solution for 18 instances (out of 21 instances) with a considerable improvement that has reached 9%, the MA gives better results than Hanshar's algorithm for 18 instances (86%) and Montemanni's algorithm for all instances (100%). The Gap measures the difference between the MA and the best result (between DVRP-ACS and DVRP-GA). We conclude that the performance of the MA is better than the performance of ACS algorithm and genetic algorithm in solving DVRP.

Table 6
MA results vs DVRP-GA and DVRP-ACS results

Instance	MA	DVRP-ACS	DVRP-GA	Gap%
c50	554.81	631.3	570.89	-2.8
c75	966.28	1009.36	981.57	-1.5
c100	908.34	973.26	961.10	-5.5
c100b	880.69	944.23	881.92	-0.1
c120	1187.71	1416.45	1303.59	-8.8
c150	1289.15	1345.73	1348.88	-4.2
c199	1543.61	1771.04	1654.51	-6.7
f71	306.19	311.18	301.79	1.4
fl34	1405.82	1513.55	1552.88	-7.1
tai75a	1675.32	1843.08	1782.91	-6.0
tai75b	1418.21	1535.43	1464.56	-3.1
tai75c	1426.85	1574.98	1440.54	-0.9
tai75d	1338.24	1472.35	1399.83	-4.4
tai100a	2206.18	2375.92	2232.71	-1.2
tai100b	2185.64	2283.97	2147.70	1.7
tai100c	1483.47	1562.3	1541.28	-3.7
tai100d	1724.89	2008.13	1834.6	-6.0
tai150a	3386.75	3644.78	3328.85	1.7
tai150b	2867.06	3166.88	2933.40	-2.3
tai150c	2572.51	2811.48	2612.68	-1.5
tai150d	2943.72	3058.87	2950.61	-0.2

6. Conclusion

In this paper, we treated the dynamic vehicle routing problem with simultaneous delivery and pickup (DVRPSDP). A memetic algorithm which combines genetic algorithm with a local search procedure is developed to solve the problem. The algorithm was tested on DVRP and DVRPSDP using problem instances reported in the literature. For the DVRP we compared the MA results with those of Montemanni et al. (2005) results and Hanshar and Beatrice (2007) results, The MA was able to outperform the ACS algorithm and 86% of the MA results are better than GA results. The DVRPSDP is not widely studied, we have not found any work to compare our results. As future work, we will test the performance of the algorithm on larger instances, and also, we will propose other metaheuristics to solve the problem and compare the results with those obtained by the MA. Another avenue of research is to add more constraints to the problem such as stock restriction, time windows and more objective functions.

References

- Aguiar, B. D. C. X. C., Siqueira, P. H., de França Aguiar, G., & de Souza, L. V. (2014). Particle swarm optimization for vehicle routing problem with fleet heterogeneous and simultaneous collection and delivery. *Applied Mathematical Sciences*, 8(77), 3833-3849.
- Ai, T. J., & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 36(5), 1693-1702.

- Angelelli, E., & Mansini, R. (2002). The vehicle routing problem with time windows and simultaneous pick-up and delivery. In *Quantitative approaches to distribution logistics and supply chain management* (pp. 249-267). Springer, Berlin, Heidelberg.
- Augerat, P., Naddef, D., Belenguer, J. M., Benavent, E., Corberan, A., & Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem.
- Ayadi, R., & Benadada, Y. (2013). Memetic Algorithm for a Multi-Objective Vehicle Routing Problem with Multiple Trips. *Int. J. Comput. Sci. Appl.*, 10(2), 72-91.
- Azadeh, A., Elahi, S., Farahani, M. H., & Nasirian, B. (2017). A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers & Industrial Engineering*, 104, 124-133.
- Barkaoui, M., Berger, J., & Boukhtouta, A. (2015). Customer satisfaction in dynamic vehicle routing problem with time windows. *Applied Soft Computing*, 35, 423-432.
- Berahhou, A., & Benadada, Y. (2020, October). Dynamic Vehicle Routing Problem with Simultaneous Delivery and Pickup: Formulation and Resolution. In *2020 5th International Conference on Logistics Operations Management (GOL)* (pp. 1-8). IEEE.
- Berbeglia, G., Cordeau, J. F., & Laporte, G. (2010). Dynamic pickup and delivery problems. *European journal of operational research*, 202(1), 8-15.
- Bouanane, K., El Amrani, M., & Benadada, Y. (2020). The Vehicle Routing Problem with Simultaneous Delivery and Pickup: A taxonomic survey. DOI: 10.1504/IJLSM.2020.10036700.
- Catay, B. (2010). A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert systems with applications*, 37(10), 6809-6817.
- Chang, M. S., Chen, S. R., & Hsueh, C. F. (2003). 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.
- Chen, J. F., & Wu, T. H. (2006). Vehicle routing problem with simultaneous deliveries and pickups. *Journal of the Operational Research Society*, 57(5), 579-587.
- Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4), 568-581.
- Christopher, Y., Wahyuningsih, S., & Satyananda, D. (2021, May). Study of variable neighborhood descent and tabu search algorithm in VRPSDP. In *Journal of Physics: Conference Series* (Vol. 1872, No. 1, p. 012002). IOP Publishing.
- Christofides, N. (1979). The vehicle routing problem. *Combinatorial optimization*.
- Christofides, N., & Beasley, J. E. (1984). The period routing problem. *Networks*, 14(2), 237-256.
- Dell'Amico, M., Righini, G., & Salani, M. (2006). A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation science*, 40(2), 235-247.
- Dethloff, J. (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pickup. *OR-Spektrum*, 23(1), 79-96.
- Elhassania, M., Jaouad, B., & Ahmed, E. A. (2014, June). Solving the dynamic vehicle routing problem using genetic algorithms. In *2014 International Conference on Logistics Operations Management* (pp. 62-69). IEEE.
- Euchi, J., Yassine, A., & Chabchoub, H. (2015). The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach. *Swarm and Evolutionary Computation*, 21, 41-53.
- Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, 11(2), 109-124.
- Gajpal, Y., & Abad, P. (2009). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research*, 36(12), 3215-3223.
- 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.
- Goksal, F. P., Karaoglan, I., & Altiparmak, F. (2013). A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & industrial engineering*, 65(1), 39-53.
- Hanshar, F. T., & Ombuki-Berman, B. M. (2007). Dynamic vehicle routing using genetic algorithms. *Applied Intelligence*, 27(1), 89-99.
- Holland, J.H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.
- Hong, L. (2012). An improved LNS algorithm for real-time vehicle routing problem with time windows. *Computers & Operations Research*, 39(2), 151-163.
- Hu, Z. H., Sheu, J. B., Zhao, L., & Lu, C. C. (2015). A dynamic closed-loop vehicle routing problem with uncertainty and incompatible goods. *Transportation Research Part C: Emerging Technologies*, 55, 273-297.
- Kalayci, C. B., & Kaya, C. (2016). An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 66, 163-175.
- 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*, 12(4), 1426-1439.
- Kilby, P., Prosser, P., & Shaw, P. (1998). Dynamic VRPs: A study of scenarios. *University of Strathclyde Technical Report*, 1(11).
- Lan, L., & Xia, W. (2015, October). Application based on PSO of hopfield neural network algorithm in dynamic VRPSDP. In *2015 IEEE 12th International Conference on e-Business Engineering* (pp. 107-113). IEEE.

- Li, J., Pardalos, P. M., Sun, H., Pei, J., & Zhang, Y. (2015). Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. *Expert Systems with Applications*, 42(7), 3551-3561.
- Liu, R., Xie, X., Augusto, V., & Rodriguez, C. (2013). Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research*, 230(3), 475-486.
- Luo, J. Y., Wang, J. Y., & Yu, H. (2011, August). A dynamic vehicle routing problem for medical supplies in large-scale emergencies. In *2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference* (Vol. 1, pp. 271-275). IEEE.
- Mańdziuk, J., & Żychowski, A. (2016). A memetic approach to vehicle routing problem with dynamic requests. *Applied Soft Computing*, 48, 522-534.
- Mes, M., van der Heijden, M., & Schuur, P. (2010). Look-ahead strategies for dynamic pickup and delivery problems. *OR spectrum*, 32(2), 395-421.
- Miller, B. L., & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3), 193-212.
- Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5), 377-386.
- Mitrović-Minić, S., & Laporte, G. (2004). Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, 38(7), 635-655.
- Montemanni, R., Gambardella, L. M., Rizzoli, A. E., & Donati, A. V. (2005). Ant colony system for a dynamic vehicle routing problem. *Journal of combinatorial optimization*, 10(4), 327-343.
- Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826, 1989.
- Necula, R., Breaban, M., & Raschip, M. (2017, June). Tackling dynamic vehicle routing problem with time windows by means of ant colony system. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2480-2487). IEEE.
- 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.
- Ouaddi, K., Mhada, F., & Benadada, Y. (2020). Memetic algorithm for multi-tours dynamic vehicle routing problem with overtime (MDVRPOT). *International Journal of Industrial Engineering Computations*, 11(4), 643-662.
- Pureza, V., & Laporte, G. (2008). Waiting and buffering strategies for the dynamic pickup and delivery problem with time windows. *INFOR: Information Systems and Operational Research*, 46(3), 165-175.
- Rios, B. H. O., Xavier, E. C., Miyazawa, F. K., Amorim, P., Curcio, E., & Santos, M. J. (2021). Recent dynamic vehicle routing problems: A survey. *Computers & Industrial Engineering*, 160, 107604.
- Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4), 455-472.
- Roy, R. K. (2010). *A primer on the Taguchi method*. Society of Manufacturing Engineers.
- Sáez, D., Cortés, C. E., & Núñez, A. (2008). Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering. *Computers & Operations Research*, 35(11), 3412-3438.
- Sarbijan, M. S., & Behnamian, J. (2022). Multi-fleet feeder vehicle routing problem using hybrid metaheuristic. *Computers & Operations Research*, 105696.
- Salhi, S., & Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the operational Research Society*, 50(10), 1034-1042.
- Schyns, M. (2015). An ant colony system for responsive dynamic vehicle routing. *European Journal of Operational Research*, 245(3), 704-718.
- Sheridan, P. K., Gluck, E., Guan, Q., Pickles, T., Balciog, B., & Benhabib, B. (2013). The dynamic nearest neighbor policy for the multi-vehicle pick-up and delivery problem. *Transportation Research Part A: Policy and Practice*, 49, 178-194.
- Srour, F. J., Agatz, N., & Oppen, J. (2018). Strategies for handling temporal uncertainty in pickup and delivery problems with time windows. *Transportation Science*, 52(1), 3-19.
- Subramanian, A., Cabral, L. D. A. F., & Carvalho, G. R. (2007, October). A hybrid metaheuristic for the vehicle routing problem with simultaneous pick-up and delivery. In *ICIEOM: XIII international conference on industrial engineering and operations management, energy that moves production: A dialogue among integration, project and sustainability* (pp. 09-11).
- Subramanian, A., Drummond, L. M. D. A., Bentes, C., Ochi, L. S., & Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11), 1899-1911.
- Swihart, M. R., & Papastavrou, J. D. (1999). A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. *European Journal of Operational Research*, 114(3), 447-464.
- Taillard, É. (1993). Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8), 661-673.
- Tao, N., Shishasha, S., Peng, Z., & Tao, G. (2021). Disruption management decision model for VRPSDP under changes of customer distribution demand. *Journal of Ambient Intelligence and Humanized Computing*, 12(2), 2053-2063.
- Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2010). An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries. *European Journal of Operational Research*, 202(2), 401-411.

Zhao, L., Liang, X., & Yang, H. (2018, June). Disruption management model for VRPSDP with new customer request for pickup. In *2018 Chinese Control And Decision Conference (CCDC)* (pp. 423-427). IEEE.

Appendix

In order to compare the results of the static VRPSDP with works in the literature, MA was tested using the data of Christofides et al. (1979) (12 instances and 50 to 199 customers), used by Salhi and Nagy (1999) to test the VRPSDP. Table 5 provides comparison of the solution quality for the static VRPSDP in terms of minimizing travel distance is done between Parallel Iterated Local Search (PILS) proposed by Subramanian et al. (2010), and the ant colony system empowered variable neighborhood search algorithm (ACSEVNS) developed by Kalayci and Kaya (2016) and the MA solution. The results prove that the efficiency of the MA is better than the precedent algorithms. The results shows that MA has been able to provide new best solutions for all 24 instances, and it improves the results of Kalayci and Kaya (2016) and Subramanian et al. (2010) with a considerable decrease of up to 6% for the CMT2X and CMT2Y instances. In terms of the number of vehicles used, we notice that the Memetic algorithm uses less vehicles for all instances. Since Kalayci and Kaya (2016) did not specify the number of vehicles in their paper, some of the best-known results in Table 5 do not contain the number of vehicles.

Table 7
Comparison of MA with PILS and ACSEVNS for static VRPSDP.

Instance		PILS		ACSEVNS		Memetic Algorithm			
CMT	N	Distance	V	Distance	V	Distance	V	Gap%	
1X	50	466.77	3	466.77	-	459.21	3	-1.6	
2X	75	684.21	6	684.21	-	644.34	4	-5.8	
3X	100	721.27	5	721.27	-	708.96	3	-1.7	
4X	150	852.46	7	852.46	-	826.37	6	-3.0	
5X	199	1029.25	10	1030.55	-	1,006.62	6	-2.2	
6X	50	-	-	555.43	-	550.09	4	-0.9	
7X	75	-	-	900.12	-	891.49	8	-0.9	
8X	100	-	-	865.50	-	852.27	6	-1.5	
9X	150	-	-	1160.68	-	1149.86	8	-0.9	
11X	120	833.92	4	833.92	-	815.29	4	-2.2	
12X	100	662.22	5	662.22	-	651.59	4	-1.6	
1Y	50	466.77	3	466.77	-	459.21	3	-1.6	
2Y	75	684.21	6	684.21	-	644.34	4	-5.8	
3Y	100	721.27	5	721.27	-	708.96	3	-1.7	
4Y	150	852.46	7	852.46	-	826.37	6	-3.0	
5Y	199	1029.25	10	1030.55	-	1,006.62	6	-2.2	
6Y	50	-	-	555.43	-	550.09	4	-0.9	
7Y	75	-	-	900.12	-	891.49	8	-0.9	
8Y	100	-	-	865.50	-	852.27	6	-1.5	
9Y	150	-	-	1160.68	-	1149.86	8	-0.9	
11Y	120	833.92	4	833.92	-	815.29	4	-2.2	
12Y	100	662.22	5	662.22	-	651.59	4	-1.6	

N: number of customers, V: number of vehicles, Gap%: Percentage difference between the best known and the best-found solution, the best solution of is highlighted in bold, -: means the value was not mentioned in the source.



© 2022 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/>).