Journal of Engineering and Applied Sciences 11 (12): 2740-2749, 2016
ISSN: 1816-949X
© Medwell Journals, 2016

Mathematic Model and an Improved Evolutionary Algorithm for Bi-Objective Vehicle Routing Problem with Dynamic Requests

[1]Messaoud Elhassania and [2]Elhilali Alaoui Ahmed
[1]Systems and Sustainable Environments Laboratory, Private University of Fez, Fez, Morocco
[2]Modeling and Scientific Computing Laboratory, Faculty of Science and Technology, University Sidi Mohammed Ben Abdellah, Fez, Morocco

Abstract: The Vehicle Routing Problem (VRP) can be described as the problem of designing optimal delivery or collection routes from one or several depots to a number of geographically scattered cities or customers, subject to side constraints. The dynamic case of this problem where the information is not completely known in advance has not received enough consideration. In our research, we consider this case which new requests are received along the day. Hence, they must be serviced at their locations by a set of vehicles in real time minimizing two objectives simultaneously: the total travel distance and the response time of customers. The main goal of this research is to propose a mathematic model and to find a solution for our problem using an improved evolutionary algorithm. The experimental results show that the proposed approach proved to be successful on a variety of benchmark instances in terms of solution quality.

Key words: Vehicle routing problem, dynamic requests, bi-objective optimization, mathematic model, improved evolutionary algorithm

INTRODUCTION

The objective of VRP is to serve a set of customers at minimum cost such that every node is visited by exactly one vehicle only once, subject to side constraints. The VRP plays a central role in the fields of physical distribution and logistics. Huge research efforts have been devoted to studying the VRP since 1959 where Dantzig and Ramser (1959) have described the problem as a generalized problem of Travelling Salesman Problem (TSP).

In the static version of this problem, it is assumed that all customers are known in advance for the planning process. However, it may be the case that customers, routing costs or service times become available in the real time once the service has begun. Due to the recent advances in information, positioning systems and communication technologies, it is now possible to study such dynamic problems. The first reference to a dynamic vehicle routing problem is due to Wilson and Colvin (1977). They studied a Dial-A-Ride Problem (DARP) with a single vehicle in which customers requests are tripping from an origin to a destination that appear dynamically. Their approach uses insertion heuristics able to perform well with low computational requirement. Later research by Psaraftis (1980) introduced the concept of immediate request: a customer requesting service always wants to be served as early as possible, requiring the immediate re-planning of the current vehicle route. They propose a dynamic-programming algorithm for the static case as well as an adaptation to the dynamic context tested on a ten-customer instance.

The most common source of dynamism in vehicle routing is the online arrival of customer requests during the operation. More specifically, requests can be a demand for goods (Minic and Laporte, 2004; Hvattum *et al*., 2007; Mes *et al*., 2007) or services (Bertsimas and Ryzin, 1991; Beaudry *et al*., 2010), travel time, a dynamic component of most real-world applications has been recently taken into account (Fleischmann *et al*., 2004; Chen *et al*., 2006; Guner *et al*., 2012) while service time has not been explicitly studied (but can be added to travel time). We cite also some recent works which consider dynamically revealed demands for a set of known customers (Secomandi, 2000; Novoa and Storer, 2009) and vehicle availability (Li *et al*., 2009; Mu *et al*., 2011) in which case the source of dynamism is the possible breakdown of vehicles.

Corresponding Author: Messaoud Elhassania, Systems and Sustainable Environments Laboratory, Private University of Fez, Fez, Morocco

In our research, we consider the Vehicle Routing Problem with Dynamic Request (VRPDR) (or Dynamic Vehicle Routing Problem DVRP) with capacity and time duration constraints which was introduced by Kilby *et al.* (1998) and further refined by Larsen and Madsen (2000) and Montemanni *et al.* (2005). These authors proposed some benchmark instances for the DVRP and presented a study on how the degree-of-dynamism affects the final travel costs. Montemanni *et al.* (2005) who extended work (Kilby *et al.* 1998), considered a DVRP as an extension to the standard VRP by decomposing a DVRP as a sequence of static VRPs and then solving them using an Ant Colony System (ACS) algorithm. We quote also some other works for the DVRP (Branchini *et al.*, 2009) presented an adaptive granular local search heuristic; (Creput *et al.*, 2012) proposed the Self-Organizing Map (SOM) neural network into a population based evolutionary algorithm (Khouadjia *et al.*, 2012) studied a Particle Swarm Optimization (PSO) and a Variable Neighborhood Search (VNS) (Messaoud *et al.*, 2013) proposed a Hybrid Ant Colony System (Larsen and Madsen, 2000) applied the relocation strategy, Branke *et al.* (2005) and Thomas (2007) implemented the waiting strategy in various frameworks for the DVRP.

As for static VRP, a lot of different versions of the DVRP have been studied. For example, the dynamic VRPTW is recognized as a standard problem well suited to allow comparative evaluations of heuristics and metaheuristics for a common set of benchmarks. To solve this problem, an Improved Large Neighborhood Search algorithm is studied by Lianxi (2012), an Ant Colony System algorithm is adapted by Elhassania *et al.* (2013), an improved variable neighborhood search algorithm is proposed by Xu *et al.* (2013), the column generation scheme is introduced by Qureshi *et al.* (2012), the adaptation of the parallel Tabu Search (TS) framework is proposed by Taillard *et al.* (1997) and the relocation strategy is applied by Bent and Hentenryck (2007), Ichoua *et al.* (2006) and Hentenryck and Bent (2006).

Various other DVRP studies exist in the literature. We cite for example, Azi *et al.* (2012) which considers a vehicle routing problem where each vehicle performs delivery operations over multiple routes during its workday and where new customer requests occur dynamically; Sambola *et al.* (2014) which studies the dynamic multiperiod vehicle routing problem with probabilistic information, an extension of the dynamic multiperiod vehicle routing problem in which at each time period, the set of customers requiring a service in later time periods is unknown but its probability distribution is available which presents an algorithm based on an ant colony system to deal with a broad range of dynamic capacitated vehicle routing problems with time windows, split delivery and heterogeneous fleets, Larsen and Madsen (2000) which examines the traveling salesman problem with time windows for various degrees of dynamism where he proposes a real-time solution method that requires the vehicle, when idle to wait at the current customer location until it can service another customer without being early and Pureza and Laporte (2008) which uses waiting and buffering strategies for the dynamic pickup and delivery problem with time windows. In the last, we cite the researches by Kaiwartya *et al.* (2015) and Ghannadpour *et al.* (2014) which study in the last years, a multiobjective dynamic vehicle routing problem by a particle swarm optimization and a genetic algorithm respectively.

In this study, we study the Bi-objective Vehicle Routing Problem with Dynamic Requests (B-VRPDR) with capacity and time duration constraints. For that, we divide the day in periods of equal duration. A request arriving during a time slice is not handled until the end of the time bucket, thus during a time slice we only consider the requests known at its beginning. Hence, an Improved Evolutionary Algorithm (IEA) is run statically during each time slice. This discretization is possible by the nature of the requests which are never urgent and can be postponed. The rest of this paper is organized as follows: The second section presents the mathematic model for the B-VRPDR, then in order to solve this problem, an Improved Evolutionary Algorithm (IEA) is proposed in the third section. The experimental results and discussion are reported in the fourth section. Finally, we conclude in the last section.

MATERIALS AND METHODS

In order to model the B-VRPDR, we use the event manager which serves as an interface between the arrival of new orders and the optimization procedure. This manager divides the working day into n_{ts} time slices $T = \{T_1, T_2, Y, T_n\}$, each with an equal length of time T/n_{ts}, where T is the length of the working day. Each of time slices represents a partial Bi-objective Vehicle Routing Problem with Static Requests B-VRPSR where the event manager runs in sequence the solving algorithm on the B-VRPSR problems. From the solutions provided by the algorithm, we decide about commitments within an advanced commitment time, t_{ac}.

The first B-VRPSR created for the first time slice considers all orders left over from the previous working day. The time cut-off, t_{co}, parameter controls the time in which new orders may arrive and thus may leave some customers unserved. These customers are carried over to the next working day. All the orders received after the t_{co} are interpreted as being customers that were not serviced the day before. This means that the optimization starts with customers who would have missed servicing yesterday because of the time cut-off.

The next B-VRPSR will consider all orders received during the previous time slice as well as those which have not been committed to drivers yet. In our simulation, each vehicle m starts from the location of the last customer committed to it with a starting time corresponding to the maximum between the end of the serving time for this customer and the beginning of the next time slice and with a capacity equal to the remaining capacity after serving all customers previously committed to vehicle m.

At the end of each time slice, the best solution is chosen and orders with a processing time (the processing time of an order starts when the vehicle assigned to it has to leave from its previous customer in order to travel to the next customer) starting within the next $T/n_{ts}+t_{ac}$ seconds are committed to their respected vehicles. When any vehicle has used all its capacity, it is sent back to the depot.

Let N_{Tl} the set of the orders known from the previous day if l = 1 and the set of the orders received at the previous time slice and the orders which have not been committed to drivers yet if l0{2, Y, n_{ts}}. At the beginning of the working day, the location of all the vehicles is set at the depot. A static problem B-VRPSR is created in each time slice, after it will be modeled and solved with the procedure that will be described in the following.

We note that O_l = {0_{lm}, ∀m∈0M} is the set of the locations for the vehicles m at the beginning of the time slice T_l where the 0_{lm} is the last customer served by the vehicle m before the beginning of the time slice T_l if; l0{2, Y, n_{ts}} and the depot that corresponds to 0 if l = 1 or if the vehicle m has not been used in the previous time slices T_l(i<l).

The static problem B-VRPSR corresponding to the time slice T_l is defined by an undirected graph $G_l = (V_l, E_l)$ where $V_l = N_{T_l} \cup \{0, O_l\}$ is the set of vertices and:

$$E_l = \{\{(i,j)/i,j \in N_{T_l} \text{ et } i \neq j\} \cup \{(i,j)/i \in O_l \text{ et } j \in N_{T_l}\} \cup \{(i,j)/i \in N_{T_l} \text{ et } j = 0\}\}$$

is the set of edges. A non-negative cost d_{ij} and a travel time t_{ij} are associated with each edge {i, j}0E_i; each vertex i0N_{Tl} has non-negative weights associated with it, namely, a demand q_i and a service time s_i. A set M of identical vehicles of capacity Q at depot 0 is used to visit the customers.

For every time slice, a set of vehicles must serve the customers of N_{Tl} at a least value of our objective and such that the total demand of the vertices visited does not exceed the vehicle capacity and the maximum route duration is limited. Let L is a very large number, Q^l_m is the total quantity ordered by customers already committed to vehicle m before the beginning of time slice T_l and D^l_m is the maximum between the end of the service for 0_{lm} and the beginning of time slice T_i. The decision variables are defined as follows:

$$x_{ij}^m = \begin{cases} 1 & \text{if vehicle m drives from customer i to customer j} \\ 0 & \text{otherwise} \end{cases}$$

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

The static B-VRPSR for each time slice T_l∈T can be formulated as the following integer program.

Objective function: We consider two different objectives. The minimization of the total distance of transport which can be expressed by:

$$f_1 = \text{Min} \sum_{m \in M} \sum_{i,j \in V_l} d_{ij} x_{ij}^m$$

The minimization of the customers response time which corresponds to the time between customers reception and service requests debut. This objective is:

$$f_2 = \text{Min} \sum_{i \in N_{T_l}} (ds_i - td_i)$$

Where:

td_i∈[0, T] = The demand occurrence time
ds_i = The time when the service starts for that demand

Our objective function is composed by two different objectives which aren't on the same scale. To optimize these objectives, we used the aggregation method which combines the various functions of the problem into a single function; so the problem is to minimize:

$$f = \text{Min} \sum_{i=1}^{2} c_i \lambda_i f_i$$

where, reflects the relative importance of the objectives and are the constants which put to the same across the various objectives. The constant is given by where is the optimal solution associated with the only objective function. The objective function is then:

$$f = \text{Min}\ \frac{\lambda_1}{f_1^*}\sum_{m\in M}\sum_{i,j\in V_1} d_{ij}x_{ij}^m + \frac{\lambda_2}{f_2^*}\sum_{i\in N_{T_1}}(ds_i - td_i) \quad i \neq j \qquad (1)$$

Constraints: Our objective is minimized under the following constraints:

$$\sum_{i\in N_{T_1}} x_{0_{lm} i}^m \leq 1 \qquad \forall m\in M \qquad (2)$$

$$\sum_{i\in N_{T_1}} x_{0_{lm} i}^m = \sum_{i\in N_{T_1}} x_{i0}^m \qquad \forall m\in M \qquad (3)$$

$$\sum_{i\in N_{T_1}} y_i^m \leq L.\sum_{j\in N_{T_1}} x_{j0}^m \qquad \forall m\in M \qquad (4)$$

$$\sum_{m\in M}\sum_{i\in N_{T_1}\cup\{O_1\}} x_{ij}^m = 1 \qquad \forall j\in N_{T_1} \qquad (5)$$

$$\sum_{j\in N_{T_1}\cup\{O_1\}} x_{ji}^m = \sum_{j\in N_{T_1}\cup\{0\}} x_{ij}^m \qquad \forall i\in N_{T_1}, \forall m\in M \qquad (6)$$

$$Q_m^1 + \sum_{i\in N_{T_1}} q_i y_i^m \leq Q \qquad \forall m\in M \qquad (7)$$

$$D_m^1 + \sum_{i,j\in V_1} x_{ij}^m t_{ij} + \sum_{i\in N_{T_1}} s_i y_i^m \leq T \qquad \forall m\in M \qquad (8)$$

$$x_{ij}^m \in \{0,1\} \qquad \forall i,j\in V_1,\ \forall m\in M \qquad (9)$$

$$y_i^m \in \{0,1\} \qquad \forall i\in V_1,\ \forall m\in M \qquad (10)$$

Constraints (Eq. 2-4) ensure that all vehicles start at the last customers have been committed to it and return to the depot. Constraint (Eq. 5) guarantees that each node, except the depot is visited by a single vehicle. Furthermore, the constraint (Eq. 6) assures that each node, except the depot is linked only with a pair of nodes, one preceding it and the other following it. Constraint (Eq. 7) ensures that vehicle cannot exceed its capacity. The maximum route duration is limited by Eq. 8. Finally, the constraints (Eq. 9 and 10) guarantee the binary of the decision variables.

An improved evolutionary algorithm for solving the B-VRPDR: The dynamic vehicle routing problem calls for online algorithms that work in real-time since the immediate requests should be served, if possible. As conventional static vehicle routing problems are NP-hard, it is not always possible to find optimal solutions to problems of realistic sizes in a reasonable amount of computation time. This implies that the vehicle routing problem with dynamic requests also belongs to the class of NP-hard problems, since a VRP with static requests should be solved each time a new immediate request is received. In our research, to solve the B-VRPDR, an Improved Evolutionary algorithm (IEA) is executed for each B-VRPSR created at each time slice as described above.

In this study, we adapt our proposed algorithm for each B-VRPSR for solving the B-VRPDR. For that, we maintain a set of solutions, through a fixed number of iterations (Fig. 1). Every solution is assigned a fitness which is directly related to the objective function of this solution. Thereafter, the population of solutions is modified to a new population by applying two operators: Recombination and mutation. At the end of the iteration, a replacement operator is applied to select the solutions for the next generation. We work iteratively by successively applying these three operators in each generation until a termination criterion is satisfied. The proposed evolutionary algorithm is a genetic algorithm with the following differences:

- We adopt a new generation of the initial population instead of doing it in a random way
- We use a new operator of the crossover adapted especially to our problem

These modifications can help us to have good solutions in the shortest time which is very important in the dynamic environment.

Encoding the solutions: The first step in defining an Evolutionary Algorithm is to link the real world to the

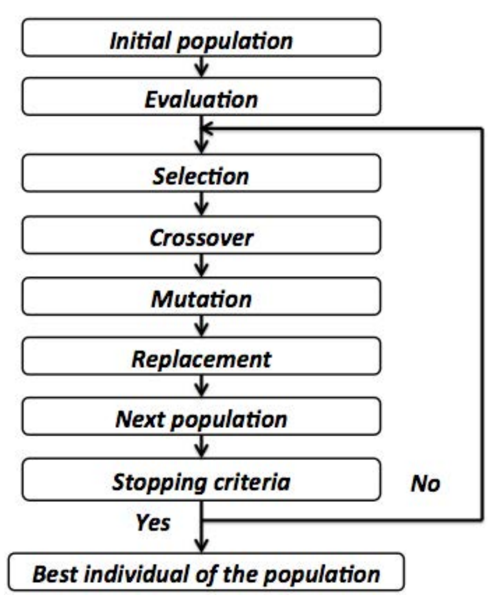

Fig. 1: General structure of an evolutionary algorithm

0_{l1}	1	5	8	0	
0_{l2}	6	3	2	7	0
0_{l3}	9	4	10	0	

Fig. 2: Example of a solution with 3 vehicles and 10 customers

evolutionary algorithm world that is to set up a bridge between the original problem context and the problem solving space where evolution will take place. The choice of an appropriate coding scheme to represent the potential solutions is the key for the proposed algorithm. The representation scheme used in our work is composed of the routes of the vehicles at each time slice T_l. Each solution contains K integers where K is the number of customers of N_{Tl}. As an example, the Fig. 2 represents a solution for a time slice T_l where the route 1 is served by vehicle 1 that visits the ordered list from the first customer, starting with 0_{l1} to the last customer 8 before it returns to the depot.

Initial population: The initial population is generated using a new procedure to create each solution (an individual in the population). In what follows we explain this procedure which generates a feasible solution of high quality to accelerate algorithmic convergence in order to have a good final solution as soon as possible for each time slice T_l: we generate randomly a partial solution S_p for N'_{Tl} which represents 50% of customers selected randomly from N_{Tl}, via the construction of the routing of available vehicles, beginning with the first one and inserting the requests in its trajectory. If the capacity of the vehicle reached the partial capacity Q_{lm}^{p} presented by the relation (Eq. 11), we take the second one, until all the customers of N'_{Tl} have been served. The construction of S_p is done for 50% of N_{Tl} that's why we use the half of capacity Q' (Q' is the total capacity plus the current capacity of the vehicle m) and since the demands for some customers of N'_{Tl} may exceed this capacity making it impossible to insert them, we define the partial capacity as follows:

$$Q_{lm}^{p} = \max(\max_{i \in N_{Tl}} q_i, Q'/2) \qquad (11)$$

This partial capacity is introduced to add the clients of N_{tl}/N'_{Tl} which are inserted into the best positions of the partial solution S_p in order to find a complete solution representing an individual of the population without constraint violations.

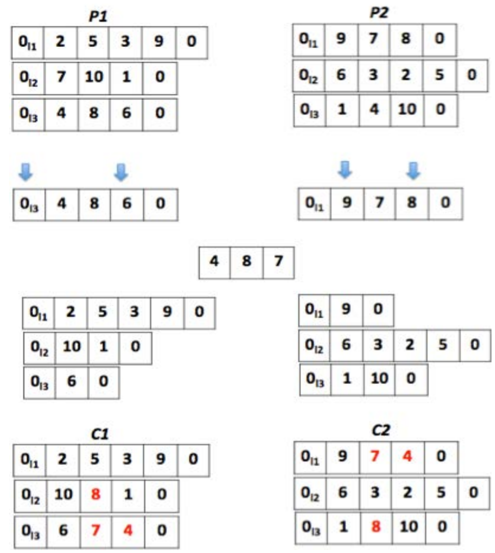

Fig. 3: Example of recombination

Evaluation function (fitness function): The role of the evaluation function is to represent the quality of the solution. It forms the basis for selection and thereby it facilitates improvements. In this research, we define the fitness function of each solution in the population by the inverse of the objective function defined by the relation (Eq. 1):

$$\frac{1}{\frac{\lambda_1}{f_1^*}\sum_{m \in M}\sum_{\substack{i,j \in V_l \\ i \neq j}} d_{ij}x_{ij}^{m} + \frac{\lambda_2}{f_2^*}\sum_{i \in N_{Tl}}(ds_i - td_i)} \qquad (12)$$

Recombination: A binary variation operator is called recombination or crossover. As the names indicate such an operator merges information from two parents into one or two offspring. The idea behind a recombination is to seek the convergence towards a solution that may be best, producing two children with two parents which are randomly selected from the population of individuals. This operator is applied, using a crossover probability p_r, as follows: a route from each parent is randomly selected and the customers presented between two points selected randomly in each route are removed from both parents and reinserted at the location which minimizes our objective in each parent. This requires the computation of the insertion cost for the selected customers in all possible locations of the solution, without constraint violation, creating a new route if no possible insertion for a particular client is found. Figure 3 helps to illustrate how

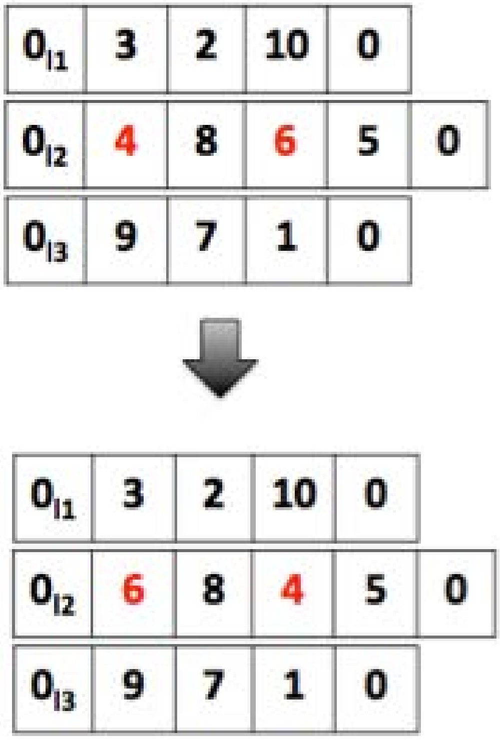

Fig. 4: Example of mutation

our recombination acts. In this example the routes 3 and 1 are selected randomly from P1 and P2, respectively then two points are chosen randomly from the selected routes. The customers 4, 8 and 7 presented between the selected points are removed from both parents, then they are reinserted to the location which minimizes our objective in each parent.

Mutation: After the recombination, the strings are subjected to mutation. This operator prevents the algorithm to be trapped in a local minimum and it has traditionally considered as a simple search operator using a mutation probability p_m. If the recombination is supposed to exploit the current solution to find better ones, the mutation is supposed to help for the exploration of the whole search space. This operator is applied on a randomly selected solution from the current population, using the principle of swapping two random clients in a randomly selected route. Figure 4 we show an example of this operator.

Replacement: During the creation of the new population, Sometimes the good chromosomes can be lost after recombination and mutation results. To avoid this, we form an intermediate population P_{inter} which comprises the current population P_n and the new solutions found from the recombination and the mutation operators, then we use the principle of elitism method where one or more of the best chromosomes are copied in the new generation. In our case, the 50% of the new population P_{n+1} contains the best solutions of P_{inter}, then the other part of P_{n+1} will be completed by randomly selected solutions from P_{inter} and which have not inserted into P_{n+1}. This replacement improves significantly our evolutionary algorithm because it allows that the best solutions can't be lost.

RESULTS AND DISCUSSION

A computational experiment has been conducted to compare the performance of the proposed algorithm with some of the best techniques designed for VRPDR. The algorithm has been tested with 22 VRPDR benchmark problems by Kilby *et al*. (1998). These benchmark instances are derived from the conventional available VRP benchmark data, namely Taillard (1994) (13 benchmarks instances), Christofides and Beasley (1984) (7 benchmarks instances) and Fisher and Jaikumar (1981) (2 benchmarks instances). The number of customers ranges in [50, 385] and the service area may consist of uniformly distributed customers, clustered customers or a combination of both. The proposed algorithm has been implemented in C++ and the experimental tests were carried out on a MacBook Pro-Core i5/ 2.4 GHz-MacOS X 10.7 Lion.

The cut-off time t_{co} and the advanced commitment time t_{ac} are set to T/2 and 0, respectively. The total length of the working day T is 1500 seconds and according to Montemanni *et al*. (2005) the best number of time slices n_{ts} is 25. To evaluate the effectiveness of the proposed approach, this latter must be compared with other works of the literature for that we will present firstly the numerical results by considering only the classic objective which minimizes the total traveled distance. In the following, we study the choice of parameters because it is so important for the success of the proposed approach. To optimize the choice of these parameters, we apply the Taguchi method which is an experimental design that analyzes the effects of several variables (parameters) on the response variable (objective function). The results obtained by the execution of the problem are evaluated by transforming the value of the objective function (Signal/Noise). The S/N rate is calculated using the formula expressed as follows:

$$S/N = -10\log\left(\frac{1}{n}\sum_{i=1}^{n} y_i^2\right) \qquad (13)$$

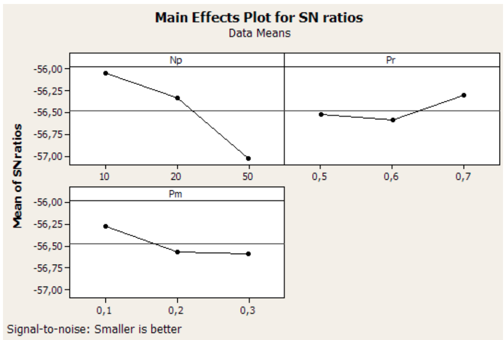

Fig. 5: Graphic of main effects

Table 1: Levels of the IEA parameters

Levels	Np	Pm	Pr
1	20	0.6	0.05
2	50	0.7	0.10
3	100	0.8	0.20

Table 2: The values of the IEA parameters

Parameters	Tp	Pr	Pm
c50	10	0.7	0.1
c75	10	0.6	0.2
c100	10	0.6	0.1
c100b	10	0.5	0.2
c120	10	0.6	0.1
c150	10	0.6	0.2
c199	10	0.5	0.2
tai75a	10	0.6	0.1
tai75b	10	0.5	0.3
tai75c	10	0.7	0.2
tai75d	10	0.5	0.3
tai100a	10	0.6	0.3
tai100b	10	0.5	0.1
tai100c	10	0.6	0.1
tai100d	10	0.6	0.2
tai150a	10	0.6	0.1
tai150b	10	0.7	0.1
tai150c	10	0.7	0.1
tai150d	10	0.5	0.2
tai385	10	0.5	0.1
f71	10	0.7	0.2
f134	10	0.6	0.2

Where:

n = the number of executions

y_i = The objective function value of the solution found during the execution i

We apply the Taguchi design with three levels which consists to define three levels for each factor. The goal is to define the effects of each factor on the response (y). We use Minitab software tool to fix different parameters of our IEA. The influential factors results in our study are defined as follows:

- x_1: population size (Np)
- x_2: mutation probability (Pm)
- x_3: recombination probability (Pr)

We define in Table 1 a 3-level of values for Np, Pmand Pr. Figure 5 shows that the best results are obtained using the following values of different parameters of our approach for the instance c50: Np = 10, Pr = 0.7, Pm = 0.1.

We precede similarly for the other instances and we obtained the results represented in Table 2 where we can see that the population size is fixed for all instances but the crossover rate and the mutation rate change from instance to another.

Table 3 shows the results of our algorithm which correspond to the best value and the average of five runs for 22 different instances of 50-385 requests. Each run is guaranteed to be independent of others by starting with different random seeds. A comparison of the solution quality in terms of minimizing travel distances is done between our IEA and other metaheuristics proposed previously in literature. These metaheuristics are GRASP and ACS algorithms proposed by Montemanni *et al.* (2005), DAPSO and VNS algorithms proposed by Khouadjia *et al.* (2012) and our ACOLNS algorithm (Messaoud *et al.*, 2013).

As we can see our approach is very competitive. It outperforms Montemanni's Grasp algorithm and Montemanni's ACS algorithm on 20 and 16 benchmark instances, respectively. Furthermore, it outperforms Khouadjia's DAPSO algorithm and Khouadjia's VNS algorithm on 8 benchmark instances and it outperforms Messaoud's ACOLNS algorithm on 14 benchmark instances.

The proposed approach is better than all these metaheuristics on 6 benchmark instances: c120, tai75a, tai100a, tai100c, tai100d et tai150c and it finds a feasible solution, unlike the GRASP, ACS, DAPSO and VNS algorithms which are not tested or feasible solutions cannot be obtained for the tai385 benchmark instance. Our approach provides also the shortest average for the travelled distance on 15 benchmark instances.

Despite the constructive aspect of the ACO algorithm has a large effect on results because each ant, during its path construction, selects stations which minimize the total distance, our IEA performs better on certain benchmarks that ACS algorithm (Montemanni *et al*., 2005) and the ACOLNS algorithm (Messaoud *et al*., 2013) due to the new construction to generate an initial population and the crossover proposed which accelerate algorithmic convergence. All these results allow us to say that our IEA is effective and shows the viability to generate very high quality solutions for the VRPDR.

Now, to evaluate our approach to solve B-VRPDR problem which minimizes the total distance and the response time of customers, Table 4 shows our experimental results for this problem where we set the

Table 3: Numerical results of our approach for VRPDR compared to other metaheuristics

	GRASP		ACS		DAPSO		VNS		ACOLNS		IEA	
Parameters	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average
c50	696.92	719.56	631.30	681.86	575.89	632.38	599.53	653.84	601.78	623.09	588.02	606.22
c75	1066.59	1079.16	1009.38	1042.39	970.45	1031.76	981.64	1040.00	1003.20	1013.47	984.92	1008.23
c100	1080.33	1119.06	973.26	1066.16	988.27	1051.5	1022.92	1087.18	987.65	1012.30	992.37	1044.97
c100b	978.39	1022.12	944.23	1023.60	924.32	964.47	866.71	942.81	932.35	943.05	897.07	937.06
c120	1546.50	1643.15	1416.45	1525.15	1276.88	1457.22	1285.21	1469.24	1272.65	1451.60	1267.54	1433.57
c150	1468.36	1501.35	1345.73	1455.50	1371.08	1470.95	1334.73	1441.37	1370.33	1394.77	1388.35	1408.21
c199	1774.33	1898.20	1771.04	1844.82	1640.40	1818.55	1679.65	1769.95	1717.31	1757.02	1707.81	1735.35
tai75a	1911.48	2005.44	1843.08	1945.20	1816.07	1935.28	1806.81	1954.25	1832.84	1880.87	1799.46	1884.81
tai75b	1582.24	1758.88	1535.43	1704.06	1447.39	1484.73	1480.70	1560.71	1456.97	1477.15	1505.52	1544.72
tai75c	1596.17	1674.37	1574.98	1653.58	1481.35	1664.4	1621.03	1746.07	1612.10	1692.00	1492.13	1546.40
tai75d	1545.21	1588.73	1472.35	1529.00	1414.28	1493.47	1446.50	1541.98	1470.52	1491.84	1447.88	1469.88
tai100a	2427.07	2510.29	2375.92	2428.38	2249.84	2370.58	2250.50	2462.50	2257.05	2331.28	2221.60	2285.34
tai100b	2302.95	2512.27	2283.97	2347.90	2238.42	2385.54	2169.10	2319.72	2203.63	2317.30	2246.07	2352.72
tai100c	1599.19	1704.40	1562.30	1655.91	1532.56	1627.32	1490.58	1557.81	1660.48	1717.61	1482.96	1570.31
tai100d	1973.03	2087.55	2008.13	2060.72	1955.06	2123.9	1969.94	2100.38	1952.15	2087.96	1857.59	1963.93
tai150a	3787.53	3899.16	3644.78	3840.18	3400.33	3612.79	3479.44	3680.35	3436.40	3595.40	3491.99	3566.89
tai150b	3313.03	3485.79	3166.88	3327.47	3013.99	3232.11	2934.86	3089.57	3060.02	3095.61	2949.83	3105.61
tai150c	3110.10	3219.27	2811.48	3016.14	2714.34	2875.93	2674.29	2928.77	2735.39	2840.69	2553.78	2702.03
tai150d	3159.21	3298.76	3058.87	3203.75	3025.43	3347.6	2954.64	3147.38	3138.70	3233.39	3099.89	3130.87
tai385	-	-	-	-	-	-	-	-	33062.06	35188.99	31041.99	32157.63
f71	359.16	376.66	311.18	348.69	279.52	312.35	304.32	325.18	311.33	320.00	321.70	335.26
f134	15433.84	16458.47	15135.51	16083.56	15875.00	16645.89	15680.05	16522.18	15557.82	16030.53	15878.91	16346.82

[Best]Best total travel distance obtained over 5 runs, [Average]Average total travel distance obtained over 5 runs, "-" The corresponding problem is not tested or there is no known feasible solution

Table 4: Numerical results of our approach for B-VRPDR

Parameters	TD*	RT*	TD	RS
c50	588.02	4901.59	570.36	5600.20
c75	984.92	6433.53	992.64	7262.28
c100	992.37	10593.35	1010.51	12172.84
c100	897.07	10107.57	935.28	11184.29
c120	1267.54	19428.52	1395.49	22031.29
c150	1388.35	14923.44	1307.26	16418.11
c199	1707.81	18083.74	1708.43	19897.41
tai75	1799.46	10347.57	1890.26	13279.50
tai75	1505.52	8732.53	1634.38	10530.12
tai75	1492.13	8862.02	1738.71	11894.51
tai75	1447.88	11528.94	1451.38	14367.02
tai100	2221.60	16513.48	2403.76	20786.02
tai100	2246.07	15222.56	2240.06	19263.98
tai100	1482.96	11746.74	1716.98	13620.96
tai100	1857.59	13570.27	2090.29	17134.74
tai150	3491.99	35117.63	3542.59	39824.94
tai150	2949.83	28119.96	3020.58	35949.49
tai150	2553.78	26436.89	2911.78	29437.02
tai150	3099.89	27685.10	3266.27	33196.26
tai385	31041.99	237281.59	33504.38	304975.22
f71	321.70	3017.37	308.03	4461.14
f134	15878.91	132737.05	15350.90	146240.70

weights of the total distance and the customers response time at 0.5 ($\lambda_1 = \lambda_2 = 50\%$). The TD* and RT* present the value of our objective when we take into account only the minimization of the total distance and the customers response time, respectively. The TD and RT give to the total distance and the customers response time respectively for the best solution, generated by the proposed approach of 5 runs for $\lambda_1 = 50\%$ and $\lambda_2 = 50\%$.

For the bi-objective optimization, the decision maker can decide which the best solution based on his/her preferences which done by the determination of the value for λ_1 and λ_2. Table 4 shows that the found results indicate the conflicting behaviour between the total distance and the customers response time where the value of our objectives for the case ($\lambda_1 = 50\%$, $\lambda_1 = 50\%$) is steadily increased compared to the TD* and RT* for most instances, at the same time our results show that in general, there isn't a big difference between the distances and the customers response time found when we take into account one objective and both objectives which confirm the robustness of our approach for B-VRPDR.

CONCLUSION

In this research, we have proposed a mathematic model and an improved evolutionary algorithm to solve the bi-objective vehicle routing problem in the dynamic environment. The proposed algorithm is applied to 22 benchmarks instances with up to 385 customers and compared to other works presented in the literature. The computational results show that our approach is competitive in terms of solution quality. As for future work, it may be interesting to combine this approach with other metaheuristics.

REFERENCES

Azi, N., M. Gendreau and J.Y. Potvin, 2012. A dynamic vehicle routing problem with multiple delivery routes. Annl. Oper. Res., 199: 103-112.

Beaudry, A., G. Laporte, T. Melo and S. Nickel, 2010. Dynamic transportation of patients in hospitals. OR Spectrum, 32: 77-107.

Bent, R. and V.P. Hentenryck, 2007. Waiting and relocation strategies in online stochastic vehicle routing. IJCAI., 7: 1816-1821.

Bertsimas, D.J. and V.G. Ryzin, 1991. A stochastic and dynamic vehicle routing problem in the Euclidean plane. Oper. Res., 39: 601-615.

Branchini, R.M., V.A. Armentano and A. Lokketangen, 2009. Adaptive granular local search heuristic for a dynamic vehicle routing problem. Comput. Oper. Res., 36: 2955-2968.

Branke, J., M. Middendorf, G. Noeth and M. Dessouky, 2005. Waiting strategies for dynamic vehicle routing. Transp. Sci., 39: 298-312.

Chen, H.K., C.F. Hsueh and M.S. Chang, 2006. The real-time time-dependent vehicle routing problem. Transp. Res. Logistics Transpor. Rev., 42: 383-408.

Christofides, N. and J.E. Beasley, 1984. The period routing problem. Networks, 14: 237-256.

Creput, J.C., A. Hajjam, A. Koukam and O. Kuhn, 2012. Self-organizing maps in population based metaheuristic to the dynamic vehicle routing problem. J. Combinatorial Optim., 24: 437-458.

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

Elhassania, M.J., B. Jaouad and E.A. Ahmed, 2013. A new hybrid algorithm to solve the vehicle routing problem in the dynamic environment. Int. J. Soft Comput., 8: 327-334.

Fisher, M.L. and R. Jaikumar, 1981. A generalized assignment heuristic for vehicle routing. Networks, 11: 109-124.

Fleischmann, B., S. Gnutzmann and E. Sandvo, 2004. Dynamic vehicle routing based on online traffic information. Trans. Sci., 38: 420-433.

Ghannadpour, S.F., S. Noori, M.R. Tavakkoli and K. Ghoseiri, 2014. A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. Appl. Soft Comput., 14: 504-527.

Guner, A.R., A. Murat and R.B. Chinnam, 2012. Dynamic routing under recurrent and non-recurrent congestion using real-time ITS information. Comput. Oper. Res., 39: 358-373.

Hentenryck, P.V. and R. Bent, 2006. Online Stochastic Combinatorial Optimization. MIT Press, Cambridge, Massachusetts,.

Hvattum, L.M., A. Lokketangen and G. Laporte, 2007. A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems. Networks, 49: 330-340.

Ichoua, S., M. Gendreau and J.Y. Potvin, 2006. Exploiting knowledge about future demands for real-time vehicle dispatching. Transp. Sci., 40: 211-225.

Kaiwartya, O., S. Kumar, D.K. Lobiyal, P.K. Tiwari and A.H. Abdullah *et al*., 2015. Multiobjective dynamic vehicle routing problem and time seed based solution using particle swarm optimization. J. Sens., 2015: 1-14.

Khouadjia, R.M., B. Sarasola, E. Alba, L. Jourdan and E.G. Talbi, 2012. A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests. Appl. Soft Comput., 12: 1426-1439.

Kilby, P., P. Prosser and P. Shaw, 1998. Dynamic VRPs: A study of scenarios. Technical Report APES-06-1998, University of Strathclyde, UK.

Larsen, A. and O.B. Madsen, 2000. The dynamic vehicle routing problem. Ph.D Thesis, Department of Transport Institut for Transport, Technical University of Denmark, Kongens Lyngby, Denmark.

Li, J.Q., P.B. Mirchandani and D. Borenstein, 2009. Real-time vehicle rerouting problems with time windows. Eur. J. Oper. Res., 194: 711-727.

Lianxi, H., 2012. An improved LNS algorithm for real-time vehicle routing problem with time windows. Comput. Opreat. Res., 39: 151-163.

Mes, M., V.D.M. Heijden and V.A. Harten, 2007. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. Eur. J. Oper. Res., 181: 59-75.

Messaoud, E., A.E.H. Alaoui and J. Boukachour, 2013. Hybridization of the Ant colony algorithm with the Large Neighbor Search algorithm for static and dynamic VRPTW resolution. INFOR. Inf. Syst. Oper. Res., 51: 41-51.

Minic, S.M. and G. Laporte, 2004. Waiting strategies for the dynamic pickup and delivery problem with time windows. Transp. Res. Methodol., 38: 635-655.

Montemanni, R., L.M. Gambardella, A.E. Rizzoli and A.V. Donati, 2005. Ant colony system for a dynamic vehicle routing problem. J. Comb. Optim., 10: 327-343.

Mu, Q., Z. Fu, J. Lysgaard and R. Eglese, 2011. Disruption management of the vehicle routing problem with vehicle breakdown. J. Oper. Res. Soc., 62: 742-749.

Novoa, C. and R. Storer, 2009. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. Eur. J. Operations Res., 196: 509-515.

Psaraftis, H.N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transp. Sci., 14: 130-154.

Pureza, V. and G. Laporte, 2008. Waiting and buffering strategies for the dynamic pickup and delivery problem with time windows. INFOR. Inf. Syst. Oper. Res., 46: 165-176.

Qureshi, A.G., E. Taniguchi and T. Yamada, 2012. A microsimulation based analysis of exact solution of dynamic vehicle routing with soft time windows. Proc. Soc. Behav. Sci., 39: 205-216.

Sambola, M.A., E. Fernandez and G. Laporte, 2014. The dynamic multiperiod vehicle routing problem with probabilistic information. Comput. Oper. Res., 48: 31-39.

Secomandi, N., 2000. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. Comput. Operat. Res., 27: 1201-1225.

Taillard, E., 1994. Parallel iterative search methods for vehicle routing problems. Networks, 23: 661-673.

Taillard, E.D., P. Badeau, M. Gendreau, F. Guertin and J.Y. Potvin, 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. Transport. Sci., 31: 170-186.

Thomas, B.W., 2007. Waiting strategies for anticipating service requests from known customer locations. Transp. Sci., 41: 319-331.

Wilson, N. and N. Colvin, 1977. Computer control of the rochester dial-a-ride system. Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Xu, Y., L. Wang and Y. Yang, 2013. Dynamic vehicle routing using an improved variable neighborhood search algorithm. J. Appl. Math., 2013: 1-12.7