Int J Adv Manuf Technol (2016) 85:2505–2523
DOI 10.1007/s00170-015-8060-8

ORIGINAL ARTICLE

Simulation optimization for the vehicle routing problem with time windows using a Bayesian network as a probability model

Ricardo Pérez-Rodríguez[1] · Arturo Hernández-Aguirre[1]

Received: 13 August 2015 / Accepted: 27 October 2015 / Published online: 19 November 2015

Abstract The main purpose of the vehicle routing problem (VRP) is to deliver a set of customers with known demands on minimum-travel routes and terminating at the same depot. The vehicle routing problem with time windows (VRPTW) requires the delivery made in a specific time window for every customer and returning to the depot before a due time. Contrary to current research, an estimation of distribution-algorithm-based approach coupled with a simulation model is proposed and developed to solve the problem and implement the solution. The approach mentioned makes use of a Bayesian network as a probability model to describe the distribution of the solution space. Furthermore, the approach taken in this study combines the key advantages of both estimation of distribution algorithms (EDA) and simulation. The simulation is used to model the VRPTW environment, while the EDA is used to guide the overall search process to identify the best performing ones. Solomon's (Oper Res 35:254–265, 1987) instances served as input and test parameters in order to show that there exists a relationship and interaction between vertices and positions on the sequence of the VRPTW solution. A better position for each vertex on the sequence can be estimated through a Bayesian network. Experimental results show that the EDA performance was better in 70 % of the cases, as average, for the number of vehicles used in all the trails with respect the other algorithms proposed as a benchmark for comparison with the EDA scheme.

Keywords Estimation of distribution algorithm · Vehicle routing problem · Bayesian network · Simulation optimization

1 Introduction

The vehicle routing problem (VRP) arises in the field of transportation, distribution, and logistics. It resembles many real-world problems faced everyday by different companies. In some market sectors, transportation means a high percentage of the value is added to goods. Therefore, the utilization of computerized methods for transportation often results in significant savings ranging from 5 to 20 % in total costs, as reported in Toth and Vigo [34]. The VRP has attracted the attention of a large number of researchers and practitioners. The practical characteristic of the VRP makes its study interesting. There are important advances and new challenges that have been raised in the last number of years due to technological innovations such as global positioning systems and radio-frequency identification. The portfolio of techniques for modeling and solving the standard VRP and its many variants has advanced significantly. Researchers and practitioners have developed faster, more accurate solution algorithms and better models that give them the ability to solve large-scale problems.

The VRP is defined on an undirected network $G = (V,E)$ with a vertex set $V = \{1, 2, \dots , n\}$ and an edge set E. Vertex 1 is the depot to be consistent with Solomon's [28] instances. Each other vertex $i>1$ represents a customer with a known service time s_i, and each edge has a non-negative travel time $t_{ij}= t_{ji}$. The VRP consists of determining a set of m vehicle trips of minimal total time, such that each vehicle starts and ends at the depot, each customer is visited exactly once, and the total demand handled by any vehicle does not exceed the vehicle's capacity Q. The VRP with time windows (VRPTW)

✉ Ricardo Pérez-Rodríguez
ricardo.perez@cimat.mx

[1] CIMAT, A. C. Callejón de Jalisco s/n, Mineral de Valenciana, C.P. 36240 Guanajuato, Mexico

is a variant of the VRP with additional time constraints. Here, the service of a customer should begin within a time window [a_i, b_i]. The vehicle should not arrive earlier than time a_i and no later than time b_i. A vehicle arriving before time a_i will produce a waiting time. A vehicle arriving after time b_i will incur a delay time. All scheduled vehicles have to return to the depot within $[0, b_1]$ where b_1 is the maximal operation time for each vehicle.

Due to its complexities and usefulness in real life, the VRPTW continues to draw attention from researchers and has been a well-known problem in network optimization. Surveys of solution methods for VRPTW can be found in Golden and Assad [14], Desrochers et al. [12], Solomon and Desrosiers [29], Bräysy and Gendreau [5], and Bräysy et al. [6]. Developments can be found in Potvin and Rousseau [24], Rochat and Taillard [26], Thangiah et al. [33], Potvin and Bengio [23], Homberger and Gehring [15], Cordeau et al. [8], Tan et al. [32], Berger and Barkaoui [4], Cordeau et al. [9], and Mester et al. [20].

2 Problem statement

Based on Toth and Vigo [34], the VRPTW can be defined on a directed graph $G=(V,E)$, where $|V| = n+2$, and the depot is represented by the two vertices 1 and $n+1$. Let $N = V \setminus \{1, n+1\}$ be the total number of customer vertices. Feasible vehicle routes correspond to paths starting in vertex 1 and ending at vertex $n+2$. Let K be a set of available vehicles, with $|K|=m$. Let s_i be the service time at vertex i, where the service time for the depot is zero, and t_{ij} is the travel time from vertex i to vertex j. There is a time window $[a_i, b_i]$ associated with all vertices including the depot to be consistent with Solomon's [28] instances. q_i denotes the demand of customer i, and let Q be the vehicle capacity. Two types of variables are presented in this problem, x_{ijk}, $(i,j) \in E$, and $k \in K$, equal 1 if and only if arc (i,j) is used by vehicle k, and continuous variables w_{ik}, $i \in N$, and $k \in K$, showing the time at which vehicle k starts servicing at vertex i. Let $S_i^+ = \{j: (i,j) \in E\}$ and $S_i^- = \{i: (i,j) \in E\}$:

$$minimize \sum_{k\in K} \sum_{(i,j)\in E} t_{ij} x_{ijk} \tag{1}$$

Subject to

$$\sum_{k\in K} \sum_{j\in S_i^+} x_{ijk} = 1, \quad i\in N, \tag{2}$$

$$\sum_{j\in S_1^+} x_{1jk} = 1, \quad k\in K, \tag{3}$$

$$\sum_{i\in S_j^-} x_{ijk} - \sum_{i\in S_j^+} x_{ijk} = 0, \quad k\in K, \ j\in N, \tag{4}$$

$$\sum_{i\in S_{n+1}^-} x_{i,n+1,k} = 1, \quad k\in K, \tag{5}$$

$$x_{ijk}\left(w_{ik} + s_i + t_{ijk} - w_{jk}\right) \le 0, \quad k\in K, (i,j)\in E, \tag{6}$$

$$a_i \le w_{ik} \le b_i, \quad k\in K, i\in V, \tag{7}$$

$$\sum_{i\in N} q_i \sum_{j\in S_i^+} x_{ijk} \le Q, \quad k\in K, \tag{8}$$

$$x_{ijk} \in \{0, 1\}, k\in K, (i,j)\in E \tag{9}$$

The objective function (1) minimizes the total routing time. Constraint (2) states that each customer vertex is visited exactly once, while constraints (3) to (5) ensure that each vehicle is used exactly once and that flow conservation is satisfied at each customer vertex. The consistency of the time variable w_{ik} is ensured through constraint (6) while time windows are imposed by (7) including the depot. These constraints also eliminate subtours. Finally, constraint (8) enforces the vehicle capacity restriction.

3 Literature review

3.1 Current research

Afifi et al. [1] present a simulated annealing-based algorithm for a variant of the VRP, in which a time window is associated with each client service and some services require simultaneous visits from different vehicles to be accomplished. The problem is called the VRPTW and synchronized visits (VRPTWSyn). The algorithm features a set of local improvement methods to deal with various objectives of the problem. Experiments conducted on the benchmark instances from the literature clearly show that their method is fast and outperforms the existing approaches. It produces all known optimal solutions of the benchmark in very short computational times and improves the best results for the rest of the instances.

Berghida and Boukra [3] present a new enhanced biogeography-based optimization (EBBO) algorithm for a complex variant of the vehicle routing problem (VRP) called vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows (HVRPMBTW). This variant is characterized by a limited number of vehicles with various capacities and costs. The vehicles serve two types of customers: linehaul customers and backhaul customers. Each customer must be visited in a specific period of time. They propose to improve and to adapt the biogeography-based optimization (BBO) approach to the problem by integrating simulated annealing algorithm to enhance solution quality at each iteration. This new approach, using Solomon's [28] instances, was tested on benchmarks and produces very satisfactory results compared to other approaches such as particle swarm optimization (PSO).

Cardoso et al. [7] propose an optimization system developed to be integrated with an existing enterprise resource planning (ERP) without causing major disruption to the current distribution process of a company. The proposed system includes a route optimization module, a module implementing the communications within and to the outside of the system, a non-relational database to provide local storage of information relevant to the optimization procedure, and a cartographic subsystem. The proposed architecture is able to deal with dynamic problems included in the specification of the project, namely, arrival of new orders while already optimizing as well as locking and closing of routes by the system administrator. A back-office graphical interface was also implemented, and some results are presented.

de Armas and Melián-Batista [11] tackle two variants of a dynamic vehicle routing problem with time windows as real-world applications of several companies in the Canary Islands, Spain. In these dynamic vehicle routing problems, customer requests can be either known at the beginning of the planning horizon or dynamically revealed during it. In particular, the problems corresponding to a delivery company and a vending machine company are taken into consideration. In addition to the dynamism feature of these problems, the companies consider several attributes that consist of a fixed heterogeneous fleet of vehicles, multiple time windows, customer priorities, and vehicle/customer constraints. This work proposes a general variable neighborhood search (GVNS) as meta-heuristic procedure to solve these problems. The computational experiments, using Solomon's [28] instances, indicate that the proposed method is feasible to solve these real-world problems.

Kabcome and Mouktonglang [16] present a mathematical model to solve the vehicle routing problem with soft time windows (VRPSTW) and distribution of products with multiple categories. In addition, we include multiple compartments and trips. Each compartment is dedicated to a single type of product. Each vehicle is allowed to have more than one trip, as long as it corresponds to the maximum distance allowed in a workday. Numerical results show the effectiveness of their model.

Kaiwartya et al. [17] identify a multi-objective dynamic vehicle routing problem (M-DVRP) and propose a time seed based solution using particle swarm optimization (TS-PSO) for M-DVRP. M-DVRP considers five objectives, namely, geographical ranking of the request, customer ranking, service time, expected reachability time, and satisfaction level of the customers. The multi-objective function of M-DVRP has four components, namely, number of vehicles, expected reachability time, and profit and satisfaction level. Three constraints of the objective function are vehicle, capacity, and reachability. In TS-PSO, first of all, the problem is partitioned into smaller-size DVRPs. Second, the time horizon of each smaller size DVRP is divided into time seeds and the problem is solved in each time seed using particle swarm optimization. The proposed solution has been simulated in ns-2 considering real road network of New Delhi, India, and results are compared with those obtained from genetic algorithm simulations. The comparison confirms that TS-PSO optimizes the multi-objective function of the identified problem better than what is offered by genetic algorithm solution.

Li et al. [18] develop a non-linear mathematical model that considered not only the vehicle routing problem with time windows but also the effect of road irregularities on the bruising of fresh fruits and vegetables. The main objective of this work was to obtain the optimal distribution routes for fresh fruits and vegetables considering different road classes with the least amount of logistics costs. An improved genetic algorithm was used to solve the problem. A fruit delivery route among the 13 cities in Jiangsu Province was used as a real analysis case. The simulation results showed that the vehicle routing problem with time windows, considering road irregularities and different classes of toll roads, can significantly influence total delivery costs compared with traditional VRP models. The comparison between four models to predict the total cost and actual total cost in distribution showed that the improved genetic algorithm is superior to the group-based pattern, CW pattern, and O-X type cross pattern.

Nalepa and Blocho [22] present an adaptive memetic algorithm to solve the VRPTW. Although memetic algorithms have been proven to be extremely efficient in solving the VRPTW, their main drawback is an unclear tuning of their numerous parameters. In their research, Nalepa and Blocho [22] introduce the adaptive memetic algorithm (AMA-VRPTW) for minimizing the total travel distance. In AMA-VRPTW, a population of solutions evolves with time. The parameters of the algorithm, including the selection scheme, population size, and the number of child solutions generated for each pair of parents, are adjusted dynamically during the search. They propose a new adaptive selection scheme to balance the exploration and exploitation of the solution space. Extensive experimental study performed on the well-known Solomon and Gehring and Homberger benchmark sets confirms the efficacy and convergence capabilities of the proposed AMA-VRPTW. They show that it is very competitive compared with other state-of-the-art techniques. Finally, the influence of the proposed adaptive schemes on the AMA-VRPTW behavior and performance is investigated in a thorough sensitivity analysis. This analysis is complemented by the two-tailed

Wilcoxon test for verifying the statistical significance of the results.

Schwarze and Voß [27] analyze the skill VRP where each vehicle is assigned a skill which represents its qualification. A vehicle is able to serve a node if its skill is sufficiently large regarding the skill requirement of the node. In their paper, they move the focus of the cost-oriented skill VRP to time-related aspects. To that end, they add time window restrictions and, second, design an alternative, time-oriented, objective function. This problem extension is motivated by an application in airport ground control where time issues play a major role. They present a mathematical model and carry out an extensive numerical study that includes load-balancing aspects as well as a multi-objective analysis.

Soonpracha et al. [30] contribute to a new challenging study by considering the customer demand as uncertain in a mix-VRPTW. This characteristic increases the difficulty for solving. The meta-heuristic algorithms proposed are developed consisting of a modification of a genetic algorithm and an adaptation of a greedy search hybridized with inter-route neighborhood search methods. Because their paper relates to uncertain customer demands, decision making is performed using the robust approach based on worst case scenarios. The results are evaluated by using the extra cost and the unmet demand against the deterministic approach to balance the decision making.

Sun and Wang [31] formulate a solution procedure for vehicle routing problems with uncertainty (VRPU) with regard to future demand and transportation cost. Unlike E-SDROA (expectation semideviation robust optimisation approach) for solving the proposed problem, the formulation focuses on robust optimization considering situations possibly related to bidding and capital budgets. Besides, numerical experiments showed significant increments in the robustness of the solutions without much loss in solution quality. The differences and similarities of the robust optimization model and existing robust optimisation approaches were also compared.

Wang et al. [36] present a vehicle route optimization model in consideration of customer characteristics with three major components: (1) A hierarchical analysis structure is developed to convert customers' characteristics into linguistic variables, and fuzzy integration method is used to map the sub-criteria into higher hierarchical criteria based on the trapezoidal fuzzy numbers; (2) a fuzzy clustering algorithm based on Axiomatic Fuzzy Set is proposed to group the customers into multiple clusters; and (3) the fuzzy technique for order preference by similarity to ideal solution (TOPSIS) approach is integrated into the dynamic programming approach to optimize vehicle routes in each cluster. A numerical case study in Anshun, China, demonstrates the advantages of the proposed method by comparing with the other two prevailing algorithms. In addition, a sensitivity analysis is conducted to capture the impacts of various evaluation criteria weights. The results indicate that their approach performs very well to identify similar customer groups and incorporate individual customer's service priority into VRP.

Yang et al. [37] investigate and compare heuristic algorithms for the VRPTW. Three heuristic algorithms were proposed firstly through combining three classical heuristic algorithms with the cross exchange method and the 2-opt exchange heuristic respectively. The proposed algorithms are then compared with the three classical algorithms based on publicly available benchmark problems. The comparison results using Solomon's [28] instances show the effectiveness of the proposed algorithms and their superiority to the classical algorithms.

Barbucha [2] proposes a new approach for the VRPTW, which integrates the asynchronous team paradigm with the island-based evolutionary algorithm concept. The process of solving the problem is carried out by the set of agents, each representing a heuristic algorithm, operating on a population of individuals stored in the common sharable memory. Agents are grouped in teams working in islands. Each team of agents periodically communicates with other teams by sharing a promising result. Computational experiment using Solomon's [28] instances confirmed effectiveness of the proposed research.

Gan et al. [13] present a variant of the VRPTW named multi-type vehicle routing problem with time windows (MT-VRPTW), which considers both multiple types of the vehicle and the uncertain number of vehicles of various types. As a consequence, the different combinations of multi-type vehicle will lead to diverse results, which should be evaluated by its own fitness function. In order to solve the proposed MT-VRPTW problem, six variants of particle swarm optimization (PSO) are used. The $2N$ dimensions encoding method is adopted to express the particle (N represents the number of demand points). In the simulation studies, the performances of the six PSO variants are compared and the obtained results are analyzed.

Li et al. [19] study a new variant of multi-depot vehicle routing problem with time windows (MDVRPTW) called multi-depot vehicle routing problem with time windows under shared depot resources (MDVRPTWSDR). In the new variant, the depot where the vehicle ends is flexible. Thus, it is not entirely the same as the depot that it starts from. An integer programming model is formulated with the minimum total traveling cost under the constraints of time window, capacity and route duration of the vehicle, the fleet size, and the number of parking spaces of each depot. As the problem is an NP-hard problem, a hybrid genetic algorithm with adaptive local search is proposed to solve it. Finally, the computational results show that the proposed method is competitive in terms of solution quality. Compared with the classic MDVRPTW, allowing flexible choice of the stop depot can further reduce total traveling costs.

3.2 EDA

The estimation of distribution algorithms (EDA) introduced by Mühlenbein and Paaß [21] is a relatively new paradigm in the field of evolutionary computation. Compared with other evolutionary algorithms, the EDA reproduces new population without using traditional evolutionary operators. In the EDA, a probability model of the most promising area is built by statistical information based on the search experience, and then the probability model is used for sampling to generate new individuals. The EDA makes use of the probability model to describe the distribution of the solution space. The updating process reflects the evolutionary trend of the population [35]. To describe the distribution of the solution space, the EDA tries to determine the relationship or interaction among variables of the problem as primary objective. The traditional evolutionary operators are replaced by the probability model which is built with information about relationships and interactions among variables. The main idea is to learn and benefit from the interaction among variables by estimating the distribution of the population and sample from this distribution of the offspring. Although the interaction may or may not be present, generally, this is explicitly unknown even in small-size VRPTW problems. In this research, an EDA is proposed in order to show that there exists a relationship and interaction between vertices and positions on the sequence of the VRPTW solution. Such a relationship should be used and exploited in order to get competitive solutions. In addition, the probability model should be able to represent the characteristics exhibited by the parents as a distribution of the solution space. Therefore, the EDA proposed does not need to hold members of the population to keep track and inherit the characteristics mentioned in each generation as other algorithms. The characteristics persist for all members of the population in the evolutionary progress by the probability model proposed. In this paper, the EDA makes use of a Bayesian network as a probability model to describe the distribution of the solution space.

The reason to propose an EDA for the VRPTW is to show that a better position for each vertex on the sequence can be estimated through a probability model that contains information on relationships and interactions among vertices and positions on the sequence. Such information allows for an avoidance of using the traditional evolutionary operators for permutation-based problems.

In all this current research, the vertices are independent of each other. It means that any vertex can be relocated to another position on the sequence (giant tour) in order to get better solutions. However, most published papers consider that the rest of the vertices are not necessarily affected (influenced) by the relocating processes because it is unknown if a dependency situation exists among them. Finally, although different algorithms have been proposed in order to solve the VRPTW problem successfully, these published algorithms consider the vertices independent of each other. It could be disadvantageous if a relation or interaction among vertices in the problem exists. To preserve any relationship or interaction using a Bayesian network as a probability model is the objective and has not been considered in all this current research. The Bayesian network paradigm is used mainly to reason in domains with an intrinsic uncertainty. The reasoning inside the model, that is, the propagation of the evidence through the model, depends on the structure reflecting the conditional (in) dependencies between the variables. In addition, we try to identify higher-order interactions to enhance the solution quality, i.e., to capture the problem structure with more precision than other evolutionary algorithms. To the best of our knowledge, probability models by means of an EDA have not been used to tackle the VRPTW problem. Furthermore, the approach taken in this study combines the key advantages of both EDA and simulation. The simulation is used to model the VRPTW environment, while the EDA is used to guide the overall search process to identify the best performing ones. In specific, the output of the simulation model built on Delmia Quest®, i.e., the total routing time, is used by the EDA written in DevC++® to provide feedback on the progress of the search for the best solution. This in turn guides further input to the simulation model. Finally, the EDA have not been developed for use as an optimization method for simulation optimization.

4 EDA—for the VRPTW problem

4.1 Solution representation

We adopted a permutation-based representation like in most genetic algorithms (GAs) for the travelling salesman problem (TSP). The representation mentioned is a vector, which shows a giant tour of all customers. In each position on the vector, a vertex represents a specific customer. Therefore, a vector is simply a sequence of vertices (permutation) π of n vertices without route delimiters. It may be interpreted as the order in which a vehicle must visit all customers, if the same vehicle performs all routes one by one. The fitness $F(\pi)$ of π is the total routing time detailed on (1) from the problem statement section. Although this solution representation without route delimiters is suitable for using traditional evolutionary operators such Prins' [25] research, our approach does not require those operators in order to get competitive offspring.

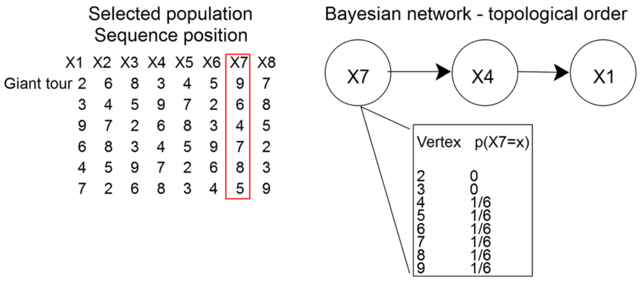

Fig. 1 A simple example to build a probability matrix *P*

4.2 Generation of the initial population

The population is implemented as an array of M vectors of size *n* vertices each one. Each vector is initialized as a random permutation of vertices (customers).

4.3 Selection rule

A subset *N* of *M* parents (where *N*<*M*) are selected by means of the bubble method. It means that *N* parents are selected according to their fitness value, the total routing time, in 55 % of the time and 45 % of the time according to any delays. We can see the delays as penalizations of the corresponding vectors in M. A delay is made when a vehicle arrives after the b_i at customer *i*.

4.4 Probability model proposed

To assess a Bayesian network, the user needs to specify

- A structure by means of a directed acyclic graph which reflects the set of conditional (in)dependencies among the variables
- The unconditional probabilities for all root nodes (nodes with no predecessors)
- Conditional probabilities for all other nodes, given all possible combinations of their direct predecessors.

The structure and conditional probabilities necessary for characterizing the Bayesian network can be provided either externally by experts or by automatic learning from a database of cases. A big number of model learning algorithms have been proposed in the literature. This research uses a Bayesian network model induction by detecting conditional (in)dependencies, i.e., the methodology for constructing the structure of a Bayesian network from data is based on constraint-based algorithms (also known as dependency analysis methods) where the possibility of representing probabilistic dependence/independence relationships in probabilistic models using graphical models. Therefore, it is possible to test the dependence/independence relationships, contained implicit in the data, with a certain conditional independence measure in order to construct such a network. The conditional independence measures used in this research are the mutual and conditional mutual information. The basic idea is that the mutual information measure can form a descending succession of variables; i.e., the variables can be ordered according to the information gain they provide to a dependent variable from maximum to minimum (ancestral ordering). If such an ordering exists, then it is possible to induce a directed acyclic graph (DAG). By definition, a Bayesian network is a DAG so, once this ancestral ordering is induced, it is possible to build a Bayesian network from that ordering. A Bayesian network is built as a probabilistic graph model to represent the dependency among variables. At a fundamental level, the algorithm for building the Bayesian network is based on Cruz-Ramírez and Martínez-Morales [10]. It works as follows:

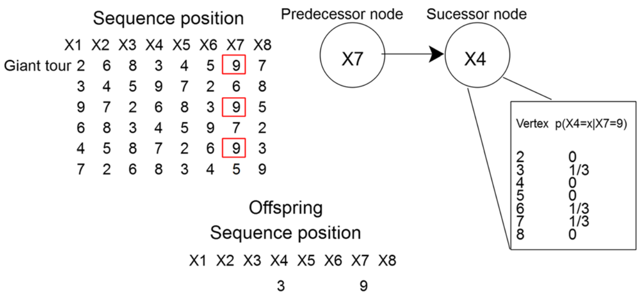

Fig. 2 A simple example to sample new offspring from the probability model

```
Let X be the dependent random variable, i.e., a position sequence, and let $Y_1, Y_2, \ldots, Y_n$ be the unordered independent variables, i.e., the rest of positions sequence.
Begin
  1. Compute the value of the mutual information $I(X|Y_i)$ that each variable $Y_i$ ($1 \leq i \leq n$) provides to X and order them according the information they provide from the largest to the smallest value. The ordered variables will be labeled now $Y_{(1)}, Y_{(2)}, \ldots, Y_{(n)}$.
  2. Draw and arc from $Y_{(1)}$ to X. Check whether the null hypothesis $H_0$ (two variables are independent of each other) holds or not. If $H_0$ does not hold, draw a directed arc from $Y_{(i)}$ $(2 \leq i \leq n)$ to X.
  3. Compute the value of the conditional mutual information $I(X, Y_{(i)} \mid Y_{(1)})$ that each variable $Y_{(i)}$ $(2 \leq i \leq n)$ provides to X given $Y_{(1)}$. Check whether the null hypothesis $H_0$ (two variables are independent from each other given a set of variables, i.e., $Y_{(1)}$) holds or not. If $H_0$ holds, remove the arc from $Y_{(i)}$ to X. If $H_0$ does not hold, leave the arc intact.
  4. Do $X = Y_{(1)}$ and delete $Y_{(1)}$ from the set of independent variables. Let the number of independent variables $n = n - 1$. If $n \geq 2$, then go to step 1; otherwise, stop.
Endbegin
```

If only the variable, which provides the biggest amount of information, is considered, then this variable will be the only one forming part of the conditioning set and, because the rest of the variables give little information, they can be discarded as forming part of that conditioning set. It will become evident why this strong assumption is, in general, not enough to produce accurate and sound results. However, it can be claimed that the Bayesian network used in this research is efficient in the sense that it does not test all the possible associations (independence tests) among variables but only those, which could be significant, based on the information gain they provide. In doing this, the Bayesian network used in this research avoids an exponential complexity on the number of such independence tests. In order to construct a Bayesian network from data, the Bayesian network used in this research first performs the necessary independence tests (marginal or conditional); then, based on those results, it checks whether the null hypothesis holds or not. If the independence hypothesis (the null hypothesis) does not hold, then the Bayesian network used in this research draws an arc from the independent variables (Y_1, Y_2,…, Y_n) to the dependent one (X). In other words, the Bayesian network used in this research first assumes that all the variables are disconnected and then starts drawing arcs among them when this is the case. This class of algorithms is known as stepwise forward algorithms.

After that, a topological order of the Bayesian network built is necessary for the sampling process.

4.5 Sampling process

For the root nodes in the Bayesian network (according to the topological order), vertex i is selected with the probability p_i. To build the probability matrix P of the root nodes, we need to compute the number of times that the vertex i appears on the root nodes' position in the selected population of size N. The result obtained

Table 1 Comparative results for R1 Solomon's [28] instances

Item	Solomon's instance	UMDA			COMIT			EDA—for the VRPTW problem		
		Vehicles	Routing time	Delays	Vehicles	Routing time	Delays	Vehicles	Routing time	Delays
1	R101.25	9	698	6	8	665	4	10	767	**0**
2	R101.50	20	1404	21	18	1296	11	**16**	**1251**	**0**
3	R102.25	9	656	6	9	653	0	**7**	732	**0**
4	R102.50	17	1473	14	17	1210	7	**14**	1560	**0**
5	R103.25	9	679	1	8	726	0	**8**	693	**0**
6	R103.50	15	1274	8	15	1545	5	**15**	1750	**0**
7	R104.25	7	693	1	8	697	0	**7**	**574**	**0**
8	R104.50	14	1461	4	13	1392	1	**13**	1412	**0**
9	R105.25	9	690	3	8	624	1	9	730	**0**
10	R105.50	16	1480	14	19	1611	4	**15**	**1444**	**0**
11	R106.25	9	752	1	9	691	0	**8**	**598**	**0**
12	R106.50	16	1497	10	15	1456	5	**14**	1680	**0**
13	R107.25	8	700	2	8	757	0	**7**	**693**	**0**
14	R107.50	14	1519	8	14	1373	3	**14**	1598	**0**
15	R108.25	7	683	1	7	729	0	**6**	**652**	**0**
16	R108.50	13	1457	1	14	1425	0	**12**	**1300**	**0**
17	R109.25	8	703	2	8	648	0	**8**	742	**0**
18	R109.50	17	1526	10	15	1399	4	16	**1265**	**0**
19	R110.25	7	737	1	7	636	0	**7**	689	**0**
20	R110.50	16	1466	6	15	1412	1	16	**1305**	**0**
21	R111.25	8	741	3	9	799	0	**7**	**666**	**0**
22	R111.50	17	1407	5	15	1523	1	**15**	1710	**0**
23	R112.25	6	664	0	7	772	0	**6**	685	**0**
24	R112.50	14	1306	1	14	1572	0	**13**	**1557**	**0**

Bold entries are the best value for the instance

UMDA Univariate Marginal Distribution Algorithmm, *COMIT* Combining Optimizers with Mutual Information Trees, *EDA* estimation of distribution algorithms, *VRPTW* vehicle routing problem with time windows

previously is divided by the number of individuals considered in the selected population, i.e., N. A simple example is shown in Fig. 1, where the node number 7 is a root node; it means that the position number 7 on the sequence should be selected at the beginning. The corresponding column contains six different vertices, i.e., 9, 6, 4, 7, 8, and 5, in the selected population. Those vertices are in red rectangle. Each vertex appears only once in the column mentioned; therefore, the probability is 1/6 for each one. Vertices 2 and 3 are not considered for the position number 7. Via sampling according to each probability matrix, new vertices are generated.

Once the root nodes have a vertex associated, the probability matrices for the successor nodes (according to the topological order) are built according to the predecessor nodes' result, i.e., the vertices sampled previously. To build the probability matrix P of each successor node, we need to compute the number of times that the all vertices $n \setminus i$ appear on the successor node's position in the selected population of size N when the set of vertices i appears on the predecessor nodes' position. The result obtained previously is divided by the number of times that the set of vertices i appears on the predecessor nodes' position. Therefore, the positions on the vector solution are sampled according to the Bayesian network. If the vertex i has already been chosen, it means the assignment of the vertex i cannot be selected again. Then, the probability matrix for each successor node should not consider the vertices chosen previously. This updating mechanism considers the previous assignments. A simple example is shown in Fig. 2, where the node number 7 is a predecessor node, which contains the vertex 9 already selected. The node number 4 is a successor node. There are three potential vertices in order to select, i.e., the vertices 3, 6, and 7 (in red rectangle). Those vertices appear only once in the corresponding column from the selected population. Therefore, the probability is 1/3 for each one.

The Bayesian network, i.e., the dependency graph model, describes the characteristics exhibited by the parents as a distribution of the solution space. Therefore, the algorithm proposed does not need to hold members

Table 2 Comparative results for R2 Solomon's [28] instances

Item	Solomon's instance	UMDA			COMIT			EDA—for the VRPTW problem		
		Vehicles	Routing time	Delays	Vehicles	Routing time	Delays	Vehicles	Routing time	Delays
1	R201.25	3	856	7	3	678	4	**3**	774	**0**
2	R201.50	7	1825	20	6	1679	16	**4**	**1807**	**0**
3	R202.25	3	743	3	2	816	1	**3**	928	**0**
4	R202.50	5	1710	11	5	1502	8	**5**	2005	**0**
5	R203.25	3	944	0	3	808	0	**3**	**721**	**0**
6	R203.50	5	1863	7	5	1612	3	**5**	2069	**0**
7	R204.25	2	802	0	2	967	0	**2**	**713**	**0**
8	R204.50	4	1812	0	4	1710	0	**3**	**1457**	**0**
9	R205.25	3	906	4	3	857	0	**3**	**855**	**0**
10	R205.50	5	1857	15	5	1781	9	**5**	1801	**0**
11	R206.25	3	890	0	3	967	0	**3**	**742**	**0**
12	R206.50	4	1793	10	5	1532	5	**4**	2024	**0**
13	R207.25	2	766	0	2	752	0	**2**	1013	**0**
14	R207.50	4	1610	4	4	1452	2	5	2066	**0**
15	R208.25	2	791	0	2	919	0	**2**	**712**	**0**
16	R208.50	4	1764	0	4	1805	0	**3**	**1680**	**0**
17	R209.25	3	855	3	3	854	0	**3**	**790**	**0**
18	R209.50	5	1811	9	4	1438	7	**3**	**1581**	**0**
19	R210.25	3	868	1	3	792	0	**3**	**657**	**0**
20	R210.50	4	1843	10	4	1623	6	**4**	**1821**	**0**
21	R211.25	2	826	1	2	875	0	**2**	868	**0**
22	R211.50	4	1767	4	4	1537	0	5	1793	**0**

Bold entries are the best value for the instance

UMDA Univariate Marginal Distribution Algorithmm, *COMIT* Combining Optimizers with Mutual Information Trees, *EDA* estimation of distribution algorithms, *VRPTW* vehicle routing problem with time windows

Table 3 Comparative results for C1 Solomon's [28] instances

Item	Solomon's instance	UMDA			COMIT			EDA—for the VRPTW problem		
		Vehicles	Routing time	Delays	Vehicles	Routing time	Delays	Vehicles	Routing time	Delays
1	C101.25	4	515	24	5	234	4	9	**425**	**0**
2	C101.50	9	1315	49	9	932	15	20	1281	**0**
3	C102.25	3	501	17	5	346	2	7	449	**0**
4	C102.50	7	1309	36	7	851	15	16	868	**0**
5	C103.25	3	554	11	4	382	0	5	500	**0**
6	C103.50	6	1359	24	7	970	6	13	**871**	**0**
7	C104.25	3	523	7	4	374	0	4	495	**0**
8	C104.50	6	1261	9	6	578	1	7	890	**0**
9	C105.25	5	508	9	5	421	3	8	456	**0**
10	C105.50	9	1374	24	9	1059	12	19	846	**0**
11	C106.25	5	419	18	5	283	4	9	465	**0**
12	C106.50	10	1163	32	10	987	13	19	**962**	**0**
13	C107.25	5	380	6	4	408	3	8	430	**0**
14	C107.50	9	1360	19	9	1040	12	18	1310	**0**
15	C108.25	4	492	6	4	357	2	7	493	**0**
16	C108.50	8	1206	17	9	886	6	16	1202	**0**
17	C109.25	3	425	7	4	373	0	**3**	703	**0**
18	C109.50	8	1209	11	8	789	4	11	**770**	**0**

Bold entries are the best value for the instance

UMDA Univariate Marginal Distribution Algorithmm, *COMIT* Combining Optimizers with Mutual Information Trees, *EDA* estimation of distribution algorithms, *VRPTW* vehicle routing problem with time windows

Table 4 Comparative results for C2 Solomon's [28] instances

Item	Solomon's instances	UMDA			COMIT			EDA—for the VRPTW problem		
		Vehicles	Routing time	Delays	Vehicles	Routing time	Delays	Vehicles	Routing time	Delays
1	C201.25	4	647	8	4	459	4	8	620	**0**
2	C201.50	6	1505	25	6	1505	18	19	**1492**	**0**
3	C202.25	3	575	5	3	559	3	5	560	**0**
4	C202.50	6	1483	13	6	1151	10	14	1488	**0**
5	C203.25	3	593	1	3	589	0	**3**	652	**0**
6	C203.50	6	1548	6	6	1132	3	10	1575	**0**
7	C204.25	3	652	0	3	528	0	**3**	570	**0**
8	C204.50	3	1415	1	3	1335	0	4	**790**	**0**
9	C205.25	3	490	7	3	496	4	7	**444**	**0**
10	C205.50	6	1460	22	6	1286	15	17	1385	**0**
11	C206.25	3	520	4	3	416	2	5	523	**0**
12	C206.50	6	1597	16	5	1125	13	14	1005	**0**
13	C207.25	2	553	8	3	590	2	4	**490**	**0**
14	C207.50	4	1298	16	5	1022	8	10	1185	**0**
15	C208.25	3	613	5	3	469	1	6	**445**	**0**
16	C208.50	5	1614	13	4	1036	12	12	**1018**	**0**

Bold entries are the best value for the instance

UMDA Univariate Marginal Distribution Algorithmm, *COMIT* Combining Optimizers with Mutual Information Trees, *EDA* estimation of distribution algorithms, *VRPTW* vehicle routing problem with time windows

Table 5 Comparative results for RC1 Solomon's [28] instances

Item	Solomon's instances	UMDA			COMIT			EDA—for the VRPTW problem		
		Vehicles	Routing time	Delays	Vehicles	Routing time	Delays	Vehicles	Routing time	Delays
1	RC101.25	9	753	4	7	410	1	8	628	**0**
2	RC101.50	23	2015	12	21	1572	5	**19**	**1604**	**0**
3	RC102.25	8	672	3	7	457	0	**8**	818	**0**
4	RC102.50	19	1966	11	19	1692	4	20	**1550**	**0**
5	RC103.25	8	616	1	7	607	0	**7**	647	**0**
6	RC103.50	18	2105	7	20	1968	2	19	**1341**	**0**
7	RC104.25	8	847	0	8	714	0	**6**	**583**	**0**
8	RC104.50	18	1781	1	18	1988	0	**14**	**1358**	**0**
9	RC105.25	9	613	2	8	524	0	**8**	703	**0**
10	RC105.50	21	1856	10	18	1583	5	**18**	1973	**0**
11	RC106.25	8	667	1	8	708	0	**8**	**655**	**0**
12	RC106.50	19	2093	10	15	1167	2	**15**	1696	**0**
13	RC107.25	8	728	2	6	439	0	7	603	**0**
14	RC107.50	20	2040	3	19	1654	0	**15**	**1364**	**0**
15	RC108.25	7	697	0	8	864	0	**6**	**491**	**0**
16	RC108.50	17	1987	4	19	1979	0	**15**	**1345**	**0**

Bold entries are the best value for the instance

UMDA Univariate Marginal Distribution Algorithmm, *COMIT* Combining Optimizers with Mutual Information Trees, *EDA* estimation of distribution algorithms, *VRPTW* vehicle routing problem with time windows

Table 6 Comparative results for RC2 Solomon's [28] instances

Item	Solomon's instances	UMDA			COMIT			EDA—for the VRPTW problem		
		Vehicles	Routing time	Delays	Vehicles	Routing time	Delays	Vehicles	Routing time	Delays
1	RC201.25	3	1035	4	3	699	3	4	1116	**0**
2	RC201.50	6	2384	20	8	2521	13	6	2853	**0**
3	RC202.25	3	943	3	3	666	0	**3**	988	**0**
4	RC202.50	6	2531	13	5	1677	9	6	2561	**0**
5	RC203.25	3	817	0	3	862	0	**3**	**814**	**0**
6	RC203.50	6	2654	6	6	1758	3	**6**	2805	**0**
7	RC204.25	2	974	0	2	979	0	**2**	**669**	**0**
8	RC204.50	5	2721	0	5	2377	0	**4**	**1272**	**0**
9	RC205.25	3	1138	6	3	757	2	4	1081	**0**
10	RC205.50	6	2561	15	6	1650	10	**6**	2718	**0**
11	RC206.25	2	856	4	3	663	0	3	1206	**0**
12	RC206.50	7	2540	9	6	1948	7	**6**	2823	**0**
13	RC207.25	3	760	2	2	634	1	3	958	**0**
14	RC207.50	6	2819	7	5	1690	3	7	2872	**0**
15	RC208.25	3	1078	0	2	998	0	**2**	**765**	**0**
16	RC208.50	5	2346	3	5	2155	0	**5**	**1628**	**0**

Bold entries are the best value for the instance

UMDA Univariate Marginal Distribution Algorithmm, *COMIT* Combining Optimizers with Mutual Information Trees, *EDA* estimation of distribution algorithms, *VRPTW* vehicle routing problem with time windows

of the population to keep track and inherit the characteristics mentioned in each generation. The characteristics persist for all members of the population in the evolutionary progress by the Bayesian network.

4.6 Split criteria

Once the sequence (permutation) π of n vertices without route delimiters was obtained by the dependency graph model, a procedure should be done in order to get the feasible routes for all vehicles required. Because a sequence can be split into many different routes, Prins [25] proposed a splitting procedure which can find an optimal splitting, i.e., a trip delimiter so that the total routing time is minimized. Prins' [25] procedure is used in this research. Without loss of generality, let $\pi = (2, 3, 4, \ldots, n)$ be a given sequence. Consider an auxiliary graph $H = (\overline{V}, \overline{E})$ where $\overline{V} = \{1, 2, 3, \ldots, n+1\}$. An arc $(i, j) \in \overline{E}$. Two labels V_j and P_j for each vertex j in π are computed. V_j is the total routing time of the shortest path from node 1 to node j in H, and P_j is the predecessor of j on this path. The minimal total routing time is given at the end by V_n. For any given i, note that the increment of j stops when the due time depot b_1 or the capacity of the vehicle Q is exceeded. The Prins' [25] algorithm can be described as follows:

```
V_0 := 0
for i := 1 to n do $V_i$ :=+∞ endfor
for i := to n do
routing time := 0; load := 0; j := i;
repeat
load := load + $q_{\pi_j}$
        if i=j then
            routing time := $t_{0,\pi_j} + s_j + t_{\pi_j,0}$
        else
            routing time := routing time − $t_{\pi_j-1,0} + t_{\pi_j-1,\pi_j} + s_j + t_{\pi_j,0}$
        endif
        if (routing time<$b_1$) and (load<Q) then
            if $V_{i-1}$+routing time<$V_j$ then
                $V_j$ := $V_{i-1}$+routing time
                $P_j$ := i − 1
            endif
            j := j+1
        endif
until (j>n) or (routing time>$b_1$) or (load>Q)
end for
```

In addition, we consider not only the due time depot b_1 but also the time window of the vertices in the procedure mentioned. The Prins' [25] algorithm modified can be described as follows:

```
V_0 := 0
for i := 1 to n do $V_i$ :=+∞ endfor
for i := to n do
routing time := 0; load := 0; j:= i;
repeat
load := load + $q_{\pi_j}$
        if i=j then
            routing time := max $[t_{0,\pi_j}, a_j] + s_j + t_{\pi_j,0}$
            arrival := max $[t_{0,\pi_j}, a_j]$
            real service := arrival+ $s_j$
        else
            routing time := max [ routing time − $t_{\pi_j-1,0} + t_{\pi_j-1,\pi_j}, a_j$ ] + $s_j + t_{\pi_j,0}$
            arrival :=max[ real service+ $t_{\pi_j-1,\pi_j}$ , $a_j$ ]
            real service := arrival+ $s_j$
        endif
        if (routing time<$b_1$) and (load<Q) and (arrival<$b_j$) then
            if $V_{i-1}$+routing time<$V_j$ then
                $V_j$ := $V_{i-1}$+routing time
                $P_j$:= i − 1
            endif
            j:= j+1
        endif
until (j>n) or (routing time>$b_1$) or (load>Q)
end for
```

The vector of labels P_j is kept with the sequence to extract the VRPTW solution, using the Prins' [25] splitting procedure shown below. It builds up to n tours (worst case with one vehicle per demand). Each tour is a list of vertices, possibly empty. The procedure *insertqueue* adds a node at the end of a tour:

```
for i := 1 to n do trip (i) := 0 endfor
t := 0
j := 0
repeat
t := t+1
t := $P_j$
for k := i+1 to j do insertqueue(trip(t),$\pi_k$) endfor
j := i
until i=0
```

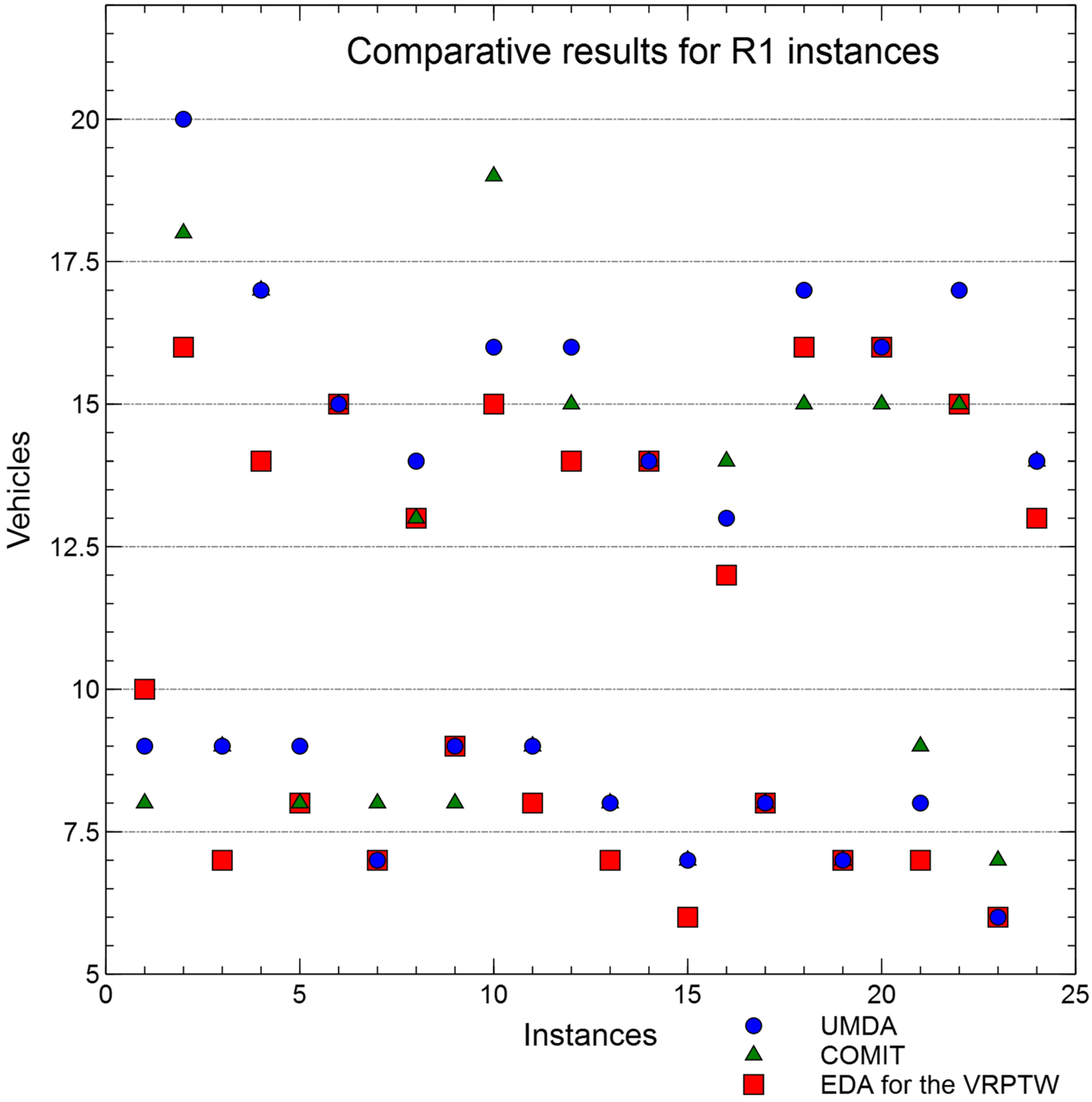

Fig. 3 Comparative results for R1 instances

4.7 EDA–simulation model

The input of the simulation model is a set of instructions in order to execute the tours according to the split criteria mentioned previously. The output of the simulation model, the total routing time, is used by EDA to provide feedback on the progress of the search for the best solution. This in turn guides further input to the simulation model.

5 Results and comparison

Solomon's [28] instances served as input and test parameters in order to show that there exists a relationship and interaction between vertices and positions on the sequence of the VRPTW solution. A better position for each vertex on the sequence can be estimate through the probability model.

There are six sets of problems in Solomon's [28] instances. Their design highlights several factors that affect the behavior of routing and scheduling algorithms. These are geographical data, the number of customers serviced by a vehicle, percent of time-constrained customers, and tightness and positioning of the time windows. The geographical data are randomly generated in problem sets R1 and R2, clustered in problem sets C1 and C2, and a mix of random and clustered structures in problem sets by RC1 and RC2. Problem sets R1, C1, and RC1 have a short scheduling horizon and allow only a few

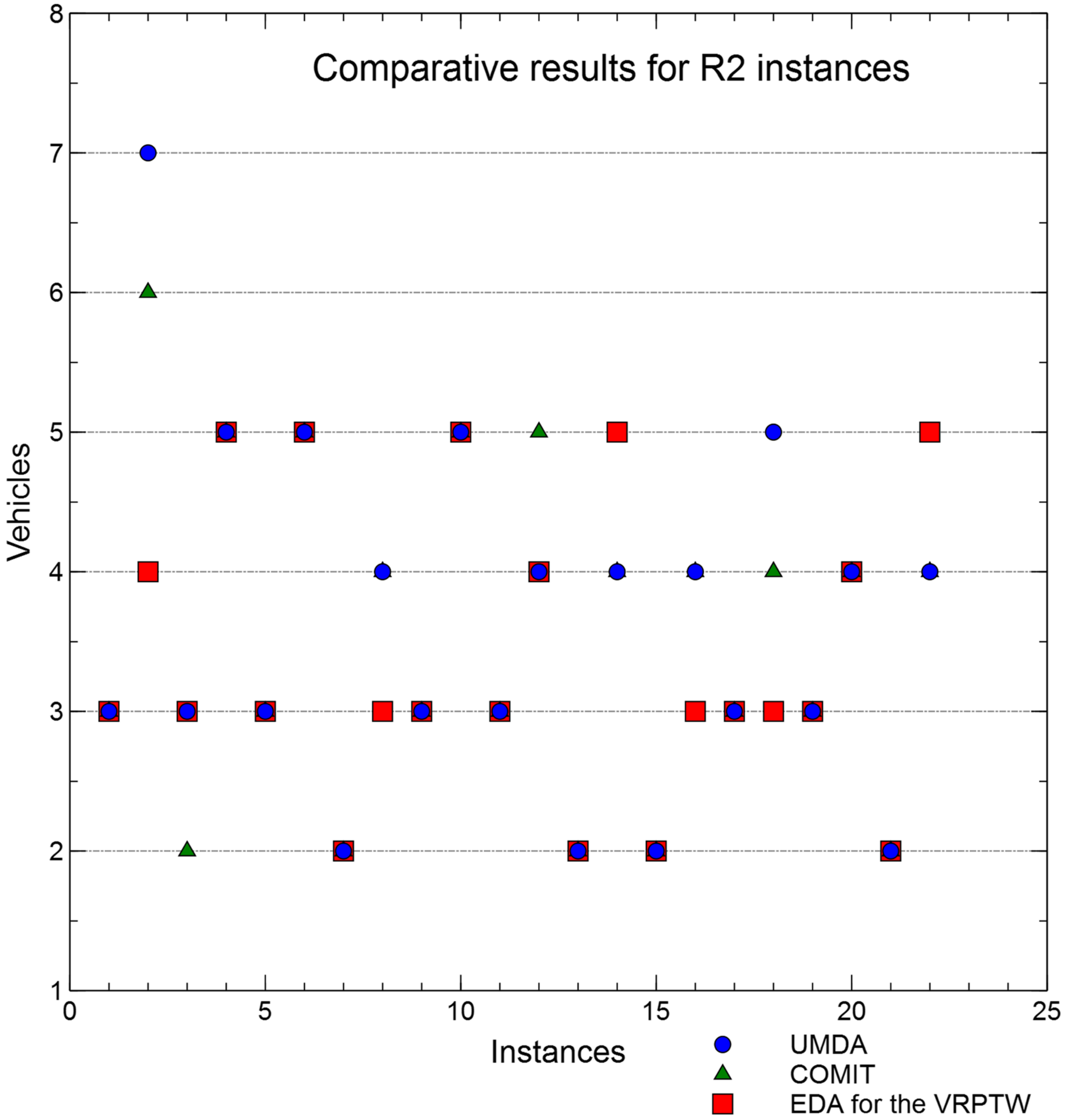

Fig. 4 Comparative results for R2 instances

customers per route. In contrast, the sets R2, C2, and RC2 have a long scheduling horizon permitting many customers to be serviced by the same vehicle. The customer coordinates are identical for all problems within one type (i.e., R, C, and RC). The problems differ with respect to the width of the time windows. Some have very tight time windows, while others have time windows which are hardly constraining. The travel times equal the corresponding distances. All these environments and conditions were executed in Delmia Quest® language.

The Univariate Marginal Distribution Algorithm (UMDA) and the Combining Optimizers with Mutual Information Trees (COMIT) are proposed as a benchmark for comparison with the EDA scheme.

From Tables 1, 2, 3, 4, 5, and 6, we can appreciate how the Bayesian network is able to detect competitive sequences for the VRPTW. In specific, the EDA was able to avoid delays for each vehicle in all kinds of the instances. Furthermore, for the instance R1, the EDA gets the same or less quantity of vehicles used in 18 of 24 cases; it means that the EDA is able to find better routes for the VRPTW in 75 % of the trails according to the R1 instance. For the routing time, the EDA was superior in only 11 of 24 cases; it is due to fewer vehicles used and these vehicles require make further paths. For the instance

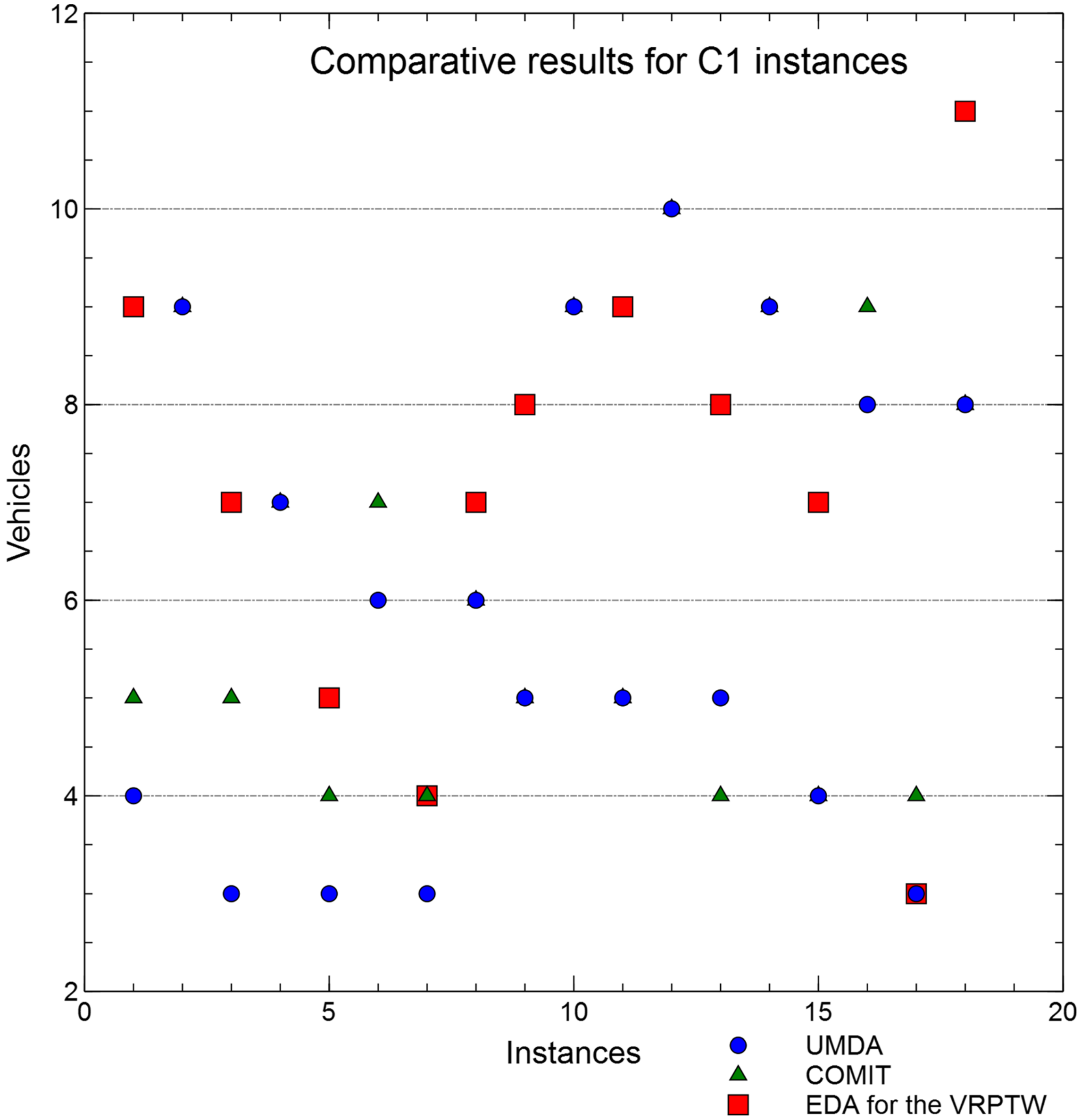

Fig. 5 Comparative results for C1 instances

R2, the EDA gets the same or less quantity of vehicles used in 20 of 22 cases, the best performance (90 %) in all the instances. It is due to the flexibility of the time windows in the R2 instances, i.e., a long scheduling horizon permits to assign more customers (vertices) in the same route or vehicle. For the routing time, the EDA was superior in 12 of 22 cases, the same situation than the R1 instance. For the instance C1, the EDA performance was not competitive and the same situation occurred in instance C2. For the instance RC1, the EDA gets the same or less quantity of vehicles used in 11 of 16 cases; it is an acceptable performance, i.e., 68 % of the trails. For the routing time, the EDA was superior in only 9 of 16 cases; it is a similar performance than the R1 instance because the time windows are very short and tight. For the instance RC2, the EDA gets the same or less quantity of vehicles used in 8 of 16 cases; it is a minimum performance, i.e., only 50 %. For the routing time, the EDA was superior in only 5 of 16 cases, a consistent result with respect other instances. It is due to that Bayesian network is shortsighted with respect the geographical data of the vertices. However, the EDA was able to avoid delays for each vehicle in all kinds of the instances.

In addition, from Figs. 3, 4, 5, 6, 7, and 8, we can find the improvement of the EDA with respect the other algorithms proposed as a benchmark for comparison with the EDA

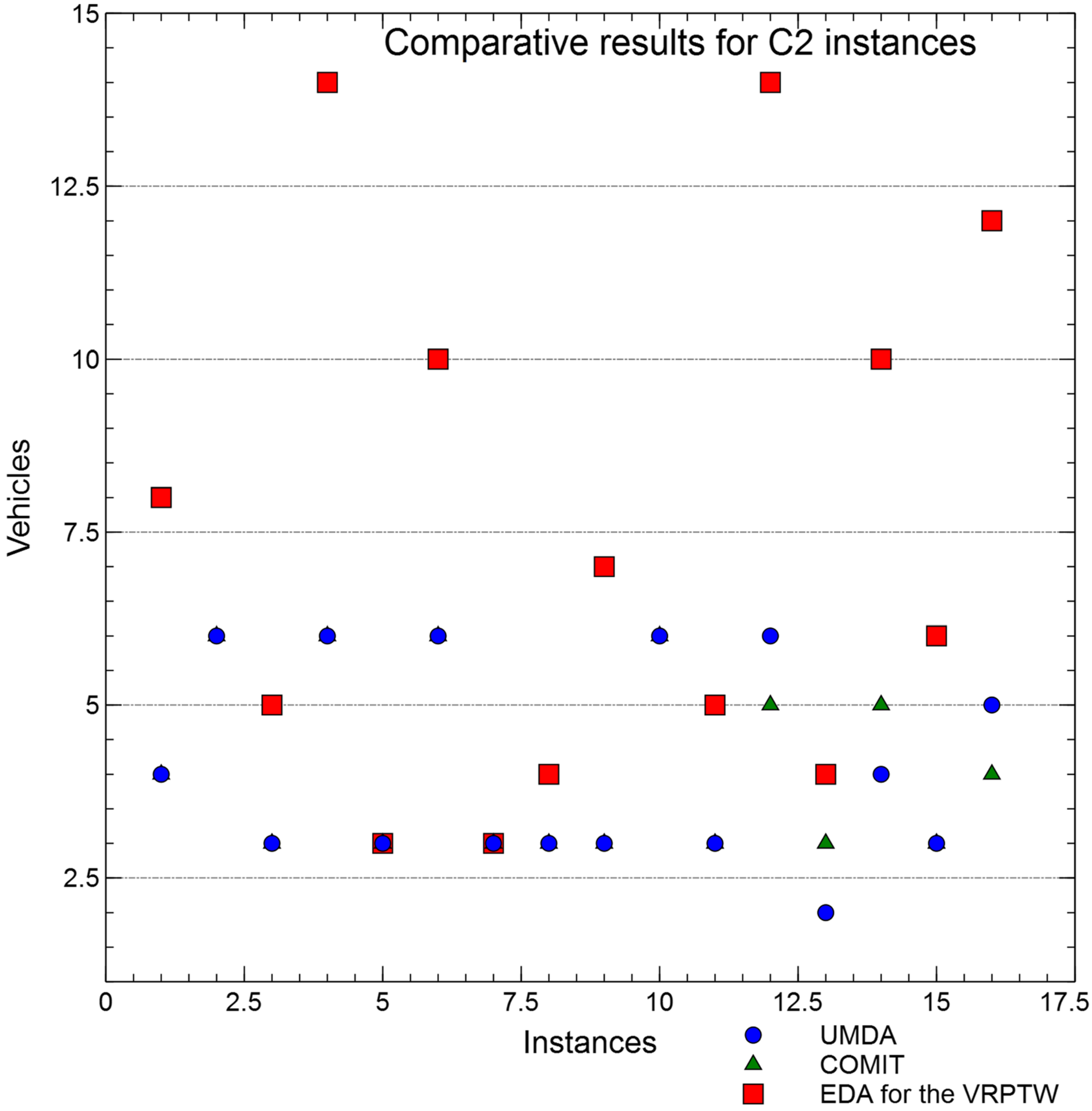

Fig. 6 Comparative results for C2 instances

scheme, where the EDA performance was better in 70 % of the cases, as average, for the number of vehicles used in all the trails.

The experimental results were analyzed comparing averages of delays between the algorithms. Table 7 details that there is a statistically significant difference between them. On the total routing time, Table 8 shows that there is a statistically significant difference between the algorithms.

6 Discussion and future research

When the amount of vertices is tractable, the EDA for VRPTW has an acceptable performance according to the results detailed in Tables 1, 2, 3, 4, 5, 6, 7, and 8. The results obtained by EDA demonstrate that the application of a probability model can be competitive for the VRPTW. Although the EDA utilized in this research is not able to handle invalid or unexpected inputs, it can be modified in order to get a useful module for specific users in industry. In addition, the EDA is able to detect relationships among vertices. We can consider that the EDA takes into account the relationship or interactions among variables of the problem as an advantage. For each generation, we know the probability that vertex *j* be selected for the *i* position. However, the probabilistic model used could be robust; it may be a disadvantage if we need to model higher interactions.

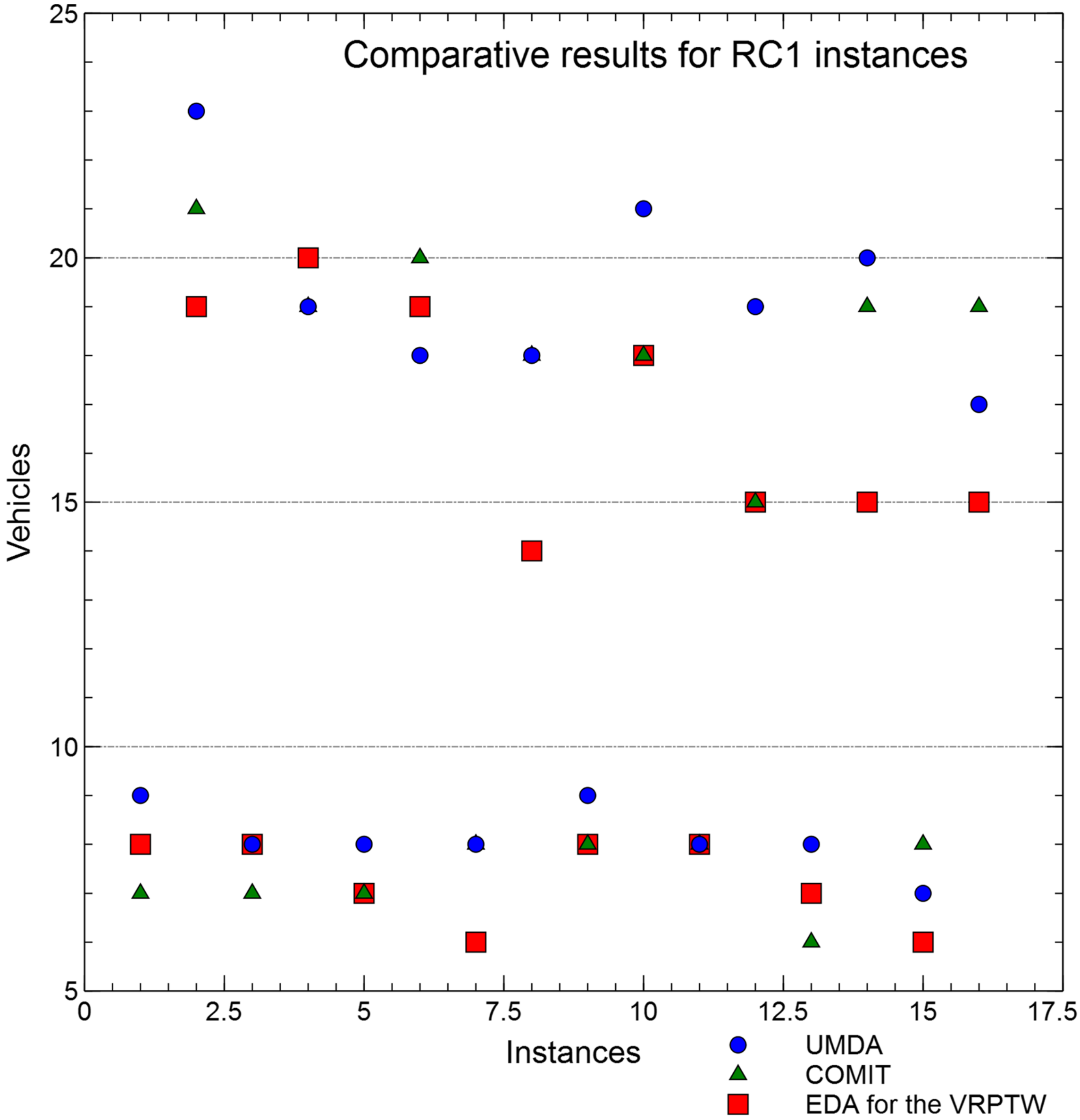

Fig. 7 Comparative results for RC1 instances

The EDA is able to produce feasible solutions according to different constraints detailed in the problem statement section of this paper. It was not necessary to repair the solutions as other algorithms used for permutation-based problems. The proposed method considers the previous results in order to avoid unfeasible solutions. On the other hand, the EDA is not flexible to handle new and unexpected requirements for the dynamic VRPTW. The proposed method is currently in the prototype phase for users. Therefore, it is expected that industrial practitioners will find the EDA beneficial to their business when it is ready for them.

According to the results above, the EDA keeps exploitative and exploratory capability in the evolutionary progress. The computational time and cost were not considered in this research because the algorithm proposed is currently in the prototype phase. Future research work would consider a module for users, and it should include computational time and cost aspects. This research is focused on only one type of vehicle, i.e., a homogeneous fleet of vehicles. Therefore, a heterogeneous fleet of vehicles appears to vary significantly in dynamic environments. Assigning the right vehicles to the right vertices may also affect travel time. This remains work for the future. Furthermore, our research should be expanded to

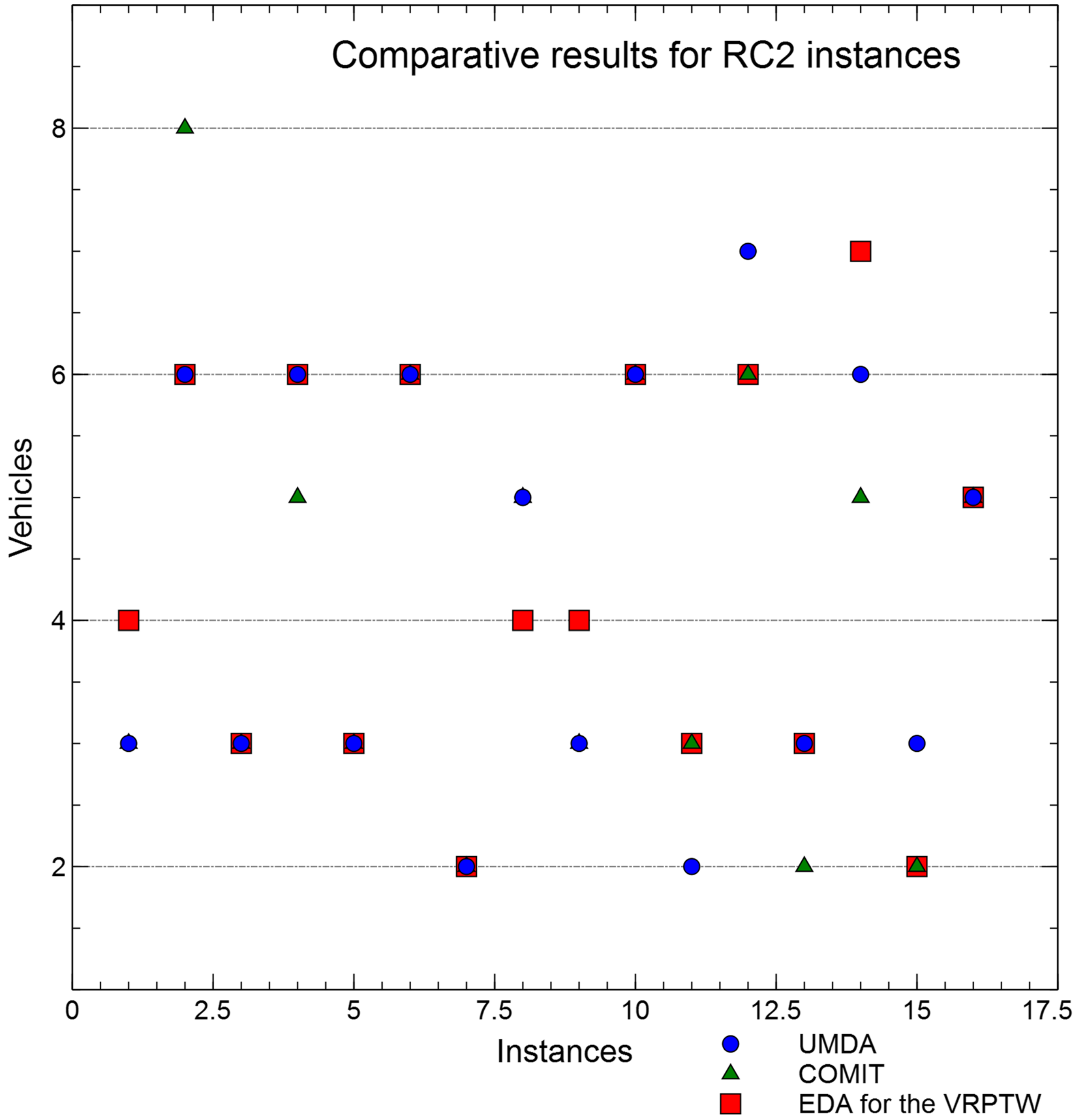

Fig. 8 Comparative results for RC2 instances

account for other delay factors, such as traffic, no-shows in drivers, and other changes that can be addresses by developing a dynamic version of the EDA.

7 Conclusions

This paper dealt with the VRPTW problem, where routes have to be constructed and released in order to minimize the total routing time for a given set of known and available vehicles. The problem is pivotal for the efficient management and control of routing systems. To solve this problem, we proposed the application of the EDA. By means of numerical experiments, it was shown that this approach generates competitive solutions. Implementing these solutions can improve customer service by delivering orders on time and by avoiding delays in other process stages. The EDA can detect relationships or interactions among vertices and positions by means of a probability model. Since the EDA presents stability, it appears very suitable for implementation in software systems for practical purposes. The potentials of the EDA approach to solving the VRPTW problem is detailed by the present results with several examples. The results encourage the development of an effective optimization method based on a probability model to resolve real-world VRPTW problems. We conclude that an effective estimation of interactions among

Table 7 Statistical analysis for delays

Instances	*t* test	*t* critic	μ UMDA	μ COMIT	μ EDA
R1	2.772*	0.0630	5.38	1.96	0
R2	1.464*	0.0630	4.95	2.77	0
C1	4.072*	0.0630	18.11	5.67	0
C2	1.446*	0.0630	9.38	5.94	0
RC1	2.848*	0.0630	4.44	1.19	0
RC2	1.412*	0.0630	5.75	3.19	0

UMDA Univariate Marginal Distribution Algorithmm, *COMIT* Combining Optimizers with Mutual Information Trees, *EDA* estimation of distribution algorithms

α=0.05 significance level

*Statistically significant difference

vertices and positions on the sequence can address the VRPTW successfully and suggest estimating relationships for other related dynamic issues (traffic, heterogeneous fleet) of the routing systems should be reconsidered in light of these results. Further research directions may deal with an extension of the EDA, in which higher probabilistic models can be used in order to model higher interactions or relationships among variables of the VRPTW performance. Effective modules for specific users in industry are required, and the study of probabilistic models would be useful to improve the VRPTW. The EDA could focus on evaluating its possible applicability on other types of routing systems, that is, pick and delivery problems.

We conclude that the interaction among vertices for VRPTW with other related dynamic issues by means of a probability model has not been considered sufficiently in the literature. Thus, it might be worthwhile providing a different approach, which integrates decisions on these dynamic issues in order to minimize the total travel time. Based on the experimental results shown, we confirmed that an appropriate modeling of the most important variables that affect the performance of the vehicle routing process should be considered in the proposed solution. We reach the conclusion that the vehicle routing performance can be improved if we take into account the relationship or interaction among variables of the problem. We deduce that the EDA for VRPTW can be an efficient algorithm to handle different vehicle routing conditions such as capacity and time windows. When the amount of vertices is tractable, the EDA for VRPTW has an acceptable performance according to the results detailed above. The probability model is useful in order to preserve sequences that appear frequently in all members of the population. The EDA is able to describe the characteristics exhibited by the parents in each generation. We conclude that the simulation optimization can be an efficient mechanism to handle different routing conditions where there are diverse variable interactions such as the VRPTW. The EDA proposed was able to find a less quantity of vehicles to use in the VRPTW with respect the other algorithms proposed as a benchmark for comparison. In addition, the EDA proposed was able to avoid delays for each vehicle used in the solutions. Moreover, this research contributes using an EDA as an optimization method for any simulation language. Finally, this research contributes by using an EDA as an optimization method for any VRP problem.

Table 8 Statistical analysis for the total routing time

Instances	*t* test	*t* critic	μ UMDA	μ COMIT	μ EDA
R1	7.664*	0.0630	1069.42	1067.13	1085.54
R2	6.837*	0.0630	1313.73	1225.27	1312.59
C1	3.468*	0.0630	881.83	626.11	745.33
C2	3.870*	0.0630	1035.19	856.13	890.13
RC1	3.456*	0.0630	1339.75	1145.38	1084.94
RC2	3.756*	0.0630	1759.81	1377.13	1695.56

α=0.05 significance level

UMDA Univariate Marginal Distribution Algorithmm, *COMIT* Combining Optimizers with Mutual Information Trees, *EDA* estimation of distribution algorithms

*Statistically significant difference

Acknowledgments We would like to express our gratitude to Elizabeth O'Shaughnessy for reviewing the manuscript.

References

1. Afifi S, Dang D-C, Moukrim A (2015) Heuristic solutions for the vehicle routing problem with time windows and synchronized visits. Optim Lett. doi:10.1007/s11590-015-0878-3
2. Barbucha D (2014). Team of A-Teams Approach for Vehicle Routing Problem with Time Windows. In: Terrazas G, Otero F, Masegosa A (eds) Nature Inspired Cooperative Strategies for Optimization (NICSO 2013), vol 512. Springer International Publishing, pp 273–286
3. Berghida M, Boukra A (2015) EBBO: an enhanced biogeography-based optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. Int J Adv Manuf Technol 77(9–12):1711–1725
4. Berger J, Barkaoui M (2004) A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Comput Oper Res 31:2037–2053
5. Bräysy O, Gendreau M (2005) Vehicle routing problem with time windows, Part II: Metaheuristics. Transp Sci 39:119–139
6. Bräysy O, Dullaert W, Gendreau M (2004) Evolutionary algorithms for the vehicle routing problem with time windows. J Heuristics 10: 587–611
7. Cardoso P, Schütz G, Mazayev A, Ey E, Correa T (2015) A solution for a real-time stochastic capacitated vehicle routing problem with time windows. Procedia Comput Sci 51:2227–2236
8. Cordeau JF, Laporte G, Mercier A (2001) A unified tabu search heuristic for vehicle routing problems with time windows. J Oper Res Soc 52:928–936

9. Cordeau JF, Laporte G, Mercier A (2004) Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. J Oper Res Soc 55:542–546
10. Cruz-Ramírez, N. and Martínez-Morales M (1997) Un algoritmo para generar redes Bayesianas a partir de datos estadísticos. Primer Encuentro Nacional de Computación, ENC 97, Querétaro México
11. de Armas J, Melián-Batista B (2015) Constrained dynamic vehicle routing problems with time windows. Soft Comput. doi:10.1007/s00500-014-1574-4
12. Desrochers M, Lenstra J, Savelsbergh M, Soumis F (1988) Vehicle routing with time windows: optimization and approximation. In: Golden B, Assad A (eds) Vehicle Routing: Methods and Studies. Elsevier Science Publishers, Amsterdam
13. Gan X, Kuang J, Niu B (2014) Multi-type Vehicle Routing Problem with Time Windows. In Huang D-S, Jo K-H, Wang L (eds) Intelligent Computing Methodologies, vol 8589. Springer International Publishing, pp 808–815
14. Golden B, Assad A (1986) Perspectives on vehicle routing: exciting new developments. Oper Res 34:803–809
15. Homberger J, Gehring H (1999) Two evolutionary meta-heuristics for the vehicle routing problem with time windows. INFOR 37: 297–318
16. Kabcome P, Mouktonglang T (2015) Vehicle routing problem for multiple product types, compartments, and trips with soft time windows. Int J Math Math Sci. doi:10.1155/2015/126754
17. Kaiwartya O, Kumar S, Lobiyal D, Tiwari P, Abdullah A, Hassan A (2015) Multiobjective dynamic vehicle routing problem and time seed based solution using particle swarm optimization. J Sens. doi: 10.1155/2015/189832
18. Li P, He J, Zheng D, Huang Y, Fan C (2015) Vehicle routing problem with soft time windows based on improved genetic algorithm for fruits and vegetables distribution. Discret Dyn Nat Soc. doi:10.1155/2015/483830
19. Li J, Li Y, Pardalos P (2014) Multi-depot vehicle routing problem with time windows under shared depot resources. J Comb Optim. doi:10.1007/s10878-014-9767-4
20. Mester D, Bräysy O, Dullaert W (2007) A multi-parametric evolution strategies algorithm for vehicle routing problems. Exp Syst Appl 32:508–517
21. Mühlenbein H, Paaß G (1996) From recombination of genes to the estimation of distributions: I. binary parameters. In: Voigt H, Ebeling W, Rechenberg I, Schwefel H (eds) Parallel problem solving from nature PPSN IV. Springer, Berlin, pp 178–187
22. Nalepa J, Blocho M (2015) Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows. Soft Comput. doi:10.1007/s00500-015-1642-4
23. Potvin JY, Bengio S (1996) The vehicle routing problem with time windows - Part II: Genetic search. INFORMS J Comput 8:165–172
24. Potvin JY, Rousseau JM (1993) A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. Eur J Oper Res 66:331–340
25. Prins C (2004) A simple and effective evolutionary algorithm for the vehicle routing problem. Comput Oper Res 31–12:1985–2002
26. Rochat Y, Taillard É (1995) Probabilistic diversification and intensification in local search for vehicle routing. J Heuristics 1:147–167
27. Schwarze S, Voß S (2015) A Bicriteria Skill Vehicle Routing Problem with Time Windows and an Application to Pushback Operations at Airports. In: Dethloff J, Haasis H-D, Kopfer H, Kotzab H, Schönberger J (eds) Logistics Management (Products, Actors, Technology - Proceedings of the German Academic Association for Business Research, Bremen, 2013). Springer International Publishing, pp 289–300
28. Solomon M (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper Res 35: 254–265
29. Solomon M, Desrosiers J (1988) Time window constrained routing and scheduling problems. Transp Sci 22:1–13
30. Soonpracha K, Mungwattana A, Manisri T (2015) A Reconstructed Meta-Heuristic Algorithm for Robust Fleet Size and Mix Vehicle Routing Problem with Time Windows under Uncertain Demands. In: Handa H, Ishibuchi H, Ong Y-S, Tan K-C (eds) Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, vol. 2. Springer International Publishing, pp 347–361
31. Sun L, Wang B (2015) Robust optimisation approach for vehicle routing problems with uncertainty. Math Probl Eng. doi:10.1155/2015/901583
32. Tan KC, Lee LH, Zhu QL, Ou K (2001) Heuristic methods for vehicle routing problem with time windows. Artif Intell Eng 15: 281–295
33. Thangiah SR, Osman IH, Vinayagamoorthy R, Sun T (1995) Algorithms for the vehicle routing problems with time deadlines. Am J Math Manag Sci 13:323–355
34. Toth P, Vigo D (2001) The vehicle routing problem. Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia
35. Wang L, Wang S, Xu Y, Zhou G, Liu M (2012) A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. Comput Ind Eng 62:917–926
36. Wang Y, Ma X, Xu M, Wang Y, Liu Y (2015) Vehicle routing problem based on a fuzzy customer clustering approach for logistics network optimization. J Intell Fuzzy Syst. doi:10.3233/IFS-151578
37. Yang C, Guo Z-X, Liu L-Y (2015) Comparison Study on Algorithms for Vehicle Routing Problem with Time Windows. In: Qi E, Shen J, Dou R (eds) Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management 2014. Atlantis Press, pp 257–260