Hybrid estimation of distribution algorithm for a multiple trips fixed fleet vehicle routing problems with time windows

Article *in* International Journal of Operational Research · October 2014

DOI: 10.1504/IJOR.2014.065611

1 author:

Jalel Euchi
Qassim University

Hybrid estimation of distribution algorithm for a multiple trips fixed fleet vehicle routing problems with time windows

Jalel Euchi

Higher Institute of Business Administration,
Gafsa University, Cité universitaire,
Rue Houcine Ben GADDOUR,
Zarroug, 2112, Gafsa, Tunisia
E-mail: euchi.jalel@yahoo.fr

Abstract: In this paper, we consider a variant of vehicle routing problem with multiple trip and time windows. In this alternative problem, we think about the scheduling of trucks to a number of customers in the presence of fixed fleet and time windows constraints. This type of problem can be described as determining a number of vehicle trips minimising total travelled distance complying with the time windows and the multiple use of vehicles. We call this problem a multiple trips fixed fleet vehicle routing problems with time windows (MTFFVRPTW). To solve this variant, a new hybrid evolutionary algorithm for transport optimisation problems, called estimation of distribution algorithm with local search is used. The proposed algorithm integrates a double structure of distribution technique, which is used to introduce the variables dependency. To evaluate our approach we provide a number of experimentation to a modified Solomon's instances for the case of multiple trips. The results show that the hybrid proposed algorithm offer a good quality solution in the term of objective functions, also in the running time.

Keywords: vehicle routing; multiple trips; time windows; estimation of distribution algorithm.

Reference to this paper should be made as follows: Euchi, J. (2014) 'Hybrid estimation of distribution algorithm for a multiple trips fixed fleet vehicle routing problems with time windows', *Int. J. Operational Research*, Vol. 21, No. 4, pp.433–450.

Biographical notes: Jalel Euchi completed his PhD in 2011 at Sfax University, Tunisia and prepared his thesis in the form of co-supervised at the University of LeHavre in France. He received his Master degree in Operational Research and Production Management from the Sfax University. Currently, he is an Assistant Professor at the Higher Institute of Business Administration of Gafsa-TUNISIA. His primary research interests are in complex routing problems, meta-heuristics algorithms to solve the NP-hard problems, computational operations research, logistics and supply chains. He has participated in different international conferences and was the corresponding author of several published papers.

1 Introduction

Until that time, logistics and transport research has habitually assumed a situation perspective as regards the relation between transport cost and strategy of optimisation. Due to the increases of the fuels prices in the world, and the increases of transportation cost, the companies are obliged to redefine and to accurate their procedures concerning the management of their vehicles fleet (e.g., Euchi et al., 2011).

Due to the globalisation, the relocation of production facilities and expansion of markets, the transportation is critical in ensuring the connections between the production facilities. Logistics related to it raises many issues, often difficult to solve optimally. The vehicle routing problem (VRP) is one of those problems that is determining the order in which to visit a spatially set of customers with a fixed fleet of vehicles, the goal is basically to visit a set of places as customers (which can be regarded as a one-time site specific, or as a lanky street) at minimum cost. It represents one of the most useful and successful fields of operations research.

Given their economic significance, there is continuing research interest in provided good solutions to real-world scheduling problems. According the importance of transport cost, it is necessarily that the fleet must be managed effectively, it is realised by the use of good software to schedule the vehicle routing.

The VRP is one of the most challenging combinatorial optimisation tasks most studied. It consists in designing the optimal set of routes for fixed fleet of vehicles in order to serve a given set of customers. As described in literature the interest in VRP is motivated by its practical relevance as well as by its considerable difficulty.

The VRP is a generic name given to a whole class of problems in which a set of routes for a fixed fleet of vehicles based at one or several depots must be determined for a number of geographically dispersed cities or customers. The objective of the VRP is to deliver a set of customers with known demands on minimum-cost vehicle routes originating and terminating at a depot (to minimise the total distance travelled).

In related works devoted to the VRP, we regards that are studied in several ways. Many variants were considered the vehicle routing problem with fixed fleet (FFVRP), vehicle routing problem with time windows (VRPTW), dynamic vehicle routing problem (DVRP), and vehicle routing problem with private fleet and common carrier (VRPPC), etc.

The VRPTW is the same problem that VRP with the additional restriction that in VRPTW a time window is associated with each customer, defining an interval wherein the customer has to be supplied (e.g., Caramia and Onori, 2008). In the case or in the presence of multiple trips, the vehicles are allowed to perform multiple routes. This problem holds the attention of numerous researchers, in the world, since many years.

A number of authors have made a literature review that deal with VRP. The related works devoted to the VRP was introduced by Dantzig and Ramser (1959) and since then it has been widely studied include those of Clarke and Wright (1964) which propose a saving heuristic to solve this variant of problem.

A state of the art is proposed by Bodin et al. (1983) with principle idea to give a good review for the routing and scheduling of vehicles and crews. In the paper of Laporte (1992), we give an overview of exact and approximate algorithms to solve the VRP.

As mentioned in the previous paragraph the VRPs are separated into various areas. Problems related to on condition that services through fixed fleets are problematical and complicated in comparison with unlimited fleet VRPs (e.g., Euchi and Chabchoub, 2011).

Several authors studied the FFVRP as in Choi and Tcha (2007) they propose a column generation approach to solve this variant. Also, an algorithm based on the record-to-record travel approach is developed by Li et al. (2007) which built an integer programming model and solved the linear relaxation by column generation.

Recently, Euchi and Chabchoub (2010) describe a tabu search algorithm embedded in the adaptive memory (TSAM) procedure to solve the HFFVRP. In this paper the authors demonstrated the role of the adaptive memory to allow a comparatively large pool of good and diversified solutions.

In the case of VRPTW, conversely, a number of successful metaheuristic solution approaches have been anticipated, which are capable to produce high-quality solutions for reasonably-sized instances in limited time. With reference to the research related to solve the VRPTW, we can cite a route-neighbourhood-based metaheuristic of Liu and Shen (1999). A literature review of VRPTW is reported in the paper of Bräysy and Gendreau (2005).

Lau et al. (2003) introduce a variant of VRPTW where a limited number of vehicles is given (m-VRPTW). The authors proposed a tabu search algorithm characterised by a holding list and a mechanism to force dense packing within a route also in this approach they introduces the notion of penalty for lateness to allow time windows to be relaxed. In the other way Potvin and Rousseau (1993) operate regret heuristic embedded in a parallel route building algorithm for the vehicle routing and scheduling problem with time windows.

In the last decades, we have observed a superficial change in search and optimisation technologies. In this problem domain when information on time windows and fixed fleet vehicle with the presence of multiple trips exist we are in the context of the most practical variant. In this work, we concentrate to the multiple trips fixed fleet vehicle routing problems with time windows (MTFFVRPTW).

A few works are available for the MTFFVRPTW in the literature. This probably depends on the high complexity of the problem itself. Originally, this variant was introduced for the first time by Fleischmann (1990). In his paper, Fleischmann propose a one phase algorithm to generate solution for the MTFFVRPTW, it incorporate a greedy-type heuristic with the need to assign route to vehicles. The routes constructed are then combined to produce workdays for the vehicle by solving a bin packing problem.

To solve the MTFFVRPTW, Taillard et al. (1996) proposed a three phase approach. This algorithm generates a large set of routes basis on tabu search metaheuristic. It starts by generating a large set of routes satisfying the VRP capacity constraints in the first phase, we combines the routes constructed into complete VRP solutions in the second phase and finishes by assembling selected routes with a bin packing heuristic into feasible working days in the third phase.

Brandao and Mercer (1998) adapted their algorithm to solve a complex real life problem of Burton's Biscuits. The authors designed a tabu search metaheuristic which starts from a constructive solution generated with nearest neighbour insertion and swap moves. Moves are defined by swapping two customers and by removing one customer from its route and inserting it into another one. Insertions are made using the GENI algorithm.

A multi phase heuristic is proposed in Petch and Salhi (2003). Feasible initial solutions are constructed with repeated executions of Yellow's savings algorithm and using a route population approach (e.g., Yellow, 1970). The algorithm consists of allocation of routes to vehicles which constructed through a route-first-cluster-second method and then again packed into working shifts using improvement methods.

Salhi and Petch (2007) designed a genetic algorithm to solve the MTFFVRPTW problem. The authors propose a new non-binary chromosome representation and an appropriate genetic operators and a scheme for chromosome evaluation adapted and applied to solve the multi trip vehicle routing solutions.

An exact algorithm to solve the multi trip VRP is proposed by Azi et al. (2007). In this work, the algorithm is divided into two phases: all non-dominated feasible routes are generated in the first phase, and then a number of routes are selected and sequenced to form the vehicle workday are described in the second phase.

Recently, in Azi et al. (2010) an exact solution based on branch-and-price approach to solve the MTFFVRPTW is developed. Lower bounds are computed by solving the linear programming relaxation of a set packing formulation, using column generation. The authors define that the pricing sub problems are elementary shortest path problems with resource constraints.

Some works have been presented for problems for which MTFFVRPTW represents a generalisation. Olivera and Viera (2007) describe an adaptive memory programming for the VRP with multiple trips. The authors introduce a tabu search algorithm embedded in adaptive memory procedure to solve the multi trip VRPs. At each iteration a subset of routes constructed in candidate list are chosen and improved through the tabu search algorithm. Next, the resulting routes are returned into the adaptive memory.

Alonso et al. (2008) developed a tabu search algorithm basis on periodic VRP (PVRP), it incorporates the fact that all vehicles can visit all customers, which leads to the site dependent VRP (SDVRP). They combine the ideas of SDVRP with multi trip SDVRP (SDVRPM).

Battarra et al. (2009) suggest an adaptive guidance approach for the heuristic solution of a minimum multiple trip VRP. They decomposed the master problem into several problems, each one is solved with a specific heuristic adapted to solve the multi trip VRPs. An adaptive guidance mechanism is worn to direct the heuristics to possibly get better the current solution.

Generally, the multi trip VRP is NP-hard. It is very difficult to find a good and effective solution for the MTFFVRPTW due to the diversity of the problem and variance of constraints devoted to this variant. Few works tackled this variant, several authors proposed various approaches, e.g., tabu search algorithm, genetic algorithm, meta-heuristic approaches, for MTFFVRPTW as metaheuristics developed in Rajmohan and Shahabudeen (2009).

We have proved in literature that the meta-heuristics maintain to demonstrate their effectiveness for solving difficult combinatorial optimisation problems appearing in a wide variety of industrial, economic, and scientific domains.

Meta-heuristics embraces the attention of numerous researchers, and everywhere in the world. Evolutionary algorithms (EAs) represent a subset of meta-heuristics. They received a growing interest in solving the multiobjective optimisation problems especially the VRPs.

The estimation of distribution algorithms is relatively recent type of optimisation and learning techniques based on the concept of using a population of tentative solutions to iteratively approach the problem region where the optimum is located (e.g., Euchi et al., 2011; Larrañaga, 2002; Bosman and Thierens, 2002). In the present paper, we examine an estimation of distribution algorithms with hybrid multi start constructive method to improve the solution quality of MTFFVRPTW.

The estimation of distribution algorithms (EDA) has been used in numerous ways to optimise numerous problems to reach the highest possible solutions. In VRP, few papers dealing with EDA are used to solve it. Recently, Euchi et al. (2011) propose an iterated density estimation EA with two-opt local search to solve the vehicle routing problem with private fleet and common carrier (VRPPC). The main idea of our paper is to solve the MTFFVRPTW in the first time with EDA, so to apply our hybrid methodology to hard benchmarks presented in literature.

The contribution of this paper is threefold: first of all is the hybridisation of the estimation of distribution algorithms (EDA) and a multi start constructive initial solution method that permits to obtain near optimal solution. Secondly, the effect of using a dual-estimation of distribution structure in the EDA algorithm in the final solution. Thirdly, a new implementation of data instances in literature with a number of customers equal to 1,000 with the good and important EDA design to solve a MTFFVRPTW that illustrate the superiority of our approach expressed by the produced solutions comparing to the results realised in literature.

The rest of the paper is organised as follows: in Section 2, we briefly give the main description and mathematical formulation of the MTFFVRPTW. Section 3 describes the solution methodology. In Section 4, we present the proposed hybrid EDA algorithm. Experimental settings and numerical results are given in Section 5. The paper concludes in Section 6 with various remarks and indications for future work.

2 Problem definition

In this paper, we address one of the most general variant of VRP is the MTFFVRPTW. The main reason behind this great attention to the problem is the role and support tools to make a decision and to specify the trip assigned of the vehicles for the delivery (e.g., Euchi and Chabchoub, 2010).

This problem is defined over a graph $G = (V, A)$ where $V = \{0, 1,\dots,n\}$ is the vertex set of customers and $A = \{(i, j)\colon i, j \in V, i \neq j\}$ is the arc set. Vertex 0 is a depot. Let c_{ij} and t_{ij} be the distance and a travel time associated with each arc $(i, j) \in A$ (distances and travel times are the same). With each customer $i \in V - \{0\}$ is associated an advantage av_i. Each customer i has a non-negative demand q_i, a service time st_i and a time windows $[a_i, b_i]$, where a_i and b_i are the earliest and latest time, respectively (we begin the service with $a_0 = 0$ and $b_0 = \infty$). Each vehicle waits if it arrives at customer i before a_i. Let $k = 1, 2,\dots,m$ be the set of identical vehicles available to serve the n customers starting from the single depot. Let Q be the capacity of each vehicle, and $L_{\max}$ be the time limit for a single trip of a vehicle. A trip is a sequence of customers starting from and finishing at the depot. A setup time σ_r necessary to load the vehicle is associated with each route $r \in R$. These routes are denoted by set R, where $|R|$ is large enough to accommodate the maximum number of routes that the fleet of vehicles can possibly perform in a day. We assume that the routes served by a same vehicle are numbered in increasing order, that is, a vehicle serves route s after route r if and only if $r < s$. The objective of the optimisation is to minimise the total travel time required by the available fleet to serve all the customers.

3 Solution methodology

The major difficulty for solving NP-hard problems as the MTFFVRPTW in the presence of large instances and with increasing number of customers we obliged to choose a meta-heuristic procedure to solve the MTFFVRPTW.

In the last few years, various meta-heuristic approaches and their hybrids have been tested to solve many problems. Taking into consideration a confirmed record of successful applications of these meta-heuristics especially the EA to solving complex MTFFVRPTW problems, we prefer to solve the MTFFVRPTW with a new class of EA known EDA.

The algorithms with estimation of distribution were originally designed as an alternative for the EAs as described in Mühlenbein and Mahnig (1999). EDA algorithm is evolutionary technique that uses probabilistic models to represent pertinent information about the search space. New solutions are sampled from the model thus built in probabilistic model.

Recently, Hauschild and Pelikan (2011) illustrate a survey of EDAs algorithms. EDA work with a population of candidate solutions (generate M points) to the problem according to the initial solution created with constructive method, then evaluate the points using the fitness function. Select a set of M' points generated in population according to a selection method. After that we calculate a probabilistic model from the selected set. Generate M'' new points sampling from the distribution represented in the model. This procedure continues until a termination criterion is met. The idea is to confine, in the form of probabilistic dependencies between the variables, information about promising areas of the search space that can be used to improve the search for better solutions. The basic EDA procedure is summarised in Algorithm 1.

Algorithm 1 Pseudo code of EDA algorithm

Set $t \leftarrow 0$
Generate M points randomly
do
    Evaluate the points using the fitness function
    Select a set of M' points according to a selection method
    Calculate a probabilistic model
    Generate M'' new points sampling from the distribution represented in the model
    $t \leftarrow t + 1$
until Termination criteria are met.

4 Proposed hybrid meta-heuristic algorithm

In this section, we propose to develop an optimisation algorithm based on EDA procedure and a multi start constructive solution for the minimisation of MTFFVRPTW.

The EDA is notated as follows: Let $X = (X_1,\ldots,X_n)$ be a vector of random variables. A value of X is denoted x.

We will use $x = (x_1,\ldots,x_n)$ to denote an assignment to the variables. S will denote a set of indices in $N = \{1,\ldots,n\}$, and XS (respectively, xS) a subset of the variables of X

(respectively, a subset of values of x) determined by the indices in S. We will work with discrete variables.

The joint probability mass function of x is represented as $p(X = x)$ or $p(x)$. $p(xS)$ will denote the marginal probability distribution for XS. We use $p(X_i = x_i \mid X_j = x_j)$ or, in a simplified form, $p(x_i \mid x_j)$, to denote the conditional probability distribution of X_i given $X_j = x_j$.

Therefore the EDA algorithm possessed the following format: With each generation t, EDA maintains a population of solutions, which $Pop(t) = \{\pi^1, \pi^2, \ldots, \pi^N\}$ and the probability matrix is

$$p(t) = \begin{pmatrix} p_{11}(t) & \ldots & p_{1n}(t) \\ \vdots & & \vdots \\ p_{n1}(t) & \ldots & p_{nn}(t) \end{pmatrix};$$

where $p(t)$ models the distribution of promising solutions in the search space. More precisely, $p_{kj}(t)$ is the probability that vehicle k is assigned to customer j in the assignment (e.g., Euchi et al., 2011). In the sequel the implementation of each element of the EDA algorithm to solve the MTFFVRPTW is described.

The hybrid EDA proposed in this paper can be roughly illustrated into seven steps: vehicle routing representation, constructive initial solution, construction of population, selection operators, probabilistic model, replacement, stopping criterion.

4.1 Vehicle routing representation

This section is devoted to the representation form of solution of MTFFVRPTW generated with EDA. Our hybrid EDA works here with the integer representation. We refer the reader to the paper of Euchi et al. (2011) which we have used the modified vehicle routing representation presented.

In practice, a representation must be chosen for the individuals of a population, an individual could be a list of integers represents the customers that should be visited in the same order as they appear consisting of multi routes. Every customer has to be a member of exactly one route.

The solution is presented by one vector of dimension $n + k$ where n is the number of customers and k is the number of trip ($n \neq k$). It is assumed that every solution starts from the depot. Each vector has a combined value of 1 to n and $k - 1$ of value 0 (the depot) to indicate the different trips.

Each value $\in [1, n]$ indicates the customer and each value 0 indicates the return to the depot (depot for the new trip) in the $k - 1$ value 0 in the sequence. For the last value 0 indicates the end of the cycle allocated to vehicle.

To exemplify the encoding solution, we suggest an example treating seven customers and one vehicle. As a result the solution is represented as follows:

Figure 1 Vehicle routing representation

3	6	0	1	4	7	0	2	5

Figure 1 present the solution with one vector. Each value between 1 and n represent the index of the customers when each customer is listed in the order in which they are visited. In our example, we have seven customers and one vehicle. The solution to the

MTFFVRPTW is defined by a set of trip realised by one vehicle. Each route has a sequence to serve customers.

The solution is represented as follows:

- trip 1: customers 3 and 6
- trip 2: customers 1, 4 and 7
- trip 3: customers 2 and 5.

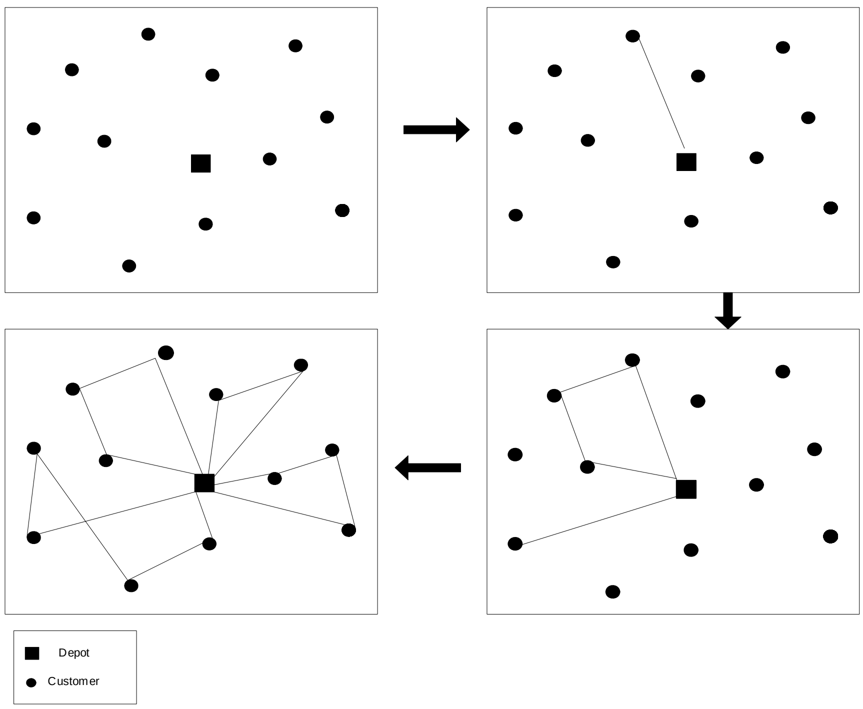
4.2 Initial constructive solution

This section is devoted to carrying out a presentation of the constructive initial solution generated. One of the most characteristics of this algorithm is the hybridisation of two methods, the path scanning algorithm and nearest neighbour procedure.

Firstly, the algorithm start by choosing the most distant customer from the depot (in terms of travel times), then we apply the path scanning algorithm of Golden et al. (1983) but in this method we apply the nearest neighbour procedure for scanning customers to construct routes. Some preliminary experiments suggested that this strategy was the most promising among those tested. The design of the initial algorithm is presented in following paragraph.

Based in the information about the travel time matrix we choose a customer which corresponds to the farthest par rapport through the depot.

Figure 2 Structure of path scanning procedure



As mentioned previously, path scanning built routes in adding customer's one by one successively. We start with an initial solution with 0 trips, it creates a new trip with the first customers, and then we apply a nearest neighbour algorithm to add the new customers. When you cannot possible to add new customers, this trip is closed to create a new one. This is done until the insertion of all customers (see Figure 2).

The nearest neighbour procedure works in the following way: In this method, the customers with the nearest neighbour through the farthest customers chosen as the first one are appended to a route. When the next to-be inserted customer's time limit exceeds the length of trip on the current route, a new trip is initiated. In this constructive heuristic, the vehicle starts at some customers and then visits the customer nearest to the starting one. From there, it visits the nearest unvisited customer, until all customers are visited, and then returns to the starting depot. A pseudo-code of initial constructive algorithm is given below:

Algorithm 2 Pseudo code of initial constructive algorithm

Set *trip* $\leftarrow$ 0.
Start with the farthest customer through the depot.
Find a node, not already on the path, which is closest to the last added node.
Repeat Evaluate the length of trip ($L_{\max}$).
until all customers belong to the path and $l < L_{\max}$.
Then, join the first and the last customers of the path.
trip $\leftarrow$ *trip* + 1
until Termination criteria are met

4.3 Construction of population

In this section, insertion heuristic is developed and used for obtaining an initial population. EDA in the initialisation start with the best solution found in the constructive solution phase obtained with the path scanning method.

In order to obtain a number of solutions to form the population we execute the initial constructive solution with different parameters for the length of trip. We adopt three values of $L_{\max}$. The N resultant solutions $\{\pi^1, \pi^2, \ldots, \pi^N\}$ constitute the initial population $Pop(0)$. The initial probability matrix $p(0)$ is set as $p_{ij} = \frac{1}{N}$. Then the probability matrix $p(t)$ can be updated as follows:

$$p_{ij}(t) = (1-\beta)\frac{1}{N}\sum_{k=1}^{N} I_{ij}\left(\pi^k\right) + \beta p_{ij}(t-1), \quad (1 \le i, j \le n)$$

where

$$I_{ij}(\pi) = \begin{cases} 1 & \text{if } \pi(i) = j \\ 0 & \text{otherwise} \end{cases}$$

$0 \le \beta \le 1$ is a learning rate. The bigger β is, the greater the contribution of the solutions in $Pop(t)$ is to the probability matrix $p(t)$.

4.4 Selection operators

The selection operators step can be considered a very important phase because meant to select the better solutions of the population. In our hybrid EDA algorithm and in each generation, the method used is the proportional selection.

This type of selection was originally proposed by Holland for the genetic algorithms. As described in Holland (1962), the expected selection λ_i of an individual i is proportional to its fitness f_i. Let μ be the population size and let λ be the total number individuals generated by the selection operator, λ_i can be expressed as:

$$\lambda_i = \frac{\lambda}{\sum_{j=1}^{\mu} f_j} f_i.$$

4.5 Probabilistic model

The learning and sampling algorithms used by our hybrid EDA depend on the class of probabilistic model used. In this section we propose a Bayesian probabilistic model which learned in each generation. The Bayesian probabilistic model is able to encode any dependencies of variables that can obtain one out of a finite set of values. The idea is to convert the minimisation problem into a search over probability distributions. It is an explicit model of currently promising regions of the search space.

For our case we have implemented a variant of probabilistic model based on the Bayesian optimisation algorithm (e.g., Pearl, 1988). Bayesian networks are probabilistic graphical models based on directed acyclic graphs. In a Bayesian network, whose directed acyclic graph is represented by *S*, where a probabilistic graphical model for $X = \{x_1,\ldots,x_n\}$ encodes a graphical factorisation of a joint probability distribution $p(x)$. It has two components:

- A structure *S* (e.g., directed acyclic graph for Bayesian networks).
- A set of local marginal probability values. *S* represents a set of conditional independence assertions between the variables.

After implementation of the first probabilistic model, the algorithm runs another time the second version of the probabilistic models. Perfectly, the repeated improvement of the probabilistic model based on representative samples of good's solutions would keep increasing the probability of generating the optimum after a practical number of iterations.

4.6 Replacement

To maintain the population size constant, it is recommended to make a replacement in the previous population. Some solutions from the population have to be substituted by some of individuals created during the probabilistic model. This can be done using the elitist replacement. An elitist strategy consists in preserving in the population, from one generation to the next one, at least the individual having the best fitness. A subset of χ individuals is selected (best fitness), and the worst one is selected for replacement (for $\chi > 1$).

4.7 Stopping criterion

We consider that the stopping criterion is natural to the complexity of used probabilistic model. As defined in literature the stopping criterion is either a predefined acceptable error or a maximum 'reasonable' number of evaluations of the objective function. It is very important to find an appropriate stopping condition and let the algorithm run forever.

In our analysis we are concentrated in choosing maximum number of iteration and maximum of non-improvement iteration in the solution as a stopping condition.

5 Experimental results

In this section, results of experiments, which goal was to compare implemented hybrid estimation of distribution algorithm with multi start constructive initial solution, are presented. The data files which have been used as test bench are derived from those of Solomon (1987) and Gehring and Homberger (1999).

5.1 Implementation and instances

For our experimental results, we consider six sets of instances to evaluate the performance of the hybrid EDA algorithm with a 100 customer instances developed by Solomon (1987), in addition to the 200, 400, 600, 800 and 1,000 customer instances of Gehring and Homberger (1999). In these Euclidean instances, the travel time between two customer locations is equal to the Euclidean distance.

As described above, we have used three different classes of instances. The first class R is composed of the data files randomly generated entitled R1 and R2 in which the coordinates of the visits are distributed at random. The second class C is composed by the clustered instances in which the visits are divided into several compact groups from a geographical point of view known C1 and C2. Moreover, the third class RC is a mix of random and clustered structures in problem sets by RC1 and RC2 for which part of the visits are grouped geographically and others are grouped at random. Problem sets R1, C1 and RC1 have a short scheduling horizon. In contrast, the sets R2, C2 and RC2 have a long scheduling horizon. The customer coordinates are identical for all problems within one type (i.e., R, C and RC).

The experimental environment described in this paper has been implemented in MATLAB 2009. Experiments are performed on a personal PC (Laptop) Intel ®2Duo 2 GHZ with 3 GB of RAM. Hence, the hybrid EDA algorithm runs five times on each instance, and all results presented below are averages over these five runs.

5.2 Parameter settings

For empirical evaluation, we attempt to find a reasonable set of the hybrid EDA parameter settings. To validate the performance of our hybrid EDA algorithm we will compare it with the results produced by the method described in Azi et al. (2010). Notice that this method solves a different objective function (number of tours minimisation). Moreover, the method presented in Azi et al. (2010) is an exact method, but it was able to solve to optimality only instances up to 40 customers. For these two reasons, the results can only be taken as a validation of the quality of the solution provided by the EDA method we propose.

In this section, we give an experiment environment of different parameters. These parameters are selected after thorough empirical testing. All parameters settings of the hybrid EDA and the multi start constructive method are described in Table 1.

Table 1 Parameters calibration

Parameters	*Description*	*Value*
$iter_{\max}$	Maximal number of iteration	100 and 200
N_{noimp}	Maximum of non-improvement iteration	50
μ	Population size	100
α	Factor used in probabilistic model	$\alpha \in \{1, 10\}$
nr	Number of runs	5
β	Learning rate	$0 \leq \beta \leq 1$

In particular, a value $L_{\max}$ to limit route duration is needed. This value was first set to 100 in the case of R and RC, and 220 in the case of C.

5.3 Numerical results and evaluation method

In this section the capabilities and the effectiveness of our proposed methodology (hybrid EDA) are shown and compared with the results presented in literature (a branch-and-price approach) with a number of customers equal to 25 and 40 in the datasets.

The above experimental results are compared with the results reported in the paper of Azi et al. (2010). In order to validate the effectiveness and competence of our proposed algorithm we present the results with a number of customers equal to 25, 40. The comparison results are carried out on an equal footing and the experimentations is achieved with respecting of the conversion factor. We give the different experimentations in the following tables.

We inform that in literature, no results are performed with a number of customers equal to 100, 200, 400, 600, 800 and 1,000 to solve the MTFFVRPTW.

Table 2 Comparison distances with optimal solutions and results of Azi et al. (2010)

Size	*Class*	*Instances*	*Optimal solutions*	*Solutions of Azi et al. (2010)*	*Our solutions*
25	R	5	844.0	844.1	844.0
	C	11	620.1	624.1	620.1
	RC	7	629.1	635.6	629.1
40	R	3	1,377.5	1,386.4	1,377.5
	C	3	1,071.6	1,289.0	1,071.6
	RC	3	1,093.0	1,105.0	1,093.0

The above experimental results show that our proposed algorithm to solve the multi trip routing problem for the number of customers equal to 25 and 40 are optimal results. Compared with those of Azi et al. (2010) we can decide that our algorithm gives the best performed results.

Table 3 Computational results for instances of class R, C, RC for 100 customers

Class	*Instances*	*Total distances*		*Total CPU time in seconds*	*Number of unserviced customers*
		Number of generation 100	*Number of generation 200*		
R	*20*	1,838.6	1,828.6	28.1	10
C	*20*	2,332.9	2,232.9	34.3	11
RC	*20*	1,909.2	1,899.2	32.5	3
All	*60*	6,080.7	5,960.7	94.9	24
Average		2,026.9	1,986.9	31.63	

Table 4 Computational results for instances of class R, C, RC for 200 customers

Class	*Instances*	*Total distances*		*Total CPU time in seconds*	*Number of unserviced customers*
		Number of generation 100	*Number of generation 200*		
R	*20*	11,134.71	10,300.54	133.98	12
C	*20*	9,845.49	8,967.03	136.21	5
RC	*20*	9,176.93	8,265.77	125.66	11
All	*60*	30,157.13	27,533.34	395.85	28
Average		10,052.37	9,177.78	131.95	

Table 5 Computational results for instances of class R, C, RC for 400 customers

Class	*Instances*	*Total distances*		*Total CPU time in seconds*	*Number of unserviced customers*
		Number of generation 100	*Number of generation 200*		
R	*20*	12,005.43	11,784.78	998.51	24
C	*20*	11,134.96	10,542.30	1,293.04	10
RC	*20*	9,844.50	8,977.51	1,265.31	20
All	*60*	32,984.89	31,304.59	3,556.86	54
Average		10,994.9633	10,434.8633	1,185.62	

Table 6 Computational results for instances of class R, C, RC for 600 customers

Class	*Instances*	*Total distances*		*Total CPU time in seconds*	*Number of unserviced customers*
		Number of generation 100	*Number of generation 200*		
R	*20*	17,623.88	16,985.35	1,137.94	42
C	*20*	14,317.59	14,089.87	1,487.91	35
RC	*20*	14,201.26	14,156.41	981.56	46
All	*60*	46,142,73	45,231.63	3,607.41	123
Average		15,380.91	15,077.21	1,202.47	

Table 7 Computational results for instances of class R, C, RC for 800 customers

Class	*Instances*	*Total distances*		*Total CPU time in seconds*	*Number of unserviced customers*
		Number of generation 100	*Number of generation 200*		
R	*20*	23,250.92	21,789.99	1,590.7	25
C	*20*	15,028.75	14,205.55	1,699.75	44
RC	*20*	17,609.01	15,987.99	1,765.00	55
All	*60*	55,888.68	51,983.53	5,055.45	124
Average		18,629.56	17,327.8433	1,685.15	

Table 8 Computational results for instances of class R, C, RC for 1,000 customers

Class	*Instances*	*Total distances*		*Total CPU time in seconds*	*Number of unserviced customers*
		Number of generation 100	*Number of generation 200*		
R	*20*	27,667.98	26,009.72	2,743.87	33
C	*20*	14,559.03	14,476.76	2,290.44	49
RC	*20*	22,133.95	21,980.09	2,598.41	51
All	*60*	64,360.96	62,466.57	7,632.72	133
Average		21,453.6533	20,822.19	2,544.24	

The results obtained in tables above have used a total of five runs for all instances. The experimental results showing the effectiveness of the hybrid estimation of distribution algorithm to solve the MTFFVRPTW.

For the HEDA the best results have been obtained with a number of generations equal to 200 iterations but when we enhance the number of generations the processing times increases.

Also from these tables, we give the average values of the total distances and the processing times results. Also we give the number of unserviced customers for each class of instances. According to this number obtained with this algorithm we conclude that our methodology maximises the number of serviced customers with the minimisation of the total travelled distances. In all table, we give a number of unserviced customers, we illustrate that the number of serviced customers are optimised which demonstrated by the results reported in the tables above. We conclude that our algorithm is efficient to solve the MTFFVRPTW.

In this experiment, we show in Tables 3, 4, 5, 6, 7 and 8 that the execution time of our proposed solutions is acceptable if we performed a number of generations equal to 100 iterations.

Further experiments are made to compare our solution with those presented in Azi et al. (2010). Table 9 presents the comparison results for the long horizon instances with type 2 (class 2). Table 10 gives the comparison results for all instances.

Table 9 Comparison results for long horizon instances

Class 2 long horizon instances							
		Results of Euchi			*Results of Azi et al. (2010)*		
		Class C2	*Class R2*	*Class RC2*	*Class C2*	*Class R2*	*Class RC2*
100	Unserviced customers (%)	11.0	10.0	3.0	0.0	10.9	24.8
	Total distance	729.2	609.5	489.8	2,221.3	1,828.1	1,894.2
	CPU time (sec)	6.0	6.4	2.9	46.7	32.7	27.9
200	Unserviced customers (%)	2.5	6.0	5.5	0.0	6.2	12.3
	Total distance	3,014.3	2,988.8	3,742.3	9,730.3	11,103.7	9,205.9
	CPU time (sec)	32.6	28.5	42.1	126.2	126.9	135.3
400	Unserviced customers (%)	2.5	6.0	5.0	0.0	6.2	22.4
	Total distance	3,756.3	3,516.9	3,099.1	10,937.2	12,657.7	10,128.0
	CPU time (sec)	425.9	309.4	435.8	340.0	427.7	460.4
600	Unserviced customers (%)	5.8	7.0	7.7	9.9	10.4	20.4
	Total distance	4,736.5	4,288.7	4,625.1	14,626.0	18,903.3	15,557.9
	CPU time (sec)	412.9	412.9	245.7	1,028.5	1,291.4	1,114.2
800	Unserviced customers (%)	5.5	3.1	6.9	24.9	7.9	24.3
	Total distance	4,935.8	4,696.6	4,287.3	14,441.1	26,136.9	20,858.5
	CPU time (sec)	456.3	328.0	333.5	1,747.4	1,678.3	1,842.8
1,000	Unserviced customers (%)	4.9	3.1	5.1	36.5	10.3	26.3
	Total distance	4,855.8	4,855.9	4,268.0	14,587.6	30,732.2	25,368.3
	CPU time (sec)	760.7	650.3	865.1	2,602.4	2,718.8	2,855.2

As we seen in Table 9, for all instances with type 2 (class R2, C2, RC2), a new best solution was found compared with solution produced by the algorithm of Azi et al. (2010). We observe that for the processing time and the percentage of number of unserviced customers, our solution methodology could obtain an best solution in the number of customers served. In fact, it gives the CPU time in seconds, which is needed by our hybrid algorithm to finds the best solution. In all Instances for type 2, we observe that the algorithm provide a good computational time.

From Table 10, we can see that the Hybrid EDA provide the best solutions than the results of Azi et al. (2010). In comparison with the total travel distance and the total CPU time en seconds, we remarks that our algorithm is the best and it gives a good results to find a best solutions.

Table 10 Comparison results for all instances

All instances							
		Euchi			*Azi et al.*		
Number of customers	*Number of instances*	*Total travel distance*	*Total CPU time in seconds*	*% of unserviced customers*	*Total travel distance*	*Total CPU time in seconds*	*% of unserviced customers*
100	20	1,828.6	28.5	48.0	1,997.2	34.5	11.9
200	20	8,966.4	96.5	28.0	10,017.4	131.1	6.2
400	20	11,154.2	993.4	27.0	11,241.0	409.4	9.5
600	20	14,089.9	1,120.9	41.0	16,431.0	1,050.6	12.2
800	20	14,118.0	1,486.4	31.0	20,478.8	1,756.2	19.0
1,000	20	14,476.8	2,001.6	26.6	23,562.7	2,725.5	24.4

6 Conclusions

We have debated a MTFFVRPTW and recommended a hybrid estimation algorithm based on the path scanning procedure to construct an initial solution. Empirical results demonstrated that our algorithm has generated heartening results.

In this paper, we have designed a good version of HEDA for multi-trip VRPTW and we have studied its performance via operated a set of instances benchmarks presented in literature.

In the development of comparing the algorithm, the contributions of the initial construction solution and the use of probabilistic model to the performance of the HEDA are also displayed.

The results of the proposed meta-heuristic are compared to the results of optimal solutions with the number of customers equal to 25 and 40 provided by Azi et al (2010) on a set of each class problems. The results display the competitiveness and accuracy of proposed HEDA algorithm.

The results explained that the solution twisted by our proposed methodology was extremely reliant on the choice of the initial solution. From the conduct experiment approved here we bring to a close that the proposed methodology is competent to solve the hard MTFFVRPTW.

References

Alonso, F., Alvarez, M.J. and Beasley, J.E. (2008) 'A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions', *Journal of the Operational Research Society*, Vol. 59, No. 7, pp.963–976.

Azi, N., Gendreau, M. and Potvin, J-Y. (2007) 'An exact algorithm for a single-vehicle routing problem with time windows and multiple routes', *European Journal of Operational Research*, Vol. 178, No. 3, pp.755–766.

Azi, N., Gendreau, M.d and Potvin, J-Y. (2010) 'An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles', *European Journal of Operational Research*, Vol. 202, No. 3, pp.756–763.

Battarra, M., Monaci, M. and Vigo, D. (2009) 'An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem', *Computers & Operations Research*, Vol. 36, No. 11, pp.3041–3050.

Bodin, L.D., Golden, B.L., Assad, A.A. and Ball, M.O. (1983) 'Routing and scheduling of vehicles and crews, the state of the art', *Computers & Operations Research*, Vol. 10, No. 2, pp.69–211.

Bosman, P.A.N. and Thierens, D. (2002) 'Multiobjective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms', *International Journal of Approximate Reasoning*, Vol. 31, No. 3, pp.259–289.

Brandao, J. and Mercer, A. (1998) 'The multi-trip vehicle routing problem', *Journal of the Operational Research Society*, Vol. 49, No. 8, pp.799–805.

Bräysy, O. and Gendreau, M. (2005) 'Vehicle routing problem with time windows, Part II: metaheuristics', *Transportation Science*, Vol. 39, No. 1, pp.119–139.

Caramia, M. and Onori, R. (2008) 'Experimenting crossover operators to solve the vehicle routing problem with time windows by genetic algorithms', *Int. J. of Operational Research*, Vol. 3, No. 5, pp.497–514.

Choi, E. and Tcha, D-W. (2007) 'A column generation approach to the heterogeneous fleet vehicle routing problem', *Computers & Operations Research*, Vol. 34, No. 7, pp.2080–2095.

Clarke, G. and Wright, J.W. (1964) 'Scheduling of vehicles from a central depot to a number of delivery points', *Operations Research*, Vol. 12, No. 4, pp.568–581.

Dantzig, G-B. and Ramser, J-H. (1959) 'The truck dispatching problem', *Management Science*, Vol. 6, No. 1, pp.80–91.

Euchi, J. and Chabchoub, H. (2010) 'A hybrid tabu search to solve the heterogeneous fixed fleet vehicle routing problem', *Logistics Research*, Vol. 2, No. 1, pp.3–11.

Euchi, J. and Chabchoub, H. (2011) 'Hybrid metaheuristics for the profitable arc tour problem', *Journal of the Operational Research Society*, Vol. 62, No. 11, pp.2013–2022.

Euchi, J., Chabchoub, H. and Yassine, A. (2011) 'New evolutionary algorithm to solve the vehicle routing problem with private fleet and common carrier', *International Journal of Applied Metaheuristic Computing*, Vol. 2, No. 1, pp.58–82.

Fleischmann, B. (1990) *The Vehicle Routing Problem with Multiple Use of Vehicles*, Fachbereich Wirtschaft swissen schaften, Universitat Hamburg, Germany.

Gehring, H. and Homberger, J. (1999) 'A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows', in Miettinen, K., Makela, M.M. and Toivanen, J. (Eds.): *Proceedings of EUROGEN99-Short course on Evolutionary Algorithms in Engineering and Computer Science, Report of the Department of Mathematical Information Technology*, No. A2/1999, University of Jyvaskyla, Finland, pp.57–64.

Golden, B.L., DeArmon, J. and Baker, E.K. (1983) 'Computational experiments with algorithms for a class of routing problems', *Computers and Operations Research*, Vol. 10, No. 2, pp.47–59.

Hauschild, M. and Pelikan, M. (2011) 'An introduction and survey of estimation of distribution algorithms', *Swarm and Evolutionary Computation*, Vol. 1, No. 3, pp.111–128.

Holland, J.H. (1962) 'Outline for logical theory of adaptive systems', *J. Assoc. Comput. Mach.*, Vol. 3, No. 3, pp.297–314.

Laporte, G. (1992) 'The vehicle routing problem: an overview of exact and approximate algorithms', *European Journal of Operational Research*, Vol. 59, No. 3, pp.345–358.

Larrañaga, P. (2002) 'A review on estimation of distribution algorithms', in Larrañaga, P. and Lozano, J.A. (Eds.): *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, pp.80–90, Kluwer, Boston, MA.

Lau, H.C., Sim, M. and Teo, K.M. (2003) 'Vehicle routing problem with time windows and a limited number of vehicles', *European Journal of Operational Research*, Vol. 148, No. 3, pp.559–569.

Li, F., Golden, B-L. and Wasil, E-A. (2007) 'A record to record travel algorithm for solving the heterogeneous fleet vehicle routing problem', *Computers & Operations Research*, Vol. 34, No. 9, pp.2734–2742.

Liu, F-H. and Shen, S-Y. (1999) 'A route-neighborhood-based metaheuristic for vehicle routing problem with time windows', *European Journal of Operational Research*, Vol. 118, No. 3, pp.485–504.

Mühlenbein, H. and Mahnig, T. (1999) 'FDA – a scalable evolutionary algorithm for the optimization of additively decomposed functions', *Evolutionary Computation*, Vol. 7, No. 4, pp.353–376.

Olivera, A. and Viera, O. (2007) 'Adaptive memory programming for the vehicle routing problem with multiple trips', *Computers & Operations Research*, Vol. 34, No. 1, pp.28–47.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, California.

Petch, R.J. and Salhi, S. (2003) 'A multi-phase constructive heuristic for the vehicle routing problem with multiple trips', *Discrete Applied Mathematics*, Vol. 133, Nos. 1–3, pp.69–92.

Potvin, J-Y. and Rousseau, J-M. (1993) 'A parallel route building algorithm for the vehicle routing and scheduling problem with time windows', *European Journal of Operational Research*, Vol. 66, No. 3, pp.331–340.

Rajmohan, M. and Shahabudeen, P. (2009) 'Metaheuristic for solving routing problem in logistics management', *Int. J. of Operational Research*, Vol. 6, No. 2, pp.223–246.

Salhi, S. and Petch, R. (2007) 'A GA based heuristic for the vehicle routing problem with multiple trips', *Journal of Mathematical Modelling and Algorithms*, Vol. 6, No. 4, pp.591–613.

Solomon, M.M. (1987) 'Algorithms for the vehicle routing and scheduling problems with time window constraints', *Operations Research*, Vol. 35, No. 2, pp.254–265.

Taillard, E.D., Laporte, G. and Gendreau, M. (1996) 'Vehicle routing with multiple use of vehicles', *Journal of the Operational Research Society*, Vol. 47, No. 9, pp.1065–1070.

Yellow, P. (1970) 'A computational modification to the savings method of vehicle scheduling', *Operational Research Quarterly*, Vol. 21, No. 2, pp.281–300.