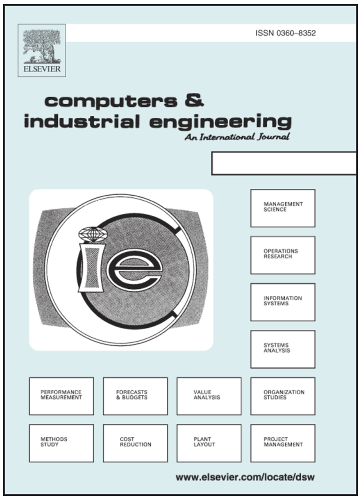

A hybrid estimation of distribution algorithm for the vehicle routing problem with time windows

Ricardo Pérez-Rodríguez, Arturo Hernández-Aguirre

PII: S0360-8352(19)30091-9
DOI: https://doi.org/10.1016/j.cie.2019.02.017
Reference: CAIE 5703

To appear in: *Computers & Industrial Engineering*

Received Date: 25 February 2018
Revised Date: 10 December 2018
Accepted Date: 13 February 2019

Title

A hybrid estimation of distribution algorithm for the vehicle routing problem with time windows

Authors

Ricardo Pérez-Rodríguez[a], Arturo Hernández-Aguirre[b]

[a]CONACYT - CIMAT, A.C. Bartolomé de las Casas 314, Barrio la Estación, 20259, Aguascalientes, Ags, México.
[b]CIMAT, A.C. Jalisco s/n, Valenciana, 36240, Guanajuato, Gto, México.

Corresponding Author

Ricardo Pérez-Rodríguez
CIMAT, A.C. Bartolomé de las Casas 314, Barrio la Estación, 20259, Aguascalientes, Ags, México.
phones +52(449)9185061
email ricardo.perez@cimat.mx

Abstract

The Vehicle Routing Problem (VRP) seeks to find minimum-travel routes for a set of vehicles. The routes contain a set of customers with known demands. Each vehicle departs and arrives at the same depot. In the vehicle routing problem with time windows (VRPTW), each vehicle has to arrive in a specific time window with each customer and also each vehicle has to return to the depot before a due time. In this research, the use of an estimation of distribution algorithm to solve the problem is proposed. The algorithm uses the generalized Mallows distribution as a probability model to describe the distribution of the solution space. Homberger-Gehring's instances are used as input and test parameters in order to show that the modification of the generalized Mallows distribution mentioned is able to produce competitive sequences for the VRPTW against some other estimation of distribution algorithms used in permutation-based optimization problems.

Keywords

Estimation of distribution algorithm; Mallows distribution; vehicle routing problem; combinatorial optimization.

1. Introduction

The vehicle routing problem (VRP) exists in many logistics enterprises as a real-world problem. The VRP arises in the field of transportation, distribution, and logistics. The distribution of merchandise is an example where the VRP and its variants arise. In some market sectors, transportation means a high percentage of the price to goods. In addition, in some economics sectors, the transportation of merchandise is considered a logistics issue. Therefore, the utilization of computerized methods for transportation often results in significant savings ranging from five per cent to twenty per cent of the total costs, as reported in Toth and Vigo (2001). 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 $V = \{0,1,2,\dots,n\}$ and an edge set E. Vertex 0 is the depot, to be consistent with Homberger-Gehring's (2005) instances. Each other vertex $i > 0$ represents a customer that requires a nonnegative demand q_i 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 minimum total travel 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) 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 vehicles scheduled have to return to the depot within $[0, b_1]$ where b_1 is the maximal operation time for each vehicle.

A complete formulation of the mathematical integer linear programming model for the VRPTW was developed by Toth and Vigo (2001). The aforementioned formulation is detailed below based on the seventh chapter of the Toth and Vigo's (2001) book.

The depot is represented by the nodes 0 and $n+1$ to be consistent with Toth and Vigo (2001). All feasible vehicle routes correspond to paths in $G=(V,E)$ that start from node 0 and end at node $n+1$. A time window is associated with nodes 0 and $n+1$, i.e., $[a_0,b_0]=[a_{n+1},b_{n+1}]=[E,L]$, where E and L represent the earliest possible departure from the depot and the latest possible arrival at the depot, respectively. Moreover, zero demands and service times are defined for these two nodes , that is, $de_0=de_{n+1}=s_0=s_{n+1}=0$. Two types of variables exist in this formulation: flow variables $x_{ijk}(i,j)\in A, k\in K,$ equal to 1 if arc (i,j) is used by vehicle k and 0 otherwise, and time variables $w_{ik}, i\in V, k\in K$, specifying the start of service at node i when serviced by vehicle k.

$\min \sum_{k\in K}\sum_{(i,j)\in A} di_{ij}x_{ijk}$ (VRPTW.1)

subject to

$\sum_{k\in K}\sum_{j\in\Delta^+(i)} x_{ijk}=1 \quad \forall i\in N$ (VRPTW.2)

$\sum_{j\in\Delta^+(0)} x_{0jk}=1 \quad \forall k\in K$ (VRPTW.3)

$\sum_{i\in\Delta^-(j)} x_{ijk}-\sum_{i\in\Delta^+(j)} x_{jik}=0 \quad \forall k\in K, j\in N$ (VRPTW.4)

$\sum_{i\in\Delta^-(n+1)} x_{i,n+1,k}=1 \quad \forall k\in K$ (VRPTW.5)

$x_{ijk}\left(w_{ik}+s_i+t_{ij}-w_{jk}\right)\leq 0 \;\; \forall k\in K, (i,j)\in A$ (VRPTW.6)

$a_i\sum_{j\in\Delta^+(i)} x_{ijk}\leq w_{ik}\leq b_i\sum_{j\in\Delta^+(i)} x_{ijk} \quad \forall k\in K, i\in N$ (VRPTW.7)

$E\leq w_{ik}\leq L \;\; k\in K, i\in\{0,n+1\}$ (VRPTW.8)

$\sum_{i\in N} de_i\sum_{j\in\Delta^+(i)} x_{ijk}\leq C \quad \forall k\in K$ (VRPTW.9)

$x_{ijk}\in\{0,1\} \;\; \forall k\in K, (i,j)\in A$ (VRPTW.10)

The objective function (VRPTW.1) expresses the total distance. Given $N=V\setminus\{0,n+1\}$ represents the set of customers. Constraints (VRPTW.2) restrict the assignment of each customer to exactly one vehicle route. Next, constraints (VRPTW.3)-(VRPTW.5) characterize the flow on the path to be followed by vehicle k. Additionally, constraints (VRPTW.6)- (VRPTW.9) guarantee schedule feasibility with respect to time considerations and capacity aspects, respectively. Note that for a given k, constraints (VRPTW.7) force $w_{ik}=0$ whenever customer i is not visited by vehicle k. Finally, conditions (VRPTW.10) impose binary conditions on the flows variables.

In this research, the hard type time window is considered. It means that it is not possible to arrive after the possible time at each vertex, i.e., b_i. In order to get feasible solutions an example is provided. At the beginning of Fig. 1 an example of a sequence of vertices (permutation) S of six vertices without route delimiters, i.e., $S=(1,4,5,2,3,6)$, viewed as a giant tour is shown. Each vertex has a demand in brackets and a time window in square brackets. To simplify, travel times are here equal to distances, i.e., $d_{ij}=t_{ij}$ for each edge $[i,j]$. Table 1 details the data for the instance used in Fig 1. In addition, Fig. 1 gives an example of splitting and is shown at the lower part of Fig. 1. The splitting scheme satisfies the set of constraints for the VRPTW, i.e., it ensures that all the vertices are assigned to a single vehicle. Furthermore, it takes care of the requirement that the sequence of vertices must respect precedence, i.e., the precedence relationships between the vertices of the sequence are not violated. In addition, the solution depicted in Fig. 1 guarantees that multiple vertices cannot be visited at the same time on any possible vehicle. The auxiliary graph is given in the middle of Fig 1, assuming a vehicle capacity $Q=50$. Each arc represents a feasible trip. For example, arc 1 models the trip reduced to vertex 1: its length 165.71 is twice the length of edge [0,1] and the corresponding service time at vertex 1. Another example, arc [1,4] is not represented, since a delay occurs when a vehicle can reach the vertex 4, at time 897.8. All the other trips, which are not represented, violate either vehicle capacity or the time window. Finally, Bellman's algorithm determines the shortest path, and determines the number of vehicles used. The labels computed are given above each node. The number of vehicles used is 4. The lower part of the figure shows the resulting solution, with departure times in brackets.

Additionally, Table 2 and Fig. 2 present an alternative sequence for the aforementioned instance. Based on Fig. 2, the performance is improved using the alternative sequence.

Table 1. A VRPTW instance.

Fig. 1. Illustration of a sequence for the VRPTW.

Table 2. Data for an alternative sequence.

Fig. 2. Illustration of an alternative sequence for the VRPTW.

As the VRP, the VRPTW requires special attention from researchers due to its peculiar complexity and interesting usefulness in real-logistics cases. Surveys of solution methods for VRPTW are available in Golden and Assad (1986), Desrochers et al. (1988), Solomon and Desrosiers (1988), and Bräysy and Gendreau (2005). Developments can be found in Potvin and Rousseau (1993), Rochat and Taillard (1995), Thangiah et al. (1995), Potvin and Bengio (1996), Homberger and Gehring (1999), Cordeau et al. (2001), Tan et al. (2001), Berger and Barkaoui (2004), Cordeau et al. (2004) and Mester et al. (2007). Recent surveys on VRP and its variants can be found in Braekers et al. (2016), Montoya-Torres et al. (2015), Gendreau et al. (2015), and Koç et al. (2016).

Evolutionary algorithms have been considered competitive procedures for tackling the VRP problem and its variants. A relatively new paradigm of evolutionary algorithms is the estimation of distribution algorithms (EDA) introduced by Mühlenbein and Paaß (1996). 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. The aforementioned information is represented by all the values contained in the solution vectors, i.e., the parents (Fig. 3(a)). Each element belonging to the solutions is considered as a variable. Identifying the values that each element (variable) has taken, it is possible to build the charts shown in (Fig. 3(a)). A probability model is used to generate new individuals (Fig. 3(b)). It is called a sampling process. Each element of a new solution is obtained through the cumulative distribution of all possible values in the corresponding element (variable). A uniform random number is used to select the value in the element. It is computed for each element of the offspring (Fig. 3(b)). 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 (Wang et al., 2012). To describe the distribution of the solution space, the EDA tries to determine the relationship or interaction among variables of the problem as a primary objective. Different estimations of the relationships mentioned are shown in Fig. 3(c). In some cases, the possible values for each element are not influenced by the values of the rest of the elements, i.e., without interaction. In other cases, the possible values for each element might be influenced by the values of some element, i.e., bivariate interaction. Finally, the possible values for each element might be influenced by the values of more than two elements, i.e., multiple interactions. Identifying the best estimation of the interaction is the main goal in this kind of algorithm.

Fig. 3. Estimation of distribution algorithm main procedure.

The traditional evolutionary operators are replaced by the probability model that is built with information on 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. However, how can we improve the estimation of the relationships among variables of the problem? How does the above mentioned estimation affect the performance of the algorithm? Different and diverse ways to estimate the relationships among variables have been proposed, such as the Mutual Information Maximizing Input Clustering algorithm (MIMIC) based on De Bonet et al. (1997). The MIMIC uses the entropy concept to estimate the relationships (Figure 4(a)). The Combining Optimizers with Mutual Information Trees Algorithm (COMIT) is based on Baluja and Davies (1997). The COMIT uses the mutual information concept to estimate the relationships (Figure 4(b)). The Bayesian Optimization Algorithm (BOA) is based on Pelikan et al. (1998). The BOA uses the Bayesian Dirichlet metric to measure of how well the network models the data (Figure 4(c)).

Fig. 4. Different types of estimation of distribution algorithms.

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 the estimation of any relationship and interaction between vertices on the sequence of the VRPTW solution can be improved. Such a relationship should be used and exploited in order to get competitive solutions. In addition, any probability model should be able to represent the characteristics exhibited by the parents as a distribution of the solution space. Therefore, the EDA proposed should not have to need to hold members of the population to keep track and inherit the characteristics exhibited by the parents mentioned in each generation as in other algorithms. In particular, the genetic algorithms with specific crossover and mutation operators have been proposed in order to preserve partial information of the members of the population. Research by Potvin (1993), Hwan (2002), Larrañaga et al. (1999) and Prins (2004) are examples of those algorithms. The proposed EDA should guarantee that the characteristics will persist for all members of the population in the evolutionary progress of the probability model proposed. Meanwhile, the probability model is updated in each generation with the potential individuals of the new population. In such an iterative way, the population evolves, and satisfactory solutions can be obtained (Wang et al., 2012).

The reason for proposing an EDA for the VRPTW is to show that a better estimation of any relationship and interaction between vertices on the sequence of the VRPTW solution can be improved using an explicit probability distribution. Such information allows for the avoidance of using the traditional evolutionary operators for permutation-based problems. The contribution of this research is to implement an EDA to identify, by statistical information, the distribution of the solution space and indicate any interaction between the vertices and their positions on the vehicle routing sequences. Therefore, we use the Mallows distribution model in this research to obtain a better estimate of the relationships between the vertices. The Mallows model was initially proposed by Mallows (1957) and later improved upon by Fligner & Verducci (1986) through the generalized Mallows distribution (GMD). A more extensive explanation of the GMD characteristics is found in Fligner & Verducci (1986) and Fligner & Verducci (1988). More recently, Ceberio et al. (2014) contributed to an initial application of the GMD to solve the flow-shop scheduling problem (FSP) coupled with an EDA.

As a popular model for rankings (Meilă et al., 2007), the GMD is suitable to utilize if there is a need to rank k vertices, in the vehicle routing problem, according to some criteria. The main challenge is to characterize suitable solutions on a random set of rankings, i.e., sequences of the VRPTW. In the GMD, the main process for ranking, the sequences, is to decompose it into $k-1$ stages, i.e., $k-1$ positions (or columns) as shown in Fig. 5.

Fig. 5. An example of a probability distribution associated with the space of possible solutions.

The GMD, as in other models, establishes a probability that one vertex is preferred to another in a specified stage, i.e., position, in the solution. Therefore, as a basic model, the most preferred vertex is chosen in the first stage with a high probability. After that, the best of remaining vertices is chosen in the second stage, with the corresponding probability, and so on until the last vertex is chosen by default. Then, each sequence of the VRPTW solution will produce a ranking. Different models have been proposed in order to compute a proper estimate of the probability of choosing a particular vertex at any stage. However, the analytic intractability of such models is a major difficulty to use in a real-world problem. The GMD avoids the analytical difficulties by assuming that the accuracy of the choice made at any stage is independent of the accuracies in the other stages (Fligner & Verducci, 1988). In order to avoid the analytical difficulties mentioned above, the GMD uses a central ranking, i.e., a consensus ranking. It is obtained from a random set of vectors, sequences of the VRPTW. A conventional definition of the central ranking is it minimizes the total number of disagreements with rankings contained in the set of rankings (Ali & Meilă, 2012). The GMD uses the Kendall's metric to compute the total number of disagreements mentioned above. The Kendall's metric has been the measure of choice in many recent applications centered on the analysis of ranked data (Cohen et al., 1998). To estimate a suitable central ranking is a difficult task, since the problem is NP-hard even for only four items (Bartholdi et al., 1989; Cohen et al., 1998). Nevertheless, once a central ranking is computed, the GMD produces a multistage model in order to estimate the probabilities in each stage.

Although the EDA was originally designed to solve integer or real-valued domain problems, it cannot produce a feasible solution from a probability model built on a permutation-based representation (Larrañaga & Lozano, 2001). Furthermore, the EDA requires being adapted to deal with permutation-based problems by means of a modification in the algorithm process. Then, the implementation of a specific probability model for this issue would be more competitive against other models. The Mallows model (1957) is a distance-based exponential probabilistic model considered analogous to the Gaussian probability distribution over the space of permutations. In the Mallows distribution, the probability value of each permutation $\sigma \in T_n$ (where T_n stands for the set of $n!$ permutations of n items) depends on just two parameters: a spread parameter θ and the distance to a central permutation σ_0. This model assigns to each permutation σ a probability that decays exponentially with respect to its distance to the central permutation σ_0. In this way, $P(\sigma)$ can be considered as an exponential non-uniform distance-based ranking model (Ceberio et al., 2014). Both parameters, i.e., the spread parameter θ and the central permutation σ_0, are estimated. Therefore, different estimates of the parameters mentioned should be considered in order to get a better performance to solve combinatorial optimization problems such as VRPTW.

Contrary to current research where diverse techniques and approaches have been proposed in order to solve the problem that is being examined, this paper contributes to the state of the art as follows:

- To introduce the GMD, detailed in Fligner and Verducci's (1986) research, for the VRPTW as a way of estimating an explicit probability distribution in the domain of permutations for any EDA.
- To apply the GMD coupled with an EDA, called HMEDA (Hybrid Mallows Estimation of Distribution Algorithm), for solving the VRPTW on a multi-objective perspective.
- To propose a different way of improving the estimation of the central permutation σ_0 in the GMD process using a Pareto-front approach to help the HMEDA to find better solutions for the VRPTW.
- The purpose of this research is to propose a different experimental technique to enhance the performance of the algorithm mentioned above for solving the VRPTW using a Pareto-front approach.

The proposed algorithm has a wide range of applications for automotive, aerospace, chemical, metallurgical, pharmaceutical, and food industries, among others. The users of this contribution are academics, engineers, managers, and practitioners related to the transportation of merchandise.

The remainder of this paper is organized as follows: in section 2, the related work is depicted. In section 3, the proposed HMEDA is detailed. In section 4, computational results and comparison with other multiobjective and recent algorithms are presented. Section 5 shows a simulation and preliminary implementation. Finally, the concluding remarks are made in section 6.

2. Related work

2.1.1. Exact methods

Exact methods often perform very poorly, i.e., in some cases taking days or more to find moderately decent, let alone optimal, solutions even to fairly small instances (El-Sherbeny, 2010).
The Lagrangiang relaxation-based methods are used in Fisher et al.´s (1997) research. The column generation approach is used for the first time in Agarwal et al. (1989) to solve the VRP problem, while Desrosiers et al. (1984) used the column generation approach to solve the m-TSP with time windows. In Desrosiers et al.'s (1984) research, the column generation approach is used for the first time to solve the VRPTW, and a more effective version of the same model with the addition of valid inequalities solves more instances to optimality in Desrochers et al.'s (1992) research.
The dynamic programming approach for VRPTW is presented for the first time in Kolen et al. (1987) and Christofides and Beasley (1984) uses the dynamic programming paradigm to solve the VRP. In Kolen et al. (1987) problems up to 15 customers are solved by this method.
The algorithm of Kohl and Madsen (1997) uses branch-and-bound to achieve optimality. Each node α in the branch-and-bound tree corresponds to three sets: F(α) which the set is of fixed feasible routes starting and finishing at the depot, P(α) which is a partially built route starting at the depot and C(α) denotes the set of customers forbidden to be next on P(α). Branching is done by selecting a customer i that is not forbidden, that is iC(α), and that does not appear on any route, that is iF(α) ∪ P(α). Branching decisions are taken on route-customer allocations. Then two branches are generated: one in which the partially build route P(α) is extended by i and one where i is forbidden as the next customer on the route that is added to C(α). Customer i is chosen as the customer the partial route P(α) was extended with in the calculation that lead to the lower bound of node α. At each branch-and-bound node dynamic programming is used to calculate a lower bound on all feasible solutions defined by F(α), P(α) and C(α).

2.1.2. Evolutionary algorithms

Variants of Genetic Algorithms (GA) have been proposed to address the VRPTW, a problem known to be NP-hard. Potvin and Bengio (1996) propose a GA for the VPRTW that directly applies genetic operators to solutions, thus avoiding coding issues. The cheapest insertion heuristic proposed by Solomon (1987) is used to create the initial population. The fitness values of the proposed approach are based on the number of vehicles and the total route time of each solution. The selection process is stochastic and highly biased towards the best solutions. For this purpose, a linear ranking scheme is implemented. The linear ranking scheme prevented individuals, with significantly better fitness values than average, from dominating the selection process. During the recombination phase, two parent solutions are merged into a single one, to guarantee the feasibility of the new solution. Two types of crossover operators are used, namely a route-based and a sequence-based crossover. The route-based crossover replaced one route of parent solution P2 with a route of parent solution P1. However, in the sequence-based crossover only a randomly defined end part in a route of parent-solution P1 has been replaced by a set of customers served by a route of parent solution P2. A repair operator is proposed to remove duplicates and insert missing customers into the solution. Finally, in order to locally optimize the solution, a mutation operator based on Or-opt exchanges is considered.

Wang et al. (2007) address the dynamic vehicle routing problem with time windows. The authors present an approach to the search for the best routes in a dynamic network. A dynamic route evaluation model for modeling the responses of vehicles to changing traffic information, a modified Dijkstra's double bucket algorithm for finding the real-time shortest paths, and an improved evolutionary algorithm for searching the best vehicle routes in the dynamic network are proposed. In the representation, a modified form of random keys is employed. A solution vector, called chromosome, consists of genes and each gene represents a customer node. The customer nodes have fixed gene positions in the chromosomes and the order in which they are visited is determined by sorting based on the gene values. The random keys have information about the vehicle number used for a service and the value for sorting, where the digit before the point represents the vehicle number and the digit after the point are used as sort keys to decode visiting sequences. The proposed approach has been evaluated by simulation experiments. It has been found that the proposed approach is quite efficient in finding real-time best vehicle routes where the customer nodes and network information change dynamically.

Kamkar et al. (2010) introduce a cellular Genetic Algorithm (cGA) to solve the VRPTW. It is a GA subclass where the population diversity and exploration are enhanced thanks to the existence of small overlapped neighborhoods. The population is structured in a specific topology, so that individuals may only interact with their neighbors. This kind of structured algorithm is well suited for complex problems. For the representation of individuals, an integer string of length N is used, where N is the number of customers. Each gene contains the integer node number assigned to that customer originally, and the sequence of the genes is the order of visiting those customers. Tournament selection is used. The authors employ the PMX crossover and inverse mutation for the variation process. The benchmark of Solomon (1987) is selected for testing the proposed cGA and compares them with some other algorithms. The results demonstrate that the proposed cGA has the potential to solve these kinds of problems. However, only a few genetic-based approaches exploit knowledge provided by the problem or by the structure of the solutions computed during the evolutionary process to explore the solution space.

2.1.3. Hybridization-based algorithms

Other approaches are the hybridization-based algorithms that have been widely implemented. In Bräysy et al. (2004) present methods that hybridize ideas of evolutionary algorithms with some other search techniques, such as Tabu Search (TS) or Simulated Annealing (SA). These techniques have been also used to solve the VRPTW. Most of the hybrid methods presented have used local search mutation instead of the random mutation operators. In the first phase, either the cheapest insertion heuristic or the sectoring based genetic algorithm creates an initial solution. The second phase is applied to one of the following search procedures that use the λ-interchange mechanism: a local search descent procedure, an SA algorithm or a hybrid SA and TS, where TS is combined with the SA-based acceptance criterion to decide which moves to accept from the candidate list. The main feature of the local search procedures is that unfeasible solutions with penalties are allowed if considered attractive (Bräysy et al. 2004).

Selma et al. (2013) deal with the two-dimensional loading capacitated VRPTW, which is a realistic extension of the well-known VRP. It consists of determining vehicle trips to deliver rectangular objects to a set of customers with known time windows, using a homogeneous fleet of vehicles, while ensuring a feasible loading of each vehicle used. Firstly, six heuristics are designed to quickly compute good solutions for realistic instances. The solutions are obtained by combining algorithms for the VRPTW with heuristics for packing rectangles. Then, a Memetic Algorithm (MA) is developed to improve the heuristic solutions. In this research, a chromosome is considered as a sequence of *n* clients, without trip delimiters. This sequence can be interpreted as a giant tour where the vehicle capacity, the time windows, and the loading constraints are relaxed. It permits reusing classical crossover operators designed for the traveling salesman problem (TSP). The Prins' (2004) procedure to split the giant tour unfeasible tours is used in this research. The quality and efficiency of the proposed heuristics and metaheuristic are evaluated by adding time windows to a set of instances.

2.1.4. Local Search-based algorithms

Local search-based algorithms and their variants are another alternatives for the VRPTW. Cordeau et al. (2001) have introduced a simple TS procedure for the VRPTW and two of its extensions, namely Periodic-VRPTW (PVRPTW) and Multi-depot VRPTW (MDVRPTW). An important feature of the approach is the possibility of exploring unfeasible solutions during the search. Braysy and Gendreau (2005) have reported a classical two-phase mechanism to solve the VRP and the VRPTW. The initial solution is first generated using the savings heuristic. Then, intra-route local searches (2-opt and Or-opt) and three inter-route operators guided with TS are used to refine the solution.

Rochat and Taillard (1995) present a probabilistic technique to diversify, intensify and parallelize a local search adapted for solving vehicle routing problems. This technique may be applied to a very wide variety of vehicle routing problems and local searches. It is shown that efficient first level taboo searches for vehicle routing problems may be significantly improved with this technique. Moreover, the solutions produced by this technique may often be improved by a post-optimization technique presented in this paper too. The solutions of nearly 40 problem instances of the literature were improved.

Homberger and Gehring (2005) introduce a two-phase hybrid metaheuristic for the VRPTW. The objective function of the VRPTW considered here combines the minimization of the number of vehicles (primary criterion) and the total travel distance (secondary criterion). The aim of the first phase is the minimization of the number of vehicles by means of a (μ,λ)-evolution strategy, whereas in the second phase the total distance is minimized using a TS algorithm. The two-phase hybrid metaheuristic was subjected to a comparative test on the basis of 356 problems from the literature with sizes varying from 100 to 1000 customers. The derived results show that the proposed two-phase approach is very competitive.

Créput et al. (2007) presented an evolutionary algorithm embedding self-organizing maps (SOM) as operators to address the VRPTW. The approach has extended and improved SOM-based neural networks (NN) applications to the VRPTW. From the point of view of NN, the evolutionary framework introduced a level of supervision; however, it corresponded to a selection principle at a higher level with the aim of allowing simplicity and flexibility and favored further parallel implantations. Operators have a similar structure based on closest point findings and simple moves performed in a Euclidean plane.

Diverse works have presented parallel co-operative methodologies in which several agents communicate through a pool of feasible solutions. The agents consist of a simple construction and local search algorithms and different metaheuristic methods, namely evolutionary algorithms and TS. The evolutionary algorithms use a probabilistic mutation and the well-known edge recombination and order crossovers, while the TS procedures are adaptations of Cordeau et al. (2001). The fitness value of solutions is based on the number of vehicles, distance and waiting times. The pool is initialized with a set of four simple construction heuristics: least successor, double-ended nearest neighbor, multiple fragments (which adds sequentially the shortest arcs) and shortest arc hybridizing (a probabilistic version of the previous). The initial and final solutions are post-optimized with an ejection chain procedure and well-known 2-opt, 3-opt, and Or-opt improvement heuristics. Alternative approaches are ant colony, bee and bat algorithms. Most of them belong to population-based algorithms with particular features.

2.1.5. The multi-objective approach

Ombuki et al. (2006) represent the VRPTW as a multi-objective problem and present a genetic algorithm solution using the Pareto ranking technique. The authors use a direct interpretation of the VRPTW as a multi-objective problem, in which the two objective dimensions are the number of vehicles and total cost (distance). An advantage of this approach is that it is unnecessary to derive weights for a weighted sum scoring formula. This prevents the introduction of solution bias towards either of the problem dimensions. The authors argue that the VRPTW is most naturally viewed as a multi-objective problem, in which both vehicles and cost are of equal value, depending on the needs of the user. A result of their research is that the multi-objective optimization genetic algorithm returns a set of solutions that fairly consider both of these dimensions. Their approach is quite effective, as it provides solutions competitive with the best known in the literature, as well as new solutions that are not biased toward the number of vehicles. A set of well-known benchmark data are used to compare the effectiveness of the proposed method for solving the VRPTW.

Chand and Mohanty (2011) proposed a multi objective genetic algorithm (MOGA) for the VRPTW in order to minimize the number of vehicles used, the total distances traveled by the vehicles, and the time window violation (routing time). In their approach, a chromosome representing route of length N, where N is the number of customers in a particular problem instance. A gene in a given chromosome indicates the original node number assigned to a customer, while the sequence of genes in the chromosome indicates the order of visitation of customers. For crossover and mutation operators, best cost route crossover techniques and exchange mutation procedures are used respectively. The authors use dominance-information of the individuals of the population by calculating for each individual i the number of alternatives X_i from which this individual is dominated. Individuals that are not being dominated by others should receive a higher fitness value than individuals that are being dominated. The authors did experimental results using the Solomon's data set.

Kumar et al. (2014) detailed a fitness aggregated genetic algorithm (FAGA) to solve the multi objective problem. The fitness considered in this research is: the number of vehicles used, the total distances traveled by the vehicles, and the route balance. The algorithm mentioned uses specialized operators like selection based on aggregate fitness value, and the best cost route crossover. The algorithm was tested on a large number of Solomon's benchmarks for a bi-objective model that is a minimization of the total distance travelled and the total number of vehicles used. After validation, the third objective that is route balance is incorporated into the bi-objective model and it is observed that FAGA produces better-balanced routes without affecting the total distance travelled and total number of vehicles used.

2.1.6. Recent work

Afifi et al. (2016) propose a simulated annealing-based algorithm for the VRPTW and synchronized visits (VRPTWSyn), where each client requires simultaneous visits from different vehicles as the main characteristic. The algorithm uses local improvement methods to deal with the problem mentioned above. Berghida and Boukra (2015) present an enhanced biogeography-based optimization algorithm (EBBO) for the HVRPMBTW (vehicle routing problem with a heterogeneous fleet, mixed backhauls, and time windows). The available vehicles have different capacities and costs. Two types of customers are served, i.e., linehaul and backhaul customers. The authors combine the biogeography-based optimization (BBO) approach with a simulated annealing algorithm to enhance solution quality. The approach mentioned produces satisfactory results compared to other approaches such as particle swarm optimization (PSO). de Armas and Melián-Batista (2015) detail a dynamic vehicle routing problem in the Canary Islands, Spain, where the demands can be either known at the beginning of the planning horizon or dynamically revealed during it. The research considers a fixed heterogeneous fleet of vehicles, customer priorities and vehicle/customer constraints. This work proposes a general variable neighborhood search (GVNS) as a meta-heuristic procedure. Nalepa and Blocho (2016) introduce an adaptive memetic algorithm for the VRPTW (AMA-VRPTW) to minimize the total travel distance. The parameters of the algorithm, including the population size, selection scheme and the number of offspring, are adjusted dynamically during the search. The performance of the algorithm is investigated in a sensitivity analysis.

Schwarze and Voß (2015) analyze a VRP extension. It is motivated by an application in airport ground control where time issues play a major role. The research includes an extensive numerical study with load-balancing aspects as well as a multi-objective analysis. Soonpracha (2015) considers the customer demand as uncertain in a mix-VRPTW. The research examines meta-heuristic algorithms based on a genetic algorithm and an adaptation of a greedy search hybridized with inter-route neighborhood search methods. Yang et al. (2015) investigate and compare heuristic algorithms for the VRPTW. These heuristic algorithms combine three classical heuristic algorithms with the cross exchange method and the 2-opt exchange heuristic respectively. Barbucha (2014) proposes a new approach for the VRPTW. It 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. Agents are grouped in teams working in islands. Each team of agents periodically communicates with other teams by sharing a promising result. Gan et al. (2014) present the multi-type vehicle routing problem with time windows (MT-VRPTW). The research considers both multiple types of the vehicle and the uncertain number of vehicles of various types. Six variants of particle swarm optimization (PSO) are used. Li et al. (2016) study the multi-depot vehicle routing problem with time windows under shared depot resources (MDVRPTWSDR). In this research, the depot where the vehicle ends is flexible. Thus, it is not entirely the same as the depot that it starts from. The authors implement a hybrid genetic algorithm with adaptive local search.

Recent authors have used well recognized and published methodologies with some modifications in order to solve the VRPTW. Figliozzi (2010) offers an algorithm based on route construction and improvement. The algorithm mentioned can be applied sequentially to solve the VRP with Soft Time Windows (VRPSTW) and Hard Time Windows (VRPHTW). Soft time windows are useful when the customers may be incapable or unwilling to set precise time windows in advance and simply prefer the flexibility to alter their pickup or delivery requests. Computational results detail the performance of the proposed algorithm against other well recognized and published solution methods using the Solomon (1987) benchmark problems.

Tas et al. (2013) study a vehicle routing problem with soft time windows and stochastic travel times. The authors propose a TS method, a well-recognized and published methodology, to solve the problem mentioned. The Solomon

(1987) instances were used to obtain competitive final solutions. In the first phase of the algorithm, an initial solution is constructed by Solomon's (1987) a heuristic method. In the second phase, the initial solution is improved by a TS method based on research by Cordeau et al. (2001). In the third phase, a post-optimization method is applied to improve the solution obtained by the second phase.

Vidal et al. (2013) address efficiently a wide range of large-scale VRP variants. The authors highlight that the most successful approaches to solving the VRPTW involve local search improvement procedures based on arc- and node-exchange neighborhoods. The authors indicate that the most competitive results are currently offered by the hybrid genetic algorithm (HGA) to solve VRP variants. The HGA combines well recognized and published methodologies such as powerful route minimization procedures, effective edge assembly cross operators, and extremely efficient local search procedures. These methods stand out in terms of simplicity and wider applicability. The authors conclude that the design of efficient local-search methods should be investigated to obtain suitable solutions.

The review mentioned above clearly underlines several gaps in the actual state of the art. The algorithms mentioned have greedy procedures to obtain promising solutions. The performance of the algorithms is implicit in the greedy procedures. Therefore, the greedy procedures are replaced by an explicit probability distribution over the set of permutations using the Mallows (1957) model. With this change, we have control to characterize the solution space explicitly. In addition, most current algorithms tackle the problem with the cluster-first route-second strategy, i.e., the procedures mentioned above determine clusters of customers compatible with the constraints, and solve the routing problem for each cluster. However, the opposite approach, called route-first cluster-second, builds a giant tour covering all customers and splits it into feasible trips, as shown Fig. 1. This opposite approach has nevertheless led to successful meta-heuristics for various vehicle routing problems in the last decade (Prins et al., 2014). In addition, the solution representation used in specific cross operators have a strong impact on the efficiency and scalability of meta-heuristics for the VRPTW. In a traditional approach to the solution representation, the vertices are independent of each other. It means that any vertex can be relocated to another position in 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 nobody knows 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 probability model is the objective and has not been considered in all the current research. To the best of our knowledge, probability models by means of an EDA have not been used to tackle the VRPTW problem.

3. HMEDA – for the VRPW

The traditional permutation-based representation is adopted in this research. The representation mentioned is a giant tour of all the locations of the customers, i.e., in each position on the representation a vertex is depicted. Therefore, a representation is simply a sequence of vertices (permutation) S of n vertices without route delimiters. If a vehicle could visit all customers one by one, the representation would detail the order in which the vehicle could perform the sequence. The fitness F(S) of S is the number of vehicles and the total distance used in the solution. This representation without route delimiters is suitable for using traditional evolutionary operators as highlighted in the research of Prins (2004). However, in this research, the traditional operators are not required in order to get competitive offspring.

To explain the representation, we provide an example by considering a problem with 7 vertices as shown in Table 3. Fig. 6 illustrates the representation used in this research.

Table 3. Data used in Fig. 6.

Fig. 6. Illustration of a sequence (giant tour) for the VRPTW.

Based on Fig. 6, a vehicle goes to vertex (customer location) number five at the beginning, after which it goes to vertex number four, and so on until it goes back to the depot (vertex zero), which does not show in the solution representation in this research.

The initial population are arrays of M vectors, where $M = 1000$, of size n vertices each one. Each vector is initialized as a random permutation of vertices (customers).

3.1. Pareto-front construction phase

In each generation, a Pareto-front approach is built with the population. A Pareto-front approach is considered in this research based on Kacem et al. (2002). The optimality notion in the Pareto approaches can be formulated as follows: the Pareto-optimal set is formed by non-dominated solutions. x vector dominates y vector if $\forall 1 \leq q \leq L, f_q(x) \leq f_q(y)$ where $f_q(\,.\,)$ is the qth objective function, L is the number of objective functions and at least one index r exists such that $f_r(x) < f_r(y)$. A solution is non-dominated if it is not dominated by any other solution. Fig. 7 details the fitness of two feasible solutions as a Pareto-optimality approach example. A non-dominated set of solutions in each generation is found and used for the selection process, as well as for the improvement of the central permutation estimate σ_0 in the GMD process.

The selection process takes the subset N from M parents (where N<M) chosen by a tournament selection which is executed based on where the candidate solutions are located on the Pareto-front approach. Different cases are possible as follows: if a solution is feasible and another is infeasible, the feasible solution is preferable. If a solution is non-dominated and another is dominated, the non-dominated solution is preferable. If both solutions are non-dominated, the solution that belongs to a better Pareto-front layer is preferable.

In order to determine which Pareto-front layer belongs to each non-dominated solution, we calculate two entities: 1) domination count n_p, the number of solutions which dominate the solution p, and 2) S_p, a set of solutions that the solution p dominates. All solutions in the first non-dominated front will have their domination count as zero. Now, for each solution p with $n_p = 0$, we visit each member (q) of its set S_p and reduce its domination count by one. In doing so, if for any member q the domination count becomes zero, we put it on a separate list Q. These members belong to the second non-dominated front. Thus, the above procedure is continued with each member and the third front is identified. This process continues until all fronts are identified.

Fig. 7. A graphical representation fitness of two feasible solutions as a Pareto-optimality approach.

3.2. Probability model proposed

Based on the related work section mentioned above, traditional evolutionary operators are used to produce a new population, i.e., performing two main processes (crossing and mutation) to different permutations (parents) in order to build a new permutation (child). These processes have an implicit contribution on the performance of the algorithm used. However, we do not have control to characterize the solution space. Therefore, those processes are replaced by an explicit probability distribution over the set of permutations using the Mallows model in this research. The Mallows model was initially proposed by Mallows (1957), and later improved by Fligner and Verducci (1986) through the generalized Mallows distribution (GMD). The Mallows model and the GMD can be used to solve permutation-based optimization problems. More recently, Ceberio et al. (2014) contribute with an initial application of the GMD to solve the flowshop scheduling problem coupled with EDAs.

Formally, the Mallows model is defined as

$$P(\sigma) = \psi(\theta)^{-1} exp(-\theta D(\sigma, \sigma_0)) \qquad (1)$$

where θ is a spread parameter, and $D(\sigma, \sigma_0)$ is the distance from σ to the central permutation σ_0, where $\psi(\theta)$ is a normalization constant. In the present work, Kendall's -τ is the distance metric with which the Mallows model is coupled.

Let's consider σ_0 = { 1 2 3 4 5 6 }
as a central permutation, i.e., a Giant tour. The number of elements in σ_0 is based on the detailed example in Table 3 and Fig. 6.
Let σ_1 = { 1 4 5 2 3 6 }
as another Giant tour with $D(\sigma_1, \sigma_0) = 4$
Let σ_2 = { 5 4 2 1 3 6 }
with $D(\sigma_2, \sigma_0) = 8$

Based on Mallows model, every permutation σ_i gets a probability that decays exponentially with respect to its distance to the central permutation σ_0. Therefore, σ_2 has less probability than σ_1. Fig. 8 depicts a visual representation of the previously detailed example.

Fig. 8. A visual representation of the Kendall's -τ distance metric.

The GMD decomposes Kendall's -τ distance metric into $m - 1$ terms in such a way that it can be expressed as $D(\sigma, \sigma_0) = D_\tau(\sigma, \sigma_0) = \sum_{q=1}^{m-1} V_q(\sigma, \sigma_0)$. The GMD is given as follows

$$P(\sigma) = \psi(\boldsymbol{\theta})^{-1} exp(\sum_{q=1}^{m-1} -\theta_q V_q(\sigma, \sigma_0)) \tag{2}$$

where $V_q(\sigma, \sigma_0)$ is the number of positions on the right of q with values smaller than the current position on the permutation (σ, σ_0). Based on the example above, for σ_1, the $V_q(\sigma_1, \sigma_0)$ values for the qth position is detailed below

qth	1	2	3	4	5	$\sim m - 1$
σ_1	1	4	5	2	3	
$V_q(\sigma_1, \sigma_0)$	0	2	2	0	0	$\sim \sum 4$

The same procedure to identify the Kendall's -τ distance metric in σ_2, i.e., the $V_q(\sigma_2, \sigma_0)$ values for the qth position is detailed below

qth	1	2	3	4	5	$\sim m - 1$
σ_2	5	4	2	1	3	
$V_q(\sigma_2, \sigma_0)$	4	3	1	0	0	$\sim \sum 8$

Instead of a single spread parameter θ as in the Mallows distribution, the GMD uses $m - 1$ spread parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_{m-1})$, every θ_q affecting a particular position q of the permutation. Based on Fligner & Verducci (1988), it is possible to determine any permutation (σ, σ_0) with the $m - 1$ integers $V_1(\sigma, \sigma_0), \ldots, V_{m-1}(\sigma, \sigma_0)$ in which Kendall's-τ distance $D_\tau(\sigma, \sigma_0)$ decomposes. Under the uniform distribution, the $V_q(\sigma, \sigma_0)$ variables that define a permutation are independent (Fligner & Verducci, 1986), and as a consequence, the probability distribution of the random variables $V_q(\sigma, \sigma_0)$ under the GMD given by eq. (2) can be written as

$$P(V_q(\sigma, \sigma_0) = r_q) = \frac{exp(-\theta_q r_q)}{\psi_q(\theta_q)} \qquad r_q \in \{0, \ldots, m - q\} \tag{3}$$

The normalization constant $\psi(\boldsymbol{\theta})$ in the GMD can be simplified as the product of $m - 1$ terms

$$\psi(\boldsymbol{\theta}) = \prod_{q=1}^{m-1} \psi_q(\theta_q) = \prod_{q=1}^{n-1} \frac{1 - exp(-\theta_q(m - q + 1))}{1 - exp(-\theta_q)} \tag{4}$$

where m is the total number of items in the permutation. When the GMD considers Kendall's-τ distance, it can be expressed as a multi-stage ranking model (Fligner & Verducci, 1988). Therefore, the probability distribution of a given permutation σ is given as

$$P(\sigma) = \prod_{q=1}^{m-1} P(V_q(\sigma, \sigma_0) = r_q) \tag{5}$$

where r_q is a possible value for the q position on the V_q.

The first stage consists of computing an approximation of the central permutation σ_0. Although the behavior of the GMD depends on the spread vector $\boldsymbol{\theta}$ that determines the shape of the distribution (Ceberio et al., 2014), future research is aimed at improving the estimation of the central permutation σ_0 to enhance the performance of the algorithm. To avoid losing diversity through modification in the central permutation is the goal. In this research, when a non-dominated set of solutions is found in the Pareto-front approach mentioned above, the corresponding vectors of the non-dominated set of solutions are used as an input parameter by the Borda algorithm (1784) to get a better approximation to the central permutation σ_0. To the best of our knowledge, a modification on the parameter described has not been developed for these kinds of algorithms. The following procedure details an example in order to estimate the central permutation σ_0 with the Borda algorithm using the corresponding vectors of the non-dominated set of solutions.

Pseudocode 1. Central permutation computation $\boldsymbol{\sigma_0}$

```
m ← number of vertices
M ← Individuals
from the non − dominated set of permutations $\{\sigma_1, \dots, \sigma_M\}$
for j := 1 to m
        Compute $\pi(j) = \sum_{i=1}^{M} \sigma_i(j)/M$
Endfor

visited = {∅}, auxiliary array
for j := 1 to m
        Find the lowest value from $\pi(i)$
                index ← $\arg\min_i\{\pi(i) | i \notin \text{visited}\}$
        Building $\sigma_0$ ...
                $\sigma_0(\text{index}) = j$
        visited = visited ∪ index
Endfor
```

Once the central permutation σ_0 is approximated using the Pareto-front approach, the second stage consists of computing the dispersion parameters θ_j by solving

$$\overline{V_j} = \frac{1}{exp(\theta_q) - 1} - \frac{m - q + 1}{exp\left(\theta_q(m - q + 1)\right) - 1}, q = 1{:}m - 1 \tag{6}$$

where $\overline{V_q} = \frac{1}{N} \sum_{i=1}^{N} V_q(\sigma_i \sigma_0)$

With the spread parameter vector generated $\boldsymbol{\theta}$ and the central permutation estimated σ_0, it is possible to create the probability matrix called P to obtain new $\boldsymbol{V}(\sigma, \sigma_0)$ vectors by eq. (3). The element $p_{jq}(l)$ of the probability matrix P represents the probability that vertex j requires r number of adjacent transpositions required to match the vertex j in the same qth position as the central permutation σ_0 at generation l.

Fig. 9 depicts a visual example, for only two positions on the sequence, in the probability matrix P. Based on the previous example above, the r_q possible values, for a new $\boldsymbol{V}(\sigma, \sigma_0)$ vector, will be between 0 and $m - q$, where q represents the positions on the sequence.

For each qth position, the probability of each r_q value, i.e., p_{jq}, refers to the importance of an operation for a specific position of the permutation-based representation. For all j $(j = 1, 2, \dots, m)$ and q $(q = 1, 2, \dots, m - 1)$.

Fig. 9. A probability matrix example.

In this way, this research characterizes the space of possible solutions explicitly through the estimation of the probability that a vertex, for a specific position of the permutation-based representation, is selected.

The process of generating the new valid permutations is decoding the $\boldsymbol{V}(\sigma, \sigma_0)$ vectors. This research uses the Meilă et al. (2007) algorithm for transforming the vectors mentioned above in valid permutations. The following procedure represents an example to get a valid permutation.

Pseudocode 2. Meilă et al. (2007) procedure

```
m ← number of vertices
Let a sample vector $\mathbf{V}(\sigma, \sigma_0) = 2\,,0\,,1$
therefore $m := 4$
        insert m in 0 → $(\sigma, \sigma_0)^{-1} = 4$
        insert $j := 3$ in $V_3 := 1$ → $(\sigma, \sigma_0)^{-1} = 4\,,3$
        insert $j := 2$ in $V_2 := 0$ → $(\sigma, \sigma_0)^{-1} = 2\,,4\,,3$
        insert $j := 1$ in $V_1 := 2$ → $(\sigma, \sigma_0)^{-1} = 2\,,4\,,1\,,3$

Every permutation σ is obtained by inverting a composing
with the central permutation $\sigma_0$, $((\sigma, \sigma_0)^{-1})^{-1}\sigma_0 = \sigma\sigma_0{}^{-1}\sigma_0 = \sigma$
```

Finally, Fig. 10 details the overall process for GMD process mentioned above.

Fig 10. GMD process for the VRPTW problem.

3.3. Split criteria

Once the sequence of vertices (permutation) S of n vertices without route delimiters is obtained by the GMD, a procedure should be done in order to get the feasible routes for all vehicles required. The best partition of the sequence of vertices mentioned can be computed using an optimal splitting procedure. The procedure of Labadi et al. (2008) is used in this research. As the authors explain, the procedure computes the shortest path in an auxiliary graph H containing one dummy node 0 and n other nodes corresponding to the n vertices. Each subsequence of vertices $(S_i, S_{i+1}, \ldots, S_j)$ corresponding to a feasible trip is modelled by a weighted arc $(i-1,\ j)$ in H. Note that any trip that violates time windows or vehicle capacity is discarded at this level. The shortest path from node 0 to node n in H can be computed using Bellman's algorithm for directed acyclic graphs. It indicates where to split S to get an optimal solution, subject to the order imposed by S. Fig. 1 gives an example of splitting for the instance defined by Table 1.

Finally, the HMEDA procedure to solve the VRPTW is illustrated below.

Pseudocode 3. HMEDA framework for the VRPTW

```
$D_0$ ← Generate M individuals
$t := 1$
Do
        $D_{t-1}$ ← Evaluate individuals
        (number of vehicles and total distance traveled by the vehicles used)
        $P_{t-1}$ ← Compute Pareto front from $D_{t-1}$
        $Ds_{t-1}$ ← Select N individuals from $Ds_{t-1}$
        Compute a matrix, called P matrix, from $Ds_{t-1}$
                Estimate central permutation $\sigma_0$ from $P_{t-1}$
                Estimate spread parameters $\boldsymbol{\theta}$ from $Ds_{t-1}$ and $\sigma_0$
        $Ds_t$ ← Sample M individuals from P matrix
        $D_t$ ← Find best individuals from $Ds_{t-1} \cup Ds_t$
        $t := t + 1$
Until (stopping criterion is met)
```

4. RESULTS AND COMPARISON

4.1. Comparison with a standard benchmarking dataset

In order to validate the scientific relevance of this paper, we compare the HMEDA results with others in a general and standard benchmarking dataset, i.e., Homberger and Gehring (2005) instances. These instances can be found at https://www.sintef.no/projectweb/top/vrptw/homberger-benchmark/200-customers/

The instances consider different factors such as geographical data, the number of customers serviced by a vehicle; the 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. Therefore, there are six sets of problems in Homberger and Gehring (2005) instances. The travel times equal the corresponding distances.

The instances mentioned above are used as input data for the comparison. The experiments are executed in a Lanix® Titan HX 4200 computer, Intel® Core™ i7 processor, 3.4 GHz, 8 GB of RAM, Windows® 10 for 64 bits to run every instance. C++ language is used for the implementation for all the comparisons. To account for the stochastic nature of the HMEDA, we run 30 trials for all the dataset. Each trial contains 50 generations, 1000 solution vectors belonging to each generation. We measure the relative percentage increase (RPI) as:

$$\mathrm{RPI}\,(\mathrm{c_i}) = (\mathrm{c_i} - \mathrm{c}^*)/\mathrm{c}^* \qquad (7)$$

where c_i is the number of vehicles obtained in the ith replication, and c^* is the best number of vehicles found and reported in the literature. The distribution of the experimental results in every interval is presented in Table 4. It is clear from the table that the results of the HMEDA algorithm are comparatively concentrated, which is mainly in the range of [0, 0.03], over the best solution found, i.e., number of vehicles, for all the instances. Based on the results, the HMEDA is a suitable approach for solving real-world routing problems.

Table 4. Comparison of results with Homberger and Gehring (2005) instances.

4.2. Comparison with other estimation of distribution algorithms

The MIMIC algorithm, the COMIT algorithm, and the BOA algorithm are proposed as a benchmark for comparison with the HMEDA scheme. These EDAs try to capture the problem structure with more precision and attempt to integrate higher-order interactions to enhance the solution quality. The abovementioned instances are used as input data for the algorithms. The experiments are executed in the same computer and language specification.

Based on the results detailed below, the modification of the GMD using the Pareto approach is able to detect competitive sequences for the VRPTW. Fig. 11 details the performance of the algorithms, based on the RPI, when we analyze the number of vehicles. The HMEDA scheme outperforms all the algorithms used in the comparison.

Fig. 11. Performance of the algorithms for the number of vehicles.

A Dunnett test was executed to show the performance of the HMEDA scheme. Dunnett's test is a multiple comparison procedure after a significant ANOVA result. It compares means from several experimental groups (algorithms) against a specific control group mean (the HMEDA scheme) to examine if there are significant statistical differences (Montgomery, 1997). Fig. 12 states that there is a statistically significant difference between the algorithms by means of a Dunnett test. The HMEDA scheme outperforms all the algorithms used in the comparative.

Fig. 12. Dunnett multiple comparison test.

4.3. Comparison with multi-objective algorithms

Furthermore, in order to enhance the novelty of the HMEDA, we consider evaluating the HMEDA with the multi-objective algorithms proposed by Srinivas and Deb (1994) called NSGA, and by Deb et al. (2002) called NSGA-II as a benchmark for comparison with the HMEDA scheme. The code of the proposed algorithms are obtained from the

Kanpur Genetic Algorithms Laboratory web. Again, the aforementioned instances are used as input data for the algorithms. The experiments are executed in the same computer and language specification.

Two conflicting objectives are analyzed: the number of vehicles and the total distance traveled by the vehicles used. The aim is to prevent a solution from assigning too much vertices on a single vehicle and to keep the balance of traveled distance distribution over the vehicles. The objectives mentioned above are used as input data to find the response variable for the experiment, i.e., we compute the area obtained from the best Pareto-front after running every algorithm.

We measure the RPI using the same eq. 7, i.e., c_i is the area obtained from the best Pareto-front obtained in the ith replication by a given algorithm configuration, and c^* is the best area found by any of the algorithm configurations. Note that for this problem, there are no known effective or exact techniques, and comparison with an optimal solution is not possible. Table 5 depicts the mean and standard deviation metrics for all the algorithms used for the comparison of over 30 independent runs per instance.

Fig. 13 includes three box plots: the NSGA, the NSGA-II, and the HMEDA performance after running all the instances. As we can see, the HMEDA is competitive in order to identify the best-performing ones. Although the GMD process is enough to find suitable solutions, the Pareto-approach helps to improve the results when the non-dominated solutions are considered in the GMD process. The HMEDA results report how the performance of the algorithm is improved.

Table 5. Comparison of the results.

Fig. 13. Comparison of results using boxplots.

The distribution of the experimental results in each interval is presented in Table 6. It is clear from the table that the results of the HMEDA algorithm are comparatively concentrated, which is mainly in the range of [0, 0.03), similar to the results of the other algorithms. In addition, we analyze whether there is a statistically significant difference between the averages of the algorithms. Fig. 14 depicts the statistical test, i.e., the Dunnett test. As we can see, the proposed HMEDA is competitive. There is a difference between the HMEDA with the other algorithms for the VRPTW. We confirm that the hybridization between the GMD process and the Pareto-front approach is suitable for the EDA scheme. Thus, the HMEDA algorithm in terms of solving the VRPTW is more robust than the other algorithms.

Table 6. Distribution of the results.

Fig. 14. Statistical test.

4.4. Comparison with recent algorithms for the VRPTW

Based on the previous results in the last comparison mentioned above, recent algorithms are proposed as a benchmark for comparison with the HMEDA scheme, i.e., the Figliozzi (2010) algorithm, the Tas et al. (2013) algorithm, and the Vidal et al. (2013) algorithm. Again, the instances mentioned in the previous section are used as input data for the algorithms. The experiments are executed in the same computer and language specification. Fig. 15 shows the performance of these algorithms based on the aforementioned RPI. As we can see, the performance of the proposed algorithm, i.e., the HMEDA outperforms all the algorithms used in the comparative.

Fig. 15. Comparative results with recent algorithms.

Fig. 16. Dunnett statistical test.

5. SIMULATION AND PRELIMINARY IMPLEMENTATION

There is a bread company from México that sells to different cities through the Guadalajara distribution center in Jalisco Province (Guadalajara is the capital of Jalisco Province), and their locations are distributed as shown in Fig. 17. The distances between customers are obtained directly on the road as detailed in Table 7. Data for the bread routing problem is depicted in Table 8.

Figure 17. Geographical distribution for distributed cities and distribution center in Jalisco Province.

Table 7. Distance (km) between different cities from the bread routing problem.

Table 8. Data for the bread routing problem.

A comparison is made between the recent algorithms and the proposed HMEDA over 30 days. The comparison is shown in Table 9, which lists the p-value of the Dunnett test. Fig. 18 details that the proposed algorithm is competitive in real-world routing problems.

Table 9. Multiple comparisons of means: Dunnett test for the bread routing problem.

Fig. 18. Dunnett multiple comparison test for the bread routing problem.

Furthermore, the bread company is considering extending its scope to other cities. The new locations are shown in Fig. 19. More than one hundred cities are located in this scenario. A new comparison, as a simulation with diverse demands, is made between the aforementioned recent algorithms and the proposed HMEDA. The comparison is shown in Table 10, which indicates the p-value of the Dunnett test. Finally, Fig. 20 shows that the proposed algorithm is competitive in real-world routing problems. There is a statistically significant difference between all the algorithms used in the comparison and the HMEDA scheme. The HMEDA outperforms all the algorithms used in the comparison.

Fig. 19. Geographical distribution for the new locations of the bread company.

Table 10. Multiple comparisons of means: Dunnett test for the bread company considering extending its scope to other cities.

Fig. 20. Dunnett multiple comparison test for the bread company considering extending its scope to other cities.

Finally, the time efficiency of the algorithms is detailed below. As we can see, the HMEDA outperforms all the algorithms used in the comparison.

Fig. 21. Time efficiency.

6. CONCLUSIONS AND FUTURE RESEARCH

This paper discusses the VRPTW, which considers 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. To solve this problem, we propose the HMEDA application scheme. By means of numerical experiments, this approach generates competitive solutions.

This novel research concludes that the GMD can be coupled with the EDA scheme in order to solve combinatorial optimization problems, such as the VRPTW from a multi-objective perspective. By using the Pareto-front approach, it is possible to obtain better solutions for the VRPTW. The computational results establish that the different probability models used for the VRPTW with large data sets are suitable. The Pareto-front approach is not only used for obtaining the best solutions for the HMEDA scheme, it is also used to propose a different way of improving the estimation of the central permutation σ_0 in the GMD process. The HMEDA provides an effective estimate of the vertices at the positions in the sequence vector for the VRPTW.

Moreover, the HMEDA scheme proposed guarantees that the characteristics will persist for all members of the population in the evolutionary progress of the probability model proposed, i.e., the GMD.

Furthermore, the HMEDA scheme is a new option for the aforementioned approach, called route-first cluster-second.

Explicit probability distributions in the domain of permutations for the VRPTW should be considered in future research. Estimating relationships for other related dynamic routing issues, e.g., vehicle maintenance, new vertices, and online environments, should be reconsidered in light of these results. Since the HMEDA presents stability, it appears very suitable for implementation in software systems for practical purposes. Further research directions may deal with an extension of the HMEDA for building effective modules for specific users in the industry. Finally, the HMEDA might be used in other types of VRP.

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 research to be carried out in the future. Furthermore, our research should be expanded to account for other delay factors, such as traffic, no-shows in drivers, and other changes that can be addressed by developing a dynamic version of the EDA.

ACKNOWLEDGEMENTS

We would like to express our gratitude to all the reviewers for their comments in improving the manuscript.

REFERENCES

Afifi, S., Dang, D.C., & Moukrim, A. (2016). Heuristic solutions for the vehicle routing problem with time windows and synchronized visits. Optimization Letters, 10(3), 511–525.

Ali, A., & Meilă, M. (2012). Experiments with Kemeny ranking: What works when?. Mathematical Social Sciences, 64(1), 28–40.

Agarwal, Y., Mathur, K., & Salkin, H. M. (1989). A set- partitioning- based exact algorithm for the vehicle routing problem. Networks, 19(7), 731–749.

Baluja, S., & Davies, S. (1997). Using Optimal Dependency-Trees for Combinatorial Optimization: Learning the Structure of the Search Space. In D. Fisher (Ed.), Proceedings of the Fourteenth International Conference on Machine Learning, (pp. 30–38).

Barbucha, D. (2014). Team of A-Teams approach for vehicle routing problem with time windows. In Nature Inspired Cooperative Strategies for Optimization (NICSO 2013) (pp. 273–286). Springer, Cham.

Bartholdi, J., Tovey, C.A., & Trick, M.A. (1989). Voting schemes for which it can be difficult to tell who won the election. Social Choice and welfare, 6(2), 157–165.

Berger, J., & Barkaoui, M. (2004). A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Computers & Operations Research, 31, 2037–2053.

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. The International Journal of Advanced Manufacturing Technology, 77(9-12), 1711–1725.

Bräysy, O., & Gendreau, M. (2005). Vehicle routing problem with time windows, Part II: Metaheuristics. Transportation Science, 39, 119–139.

Bräysy, O., Dullaert, W., & Gendreau, M. (2004). Evolutionary Algorithms for the Vehicle Routing Problem with Time Windows. Journal of Heuristics, 10, 587–611.

Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. Computers & Industrial Engineering, 99, 300–313.

Borda, J.D. (1784). Mémoire sur les élections au scrutin. Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784).

Ceberio, J., Irurozki, E., Mendiburu, A., & Lozano, J. (2014). A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem. IEEE Transaction on evolutionary computation, 18(2), 286–300.

Chand, P., & Mohanty, J.R. (2011). Multi objective genetic approach for solving vehicle routing problem with time window. In Trends in Computer Science, Engineering and Information Technology (pp. 336–343). Springer, Berlin, Heidelberg.

Christofides, N., & Beasley, J.E. (1984). The period routing problem. Networks, 14(2), 237–256.

Cohen, W.W., Schapire, R.E., & Singer, Y. (1998). Learning to order things. Journal of Artificial Intelligence Research, 10, 243–270.

Cordeau, J.F., Laporte, G. & Mercier, A. (2004). Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. Journal of Operational Research Society, 55, 542–546.

Cordeau, J.F., Laporte, G. & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society, 52, 928–936.

Créput, J.C., Koukam, A., & Hajjam, A. (2007). Self-Organizing Maps in Evolutionary Approach for the Vehicle Routing Problem with Time Windows. International Journal of Computer Science and Network Security, 7(1), 103–110.

de Armas, J., & Melián-Batista, B. (2015). Constrained dynamic vehicle routing problems with time windows. Soft Computing, 19(9), 2481–2498.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T.A.M.T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182–197.

De Bonet, J., Isbell, C., & Viola, P. (1997). MIMIC: Finding Optima by Estimation Probability Densities. In Advances in Neural Information Processing Systems (pp. 424–430).

Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. Operations research, 40(2), 342–354.

Desrochers, M., Lenstra, J., Savelsbergh, M. & Soumis, F. (1988). Vehicle Routing with Time Windows: Optimization and Approximation, In: Golden, B., and Assad, A. (eds.) Vehicle Routing: Methods and Studies, Amsterdam: Elsevier Science Publishers.

Desrosiers, J., Soumis, F., & Desrochers, M. (1984). Routing with time windows by column generation. Networks, 14(4), 545–565.

El-Sherbeny, N.A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. Journal of King Saud University-Science, 22(3), 123–131.

Fisher, M.L., Jörnsten, K.O., & Madsen, O.B. (1997). Vehicle routing with time windows: Two optimization algorithms. Operations research, 45(3), 488–492.

Figliozzi, M. A. (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. Transportation Research Part C: Emerging Technologies, 18(5), 668–679.

Fligner, M., & Verducci, J. (1988). Multistage ranking models, J. Amer. Stat. Assoc., 83(403), 892–901.

Fligner, M., & Verducci, J. (1986). Distance based ranking models. J. Royal Stat. Soc., 48(3), 359–369.

Gan, X., Kuang, J., & Niu, B. (2014, August). Particle swarm optimizations for multi-type vehicle routing problem with time windows. In International Conference on Intelligent Computing (pp. 808–815). Springer, Cham.

Gendreau, M., Ghiani, G., & Guerriero, E. (2015). Time-dependent routing problems: A review. Computers & operations research, 64, 189–197.

Golden, B. & Assad, A. (1986). Perspectives on Vehicle Routing: Exciting New Developments, Operations research. 34. 803–809.

Homberger, J., & Gehring, H. (2005). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. European Journal of Operational Research. 162(1), 220–238.

Homberger, J., & Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. INFOR, 37, 297–318.

Hwang, H. S. (2002). An improved model for vehicle routing problem with time constraint based on genetic algorithm. Computers & Industrial Engineering, 42(2-4), 361–369.

Kacem, I., Hammadi, S., & Borne, P. (2002). Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. Mathematics and computers in simulation, 60(3), 245–276.

Kamkar, I., Poostchi, M., & Totonchi, M. R. A. (2010). A cellular genetic algorithm for solving the vehicle routing problem with time windows. In Soft Computing in Industrial Applications (pp. 263–270). Springer, Berlin, Heidelberg.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. European Journal of Operational Research, 249(1), 1–21.

Kohl, N., & Madsen, O.B. (1997). An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation. Operations research, 45(3), 395–406.

Kolen, A., Rinnooy-Kan, A., & Trienkens, H. (1987). Vehicle routing problem with time windows. Operation Research, 35, 266–273.

Kumar, V.S., Thansekhar, M.R., Saravanan, R., & Amali, S.M.J. (2014). Solving multi-objective vehicle routing problem with time windows by FAGA. Procedia Engineering, 97, 2176–2185.

Labadi, N., Prins, C., & Mohamed, R. (2008). A memetic algorithm for the vehicle routing problem with time windows. RAIRO-Operations Research, 42, 415–431.

Larrañaga, P., & Lozano, J. A. (Eds.). (2001). Estimation of distribution algorithms: A new tool for evolutionary computation (Vol. 2). Springer Science & Business Media.

Larranaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., & Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: a review of representations and operators. Artificial Intelligence Review, 13, 129–170.

Li, J., Li, Y., & Pardalos, P. (2016). Multi-depot vehicle routing problem with time windows under shared depot resources. Journal of Combinatorial Optimization, 31(2), 515–532.

Mallows, C. (1957). Nonnull ranking models. Biometrika, 44(1-2), 114–130.

Meilă, M., Phadnis, K., Patterson, A., & Bilmes, J. (2007). Consensus ranking under the exponential model, Proc. 22nd Conf. Uncertainty Artif. Intell., (Vancouver, 2007), pp. 285–294.

Mester, D., Bräysy, O., & Dullaert, W. (2007). A multi-parametric evolution strategies algorithm for vehicle routing problems. Expert Systems with Applications, 32, 508–517.

Montgomery, D.C. (1997). Design and Analysis of Experiments. John Wi1ey & Sons.

Montoya-Torres, J., López-Franco, J., Nieto-Isaza, S., Felizzola-Jiménez, H., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. Computers & Industrial Engineering, 79, 115–129.

Mühlenbein, H., & Paaß, G. (1996). From recombination of genes to the estimation of distributions: I. binary parameters. In International conference on parallel problem solving from nature (pp. 178–187). Springer, Berlin, Heidelberg.

Nalepa, J. & Blocho, M. (2016). Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows. Soft Computing, 20(6), 2309–2327.

Ombuki, B., Ross, B.J., & Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. Applied Intelligence, 24(1), 17–30.

Pelikan, M., Goldberg, D., & Cantú-Paz, E. (1998). Linkage problem, distribution estimation, and Bayesian networks. The university of Illinois, Illinois Genetic Algorithm Laboratory, Urbana-Champaign.

Potvin, J.Y., & Bengio, S. (1996). The vehicle routing problem with time windows - Part II: Genetic search. INFORMS Journal on Computing, 8, 165–172.

Potvin, J.Y. (1993). Genetic algorithms for the traveling salesman problem. Annals of Operations Research, 63, 339–370.

Potvin, J.Y., & Rousseau, J.M. (1993). A Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows. European Journal of Operational Research, 66, 331–340.

Prins, C., Lacomme, P., & Prodhon, C. (2014). Order-first split-second methods for vehicle routing problems: A review. Transportation Research Part C, 40, 179–200.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research, 31-12, 1985–2002.

Rochat, Y., & Taillard, É. (1995). Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics, 1, 147–167.

Selma, K-H., Prins, C., Yalaoui, A., & Reghioui, M. (2013). Heuristics and memetic algorithm for the two-dimensional loading capacitated vehicle routing problem with time windows. Central European Journal of Operations Research, 21(2), 307–336.

Schwarze, S., & Voß, S. (2015). A bicriteria skill vehicle routing problem with time windows and an application to pushback operations at airports. In Logistics Management (pp. 289–300). Springer, Cham.

Solomon, M., & Desrosiers, J. (1988). Time Window-Constrained Routing and Scheduling Problems, Transportation Science, 22, 1–13.

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research, 35, 254–265.

Soonpracha, K., Mungwattana, A., & Manisri, T. (2015). A re-constructed meta-heuristic algorithm for robust fleet size and mix vehicle routing problem with time windows under uncertain demands. In Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems-Volume 2 (pp. 347–361). Springer, Cham.

Srinivas, N., & Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary computation, 2(3), 221–248.

Tan, K.C., Lee, L.H., Zhu, Q.L. & Ou, K. (2001). Heuristic methods for vehicle routing problem with time windows. Artificial Intelligence in Engineering, 15, 281–295.

Tas, D., Dellaert, N., van Woensel, T., & Kok, T. (2013). Vehicle routing problem with stochastic travel times including soft time windows and service costs. Computers & Operations Research, 40(1), 214–224.

Thangiah, S.R., Osman, I.H., Vinayagamoorthy, R. & Sun, T. (1995). Algorithms for the vehicle routing problems with time deadlines. American Journal of Mathematical and Management Sciences, 13, 323–355.

Toth, P., & Vigo, D. (2001). The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications. SIAM. Philadelphia.

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Computers & operations research, 40(1), 475-489.

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. Computers & Industrial Engineering, 62(4), 917–926.

Wang, J-q., Tong, X-n., & Li, Z-m. (2007). An improved evolutionary algorithm for dynamic vehicle routing problem with time windows. In International Conference on Computational Science (pp. 1147–1154). Springer, Berlin, Heidelberg.

Yang, C., Guo, Z.x., & Liu, L.y. (2015). Comparison study on algorithms for vehicle routing problem with time windows. In Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management 2014 (pp. 257-260). Atlantis Press, Paris.

Table 1. Data for the instance used in Fig. 1.

Vertex	**Xcoord**	**Ycoord**	**Demand**	**Ready time**	**Due date**	**Service time**	**Edge**	$\boldsymbol{d_{ij} = t_{ij}}$
0 (depot)	70	70	0	0	1351	0	[0,1]	37.85
1	33	78	20	750	809	90	[0,4]	78.23
2	59	52	20	553	602	90	[0,5]	62.94
3	10	137	30	147	219	90	[0,2]	21.10
4	4	28	10	616	661	90	[0,3]	89.94
5	25	26	20	128	179	90	[0,6]	36.67
6	86	37	10	478	531	90	[1,4]	57.80
$\boldsymbol{S = (1, 4, 5, 2, 3, 6)}$			$\boldsymbol{Q = 50}$				[4,5]	21.10
							[5,2]	42.80
							[2,3]	98.11
							[3,6]	125.60

Table 2. Data for the instance used in Fig. 2.

Vertex	**Xcoord**	**Ycoord**	**Demand**	**Ready time**	**Due date**	**Service time**	**Edge**	$\boldsymbol{d_{ij} = t_{ij}}$
0 (depot)	70	70	0	0	1351	0	[0,5]	62.93
1	33	78	20	750	809	90	[0,4]	78.23
2	59	52	20	553	602	90	[0,2]	21.10
3	10	137	30	147	219	90	[0,1]	37.85
4	4	28	10	616	661	90	[0,3]	89.94
5	25	26	20	128	179	90	[0,6]	36.67
6	86	37	10	478	531	90	[5,4]	21.09
$\boldsymbol{S = (5, 4, 2, 1, 3, 6)}$			$\boldsymbol{Q = 50}$				[4,2]	60.00
							[2,1]	36.77
							[1,3]	63.32
							[3,6]	125.60

Table 3. Data used in Fig. 6.

Vertex	**Xcoord**	**Ycoord**
0 (depot)	70	70
1	33	78
2	59	52
3	10	137
4	4	28
5	25	26
6	86	37
$\boldsymbol{S} = (\mathbf{5}, \mathbf{4}, \mathbf{2}, \mathbf{1}, \mathbf{3}, \mathbf{6})$		

Table 4. Comparison of results with Homberger and Gehring (2005) instances.

		Intervals					
Instances	**Trials**	[0, 0.031)	[0.031, 0.062)	[0.062, 0.093)	[0.093, 0.124)	[0.124, 0.155)	[0.155, 0.186)
120	3000	2095 69.83%	678 22.62%	226 7.53%	1 0.02%	-	-

Table 5. Comparison of the results.

Algorithm	**Global Performance** mean(std)
NSGA	0.777(2.056)
NSGA-II	0.744(1.841)
HMEDA for the VRPTW	0.315(0.278)

Table 6. Distribution of the results.

Algorithm	**Intervals**					
	[0 , 0.031)	[0.031 , 0.062)	[0.062 , 0.093)	[0.093 , 0.124)	[0.124 , 0.155)	≥ 0.155
NSGA	60.20%	10.23%	4.07%	3.90%	3.83%	*
NSGA-II	59.07%	11.07%	4.20%	4.10%	4.07%	*
HMEDA for the VRPTW	60.03%	10.40%	6.50%	6.43%	5.87%	*

* rest of data

Table 7. Distance (km) between different cities from the bread routing problem

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	-	62.963	72.99	49.081	98.326	168.864	49.275	156.031	187.683	269.92	112.979	129.256	96.634	228.175	91.282	90.93	53.723	113.984	219.997	92.068	99.143	117.247	145.035	163.578	371.619	192.731
1	63.654	-	59.701	106.495	122.444	226.078	18.256	142.742	244.898	256.631	99.69	153.374	153.849	249.453	29.574	115.048	105.691	138.102	241.275	122.055	123.261	141.365	169.153	187.696	392.898	179.442
2	72.718	59.408	-	83.389	92.615	237.353	50.359	84.956	256.172	198.845	41.904	123.545	139.773	283.095	70.366	85.218	120.603	108.89	274.917	79.31	93.432	112.152	139.941	157.867	426.54	121.656
3	48.363	106.493	84.303	-	109.639	187.5	92.805	167.344	206.32	281.233	124.292	140.569	68.858	271.705	134.812	102.243	113.777	88.023	263.527	103.381	110.456	91.286	119.074	174.891	415.15	204.044
4	97.15	134.437	92.006	107.821	-	261.784	106.986	175.047	280.604	298.911	131.995	77.202	139.564	307.527	145.394	38.875	145.034	36.91	299.348	80.296	42.961	40.173	41.715	111.524	450.971	193.004
5	175.823	234.147	246.586	200.943	271.922	-	220.459	329.627	44.887	443.516	286.575	302.852	146.407	328.102	262.467	264.525	191.721	240.7	319.924	265.663	272.738	243.963	271.751	263.74	433.861	366.327
6	49.338	17.683	50.761	92.179	106.617	211.762	-	133.802	230.582	247.691	90.75	137.547	139.533	235.137	46.002	99.221	91.375	122.275	226.959	100.359	107.434	125.538	153.326	171.869	378.582	170.502
7	156.168	142.858	84.692	166.839	176.065	320.803	133.809	-	339.622	143.259	46.303	161.532	223.223	366.545	153.816	168.668	204.053	192.34	358.367	162.76	176.882	195.602	223.391	190.966	509.99	66.07
8	185.854	244.178	256.617	210.974	281.952	45.003	230.49	339.658	-	453.546	296.606	312.882	230.162	298.19	272.497	274.556	210.384	297.61	290.012	275.694	282.769	300.873	328.661	347.205	403.949	376.358
9	269.919	256.608	198.442	280.59	267.344	434.553	247.559	143.154	453.373	-	160.053	196.028	336.973	480.296	267.566	234.052	317.803	305.69	472.117	227.016	216.517	308.952	288.161	235.719	623.74	78.397
10	113.078	99.768	41.602	123.749	132.975	277.712	90.719	46.264	296.532	160.153	-	163.904	180.133	323.455	110.726	125.578	160.962	149.249	315.277	119.67	133.791	152.512	180.3	198.227	466.9	82.965
11	129.323	166.61	124.179	139.994	76.243	293.957	139.159	161.29	312.777	195.962	164.168	-	196.378	339.7	177.568	42.952	177.207	114.589	331.522	108.3	52.819	117.852	117.958	36.937	483.144	120.296
12	95.145	153.469	147.201	70.141	138.417	146.438	139.781	230.242	191.79	344.131	187.19	203.466	-	311.271	181.788	165.14	143.327	122.301	303.093	166.278	173.353	125.563	153.352	145.341	454.715	266.942
13	228.211	248.815	282.394	272.833	307.729	325.158	235.127	365.434	297.919	479.323	322.383	338.659	318.405	-	277.134	300.333	209.094	323.387	46.527	301.471	308.546	326.65	354.438	372.982	143.647	402.135
14	92.289	29.79	71.091	135.13	151.079	254.713	46.891	154.132	273.532	268.02	111.08	182.009	182.483	278.088	-	143.682	134.325	166.737	269.91	133.444	151.896	170	197.788	216.331	421.533	190.832
15	90.016	127.303	84.873	100.688	36.937	254.651	99.853	167.914	273.471	265.516	124.862	43.807	157.071	300.393	138.261	-	137.901	75.283	292.215	68.437	12.956	78.546	78.651	78.129	443.838	159.609
16	53.652	98.658	121.694	104.562	147.03	191.563	84.969	204.735	227.812	318.624	161.683	177.96	145.603	209.618	126.977	139.634	-	162.688	201.439	140.772	147.847	165.951	193.739	212.282	353.062	241.435
17	114.103	139.099	109.925	87.549	37.144	240.738	123.939	192.966	289.878	337.949	149.914	116.239	123.721	324.48	163.313	77.913	161.987	-	316.301	119.334	81.999	3.328	25.335	62.385	467.924	232.042
18	220.026	240.631	274.209	264.648	299.545	316.974	226.943	357.25	289.735	471.139	314.198	330.474	310.221	46.521	268.95	292.148	200.91	315.203	-	293.286	300.361	318.465	346.254	364.797	189.965	393.95
19	92.059	122.834	80.403	102.73	80.378	256.693	101.895	163.444	275.513	227.227	120.392	107.5	159.114	302.436	133.792	68.436	139.943	118.724	294.257	-	55.093	121.987	122.092	141.823	445.88	151.562
20	99.882	137.168	94.738	110.553	42.933	264.516	109.718	177.779	283.336	216.095	134.727	52.018	166.936	310.259	148.126	12.954	147.766	81.279	302.08	55.095	-	84.542	84.648	86.341	453.703	140.43
21	117.083	142.08	112.906	90.529	40.125	243.719	126.919	195.947	292.859	340.929	152.895	119.22	126.701	327.46	166.294	80.894	164.968	3.328	319.282	122.314	84.98	-	29.839	65.713	470.905	235.023
22	144.9	169.896	140.722	118.346	41.722	271.535	154.736	223.763	320.675	287.835	180.711	118.924	154.518	355.277	194.11	80.598	192.784	25.335	347.098	122.018	84.683	29.839	-	53.571	498.721	234.726
23	163.272	200.559	158.129	173.943	110.193	263.436	173.108	190.884	346.726	235.31	198.117	37.037	146.418	373.649	211.517	76.901	211.157	62.334	365.471	142.249	86.768	65.662	53.594	-	517.094	149.891
24	371.674	392.278	425.856	416.296	451.192	432.068	378.59	508.897	404.829	622.786	465.845	482.122	461.868	143.665	420.597	443.795	352.557	466.85	189.989	444.934	452.009	470.112	497.901	516.444	-	545.597
25	192.427	179.117	120.951	203.098	191.166	357.062	170.068	65.662	375.881	78.668	82.562	119.85	259.482	402.804	190.075	157.875	240.312	229.512	394.626	150.839	140.34	232.775	232.88	149.285	546.249	-

Table 8. Data for the bread routing problem

Vertex	Demand (Percentaje of the capacity)	Ready time	Due date	Service time
0 (depot)	0	0	3598	0
1	20	2448	3328	90
2	20	308	1188	90
3	10	661	1541	90
4	10	2457	3337	90
5	30	1562	2442	90
6	30	57	937	90
7	10	2237	3117	90
8	20	546	1426	90
9	20	1457	2337	90
10	20	63	943	90
11	10	2550	3430	90
12	20	849	1729	90
13	20	757	1637	90
14	10	2584	3464	90
15	10	557	1437	90
16	10	1043	1923	90
17	10	2140	3020	90
18	20	475	1355	90
19	20	1554	2434	90
20	40	1277	2157	90
21	20	2435	3315	90
22	30	941	1821	90
23	30	2602	3482	90
24	10	1560	2440	90
25	10	985	1865	90

Table 9. Multiple comparisons of means: Dunnett test for the bread routing problem

Hypotheses	Estimate	Std. Error	t value	Pr (> | t |)
Total duration				
Figliozzi - HMEDA for the VRPTW == 0	0.81556	0.01698	48.02	<2e-16
Tas et al - HMEDA for the VRPTW == 0	0.82333	0.01698	48.48	<2e-16
Vidal et al - HMEDA for the VRPTW == 0	0.82222	0.01698	48.41	<2e-16

Table 10. Multiple comparisons of means: Dunnett test for the new locations of the bread routing problem

Hypotheses	Estimate	Std. Error	t value	Pr (> | t |)
Total duration				
Figliozzi - HMEDA for the VRPTW == 0	0.069222	0.005904	11.72	<2e-16
Tas et al - HMEDA for the VRPTW == 0	0.079557	0.005904	13.48	<2e-16
Vidal et al - HMEDA for the VRPTW == 0	0.071217	0.005904	12.06	<2e-16

Highlights

- A new way of estimating an probability distribution for routing problems
- The Mallows distribution solves vehicle routing problems efficiently
- A Pareto-front approach helps to find better solutions for routing problems
- An effective estimate of the vertices at the positions in the routing sequence
- A new option for routing problems, called route-first cluster-second