

# On Optimizing a Demand Responsive Transport with an Evolutionary Multi-Objective Approach

Rémy Chevrier, Arnaud Liefoghe, Laetitia Jourdan, Clarisse Dhaenens

► **To cite this version:**

Rémy Chevrier, Arnaud Liefoghe, Laetitia Jourdan, Clarisse Dhaenens. On Optimizing a Demand Responsive Transport with an Evolutionary Multi-Objective Approach. IEEE Intelligent Transportation Systems Conference, 2010, Madeira, Portugal. pp.575–580. hal-00469418v2

**HAL Id: hal-00469418**

**<https://hal.archives-ouvertes.fr/hal-00469418v2>**

Submitted on 12 Oct 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# On Optimizing a Demand Responsive Transport with an Evolutionary Multi-Objective Approach

Rémy Chevrier

Université Lille Nord de France – INRETS ESTAS  
20 rue Élisée Reclus – 59650 Villeneuve d’Ascq, France  
Email: remy.chevrier@inrets.fr

Arnaud Liefoghe, Laetitia Jourdan, Clarisse Dhaenens

Université Lille Nord de France – INRIA Lille Nord Europe  
40 avenue Halley – 59650 Villeneuve d’Ascq France  
Email{arnaud.liefoghe,laetitia.jourdan,clarisse.dhaenens}@lifl.fr

**Abstract**—This paper deals with a dial-a-ride problem with time windows applied to a demand responsive transport service. An evolutionary approach as well as new original representation and variation operators are proposed and detailed. Such mechanisms are used with three state-of-the-art multi-objective evolutionary algorithms: NSGA-II, IBEA and SPEA2. After introducing the general problem, the solution encoding and the algorithm mechanisms are depicted. The approach is assessed by applying the algorithms to both random and realistic dial-a-ride instances. Then a statistical comparison is provided in order to highlight the most suited evolutionary algorithms to optimize real-life transportation problems.

**Index Terms**—Dial-a-ride, Demand responsive transport, Multi-objective, Evolutionary algorithm, Optimization

## I. INTRODUCTION

The demand responsive transport (DRT) is a transport service which involves the satisfaction of a set of travel requests. A request corresponds to a customer to be carried from a pick-up location to a delivery at a predetermined time. In its usual form, DRT is related to the dial-a-ride problem (DARP) or to the vehicle routing problem (VRP) which both consist in optimizing the vehicles routes by reducing the number of vehicles and the travel durations [1], [2]. Nevertheless, the main differences between the DARP and the VRP consist of the precedence constraints imposed by the customer journeys and their tolerance to the delays (quality of service [3]). A DRT service is a specific case of the DARP, the latter being the academical formulation of routing with passengers.

The DARP involves antagonist objectives which have to be optimized concurrently. That is why the proposed approach has to cope with a multi-objective problem in which the criteria under study are not hierarchized. Furthermore, since the DARP is *NP*-hard [1], we propose to use an Evolutionary Algorithm (EA) which is known to efficiently cope with scalable problems. Besides, an EA can produce multiple distinct solutions in a single run which could help the decision making. Given that it is almost unlikely to have a single optimal solution in a multi-criteria problem [4], the multi-objective algorithm has to deal with a set of incomparable solutions. In this case, a Pareto approach is preferred for assessing incomparable solutions. Moreover using an EA paired to a Pareto approach has already been shown to efficiently solve multi-objective real-life problems [5], [4].

Although a lot of metaheuristic methods exist for solving the DARP [6], [1], EA are generally used as single-criterion optimization methods aggregating multiple objectives in order to make the computation easier so that a single solution has to be found [7], [8], [9]. As far as we know, only Chevrier [10] used a Pareto-based three-objectives model to cope with the DARP with time windows (DARPTW).

In this paper, a new encoding mechanism based on a two-dimensional representation as well as original operators are proposed for the DARPTW. They are integrated in three state-of-the-art algorithms: the Non dominated Sorting Genetic Algorithm II (NSGA-II) [11], Strength Pareto EA 2 (SPEA-2) [12] and Indicator Based EA (IBEA) [13]. Then they are applied to benchmark instances and pairwise compared.

The paper is organized as follows. The problem formulation is described in Section II. Section III introduces the evolutionary multi-objective principles. Then the proposed encoding to represent the DARP, as well as the variation operators, are presented in Section IV. Experimental results for Chevrier’s instances [10] are provided and discussed in Section V. Finally, Section VI concludes the paper.

## II. PROBLEM OVERVIEW

### A. Formulation

The multi-objective problem (MOP) under study needs to be formulated as a set of three objective functions to be achieved ( $f = (f_1, f_2, f_3)$ ) and a set of constraints to be taken into account. The problem solving is based on specific parameters such as a relaxation and time windows which introduce more tolerance to slight delays and hence flexibility. For helping the reader, the symbols used in the paper are summarized in Table I. Since the DRT problem contains a DARPTW, the reader can refer to [1] to have a mathematical model. Due to the lack of space in the paper, we only detail the specificities of the DRT problem and its multi-objective formulation.

$$f = (f_1, f_2, f_3) \quad (1)$$

$$f_1 = \min |\Lambda| \quad (2)$$

$$f_2 = \min \sum_{v \in \Lambda} t_v \quad (3)$$

$$f_3 = \min \sum_{v \in \Lambda} d_v \quad (4)$$

TABLE I  
DEFINITION OF SYMBOLS USED FOR THE DARPTW

Input data	
$f_i$	Objective function $i$
$\Lambda$	Set of the used vehicles
$V$	Set of the pick-up ( $V^+$ ) and delivery ( $V^-$ ) points such that $V = V^+ \cup V^-$
$x, y$	Arbitrary points such that $\{x, y\} \in V$
$t_{x \rightarrow y}$	Travel duration from $x$ to $y$
$d_x$	Delay at a point $x$
$R$	Set of the requests
$r$	A request such that $r \in R$
$r^+$ (resp. $r^-$ )	Pick-up (resp. delivery) point of the request $r$ such that $r^+ \in V^+, r^- \in V^-$
$r^*$	One point of request $r$ : $r^* = \{r^+   r^-\}$
$w_r$	Number of people of request $r$ to be carried
$v$	A vehicle. At a point $x$ , it is denoted: $v_x$
$Q_v$	Capacity of vehicle $v$ , $v \in \Lambda$
$h_{r^+}$	Desired pick-up time
$h_{r^-}$	Theoretical arrival time
$k_r$	Relaxation coefficient
$k_w$	Coefficient for the time windows
Variables	
$t_v$	Amount of each travel duration between each point visited by vehicle $v$
$d_v$	Sum of each delay of vehicle $v$ at delivery points
$p_v$	Number of passengers in vehicle $v$
$H_x$	Effective arrival time at a point $x$
$tw_x$	Time window at a point $x$

## B. Objectives

In the DRT service under study, we aim at minimizing three objectives. The first ( $f_1$ ) is *economic* and consists in minimizing the number of required vehicles (eq. (2)). The second corresponds to the duration reduction of the vehicle journeys ( $f_2$ , eq. (3)) which could correspond to an *environmental* objective in so far as we look to limiting the emission of pollutants, and also in expectation of a hypothetical carbon tax. The last objective minimizes the likely delays (*quality of service*:  $f_3$ , eq. (4)).

## C. Introduction of delay tolerance and time windows

A usual DRT service uses the tolerance of the customers to accept more or less delays. Making detours allows a vehicle to group the customers more easily while producing a few delays. To model these delays, we introduce a coefficient of relaxation  $k_r$  applied to the travel duration to define a maximal delivery time. Let  $t'_{r^+ \rightarrow r^-}$  be the slackened travel duration when  $k_r > 1$ :  $t'_{r^+ \rightarrow r^-} = k_r \cdot t_{r^+ \rightarrow r^-}$ . Consequently, the maximal delivery time  $h'_{r^-}$  is defined as follows:  $h'_{r^-} = h_{r^+} + t'_{r^+ \rightarrow r^-}$ .

In the case of VRP, time windows are generally used as time slots to make delivery easier. In the case of DARP, a time window facilitates the picking-up when a travel duration slightly exceeds the picking-up time. With such a mechanism, it is easier to bring together customers into a same vehicle. So, a time window at a point  $r^+$  is denoted  $tw_{r^+}$  and is proportional to the theoretical travel duration to the point  $r^-$ :  $tw_{r^+} = k_w \cdot t_{r^+ \rightarrow r^-}$ , where  $k_w$  is a coefficient indicating the percentage of the duration allocated to the time window.

## D. Constraints

The flexibility is introduced by using relaxation and time windows while authorizing delays on the travels. Nevertheless, for limiting delays and for keeping a good quality of service, the likely delays must be limited by adding a set of constraints. Firstly, it is necessary to define what a feasible travel is. We denote  $x, y$  two arbitrary points to be potentially connected and  $H_x$  the effective starting time at Point  $x$ . Indeed, there may be some accumulated delays that  $H_x$  takes into account. The travel from  $x$  to  $y$  is feasible *iff*:

$$\begin{aligned} H_x + t_{x \rightarrow y} &\leq h_y + tw_y && \text{if } y \in V^+ \\ H_x + t_{x \rightarrow y} &\leq h'_y && \text{if } y \in V^- \end{aligned}$$

Note that  $H$  is evaluated after each travel between two points and depends on the travel durations and whether the vehicle arrives before or after a theoretical starting time. Therefore, two cases may arise for determining  $H_y$  which corresponds to the effective time at Point  $y$  coming from Point  $x$ :

$$H_y = \begin{cases} h_y & \text{if } H_x + t_{x \rightarrow y} < h_y \\ H_x + t_{x \rightarrow y} & \text{otherwise.} \end{cases}$$

After calculating the effective visiting time, it is possible to evaluate the delay at a delivery point. A delay  $d$  corresponds to the difference between  $H$  and the theoretical arrival time, thus in a delivery point  $y$ , the delay  $d_y$  is defined as follows

$$d_y = \begin{cases} H_y - h_y & \text{if } H_y > h_y \\ 0 & \text{otherwise.} \end{cases}$$

The number of passengers in a vehicle is indicated by  $p_v$  (initially,  $p_v = 0$ ).  $p_v$  is updated whenever customers are delivered or picked-up. In this latter case, the capacity constraint must be checked if customers ( $w_r$ ) are picked-up at a point  $r^+$ :  $p_v + w_r \leq Q_v$

## III. EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION

### A. Multi-objective Optimization

A general MOP can be defined by a set of  $n$  objective functions ( $f_1, f_2, \dots, f_n$ ), a set  $X$  of feasible solutions in the decision space, and a set  $Z$  of feasible points in the objective space. Without loss of generality, we here assume that each objective function is to be minimized. To each solution  $x \in X$  is assigned an objective vector  $z \in Z$  on the basis of the vector function  $f : X \rightarrow Z$  with  $z = f(x) = (f_1(x), f_2(x), \dots, f_n(x))$  as illustrated by Figure 1. An objective vector  $z \in Z$  is said to *dominate*<sup>1</sup> another objective vector  $z' \in Z$  *iff*  $\forall i \in \{1, 2, \dots, n\}, z_i \leq z'_i$  and  $\exists j \in \{1, 2, \dots, n\}$  such as  $z_j < z'_j$ . An objective vector  $z \in Z$  is said to be *non-dominated* *iff* there does not exist another objective vector  $z' \in Z$  such that  $z'$  dominates  $z$ . A solution  $x \in X$  is said to be *efficient* if its mapping in the objective space results in a non-dominated point. The set of all efficient solutions is the *efficient set*, denoted by  $X_E$ . The set of all non-dominated vectors is the *Pareto front*, denoted

<sup>1</sup>We will also say that a decision vector  $x \in X$  *dominates* a decision vector  $x' \in X$  if  $f(x)$  dominates  $f(x')$ .

by  $Z_N$ . A possible approach in MOP solving is to find the minimal set of efficient solutions, *i.e.* one solution  $x \in X_E$  for each non-dominated vector  $z \in Z_N$  such as  $f(x) = z$ . However, generating the entire efficient set is usually infeasible due to the complexity of the underlying problem. Therefore, the overall goal is often to identify a good approximation of it. EAs are commonly used to this end as they are able to find multiple and well-spread non-dominated solutions in a single simulation run [5].

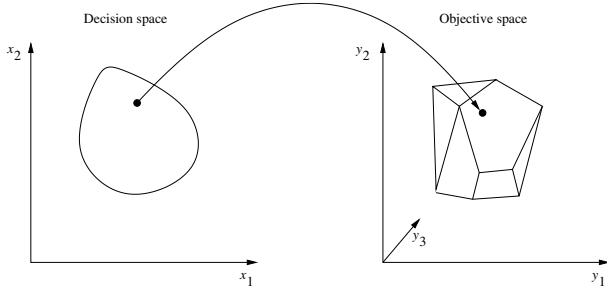


Fig. 1. Representation of a solution  $(x_1, x_2)$  in the decision space and the corresponding values in the objective space:  $(y_1, y_2, y_3) = f(x_1, x_2)$ .

## B. Evolutionary Multi-objective Algorithms

1) *NSGA-II*: [11] is probably the most widely used multi-objective resolution method. At each NSGA-II generation, solutions from the current population are ranked into several classes. Individuals mapping to vectors from the first front all belong to the best efficient set; individuals mapping to vectors from the second front all belong to the second best efficient set; and so on. Two values are then assigned to the population members. The first one corresponds to the *rank* the corresponding solution belongs to, and represents the quality of the solution in terms of convergence. The second one, the *crowding distance*, consists in estimating the density of solutions surrounding a particular point of the objective space, and represents the quality of the solution in terms of diversity. A solution is said to be better than another one if it has a best rank value, or in case of equality, if it has the best crowding distance. The selection strategy is a deterministic tournament between two random solutions. At the replacement step, only the best individuals survive, with respect to a predefined population size. Furthermore, note that, in addition to the original NSGA-II, we add an external population, the so-called *archive*, in order to store the whole set of potentially efficient solutions found during the search.

2) *IBEA*: [13] is a more recent method that is a good illustration of the new trend dealing with indicator-based search, and started to become popular in recent years. The main idea of IBEA is to introduce a total order between solutions by means of a binary quality indicator. Its fitness assignment scheme is based on a pairwise comparison of solutions from the current population with regards to an arbitrary indicator  $I$ . To each individual  $x$  is assigned a fitness value  $F(x)$  measuring the 'loss in quality' if  $x$  was removed from the current

population  $P$ , *i.e.*  $F(x) = \sum_{x' \in P \setminus \{x\}} (-e^{-I(x',x)/\kappa})$ , where  $\kappa > 0$  is a user-defined scaling factor. Different indicators can be used for such a purpose, and we choose to use the binary additive  $\epsilon$ -indicator ( $I_{\epsilon+}$ ) as defined in [13].  $I_{\epsilon+}(x, x')$  gives the minimum value by which a solution  $x \in X$  has to or can be translated in the objective space to weakly dominate another solution  $x' \in X$ . Selection for reproduction consists of a binary tournament between randomly chosen individuals. Selection for replacement consists in iteratively removing the worst solution from the current population until the required population size is reached; fitness information of the remaining individuals is updated each time there is a deletion. Similarly, we added an archive to store non-dominated solutions in order to prevent their loss during the stochastic search process.

3) *SPEA2*: [12] is an extension of SPEA, where an improved fitness assignment strategy is proposed. It intrinsically handles an internal archive of fixed size that is used during the selection step to create offspring solutions. At a given iteration of SPEA2, to each population and archive member  $x$  is assigned a strength value  $S(x)$  representing the number of solutions it dominates. Then, the fitness value  $F(x)$  of solution  $x$  is calculated by summing the strength values of all individuals solution  $x$  currently dominates. Additionally, a diversity preservation strategy, based on a nearest neighbor technique, is incorporated. The selection step consists of a binary tournament with replacement applied on the internal archive only. At last, given that the SPEA2 archive has a fixed size storage capacity, a bounding mechanism, based on fitness and diversity information, is used when the size of the non-dominated set is too high. On the contrary, when the size of the non-dominated set is too small, some dominated solutions are allowed to be incorporated. As well, we added an external archive to store the whole set of non-dominated solutions found during the search.

## IV. APPLICATION TO THE DARPTW

### A. Solution encoding and initialization

1) *Representation*: The main idea of the representation is to simplify the reading of the vehicle routes. So, we use a two-dimensional representation (not a matrix but a vector of vectors) in which a gene corresponds to a single vehicle route (one vector). The sequence of data literally indicates the order of the points traversed by a vehicle. In order to ensure the points precedence, a cell indicates a request identifier and not the point itself in such a way that we can retrieve the associated point by counting the number of times the corresponding request identifier is encountered. If a request  $r$  appears for the first time, it is necessarily the pick-up point  $r^+$ , otherwise it is the delivery point  $r^-$ . Given that there are two points per request, a gene has necessarily an even number of cells.

The examples of Figures 2(b,c,d) are solutions to a DARPTW instance (Fig. 2(a)). For each solution, the associated encoding is depicted. Each gene  $i$  is a vector translating the route of vehicle  $v_i$ . The order of the cell values indicates the sequence of the points visited by a vehicle, such as in the

example 2(b), where the sequence  $(A, B, A, C, B, C)$  is the counterpart of the path  $(A^+, B^+, A^-, C^+, B^-, C^-)$ .

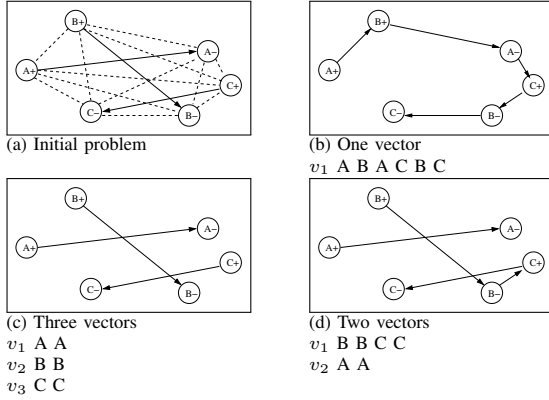


Fig. 2. Examples of solution encoding: (a) Initial problem with three customers A, B, C, (b) a solution with one vehicle (one vector), (c) a solution with three vehicles (three vectors), (d) a solution with two vehicles (two vectors).

2) *Population initialization*: The population is composed of  $N$  feasible solutions. To be sure of their feasibility we assign one request *per* vehicle, so that  $|\Lambda| = |R|$ . Hence, the DARP instance (Fig. 2(a)) requires three vehicles and the population  $P = \{S_i\}$  could be initialized as follows:

	$S_1$	$S_2$	$S_3$	$S_4$
$v_1$	A A	C C	A A	B B
$v_2$	C C	A A	B B	A A
$v_3$	B B	B B	C C	C C

### B. Evaluation

The evaluation function  $f = (f_1, f_2, f_3)$  applied to a solution  $S$  computes three values corresponding to the respective objectives:

- $f_1 = \min |\Lambda|$ ;
- $f_2 = \min \sum_{v \in \Lambda} t_v$ , that is the minimization of the sum of each vehicle route duration  $t_v$ .
- $f_3 = \min \sum_{v \in \Lambda} d_v$ , that is the minimization of the amount of each vehicle delay  $d_v$ .

If a solution is not feasible and marked as wrong, its score is penalized in such a way that the solution is unlikely to be kept in the next population.

### C. Crossover

The crossover aims at producing new individuals (offspring) from two individuals of the current population (parents). Let  $P_1, P_2$  be two parents in the population and  $C_1, C_2$  two solutions built from  $P_1, P_2$ .  $C_1$  (resp.  $C_2$ ) is the copy of  $P_1$  (resp.  $P_2$ ) with elements from  $P_2$  (resp.  $P_1$ ).

After creating  $C_1$ , the next step consists in randomly choosing a gene of  $P_2$  at position  $\lambda$  (vector  $v_\lambda$ ).  $v_\lambda$  contains the values that will be reassigned to  $C_1$ . But, in order to avoid duplicated data, the values of  $C_1$  matching those of  $v_\lambda$  are removed. Then, the data of  $v_\lambda$  can be inserted in Gene  $\lambda$  of  $C_1$  at a random position.

In the example of Figure 3, Vector  $v_1$  of  $P_2$  is randomly chosen and the values (C C) have to be assigned to the

counterpart vector of  $C_1$ . These values are randomly inserted into Vector  $v_1$  of  $C_1$  and removed from Vector  $v_2$ . If the solutions to be crossed do not have the same size, *i.e.* the vector to be filled does not exist, a new vector is created before filling it. Producing  $C_2$  follows the same process by inverting  $P_1$  and  $P_2$ .

	Copying $P_1$ and choosing $v_1$		Assigning (C C) to $v_1$
	$C_1 = P_1$	$P_2$	$C_1$
$v_1$	B A A B	C C	B A C A C B
$v_2$	C C D D	A D D A	D D
$v_3$		B B	

Fig. 3. Example of crossover of two solutions  $P_1, P_2$  for producing a new individual  $C_1$ .

### D. Mutation

The mutation generally aims at bringing diversity into a population and prevents the population to converge towards local optima. Our operator mixes two mechanisms, each one used in half cases:

- an analogy to the well-known **2-OPT** exchange operator which reverses a sequence of values (random size) of a randomly chosen gene (Fig. 4(a)). For example a sequence (C A C B) becomes (B C A C);
- a shifting of a random customer from a vector to another (Fig. 4(b)), the pair values are randomly inserted in the host vector.

(a) Analogy to the <b>2-OPT</b> operator	
$C_1$	$C'_1$
B A C A C B	B A B C A C
D D	D D

(b) Shifting of one customer	
$C_1$	$C''_1$
B A C A C B	A C A C
D D	B D B D

Fig. 4. Example of mutation of chromosome  $C_1$  into chromosomes  $C'_1$  and  $C''_1$

## V. EXPERIMENTAL RESULTS

### A. Benchmark Test Instances

Two sets of instances are used to evaluate our modeling on existing algorithms. Each instance has 100 trips and is built on geographically realistic data [10]. All picking-up or delivery locations are geolocalized stops and the shortest duration matrix  $M$  is built using a GIS<sup>2</sup>.

The first set, denoted '*Rnd100*', is composed of 10 randomly generated instances, whereas the instances of the second set, named '*Gravit100*', are generated by a *geographical* model of flows. The '*gravity model*' [14], as well as other interaction models, allows to quantify people flows in a geographical space according to a calibration based on inhabitant densities, the time dimension, the transport facilities and attractive areas (malls, business parks...)

<sup>2</sup>Geographical Information System

Set 'Rnd100' has instances with an almost homogeneous distribution of customers while set 'Gravit100' has instances with a non-homogeneous distribution representing flows between 8:00am and 9:00am. This effect results from the existence of attractive areas which draw most flows instead of less attractive areas such as residential downtowns which are rather emission areas.

The algorithms the most suited to real-life applications can be highlighted by using realistic instances. Indeed, optimizing transport applications need efficient and robust algorithms.

### B. Parameter Setting and Performance Assessment

The population of 100 individuals evolves over 10,000 generations. A set of 20 runs *per* instance has been performed for each EA by using the ParadisEO-MOEO framework [15]. In order to evaluate the quality of the approximations for every instance we solved, we follow the protocol proposed in [16]. For a given instance, let  $Z^{all}$  denote the union of the outputs we obtained during all our experiments. We first compute a reference set  $Z_N^*$  containing all the non-dominated points of  $Z^{all}$ . Second, we define  $z^{min} = (z_1^{min}, \dots, z_n^{min})$  and  $z^{max} = (z_1^{max}, \dots, z_n^{max})$ , where  $z_k^{min}$  (resp.  $z_k^{max}$ ) denotes the lower (resp. upper) bound of the  $k^{th}$  objective for all the points contained in  $Z^{all}$ . In order to give a roughly equal range to the objective functions, values are normalized with respect to  $z^{min}$  and  $z^{max}$ . Then, to measure the quality of an output set  $A$  in comparison to  $Z_N^*$ , we compute the difference between these two sets by using the unary hypervolume metric [17],  $z^{max}$  being the reference point. The hypervolume difference indicator ( $I_H^-$ ) computes the portion of the objective space that is dominated by  $Z_N^*$  and not by  $A$ . Furthermore, we also consider the additive  $\epsilon$ -indicator proposed in [17]. The unary additive  $\epsilon$ -indicator ( $I_{\epsilon+}^1$ ) gives the minimum factor by which an approximation  $A$  has to be translated in the objective space to dominate the reference set  $Z_N^*$ . Note that both  $I_H^-$  and  $I_{\epsilon+}^1$ -values are to be minimized.

Thus, for each test instance, we obtain 20  $I_H^-$  measures and 20  $I_{\epsilon+}^1$  measures corresponding to the 20 simulation runs *per* algorithm. Once all these values are computed, we perform a statistical analysis for a pairwise comparison of methods. To this end, we use the Wilcoxon signed rank test. For a given test instance, and with respect to a  $p$ -value of 0.05 and to the metric under consideration, this statistical test reveals if the sample of approximation sets obtained by a given search method is significantly better than the one of another search method, or if there is no significant difference between both of them.

### C. Results and Discussion

Table II details a statistical comparison of each algorithm vs each other (symbols  $\prec$ ,  $\succ$ ,  $\equiv$  resp. indicate that the algorithm of a specific column is significantly 'better', 'worse' or 'equivalent' than the one of a specific row). The obtained results confirm that NSGA-II and IBEA are globally equivalent according to  $I_H^-$ , but that NSGA-II is always better (or at least equivalent) than IBEA according to  $I_{\epsilon+}^1$ . SPEA2 is always

outperformed by the other methods. Thus, it appears that NSGA-II is better than both IBEA and SPEA2 for optimizing randomly generated instances of DARPTW.

Table III is the counterpart for the realistic instances. Contrary to Table II, IBEA is shown to be the best algorithm with respect to both  $I_H^-$  and  $I_{\epsilon+}^1$  metrics (except for instance *Gravit100-4*). These unambiguous results indicate that IBEA is the most efficient (or at least equivalent) algorithm for optimizing realistic DARPTW instances while comparing pairwise algorithms. If we focus on the comparison of the two other algorithms, we see that SPEA2 is always better than NSGA-II with respect to  $I_H^-$ , while they are almost equivalent with respect to  $I_{\epsilon+}^1$ .

The main differences between the two kinds of instances are the densities of customers on the territory. Given the level of homogeneity of the distribution of the requests, the algorithms differently cope with the clustering effects. Hence, it seems that IBEA which uses a selection by indicator, appears to be more robust to the clustering effects due to the heterogeneous distribution of the customers.

## VI. CONCLUSION AND PERSPECTIVES

This paper is devoted to solve a DARPTW applied to a Demand Responsive Transport service. After having formalised the general problem, an original evolutionary multi-objective approach was presented. Thanks to the Pareto-dominance principles, our approach is able to build a set of distinct non-dominated solutions which are provided to the decision maker in order to support and help his final choice. To fit the evolutionary algorithms to be compared, an original representation and variation operators were proposed and developed.

The candidate algorithms are: NSGA-II, IBEA and SPEA2. They were performed on two sets of transport instances. The first set is composed of random instances and the second of realistic instances. Such instances allow to assess the algorithm efficiency to cope with real-life applications.

The analysis of the results underline the more or less adapted behavior of the algorithms. It results of this comparative study that NSGA-II generally seems to be more efficient than IBEA for the random instances and SPEA2 is always outperformed by its counterparts. However, if we focus on the realistic instances, the results bring to light that IBEA is the best algorithm for solving realistic DARPTW.

Given that the algorithms are sensitive to the distribution of customers and flows, a preemptive and relevant work would be to integrate the geographical dimension. Indeed, the better results obtained by IBEA on the realistic instances invite us to take this dimension into account. Therefore, a lot of complementary analysis will have to be done in order to improve our understanding of the algorithm sensitivity. These analysis concern the inhabitants densities and the distribution of the stops. Indeed, the experiments on realistic instances seem to indicate the algorithm sensitivity to a specific topology. Then, it is necessary to evaluate the algorithms behavior according to the problem structures. Besides, an uncertainty integration

[18] in the method will help us to develop dynamic [19] and multiobjective DARPTW services at the same time.

#### ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their contribution and their relevant remarks.

#### REFERENCES

- [1] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: Models and algorithms," *Annals of Operations Research*, vol. 153, no. 1, pp. 29–46, 2007.
- [2] B. Golden, S. Raghavan, and E. Wasil, Eds., *The Vehicle Routing Problem: Latest Advances and New Challenges*. New York: Springer, 2008, 589 p.
- [3] J. Paquette, J.-F. Cordeau, and G. Laporte, "Quality of service in dial-a-ride operations," *Computers & Industrial Engineering*, vol. 56, no. 4, pp. 1721 – 1734, 2009.
- [4] E. G. Talbi, *Metaheuristics: from design to implementation*. Wiley, 2009, 624 p.
- [5] K. Deb, *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, 2001, 517 p.
- [6] O. Bräysy, W. Dullaert, and M. Gendreau, "Evolutionary algorithms for the vehicle routing problem with time windows," *Journal of Heuristics*, vol. 10, pp. 587–611, 2004.
- [7] R. Chevrier, P. Canalda, P. Chatonnay, and D. Josselin, "Comparison of three algorithms for solving the convergent demand responsive transportation problem," in *ITSC'2006, 9th Int. IEEE Conf. on Intelligent Transportation Systems*, Toronto, Canada, Sep. 2006, pp. 1096–1101.
- [8] R. Jørgensen, J. Larsen, and K. Bergsvindottir, "Solving the dial-a-ride problem using genetic algorithms," *Journal of the Operational Research Society*, vol. 58, pp. 1321–1331, 2007.
- [9] C. Cubillos, N. Rodriguez, and B. Crawford, *Bio-inspired Modeling of Cognitive Tasks*, ser. Lecture Notes in Computer Science. Berlin / Heidelberg: Springer, 2007, ch. A Study on Genetic Algorithms for the DARP Problem, pp. 498–507.
- [10] R. Chevrier, "Optimization of demand responsive transport in polarized territories," Ph.D. dissertation, UMR ESPACE (CNRS 6012), University of Avignon (France), november 2008, 244 p.
- [11] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, *Parallel Problem Solving from Nature PPSN VI*. Berlin/Heidelberg: Springer, 2000, vol. 1917/2000, ch. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II, pp. 849–858.
- [12] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Computer Engineering and Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Tech. Rep. 103, 2001.
- [13] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *LNCS - Parallel Problem Solving from Nature - PPSN VIII*, vol. 3242/2004, 2004, pp. 832–842.
- [14] P. Haggett, *Locational Analysis in Human Geography*. London: Edward Arnold, 1977, 2nd edition, Volume I: "Locational Models" (298 pages) and Volume II: "Locational Methods" (346 pages).
- [15] A. Liefvooghe, L. Jourdan, and E.-G. Talbi, "A unified model for evolutionary multiobjective optimization and its implementation in a general purpose software framework: ParadisEO-MOEO," INRIA, Research Report RR-6906, 2009.
- [16] J. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, Tech. Rep., 2006, (revised version).
- [17] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [18] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments - a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, pp. 303–317, 2005.
- [19] Z. Xiang, C. Chu, and H. Chen, "The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments," *European Journal of Operational Research*, vol. 185, no. 2, pp. 534–551, 2008.

TABLE II  
STATISTICAL COMPARISON FOR RANDOMLY GENERATED INSTANCES.

		$I_H^-$			$I_{\epsilon^+}^+$		
		NII	I	S2	NII	I	S2
R0	NII	.	.	.	.	.	.
	I			Y		Y	
	S2	Y	Y	.	Y		.
R1	NII	.	.	.	.	.	.
	I			Y		Y	Y
	S2	Y	Y	.	Y		.
R2	NII	.	.	.	.	.	.
	I			Y			Y
	S2	Y	Y	.	Y		.
R3	NII	.	.	.	.	.	.
	I	Y	.	Y		Y	Y
	S2	Y	Y	.	Y		.
R4	NII	.	.	.	.	.	.
	I			Y		Y	
	S2	Y	Y	.	Y	Y	.
R5	NII	.	.	.	.	.	.
	I			Y		Y	
	S2	Y	Y	.	Y		.
R6	NII	.	.	.	.	.	.
	I			Y		Y	
	S2	Y	Y	.	Y		.
R7	NII	.	.	.	.	.	.
	I			Y		Y	
	S2	Y	Y	.	Y		.
R8	NII	.	.	.	.	.	.
	I			Y		Y	Y
	S2	Y	Y	.	Y		.
R9	NII	.	.	.	.	.	.
	I			Y		Y	
	S2	Y	Y	.	Y		.

For each instance, the algorithms are opposed and compared by twos (NII: NSGA-II, I: IBEA, S2: SPEA2); Instance  $R_i$  indicates instance  $Rnd100-i$

TABLE III  
STATISTICAL COMPARISON FOR REALISTIC INSTANCES.

		$I_H^-$			$I_{\epsilon^+}^+$		
		NII	I	S2	NII	I	S2
G0	NII	.	Y	Y	.	Y	
	I	Y	.	Y		.	Y
	S2	Y	Y	.		Y	.
G1	NII	.	Y		.	Y	
	I	Y	.	Y		.	Y
	S2		Y	.		Y	.
G2	NII	.	Y	Y	.	Y	Y
	I	Y	.	Y		.	Y
	S2	Y	Y	.		Y	.
G3	NII	.	Y		.	Y	
	I	Y	.	Y		.	Y
	S2		Y	.		Y	.
G4	NII	.	Y		.		Y
	I	Y	.	Y		.	
	S2		Y	.			.
G5	NII	.	Y	Y	.	Y	
	I	Y	.	Y		.	Y
	S2	Y	Y	.		Y	.
G6	NII	.	Y	Y	.	Y	
	I	Y	.	Y		.	Y
	S2	Y	Y	.		Y	.
G7	NII	.	Y		.	Y	
	I	Y	.	Y		.	Y
	S2		Y	.		Y	.
G8	NII	.	Y		.	Y	
	I	Y	.	Y		.	Y
	S2		Y	.		Y	.
G9	NII	.	Y	Y	.	Y	
	I	Y	.	Y		.	Y
	S2	Y	Y	.		Y	.

For each instance, the algorithms are opposed and compared by twos (NII: NSGA-II, I: IBEA, S2: SPEA2); Instance  $G_i$  indicates instance  $Gravit100-i$