


## Article

# An Evolutionary Descent Algorithm for Customer-Oriented Mobility-On-Demand Problems

Sonia Nasri <sup>1,\*</sup>, Hend Bouziri <sup>2</sup> and Wassila Aggoune-Mtalaa <sup>3</sup> 

- <sup>1</sup> LARODEC Laboratory, Business Higher School of Tunis, Manouba University, Manouba 2010, Tunisia  
<sup>2</sup> LARODEC Laboratory, Higher School of Economics and Business, Tunis University, Tunis 1002, Tunisia; hend.bouziri@gmail.com  
<sup>3</sup> Luxembourg Institute of Science and Technology, L-4362 Esch-sur-Alzette, Luxembourg; wassila.mtalaa@list.lu  
\* Correspondence: sonia15.nasri@gmail.com

**Abstract:** This paper is addressing a new class of on-demand transport problems oriented toward customers. A mixed-integer linear programming model is proposed with new effective constraints that contribute to enhancing the quality of service. An exact resolution has been achieved, leading to lower bounds of the solution space of real cases of on-demand transport problems. To overcome the exponential computational time of the exact resolution, an evolutionary descent method is developed. It relies on a new operator for perturbing the search. The comparative results between the new method and the branch and bound show low gaps for almost all the instances tested with lower execution times. The results of the evolutionary descent method are also compared with the results of two different heuristics, namely a Tabu Search and an Evolutionary Local Search. Our evolutionary method demonstrates its effectiveness through competitive and promising results.



**Citation:** Nasri, S.; Bouziri, H.; Aggoune-Mtalaa, W. An Evolutionary Descent Algorithm for Customer-Oriented Mobility-On-Demand Problems. *Sustainability* **2022**, *14*, 3020. <https://doi.org/10.3390/su14053020>

Academic Editors: Panagiotis Georgakakis, Efthimios Bothos, Babis Magoutas and Michiel de Bok

Received: 31 January 2022

Accepted: 2 March 2022

Published: 4 March 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** mobility-on-demand problem; quality of service; dial-a-ride problem; customer service design; metaheuristics; exact optimization

## 1. Introduction

This paper aims at addressing a sustainable mobility-on-demand problem for enabling fair access to employment, social life, markets, and services to people. It is in line with the 11th Sustainable Development Goal (SDG) of the United Nations for “Making Cities and Human Settlements Inclusive, Safe, Resilient and Sustainable”. The sustainable transport agenda for 2030 aims at enhancing mobility, including urban mobility ensuring access to safe, accessible, and sustainable transportation systems for all at an affordable cost. Therefore, on-demand mobility services should be designed as sustainable transport options for enhancing accessibility to transport, meeting various needs of people in their personal and economic lives. Such systems are part of a growing sector named Mobility as a Service (MaaS) which advocates accessibility to means of transport and personalization of services.

In this study, we focus on a Dial-A-Ride Problem (DARP) which was originally proposed by the authors of [1]. The DARP is also known as an on-demand transportation problem in the work of [2], where a transportation demand is a request made by a customer who wants to be transported from an origin to a destination site. A DARP is characterized by the quality of service offered to the customers. The most common features which reflect this service level are both the time windows imposed at different locations and the maximum riding time which is the time needed to fulfil a request of transport. However, new advances are required to deal with other citizens' needs which are more expressed and varied according to the authors of [3]. A better customized and effective design of the service quality is highlighted as beneficial for both the service provider and the customer, see the work of [4].

In this work, our contribution is twofold. On the one hand, we propose a new problem providing more effective on-demand transport plans. In this model, the customers' requirements are expressed and used to fulfil the expected demand and improve the service quality. More precisely, a customer-oriented design of the time windows is proposed for minimizing passengers' waiting times. Then, the resulting mixed-integer linear programming (MILP) model is implemented using the CPLEX optimizer tool which uses the branch and bound (B&B) exact method. The experimentation is performed on real-life benchmark data.

On the other hand, as computation times required by CPLEX are too long when solving complex and real-life problems (exponential with regards to the heuristic approaches), a new evolutionary-based approach called the Evolutionary Descent Algorithm (EDA) is developed. With the use of this new EDA, near-optimal solutions are obtained for real-life on-demand transport problems. Two types of experiments are set. In the first part, the lower bounds of the solution space for the studied problem are compared against near-optimal solutions produced by our EDA. Then, a deeper comparison of the EDA algorithm with other existing metaheuristic-based approaches from the literature is achieved.

The paper is organized as follows. Section 2 presents the relevant literature linked with the studied problem. Section 3 defines a mathematical formulation of the problem. Section 4 describes the EDA for the resolution of the problem. Section 5 reports numerical results obtained by the EDA on real-life instances with a comparative study with existing methods of the literature. Finally, a discussion on the results of the proposed methodology is undertaken in Section 6, followed by a conclusion.

## 2. Relevant Literature on Dial-A-Ride Problems

### 2.1. The Classical Dial-A-Ride Problem

The DARP is a mobility-on-demand problem marked by the quality of service offered to the customers. This problem was initially dedicated to the elderly or those with reduced mobility (e.g., [5,6]). Nowadays, this problem is evolving to be applied in new fields like integrated transport in [7,8], the healthcare domain in [9,10], and ride-sharing transport in [11]. Many variants are provided in the work of [2] including two routing transportation issues involving static or dynamic frameworks with single or multiple vehicles. The quality of service is defined in [3] through several terms which are related to customer perceptions and technical approaches. From the customer perception viewpoint, the quality of service is linked to safety, comfort, reliability, and vehicle access. From the technical perspective, the quality of service is defined with operational terms such as maximum ride time, total travel time, user riding time, and waiting time. These terms are expressed in the form of constraints as in the works of [12–14] or directly in the objective function as in the works of [15,16]. The different expressions of quality of service come from the variety of models which try to find a balance between the travel costs and the quality of service for the customer.

### 2.2. The Dial-A-Ride Problem with a Customized Design of the Service Quality

Customers' access to mobility has evolved rapidly due to technological advancements and changing customer preferences. Several factors such as the price, waiting time, travel time, convenience, and traveller experience are included when selecting the appropriate mode of transport by the mobility service consumers as in [17]. These factors should be well addressed to ensure the improvement of the quality of service. This quality provided in [18] is measured through waiting time metrics and riding time according to customers' preferences for improving both the service quality and the distances travelled by the vehicles. As highlighted by the authors of [19], the transportation of customers should be improved through the use of new information and communication technologies. Their expectation was to draw new public transport services to customers.

Thus, one way to address the new demands of transport nowadays is to be able to align with the diversified needs of the customers for transport. This is what is highlighted in [4], fostering the modelling of mobility-on-demand problems according to a customer-oriented

design. This issue is thoroughly studied in the survey of [20] which gathers two main fields of research related to the quality of service levels in DARPs. In the first group of works, the DARP is qualified as classical when it aims at minimizing the travel costs ensuring a common design of the service quality for all the users, (e.g., [5,21–24]). In the second trend of works, the quality of service is studied from a customer-oriented point of view. In this second group of researches, the authors of [25] assessed the variation of different features related to the quality of service on operational costs. The authors of [26] proposed time windows, and a maximum riding time for each user. A maximum ride time is specified for each customer and investigated for designing time windows in other works of [13,27]. A recent example of another customer-oriented computation of the time windows is made using the maximum riding time and the beginning of service in the work of [14]. In the problem of [28], a request is split into a set of groups of persons and has a revenue. Time windows are redefined for only the delivery location using a service time, a transit time and a maximum ride time. Based on multi-criteria models, the authors of [29] stressed that expanding the size of the time window for a transport service is beneficial for the system provider but not for the customers, which is also emphasized by the authors of [30].

In this paper, we consider a customer-oriented service quality in which each customer specifies his/her own riding time. In this model, the time windows are designed to minimize the waiting time of the users.

### 2.3. Resolution Methods for the Dial-A-Ride Problem

The most successful method for solving a wide range of Dial-Ride Problems is the Tabu Search in [31]. This local-based method is extensively applied for artificial cases of DARP (e.g., [22,32,33]). However, this method needs to be adapted to deal with real-life applications and customized designs of DARPs. Some works (e.g., [10,34]) in the literature address the quality of service in DARPs from a customer point of view. The authors in [34] proposed a Tabu Search based on a neighbourhood strategy that operates a tactical move operation. The latter consists in altering the requests schedule by allocating a random request to its nearest predecessor in another route. An insertion heuristic relies on iterative checks of the constraints. In order to diversify the search, the Tabu list is emptied after a selected number of TS iterations. Real-life cases of on-demand transport are tackled and the results for small instances are promising.

In the field of population-based approaches, few evolutionary algorithms are found in the literature for solving mobility-on-demand problems (e.g., [35–37]). Several operators, namely the mutation and the crossover ones, are proposed. In [35], the crossover consists in randomly incorporating a customer in the current solution. A cluster mutation and a route mutation are used. The artificial instances in [38] were used for the experimentation. As it was stated by the authors of [37], a crossover aimed at generating a non-legal solution by randomly selecting a cluster of nodes from two different parents. Then the mutation operation consisted of a local modification in the clusters of nodes. Experiments were performed and compared to instances in [1].

To the best of our knowledge, the only evolutionary approach devoted to solving DARPs including the customer viewpoint is the Evolutionary Local Search (ELS) of [13]. The mutation consists in removing a transportation request from a trip to be inserted into another. Next, transportation requests are exchanged between two trips. This action is considered the crossover operator. The crossover is applied to locally alter a mutated solution. Each mutated solution is improved by a local search which is based on six neighbourhood structures. These structures are dynamically managed during the hybridized algorithm iterations. This management is provided by the mean of several probabilities. With this regard, a local search is altered through the evolutionary computations increasing the level of intensification of the search process.

New evolutionary algorithms should be applied on real-life cases of DARPs such as the 96 instances of [39] considering customer expectations. For these problems, the ELS

of [13] performed well. This is on some of these instances that the new EDA algorithm proposed in this paper is tested (see Section 5).

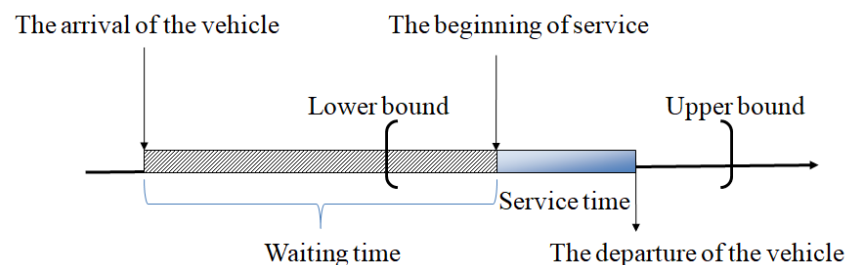
### 3. Problem Definition and Formulation

#### 3.1. The Customer-Oriented DARP

The customer-oriented DARP is an on-demand transport problem. It consists of satisfying a set of prior known transportation requests using a set of homogeneous vehicles with a maximum capacity. A request corresponds to a limited number of people who require to be transported from one location to another. This number is the same on the pickup and the delivery sites. Thus, the set of requests corresponds to that of the pickup nodes. The depot is concerned by the departure and the arrival of all the vehicles and no demand is assigned to it. In addition, a maximum total duration is fixed to limit the total vehicle travel time. Each customer has a riding time which is the time interval from the time of his/her pickup to that of the delivery including waiting times which occurred during the vehicle route.

The customer-oriented DARP is different from classical DARPs in terms of the design of the service quality which is customer oriented. Firstly, a maximum riding time is specified for each customer considering the distance between the customer's origin and that of the destination. Secondly, new customized bounds are produced for time windows at origin and destination sites separately. The redefinition of the new bounds is based on customers' expectations including a lower riding time (time spent aboard the vehicle), service time (the time taken to load or discharge people from the vehicle), and a restricted maximum riding time. The major contribution of the customized design of time windows relies on reducing the customers' waiting times.

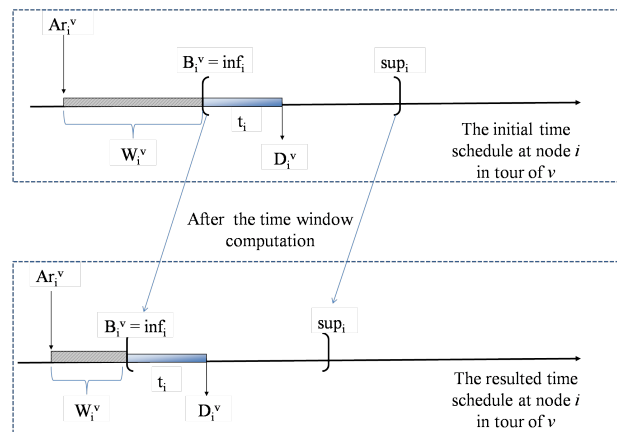
A potential waiting time at a node is illustrated by Figure 1. In this figure, the time window is defined by its lower and upper bound.



**Figure 1.** A time schedule at a visited node.

Inside this time window, the beginning of the service is bounded. In addition, the arrival time and the departure time are defined at the node. Note that a beginning of service is always considered as the maximum value between the arrival time and the earliest service bound. The waiting time is the time between the vehicle's arrival time and the beginning of service at a node. The vehicle's departure time is the time that starts after the end of the service time for loading or delivering people.

In the studied problem, particular attention is given to the time windows for improving the waiting time as well as the total duration, see the example in Figure 2.



**Figure 2.** A case of a reduced waiting time at a visited node  $i$  in the tour of  $v$ .

### 3.2. Mathematical Formulation of the Customer-Oriented DARP

The customer-oriented DARP is defined on a complete directed graph  $G = (N, A)$ , where  $N$  is the set of vertices and  $A$  is the set of arcs. The set of nodes  $i \in N$  is defined as  $\{0 \dots 2n + 1\}$  which includes the depot in two copies ( $i = 0$ ) at the departure and ( $i = 2n + 1$ ) at the arrival, the set of pickup nodes  $\{1 \dots n\}$ , and the set of delivery nodes  $\{n + 1 \dots 2n\}$ . The set of arcs is defined as  $\{(i, j)$  with  $i, j \in N$ , and  $i \neq j\}$ . We assume that a fleet of homogeneous vehicles is available at the depot to satisfy a given number of requests. A request is defined by an origin node  $i \in \{1 \dots n\}$  and a destination one  $(i + n) \in \{n + 1 \dots 2n\}$ .

#### 3.2.1. Notations Used in the Formulation

The problem under consideration is NP-Hard, as it was proved by [40]. It can be modelled as a mixed-integer linear program (MILP) using the following decisions variables: The first one is a routing variable whereas the others are scheduling variables.

- $x_{(i,j)}^v$  is a binary variable equal to 1 if and only if the vehicle  $v$  goes from node  $i$  to node  $j$ , equal to 0 otherwise.
- $B_i^v$ ,  $Ar_i^v$ ,  $W_i^v$ , and  $R_i^v$  are real variables that indicate the starting time of service, the arrival time, the waiting time, and the riding time at node  $i$  if the latter is visited by a vehicle  $v$ .

The set of notations used in the problem formulation is in Table 1.

**Table 1.** The problem notations.

Parameters	Description
$n$	Total number of requests
$m$	Total number of vehicles
$l_i$	Number of persons to charge or discharge at node $i \in N$
$t_i$	Service time at node $i \in N$
$\delta$	Penalty term on waiting time
$tr_{(i,j)}$	Transit time on an arc $(i, j) \in A$
$C$	Maximum capacity of the vehicles
$mrt_i$	Maximum riding time of a request $i \in \{1 \dots n\}$
$dr$	Maximum total tour duration of the vehicles
$[inf\_i, sup\_i]$	Time window at node $i \in N$

The vehicle's arrival time  $Ar_i^v$  at node  $i$  is the time spent from its departure to this node. Note that this arrival is allowed before the beginning of service  $B_i^v$  but not after. Therefore, only a positive value for the waiting time at node  $i$  is considered. It is equal to the time between the arrival time  $Ar_i^v$  and the starting of the service  $B_i^v \in [inf_i, sup_i]$  where  $inf_i$  and  $sup_i$  are the earliest and the lasted service time respectively.

The departure time at node  $i$  is  $D_i^v = B_i^v + t_i$ . In this problem the service time  $t_i$  is equal to the passengers' loading time. Given that no demand is assigned at the depot, the departure time at the depot is equal to the beginning of service; thus  $B_0^v = D_0^v$  and  $B_{2n+1}^v = D_{2n+1}^v$ . In addition, the arrival time at the depot is initialized as  $B_0^v = Ar_0^v$  as well as  $B_{2n+1}^v = A_{2n+1}^v$ . Last, we consider that the parameters  $t_i = t_{i+n}$ ,  $l_i = l_{i+n}$ , and  $mrt_i = mrt_{i+n}$  are the same at the origin and the destination sites of a request.

### 3.2.2. Mathematical Formulation of the Customer-Oriented DARP

The objective function which has to be minimized, see (1), consists of the total travel costs generated by the visited arcs in the vehicle tours. A travel cost  $c_{(i,j)}$  on a visited arc  $(i, j)$  is the sum of the transit time  $tr_{(i,j)}$  and the service time  $t_i$  at node  $i$ . Besides, the total travel cost includes the total penalties on waiting times when visiting nodes. The value of  $\delta$  is chosen between 0.05 and 0.1 as it is used in [41]. This variable increases in the event of a waiting time otherwise it decreases. The penalty growth is defined by  $(\delta = \delta * (1 + \delta))$ . Otherwise, the penalty term follows  $(\delta = \delta \setminus (1 + \delta))$ .

$$Min f(S) = \sum_{v=1}^{v=m} \sum_{(i,j) \in A} (c_{(i,j)} * x_{(i,j)}^v + \delta * W_i^v) \tag{1}$$

Subject to:

$$\sum_{v=1}^{v=m} \sum_{i=1}^{i=n} x_{(i,j)}^v = 1, \quad \forall j \in N \tag{2}$$

$$\sum_{j \in N} x_{(i,j)}^v - \sum_{j \in N} x_{(j,i)}^v = 0, \quad \forall i \in N, \quad \forall v \in \{1 \dots m\} \tag{3}$$

$$\sum_{(i,j) \in A} x_{(i,j)}^v - \sum_{(i,j) \in A} x_{(i+n,j)}^v = 0, \quad \forall v \in \{1 \dots m\}, \quad \forall i \in \{1 \dots n\} \tag{4}$$

$$\sum_{j \in N \setminus \{0, 2n+1\}} x_{(0,j)}^v = 1, \quad \forall v \in \{1 \dots m\} \tag{5}$$

$$\sum_{i \in N \setminus \{0, 2n+1\}} x_{(i, 2n+1)}^v = 1, \quad \forall v \in \{1 \dots m\} \tag{6}$$

$$0 \leq \sum_{i \in N} l_i * x_{(i,j)}^v \leq C, \quad \forall (i, j) \in A, \quad \forall v \in \{1 \dots m\} \tag{7}$$

$$\sum_{j \in N} l_j * x_{(i,j)}^v \geq \sum_{i \in N} l_i * x_{(i,j)}^v + l_j, \quad \forall (i, j) \in A, \quad \forall v \in \{1 \dots m\} \tag{8}$$

$$B_{2n+1}^v - B_0^v \leq dr, \quad \forall v \in \{1 \dots m\} \tag{9}$$

$$tr_{(i,i+n)} \leq R_i^v \leq mrt_i, \quad \forall i \in \{1 \dots n\}, \quad \forall v \in \{1 \dots m\} \tag{10}$$

$$B_{i+n}^v - (B_i^v + t_i) = R_i^v, \quad \forall i \in \{1 \dots n\}, \quad \forall v \in \{1 \dots m\} \tag{11}$$

$$B_j^v \geq B_i^v + tr_{(i,j)} + t_i + W_j^v, \quad \forall v \in \{1 \dots m\}, \quad \forall (i, j) \in A \tag{12}$$

$$B_{i+n}^v \geq B_i^v + tr_{(i,i+n)} + t_i + W_{i+n}^v, \quad \forall v \in \{1 \dots m\}, \quad \forall i \in \{1 \dots n\} \tag{13}$$

$$Ar_{(j)}^v \geq B_{(i)}^v + t_i + tr_{(i,j)}, \quad \forall (i, j) \in A, \quad \forall v \in \{1 \dots m\} \tag{14}$$

$$Min \left( inf_{(i+n)} - mrt_i - t_i, inf_i \right) \leq B_i^v + M * (1 - var), \tag{15}$$

$$\forall i \in \{1 \dots n\}, \forall v \in \{1 \dots m\}$$

$$B_i^v - M * var \leq Min \left( sup_{(i+n)} - t_i - R_i^v, sup_i \right), \tag{16}$$

$$\forall i \in \{1 \dots n\}, \forall v \in \{1 \dots m\}$$

$$\text{Min}\left(\text{inf}_i - \text{mrt}_i - t_i, \text{inf}_{(i+n)}\right) \leq B_{(i+n)}^v + M * (1 - \text{var}), \quad (17)$$

$$\forall i \{1 \dots n\}, \forall v \{1 \dots m\}$$

$$B_{(i+n)}^v - M * \text{var} \leq \text{Min}\left(\text{sup}_{(i)} - t_i - R_i^v, \text{sup}_{(i+n)}\right), \quad (18)$$

$$\forall i \{1 \dots n\}, \forall v \{1 \dots m\}$$

$$W_{(i)}^v = B_{(i)}^v - Ar_{(i)}^v \quad \forall i \in N \quad \forall v \{1 \dots m\} \quad (19)$$

$$x_{(i,j)}^v \in \{0, 1\}, \quad \forall (i, j) \in A, \quad \forall v \in \{1 \dots m\} \quad (20)$$

$$B_{(i)}^v \geq 0, Ar_{(i)}^v \geq 0, W_{(i)}^v \geq 0, R_{(i)}^v \geq 0, \quad \forall i \in N, \quad \forall v \in \{1 \dots m\} \quad (21)$$

The first constraints of the problem are common with standard DARP constraints. Indeed, in equality (2), we express that a request is allocated to exactly one vehicle. Besides, Equation (3) makes sure that each vehicle that visits a node also leaves it. Equation (4) expresses the balance between the pickups and the deliveries in a vehicle's tour. Equations (5) and (6) assume the depot as the beginning and end of each tour. The constraints (7) and (8) ensure that the vehicle's capacity is respected while charging and discharging persons at the pickup and delivery nodes. A positive load is supposed in the case of a pickup and the load is negative otherwise. The total travel duration is expressed by the set of constraints (9). Let us note that the travel time of each vehicle starting from the depot and returning to it must respect this common bound.

Other constraints are defined for enhancing the quality of the service provided to the customers. The riding time expressed by (11) is the time spent between the vehicle's departure  $D_i^v$  at the origin and the beginning of service  $B_{i+n}^v$  at the destination. This riding time is limited by a customer-oriented maximum riding time  $\text{mrt}_i$  in (10). The precedence constraint between the pickup and delivery stations is mentioned in (13) and that between the nodes is defined in (12). The arrival time  $Ar_i^v$  is computed by (14).

The time windows are separately designed for the origin nodes and those of the destination. Equations (15) and (16) successively present the lower and upper bounds for the beginning of service at the origin sites. The time window at an origin is tightened by taking the minimum value between the initial earliest service time  $\text{inf}_{(i)}$  supposed for a node of pickup, and the new derived one by  $(\text{inf}_{(i+n)} - \text{mrt}_i - t_i)$ . Consequently, the beginning of service is bounded by a new resulted earliest service time in (15).

Equations (17) and (18) define the lower and upper bounds for the beginning of service at the destinations. The new bounds are the minimal values between the initial time windows  $[\text{inf}_i, \text{sup}_i]$  and  $[\text{inf}_{(i+n)}, \text{sup}_{(i+n)}]$  and those adjusted to customers' specifications. These latter are the effective customer riding time  $R_i^v$ , the service time  $t_i$ , and the maximum ride time  $\text{mrt}_i$  which are used in this redefinition. With this time windows customization, a new beginning of service is calculated leading to a minimization of the waiting time expressed by (19).

To achieve the linearization of the time window constraints, we use a large constant  $M$  and a Boolean's variable  $\text{var}$ . This constant is generally fixed as 1000, see the work of [1,42], which used this value successfully for the linearization of their problem constraints. Moreover, Equation (20) ensures that the binary nature of the decision variables  $x_{(i,j)}^v$  is respected. Finally, inequality (21) impose the positivity of the scheduling decision variables.

Figure 2 shows two cases of a time schedule at a node  $i$  in a tour of  $v$ : before and after the computation of the time window. In the second case, we illustrate an enhanced lower bound  $\text{inf}_i$  and upper bound  $\text{sup}_i$  at the node  $i$ . Thus, for the same arrival time  $Ar_i^v$  computed by (14), an enhanced value of  $B_i^v \in [\text{inf}_i, \text{sup}_i]$  can be obtained by the time windows Equations (15)–(18). Note that we assumed having  $B_i^v = \text{inf}_i^v$  at the node  $i$  visited by the vehicle  $v$ .

In addition, the new beginning of service impacts the departure time at node  $i$  and the arrival time at the next visited node  $j$ . Moreover, the total travel duration of the vehicle is also influenced as it is expressed by (9). As a consequence, both the customer and the

service provider may take benefit from the reduction of the waiting times as well as the unnecessary delays during a route.

The equations of the model are further implemented and tested on real instances in the experimental section. Indeed, although metaheuristics [43–46] and especially hybrid ones [47,48] are effective and fast for solving complex problems such as scheduling and transportation ones, we want here to have a comparative experimentation with an exact resolution [49,50] to find the optimal solution of each of the customer-oriented DARP instances foreseen. This exact resolution will be compared to the new evolutionary method which is proposed to solve the problem.

### 3.3. The Relevance of the Customized Time Windows on a Real-Life Case of On-Demand Transport

This section shows how tightened time windows improve the waiting times of the customers on a real-life example of on-demand transport. Therefore we use a problem instance of [39]. In this example, we consider a single-vehicle tour that satisfies two requests, one of which generates a long waiting time. This vehicle has a maximum capacity of 8 passengers and a maximum total duration of 480. There are two persons who have to be transported from the origin location to the destination in the first request and four for the second request.

In order to illustrate the influence of the time windows, we present two successive cases with and without the consideration of the new constraints. Two figures are proposed indicating time windows and waiting times. Figure 3 shows the case of a vehicle’s tour satisfying the two requests with the initial time windows proposed in the instance of [39] and with a waiting time which occurs at node 2. Figure 4 shows the case of the same tour but with the application of the time windows computed by the new Equations (15)–(18).

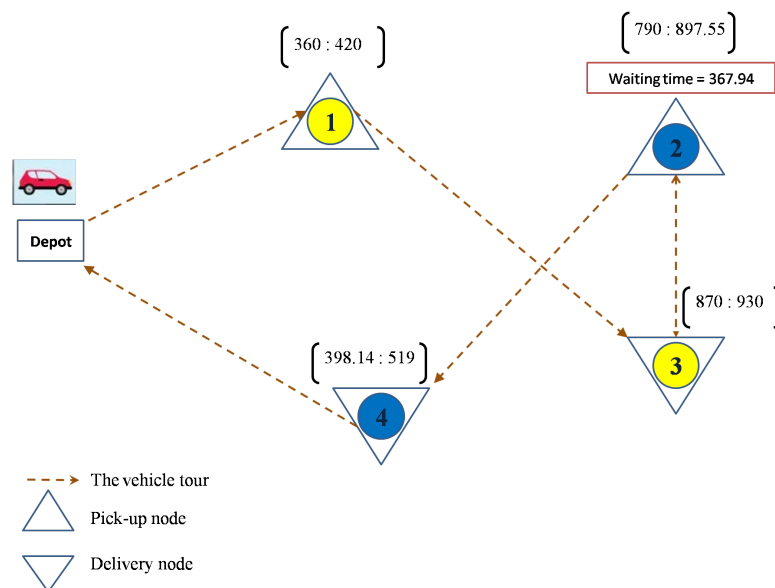


Figure 3. The tour without the time windows tightening.

In Table 2, we indicate the parameters related to the vehicle’s tour example of Figure 3, especially those linking request 2 with origin node 2 and destination node 4. This request has a maximum riding time equal to 76.

Table 2. Initial parameters of request 2.

Visited Nodes	$l_i$	$tr_{(i,j)}$	$Ar_i^v$	$B_i^v$	$D_i^v$
2	4	28.45	469.62	837.55	841.55
4	−4	3.2	870	870	874



The riding time of request 2 is equal to ( $R_2^v = 28.45$ ). The only node which is concerned by a waiting time is node ( $i = 2$ ) where we have ( $W_2^v = 367.94$ ). This waiting time  $W_2^v$  delays the arrival time at the successive visited nodes and thus impacts the total vehicle's tour duration equal to 470.15.

In order to reduce unnecessary waiting times, new time windows are computed using Equations (15)–(18). Given that the time windows are correlated, the bounds on the origin are based on those provided for the destination. Therefore, at node 4, we have  $\min(790-4-76;870) = 710$  as the lower bound and  $\min(897.55-4-28.45;930) = 865.1$  as the upper bound. Whereas for node 2,  $\min(710-4-76;790) = 630$  is computed as the lower bound and  $\min(865.1-4-28.45;897.55) = 832.65$  as the upper bound. The beginning of service  $B_2^v$  is then calculated using Equation (11) based on the new earliest service time  $B_4^v = 710$ . As a result,  $B_2^v = 677.55 \in [630, 832.66]$ .

Given Equation (1), the new waiting time  $W_2^v$  is reduced as ( $B_2^v - A_2^v = 677.55 - 469.62 = 207.93$ ). Consequently, the arrival time  $A_4^v$  is changed by ( $A_4^v = 469.62 + 207.93 + 28.45$ ), with  $tr_{(2,4)} = 28.45$ ,  $W_2^v = 207.93$ , and  $A_2 = 469.62$ . Then, we obtain ( $A_4 = 706$ ) and a new lower waiting time at node 4 ( $W_4^v = 4$ ). Globally, with the time windows tightening, the total waiting time is less than the initial one  $W_2^v = 367.94$  indicated in Figure 3. Indeed, we obtain ( $W_2^v + W_4^v = 207.93 + 4 = 211.93$ ).

The vehicle tour after the new time windows calculation is illustrated in Figure 4.

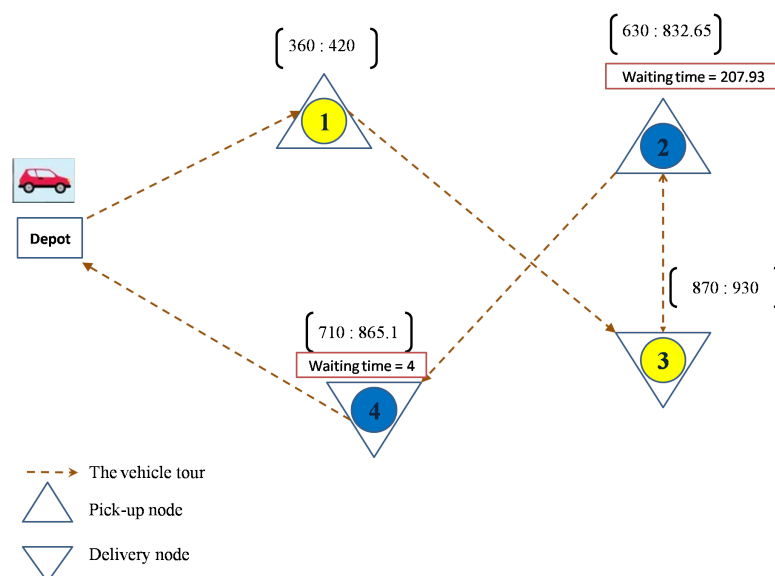


Figure 4. The case with the time window recalculation.

The minimization of unnecessary waiting times contributes to also minimizing the total duration since the new arrival times at nodes 2, 4, and 5 (the depot) are enhanced.

#### 4. The Evolutionary Descent Algorithm for the Customer-Oriented DARP

To produce good solutions to the problem in shorter computational times, we propose an evolutionary descent approach (EDA). The main steps of the algorithm are as follows.

**Step 1:** Construct an initial solution of the EDA using the customized insertion heuristic, see Section 4.2.

**Step 2:** At each iteration in the EDA, construct a population of individuals by perturbing the initial solution.

**Step 3:** For each individual in the population, apply the crossover (see Section 4.3).

**Step 4:** Evaluate each solution in the population using the objective function (1) of Section 3.2 and save the best individual.

**Step 5:** If the fitness of the best individual is the best encountered one in the overall research, consider it as a new initial solution and save it as a temporary best solution in the search space.

**Step 6:** If the maximum number of iterations is not reached, then go to step 2; otherwise, go to step 7.

**Step 7:** Keep the final temporary best solution as the best one in the EDA.

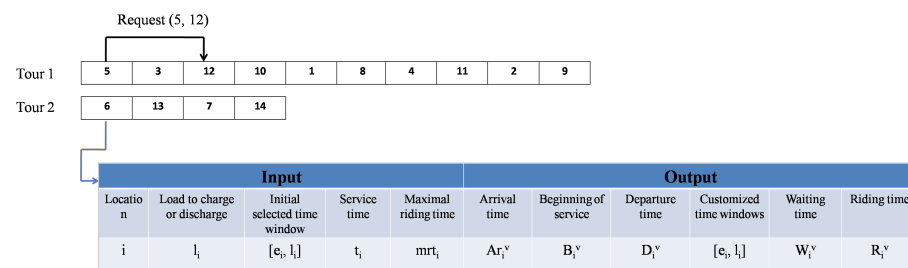
To start the search in the EDA, an initial solution is constructed by the mean of a customized heuristic. Next, in order to create individuals composing the initial population, a simple perturbation procedure is applied to the initial solution. It consists of the prohibition of a random request to be assigned to a dispatched vehicle. This action allows the allocation of the request to other vehicles and permits to produce new feasible solutions in the population.

The population is then diversified by the use of the crossover (see Section 4.3). The produced solutions are evaluated and the best one is selected for the next iteration in the EDA. The descent mechanism is iteratively repeated through the perturbation to generate the population, the crossover, and the selection. The algorithm is stopped when the number maximum of iterations is reached and the final solution is kept as the best.

#### 4.1. Representation of a Solution

In order to represent a solution to the problem, we use an array of the allocation of the customers to a vehicle. Each vector represents a vehicle’s tour satisfying a set of requests. The number of tours is equal to the number of vehicles. The elements of a vector are a succession of visited nodes from the depot to it. The depot is not represented in Figure 5, but it is considered in the implementation of the EDA. In this example, seven requests are served by two vehicles. Each satisfied request is represented by a pair of origin and destination nodes such as the pair (5, 12) for request 5 in tour 1.

A solution is said to be feasible if it results in the routes satisfying all the problem constraints. Thus, the output variables present the time schedule for each visited node in a tour as in Figure 1. In what follows, we explain how these variables are calculated to produce an effective routing plan which considers the customized design of the time windows



**Figure 5.** An example of a solution in the customer-oriented DARP.

#### 4.2. The Customized Insertion Heuristic

The customized insertion heuristic builds a solution that satisfies all the requests using a set of vehicles. Its specific feature relies on the customized design of the transport schedule ensuring the enhancement of the quality of service. The steps of the customized insertion heuristic are as follows.

**Step 1:** Initially, each tour is empty and all the vehicles are available.

**Step 2:** Sort the list of non-allocated requests in the increasing order of the upper bound of their initial time window at the pickup nodes.

**Step 3:** Select a vehicle from the list of available ones, according to its capacity considering the load of the selected request  $i$ .

**Step 4:** Select a request from the ordered list of non-allocated requests.

**Step 5:** Before any new request’s insertion, check if there are some previous allocated requests in the tour, then execute some deliveries to ensure that the maximum riding time (10) is respected, and go to step 7.

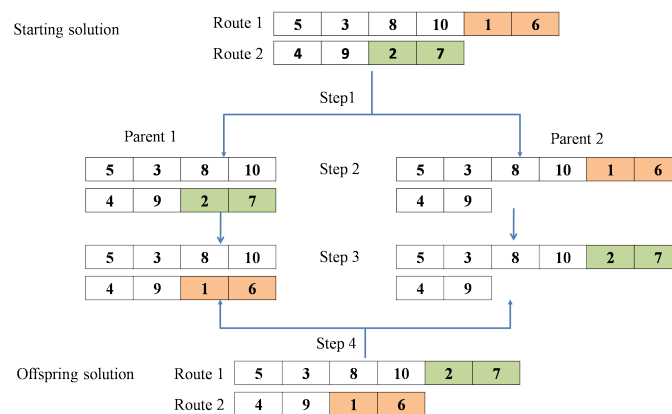
- Compute the customized time windows with (17) and (18).
- Update the time schedule with the customized beginning of service.

- Update the vehicle’s capacity and the vehicle’s total tour duration.
- **Step 6:** Check the feasibility of the insertion of the request in the tour according to the current time schedule, the customized time window (15) and (16) of the request, and the total duration of the vehicle tour.
- In the case of a feasible insertion, insert the pickup node of the request in a valid position in the tour.
- Update the vehicle’s capacity and total tour duration, delete the request from the list of non-assigned requests and go to step 4.
- When the vehicle capacity is completed, deliver some requests in the descending order of their maximum riding times and update the time schedule with new customized time windows (17), (18); go to step 7.
- Go to Step 3 when the insertion is not feasible.
- **Step 7:** Check the maximum duration of the vehicle tour and set the dispatched vehicle as non-available in the case of a potential violation of the constraint; go to step 3.

The insertion algorithm allocates iteratively customer requests while problem constraints are satisfied. Thus, given a set of requests, the algorithm looks for valid positions on each tour. When the insertion is not possible, the heuristic looks for another vehicle. In addition, the feasibility of the insertion is linked with the maximum total duration of the tour trip. If it is a valid insertion, all the vehicles, requests and schedule parameters are managed and updated. Moreover, to enhance the quality of the routing plan, the time windows are customized. With this regard, new time windows are computed by reducing unnecessary delays. Hence, the customized time windows are computed using Equations (15)–(18), see Section 3.2. Consequently, additional requests can be accepted for insertion into a vehicle’s tour.

### 4.3. The Crossover Operator for the Customer-Oriented DARP

The crossover is a classic gene-changing process by which the offspring inherit a genetic code slightly different from the parents. The crossover operator in this work consists in creating an asymmetry between two copies of an individual. An example of the crossover operator is illustrated by Figure 6. In this example, we consider five requests that are satisfied by two vehicles.



**Figure 6.** An overview of the crossover operator.

First, an individual is randomly selected from the current population. In step 1 of Figure 6, a copy of this solution is created. Hence, two parents are considered for our crossover operator. Next, from each route, a complete sub-route is chosen (with the corresponding pickup and delivery nodes). In the example, we select the sub-route from node 1 to node 6 from route 1, and we select as an example, the sub-route from nodes 2 to 7 from route 2. Then, in step 2, the sub-sequence from 1 to 6 is removed from parent 1, and the sub-sequence from 2 to 7 is deleted from parent 2. Next, in step 3, two new

solutions are constructed by exchanging the selected sub-routes respecting their exchanged positions in the tours. This exchange is guaranteed if it maintains the feasibility of the solution. With this regard, all the constraints of the problem must be checked at this stage. Finally, in step 4, an offspring solution is constructed by the union of the two resulting parts of solutions.

## 5. Experimental Study

In this section, an experimental study is conducted in order to assess the performance of the EDA method in terms of reliability and efficiency.

### 5.1. Description of the Benchmark Test Instances

The benchmark test instances which have been chosen for the study are mobility-on-demand problems taken from the work of [39]. These instances are built upon realistic geographical data where pickup and delivery points are located sites defined by a geographical information system. Large distances separate the locations and the networks are of different complexities.

These instances include customer-oriented values with parameters related to the quality of service. Indeed, maximum riding times are specified for each customer as well as time windows which are supposed for pickup and delivery nodes. For these requests, the time windows are of different sizes depending on the customers' preferences. The maximum riding times are also expressed according to the distances between the requests of the customers. A homogeneous fleet of vehicles is available at the depot with a maximum capacity of eight persons. A maximum total tour duration equal to 480 is considered. There are 96 instances, but only 15 examples are reported for this experimentation. These instances were selected considering various problem sizes with between 10 and 47 requests. From each set of problems with the same number of requests, one instance was selected. The selected instances are those solved by the branch and bound method embedded in the CPLEX software within a limit of four hours.

### 5.2. Experimental Protocol

The EDA was implemented in C++ language using the DevC++ compiler. The mixed-integer linear formulation of the Customer-Oriented DARP was implemented in OPL language in the commercial software CPLEX 12.6. This latter uses the branch and bound technique for an exact resolution of the problem. The two methods were executed on an Intel(R) Core(TM) computer (2.4 GHz) with 4 GB RAM.

Three types of experiments were conducted. In the first level, a comparison is made between near-optimal solutions obtained with the EDA and the lower bounds of the solution space provided by CPLEX. In order to assess the performance of the EDA, both the execution time for finding the exact and approximate solutions are provided as well as these solutions. In this work, we limit the calculations with CPLEX to four hours of *cpu* time.

In the second level of experimentation, we compare our evolutionary method EDA with another local search based metaheuristic which is a Tabu Search method dedicated to DARPs with customer specifications, see the work of [34]. In the third level of the experimentation, we compare our EDA method with a more complicated evolutionary technique where the genetic schema is hybridized with a local search dynamically managed over the search. This hybridized method is the ELS of [13].

In order to fairly compare the different methods, a parameter *gap* (%) is introduced. It is calculated as  $((cost_2 - cost_1) / cost_1) * 100$  where  $cost_1$  is the cost of the execution of a  $method_1$  and  $cost_2$  is the cost of a  $method_2$  compared to the first one. Let us note that a negative value of *gap* expresses an improvement of the EDA as compared with the other methods.

### 5.3. Comparison of the Results Obtained with the EDA Method and CPLEX

The computational results of the first testing phase are presented in two main tables. Table 3 reports the results obtained in less than two hours. In Table 4, the results of CPLEX take more than two hours. In the three first columns, the instances are described with their names, the total number of nodes, and the total number of vehicles. All the instances of the tables have been tested with CPLEX which gave the optimal solutions and with the EDA which provided near-optimal ones.

The comparison is expressed in terms of gaps in the value of the total costs and the *cpu* times. Table 3 presents problems with up to 58 nodes with 29 requests. In Table 4, between 68 and 94 nodes are considered and thus 34 to 47 requests have to be addressed.

**Table 3.** Comparison of the results obtained with CPLEX and the EDA in less than two hours *cpu* time.

Instances	<i>n</i>	<i>m</i>	CPLEX		EDA		Gap (%)
			<i>f</i> (S)	Time (mn)	<i>f</i> (S)	Time (mn)	
SF_RL_d75	10	2	150.01	30.12	150.05	2.15	0.02%
SF_RL_d92	17	2	320.92	22.38	323.16	2.24	0.70%
SF_RL_d93	20	2	394.5	30.22	412.35	3.22	4.53%
SF_RL_d94	23	2	317.29	40.30	325.98	2.18	2.74%
SF_RL_d55	28	4	1365.92	50.51	1396.29	2.21	2.22%
SF_RL_d52	29	4	1481.30	58.57	1527.96	3.08	3.15%

**Table 4.** Comparison of the results obtained with CPLEX and the EDA in up to four hours *cpu* time.

Instances	<i>n</i>	<i>m</i>	CPLEX		EDA		Gap (%)
			<i>f</i> (S)	Time (mn)	<i>f</i> (S)	Time (mn)	
SF_RL_d10	34	4	1362.15	71.34	1401.84	3.18	2.91%
SF_RL_d39	38	6	2061.61	73.73	2073.39	2.50	0.57%
SF_RL_d70	39	6	2000.3	83.58	2012.05	5.47	0.58%
SF_RL_d82	39	6	1830.12	76.35	1920.25	3.47	4.92%
SF_RL_d08	42	7	1682.52	85.48	1832.94	2.48	8.94%
SF_RL_d36	42	6	2180.52	79.16	2197.05	5.42	0.76%
SF_RL_d43	43	6	2033.64	80.36	2097.47	12.34	3.14%
SF_RL_d01	46	7	2339.41	119.00	2477.88	13.48	5.92%
SF_RL_d11	47	6	2499.62	104.19	2500.07	13.73	0.02%

Regarding the costs obtained with the EDA, the calculated *gaps* are smaller than 5% in both Tables 3 and 4, except the case of SF\_RL\_d08. However, for CPLEX, the problem resolution implies longer times, especially for problems involving more than 20 requests. For instance, the resolution of the instance SF\_RL\_d36 requires more than two hours for CPLEX whereas about five minutes are consumed by the EDA. As a consequence, the EDA can provide near-optimal solutions in a much shorter time for hard real life instances.

### 5.4. Comparison of the Results Obtained with the EDA Technique and Existing Methods from the State of the Art

In this part of the study, we show the quality of the solutions produced by the EDA by comparing them to other metaheuristic based approaches.

In the first Table 5, we compare our EDA to a local search-based method which is a Tabu Search dedicated to DARPs addressing some customers' expectations, see the work of [34]. In Table 6, we observe the behaviour of our evolutionary method as compared with the ELS of [13] which is more complicated than our method since the hybridization of the genetic schema is managed using six parameters which trigger the diversification and intensification operators dynamically during the run. As no computational time was given in the reference presenting the results of Chassaing et al. and tested on the same instances,

in this paper, the *cpu* times are avoided from Tables 5 and 6. Similarly to the previous experimental part, both the tables are divided into two parts separated by a horizontal line which separates the problems which consider more or less than 29 requests.

#### 5.4.1. Comparison of the Results Obtained with the EDA and a Tabu Search

Table 5 presents a comparison of the results obtained with the EDA and a Tabu Search specially dedicated to customer-oriented DARPs.

**Table 5.** Comparison of the results obtained with the EDA and the Tabu Search.

Instances	<i>n</i>	<i>m</i>	Tabu Search	EDA	Gap (%)
			<i>f</i> (S)	<i>f</i> (S)	
SF_RL_d75	10	2	150.22	<b>150.05</b>	−0.11%
SF_RL_d92	17	2	324.01	<b>323.16</b>	−0.26%
SF_RL_d93	20	2	430.01	<b>412.35</b>	−4.11%
SF_RL_d94	23	2	345.82	<b>325.98</b>	−5.73%
SF_RL_d55	28	4	1469.28	<b>1396.29</b>	−4.97%
SF_RL_d52	29	4	1528.45	<b>1527.96</b>	−0.03%
SF_RL_d10	34	4	1375.86	1401.84	1.89%
SF_RL_d39	38	6	2091.61	<b>2073.39</b>	−0.90%
SF_RL_d70	39	6	2148.97	<b>2012.05</b>	−6.37%
SF_RL_d82	39	6	1918.25	1920.25	0.10%
SF_RL_d08	42	7	1992.69	<b>1832.94</b>	−8.02%
SF_RL_d36	42	6	2198.22	2197.05	−0.05%
SF_RL_d43	43	6	1995.23	2097.47	5.12%
SF_RL_d01	46	7	2506.84	<b>2477.88</b>	−1.15%
SF_RL_d11	47	6	2661.76	<b>2500.07</b>	−6.07%

The upper part of the Table 5 shows that there are better results with the EDA than with the TS. This is reflected by the negative values of the gaps. Additional negative gaps appear in the second part of the table. As an example, on the instances SF\_RL\_d08 and SF\_RL\_d01, the negative deviations −8.02% and −1.15% are obtained. Therefore, one can observe a positive impact of the descent technique in the EDA. Results obtained with the EDA are better thanks to a faster archival of the best population along the run in the descent process. Globally, the EDA has more improved results for problems with more than 29 requests.

#### 5.4.2. Comparison of the Results Obtained with the EDA and the ELS Methods

The second study is performed on the same realistic instances to highlight the efficacy of our algorithm as compared with the hybridized ELS which counts six neighbourhoods in the local search.

In Table 6, one can observe that the behaviour of the EDA is aligned with that of the ELS when the size of the problems grows. Moreover, for instances with less than 58 nodes (29 requests), all the results obtained with the EDA are better than those of the ELS. For problems with more than 68 nodes, the quality of the solutions is relatively good, as was also observed in Table 5. However, there are only two instances where an improvement of the solution is obtained. It is for the instances SF\_RL\_d08 and SF\_RL\_d11 where the negative gaps −1.31% and −1.50% are produced respectively.

Moreover, for the same size of the network as the one with 42 requests, the costs obtained can be different. Two different gaps, one negative −1.31% and the other positive 2.69% are found on the instances SF\_RL\_d08 and SF\_RL\_d36. This fluctuation of the costs can be explained by the complexity of the on-demand transportation network.

Furthermore, even if the results seem to be less effective for large instances with more than 29 requests, the EDA is simpler and faster than the ELS since it uses only one search

space perturbation operator. Besides, the positive gaps never exceed 5% in experiments carried out on problems with a high complexity.

**Table 6.** Comparison of the results obtained with the EDA and the ELS of [13].

Instances	$n$	$m$	ELS	EDA	Gap
			$f$ (S)	$f$ (S)	(%)
SF_RL_d75	10	2	150.91	<b>150.05</b>	−0.57%
SF_RL_d92	17	2	347.01	<b>323.16</b>	−6.88%
SF_RL_d93	20	2	418.76	<b>412.35</b>	−1.53%
SF_RL_d94	23	2	352.25	<b>325.98</b>	−7.46%
SF_RL_d55	28	4	1516.71	<b>1396.29</b>	−7.94%
SF_RL_d52	29	4	1607.74	<b>1527.96</b>	−4.96%
SF_RL_d10	34	4	1341.02	1401.84	4.53%
SF_RL_d39	38	6	2030.44	2073.39	2.12%
SF_RL_d70	39	6	2006.05	2212.05	0.30%
SF_RL_d82	39	6	1842.84	1920.25	4.20%
SF_RL_d08	42	7	1857.35	<b>1832.94</b>	−1.31%
SF_RL_d36	42	6	2139.52	2197.05	2.69%
SF_RL_d43	43	6	2002.15	2097.47	4.76%
SF_RL_d01	46	7	2396.55	2477.88	3.39%
SF_RL_d11	47	6	2538.18	<b>2500.07</b>	−1.50%

## 6. Discussion

In this paper, a customer-oriented DARP has been tackled by a new evolutionary descent approach. In this approach, a descent technique aims at optimizing the quality of the population which evolves during the search. The exploration of the search space is ensured using new evolutionary operators.

The comparative results between the EDA and CPLEX show low gaps for almost all the instances tested ensuring lower execution times for a high quality of the solutions. Despite the use of a single evolutionary operator, the EDA method outperforms results obtained with other metaheuristics of the literature, namely the ELS and the TS. When compared with the TS, twelve out of the fifteen instances tested have been improved by our method. As compared with the hybrid evolutionary local search, the results are also promising with gaps reduced to 5% on instances of considerable size and complexity.

Indeed, for small instances, EDA outperforms ELS. For larger instances with more than 29 requests, the results are competitive. Let us note that the EDA is however faster and simpler to implement whereas the ELS uses six operators to manage the search for good solutions. Thus, an enhanced hybridized evolutionary algorithm with more elaborated operators is a promising direction for future works to cope with the specificity of the customers designed according to their preferences and the new requirements of current transport on demands. In addition, the proposal of the new time windows constraints makes the problem more difficult to address, especially when it is applied to real transport cases as presented in this study.

## 7. Conclusions

This paper describes a customer-oriented Dial-A-Ride Problem with specific customer-oriented constraints. A modelization of the problem is proposed using a mixed-integer linear program. For this model, a computation of lower bounds of the solution space is provided by the branch and bound method of CPLEX. To cope with the exponential *cpu* times, an evolutionary descent algorithm is developed relying on a new evolutionary operator. The comparison between the two methods indicates low deviations. When our EDA method is compared to other metaheuristics such as the Tabu Search method or another hybrid evolutionary one, promising results are obtained. For more complex transport problems, the results are less spectacular, while they can be obtained in a faster

time since it depends on only one operator for the crossover phase. This saved time can be used to test additional evolutionary operators. Moreover, as the Tabu Search is known to address efficiently DARPs, another hybridization of the evolutionary schema with a Tabu Search can be envisaged to tackle a wider range of mobility-on-demand problems.

**Author Contributions:** Conceptualization, H.B.; Formal analysis, W.A.-M.; Methodology, S.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** The project is funded in part by the National Institutes of Health, under Grant No. 5R01CA136535.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Cordeau, J.F.; Laporte, G. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transp. Res. Part B Methodol.* **2003**, *37*, 579–594. [CrossRef]
2. Cordeau, J.F.; Laporte, G. The dial-a-ride problem: Models and algorithms. *Ann. Oper. Res.* **2007**, *153*, 29–46. [CrossRef]
3. Paquette, J.; Cordeau, J.F.; Laporte, G. Quality of service in dial-a-ride operations. *Comput. Ind. Eng.* **2009**, *56*, 1721–1734. [CrossRef]
4. Paquette, J.; Bellavance, F.; Cordeau, J.F.; Laporte, G. Measuring quality of service in dial-a-ride operations: The case of a Canadian city. *Transportation* **2012**, *39*, 539–564. [CrossRef]
5. Lehuédé, F.; Masson, R.; Parragh, S.N.; Péton, O.; Tricoire, F. A multi-criteria large neighbourhood search for the transportation of disabled people. *J. Oper. Res. Soc.* **2014**, *65*, 983–1000. [CrossRef]
6. Masson, R.; Lehuédé, F.; Péton, O. The dial-a-ride problem with transfers. *Comput. Oper. Res.* **2014**, *41*, 12–23. [CrossRef]
7. Häll, C.H.; Andersson, H.; Lundgren, J.T.; Värbrand, P. The integrated dial-a-ride problem. *Public Transp.* **2009**, *1*, 39–54. [CrossRef]
8. Ma, T.Y.; Rasulkhani, S.; Chow, J.Y.; Klein, S. A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers. *Transp. Res. Part E Logist. Transp. Rev.* **2019**, *128*, 417–442. [CrossRef]
9. Cappart, Q.; Thomas, C.; Schaus, P.; Rousseau, L.M. A constraint programming approach for solving patient transportation problems. In *International Conference on Principles and Practice of Constraint Programming*; Springer: Cham, Switzerland, 2018; pp. 490–506. [CrossRef]
10. Detti, P.; Papalini, F.; de Lara, G.Z.M. A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. *Omega* **2017**, *70*, 1–14. [CrossRef]
11. Çetin, T. The rise of ride sharing in urban transport: Threat or opportunity? In *Urban Transport Systems*; IntechOpen: London, UK, 2017; Volume 191. [CrossRef]
12. Wong, K.I.; Han, A.; Yuen, C. On dynamic demand responsive transport services with degree of dynamism. *Transp. A Transp. Sci.* **2014**, *10*, 55–73. [CrossRef]
13. Chassaing, M.; Duhamel, C.; Lacomme, P. An ELS-based approach with dynamic probabilities management in local search for the Dial-A-Ride Problem. *Eng. Appl. Artif. Intell.* **2016**, *48*, 119–133. [CrossRef]
14. Ho, S.; Nagavarapu, S.C.; Pandi, R.R.; Dauwels, J. An improved tabu search heuristic for static dial-a-ride problem. *arXiv* **2018**, arXiv:1801.09547. [CrossRef]
15. Carotenuto, P.; Martis, F. A double dynamic fast algorithm to solve multi-vehicle Dial a Ride Problem. *Transp. Res. Procedia* **2017**, *27*, 632–639. [CrossRef]
16. Gökay, S.; Heuvels, A.; Krempels, K.H. On-demand ride-sharing services with meeting Points. In *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2019)*, Heraklion, Greece, 3–5 May 2019. [CrossRef]
17. Shaheen, S.; Cohen, A.; Yelchuru, B.; Sarkhili, S.; Hamilton, B.A. Mobility on Demand Operational Concept Report. 2017. Available online: <https://rosap.ntl.bts.gov/view/dot/34258> (accessed on 1 January 2022).
18. Hyland, M.; Mahmassani, H.S. Operational benefits and challenges of shared-ride automated mobility-on-demand services. *Transp. Res. Part A Policy Pract.* **2020**, *134*, 251–270. [CrossRef]
19. Vij, A.; Ryan, S.; Sampson, S.; Harris, S. Consumer preferences for on-demand transport in Australia. *Transp. Res. Part A Policy Pract.* **2020**, *132*, 823–839. [CrossRef]
20. Nasri, S.; Bouziri, H.; Aggoune-Mtalaa, W. Customer-Oriented Dial-A-Ride Problems: A Survey on Relevant Variants, Solution Approaches and Applications. In *Emerging Trends in ICT for Sustainable Development*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 111–119. [CrossRef]



21. Parragh, S.N. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 912–930. [CrossRef]
22. Paquette, J.; Cordeau, J.F.; Laporte, G.; Pascoal, M.M. Combining multicriteria analysis and tabu search for dial-a-ride problems. *Transp. Res. Part B Methodol.* **2013**, *52*, 1–16. [CrossRef]
23. Jorgensen, R.M.; Larsen, J.; Bergvinsdottir, K.B. Solving the dial-a-ride problem using genetic algorithms. *J. Oper. Res. Soc.* **2007**, *58*, 1321–1331. [CrossRef]
24. Melachrinoudis, E.; Min, H. A tabu search heuristic for solving the multi-depot, multi-vehicle, double request dial-a-ride problem faced by a healthcare organisation. *Int. J. Oper. Res.* **2011**, *10*, 214–239. [CrossRef]
25. Molenbruch, Y.; Braekers, K.; Caris, A. Operational effects of service level variations for the dial-a-ride problem. *Cent. Eur. J. Oper. Res.* **2017**, *25*, 71–90. [CrossRef]
26. Braekers, K.; Kovacs, A.A. A multi-period dial-a-ride problem with driver consistency. *Transp. Res. Part B Methodol.* **2016**, *94*, 355–377. [CrossRef]
27. Molenbruch, Y.; Braekers, K.; Caris, A. Benefits of horizontal cooperation in dial-a-ride services. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *107*, 97–119. [CrossRef]
28. Parragh, S.N.; Doerner, K.F.; Hartl, R.F.; Gandibleux, X. A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. *Netw. Int. J.* **2009**, *54*, 227–242. [CrossRef]
29. Marković, N.; Nair, R.; Schonfeld, P.; Miller-Hooks, E.; Mohebbi, M. Optimizing dial-a-ride services in Maryland: Benefits of computerized routing and scheduling. *Transp. Res. Part C Emerg. Technol.* **2015**, *55*, 156–165. [CrossRef]
30. Hu, T.Y.; Chang, C.P. A revised branch-and-price algorithm for dial-a-ride problems with the consideration of time-dependent travel cost. *J. Adv. Transp.* **2015**, *49*, 700–723. [CrossRef]
31. Glover, F. *Tabu Search Fundamentals and Uses*; Graduate School of Business, University of Colorado Boulder: Boulder, CO, USA, 1995.
32. Cordeau, J.F.; Laporte, G. The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. *Q. J. Belg. Fr. Ital. Oper. Res. Soc.* **2003**, *1*, 89–101. [CrossRef]
33. Torgal, M.; Dias, T.G.; Fontes, T. A multi objective approach for DRT service using tabu search. *Transp. Res. Procedia* **2021**, *52*, 91–98. [CrossRef]
34. Nasri, S.; Bouziri, H. Improving total transit time in dial-a-ride problem with customers-dependent criteria. In Proceedings of the 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, Tunisia, 30 October–3 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1141–1148. [CrossRef]
35. Cubillos, C.; Urrea, E.; Rodríguez, N. Application of genetic algorithms for the DARPTW problem. *Int. J. Comput. Commun. Control* **2009**, *4*, 127–136. [CrossRef]
36. Sun, B.; Wei, M.; Yang, C.; Xu, Z.; Wang, H. Personalised and coordinated demand-responsive feeder transit service design: A genetic algorithms approach. *Future Internet* **2018**, *10*, 61. [CrossRef]
37. Rekiek, B.; Delchambre, A.; Saleh, H.A. Handicapped person transportation: An application of the grouping genetic algorithm. *Eng. Appl. Artif. Intell.* **2006**, *19*, 511–520. [CrossRef]
38. Cordeau. Instances of Cordeau. 2003. Available online: <http://neumann.hec.ca/chairedistributique/data/darp/> (accessed on 19 April 2020).
39. Chassaing. Instances of Chassaing. 2020. Available online: <http://fc.isima.fr/~lacomme/Maxime/> (accessed on 19 July 2020).
40. Healy, P.; Moll, R. A new extension of local search applied to the dial-a-ride problem. *Eur. J. Oper. Res.* **1995**, *83*, 83–104. [CrossRef]
41. Parragh, S.N.; Doerner, K.F.; Hartl, R.F. Variable neighborhood search for the dial-a-ride problem. *Comput. Oper. Res.* **2010**, *37*, 1129–1138. [CrossRef]
42. Parragh, S.N.; Cordeau, J.F.; Doerner, K.F.; Hartl, R.F. Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR Spectr.* **2012**, *34*, 593–633. [CrossRef]
43. Rezgui, D.; Siala, J.C.; Aggoune-Mtalaa, W.; Bouziri, H. Towards smart urban freight distribution using fleets of modular electric vehicles. In *Innovations in Smart Cities and Applications: Proceedings of the 2nd Mediterranean Symposium on Smart City Applications, Tangier, Morocco, 15–27 October 2017*; Lecture Notes in Networks and Systems; Springer: Berlin/Heidelberg, Germany, 2017; Volume 37, pp. 602–612. [CrossRef]
44. Djenouri, Y.; Habbas, Z.; Aggoune-Mtalaa, W. Bees swarm optimization metaheuristic guided by decomposition for solving MAX-SAT. In Proceedings of the 8th International Conference on Agents and Artificial Intelligence—ICAART 2016, Rome, Italy, 24–26 February 2016; Volume 2, pp. 472–479. [CrossRef]
45. Aggoune-Mtalaa, W.; Aggoune, R. An optimization algorithm to schedule care for the elderly at home. *Int. J. Inf. Sci. Intell. Syst.* **2014**, *3*, 41–50.
46. Nasri, S.; Bouziri, H.; Aggoune-Mtalaa, W. Dynamic on Demand Responsive Transport with Time-Dependent Customer Load. In *Innovations in Smart Cities Applications Volume 4: The Proceedings of the 5th International Conference on Smart City Applications*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 395–409. [CrossRef]
47. Rezgui, D.; Bouziri, H.; Aggoune-Mtalaa, W.; Siala, J.C. A Hybrid Evolutionary Algorithm for Smart Freight Delivery with Electric Modular Vehicles. In Proceedings of the 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), Aqaba, Jordan, 28 October–1 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–8. [CrossRef]

48. Rezgui, D.; Bouziri, H.; Aggoune-Mtalaa, W.; Siala, J.C. An Evolutionary Variable Neighborhood Descent for Addressing an Electric VRP Variant. In *International Conference on Variable Neighborhood Search; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 11328, pp. 216–231. [CrossRef]*
49. Amroun, K.; Habbas, Z.; Aggoune-Mtalaa, W. A compressed Generalized Hypertree Decomposition-based solving technique for non-binary Constraint Satisfaction Problems. *AI Commun.* **2016**, *29*, 371–392. [\[CrossRef\]](#)
50. Bennekrouf, M.; Aggoune-Mtalaa, W.; Sari, Z. A generic model for network design including remanufacturing activities. *Supply Chain Forum* **2013**, *14*, 4–17. [\[CrossRef\]](#)